



Grado en Ingeniería en Tecnologías Industriales

Trabajo Fin de Grado

**OneMap+: evaluación, extensiones y despliegue
eficiente de un mapeo semántico
open-vocabulary para navegación multi-objeto
en tiempo real**

Autor

Mauro Liz Soto

Dirigido por

Dr. Joaquim de Curtò i Díaz

Madrid

Junio de 2026

Declaración de originalidad

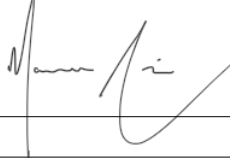
Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título **OneMap+: evaluación, extensiones y despliegue eficiente de un mapeo semántico open-vocabulary para navegación multi-objeto en tiempo real** en la ETS de Ingeniería – ICAI de la Universidad Pontificia Comillas en el curso académico 2025/2026 es de mi autoría, original e inédito y no ha sido presentado con anterioridad a otros efectos. El Proyecto no es plagio de otro, ni total ni parcialmente, y la información que ha sido tomada de otros documentos está debidamente referenciada.

Uso de Inteligencia Artificial¹

Declaro bajo mi responsabilidad que (indicar la opción correcta):

- ☐ No he utilizado Inteligencia Artificial en la elaboración del presente documento.
- ☒ He utilizado Inteligencia Artificial en la elaboración del presente documento y/o del Anexo B siempre en las condiciones permitidas por la Universidad Pontificia Comillas, es decir, aplicando el Nivel 2 de la Escala de Evaluación de Perkins et al. (2024): *“La IA puede utilizarse para actividades previas a la tarea, como la lluvia de ideas, la descripción y la investigación inicial. Este nivel se centra en el uso de la IA para la planificación, la síntesis y la generación de ideas, pero las evaluaciones deben hacer hincapié en la capacidad de desarrollar y refinar estas ideas de forma independiente”*. En concreto, la Inteligencia Artificial ha sido empleada para:

La inteligencia artificial se empleó en cuestiones de formato en LaTeX, en la interpretación de errores en el código desarrollado, en la revisión de gramática, ortografía y tono académico, y como apoyo en la producción de scripts para generar gráficas a partir de los datos obtenidos.


Fdo.: Mauro Liz Soto
Fecha: 08 / 07 / 2026

Autorización para la entrega del Proyecto

	J. de Curtò	Digitally signed by J. de Curtò Date: 2026.07.10 21:32:57 +03'00'
Fdo.: Dr. Joaquim de Curtò i Díaz		
Fecha: 08 / 07 / 2026		

¹Esta declaración se refiere al uso de la Inteligencia Artificial generativa para realizar los documentos del Proyecto, incluyendo la memoria y el Anexo B. No aplica a proyectos donde, por su naturaleza, deban emplear inteligencia artificial como parte de los mismos, como la aplicación de técnicas de aprendizaje automático, redes neuronales o análisis de datos.



Grado en Ingeniería en Tecnologías Industriales

Trabajo Fin de Grado

**OneMap+: evaluación, extensiones y despliegue
eficiente de un mapeo semántico
open-vocabulary para navegación multi-objeto
en tiempo real**

Autor

Mauro Liz Soto

Dirigido por

Dr. Joaquim de Curtò i Díaz

Madrid

Junio de 2026

ONEMAP+: EVALUACIÓN, EXTENSIONES Y DESPLIEGUE EFICIENTE DE UN MAPEO SEMÁNTICO OPEN-VOCABULARY PARA NAVEGACIÓN MULTI-OBJETO EN TIEMPO REAL

Autor: Liz Soto, Mauro.

Director: Dr. de Curtò i Díaz, Joaquim.

Entidad Colaboradora: ICAI – Universidad Pontificia Comillas / BARCELONA Supercomputing Center.

RESUMEN DEL PROYECTO

1. Introducción

Los sistemas de navegación autónoma modernos no solo necesitan detectar obstáculos, sino también comprender escenas y objetivos expresados en lenguaje natural. Esta necesidad exige pasar de una representación puramente geométrica del entorno a otra capaz de interpretar qué representan distintos objetos dentro de una escena. *OneMap* [1] aborda precisamente esta cuestión mediante una metodología orientada a dotar a un agente robótico de la capacidad de formar una representación conjunta semántica y espacial del entorno, con el fin de guiar de manera eficiente la búsqueda de objetos.

En contextos como *OneMap*, los modelos de visión-lenguaje emergen como herramientas eficaces para relacionar información visual con consultas formuladas mediante texto. No obstante, en este tipo de aplicaciones, el idioma en el que se formulan dichas consultas deja de ser un simple medio de comunicación con el usuario para convertirse en un factor que interviene en la interpretación del sistema. Se plantea, por tanto, una situación en la que el idioma afecta a los resultados de las tareas desarrolladas por agentes autónomos, con posibles implicaciones sobre el rendimiento, la consistencia de las respuestas y el coste energético asociado al procesamiento de las consultas.

A esta dependencia lingüística se suma otra dimensión relevante: el coste energético de los modelos empleados. Este análisis supone una brecha de investigación en la literatura existente, pues la mayoría de enfoques modernos se centran en *benchmarks* basados en métricas de rendimiento orientadas a tareas específicas. Esto permite comparar capacidades entre sistemas, pero ha relegado dimensiones clave como la eficiencia energética.

Considerando esto, se plantea la necesidad de evaluar modelos de lenguaje y visión-lenguaje no solo a través de métricas tradicionales de rendimiento, sino incorporando también elementos como el multilingüismo y el coste energético y computacional asociado a su ejecución. Por lo tanto, el enfoque de este trabajo se puede encapsular en la siguiente pregunta de investigación:

¿En qué medida el idioma condiciona el rendimiento y la eficiencia energética de modelos de lenguaje y visión-lenguaje utilizados como componentes perceptivos en tareas de navegación semántica autónoma?

2. Metodología

Para abordar la pregunta de investigación, se realizan una serie de experimentos. Estos no deben entenderse como estudios independientes, sino como aproximaciones complementarias a un problema común. Inicialmente, la pregunta de investigación se aborda desde una perspectiva puramente textual; seguidamente, desde un enfoque multimodal aplicado a escenas de navegación; y, finalmente, desde un marco próximo al mecanismo de consulta semántica empleado en *OneMap*: el *grounding* visual. Cada sección se apoya en la anterior, de manera que el desarrollo experimental de este trabajo presenta una evolución progresiva y refleja, a su vez, la organización cronológica de los experimentos.

1. Evaluación multilingüe de modelos de lenguaje

Se desarrolla un marco experimental para evaluar conjuntamente el consumo energético y el rendimiento de LLMs en tareas de razonamiento. En este contexto, se analizan cinco modelos de lenguaje en trece idiomas distintos, empleando un conjunto de 79 problemas de razonamiento. El análisis se realiza bajo una configuración de *hardware* controlada, basada en GPUs NVIDIA H200. Se propone el *AI Energy Score* como *benchmark* de evaluación para distintas combinaciones de idioma y modelo. Esta métrica integra, bajo una misma formulación, tanto el rendimiento, puntuado a partir de una serie de tareas de razonamiento, como el consumo energético durante el proceso de inferencia, permitiendo comparaciones bidimensionales entre LLMs.

2. Evaluación multilingüe de modelos de visión-lenguaje

Se desarrolla un marco experimental para evaluar sistemas multimodales en tareas de percepción visual aplicadas a la navegación autónoma. Se analiza si las tendencias lingüísticas observadas en modelos de lenguaje se extienden a tareas de VQA. En este caso, en lugar de plantear problemas puramente textuales, las consultas

incorporan una componente visual. Se evalúa, por tanto, la capacidad de distintos modelos para responder a preguntas formuladas a partir de una imagen de entrada. El conjunto de datos empleado consiste en una colección de fotografías de escenarios de conducción procedentes del *dataset* BDD100K [2]. De este modo, la evaluación se plantea en un contexto de navegación autónoma, tomando como referencia el enfoque de *OneMap*.

3. Estudio del grounding visual condicionado por el idioma

Se estudia la influencia del idioma en los resultados del *grounding* visual mediante arquitecturas CLIP. Esta contribución permite evaluar la capacidad de distintos modelos e idiomas para relacionar información espacial y semántica. Se emplea el mismo conjunto de idiomas e imágenes y se evalúa si consultas equivalentes formuladas en distintos idiomas permiten no solo identificar un concepto visual dentro de una imagen, sino también localizarlo en regiones similares de la escena. Mediante este estudio, los conceptos abordados en los experimentos anteriores se aterrizan ya en el contexto específico de *OneMap*, analizando un proceso de construcción de mapas de similitud semántica.

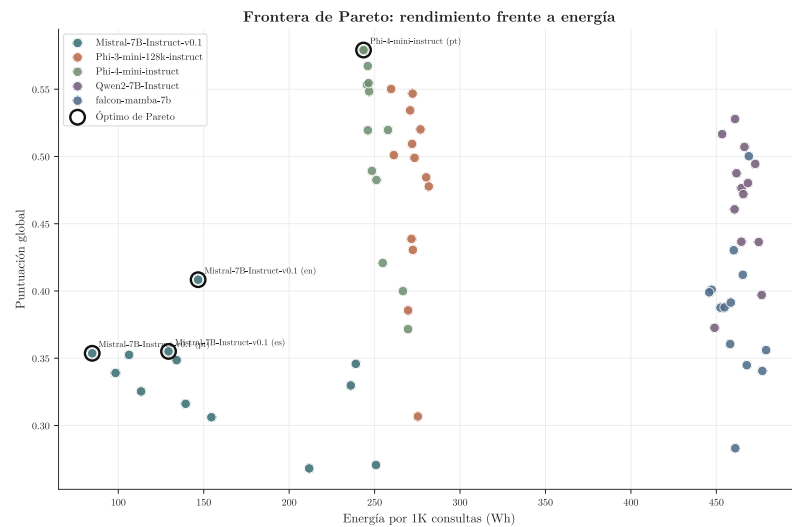
Resumen de los bloques experimentales desarrollados.

Bloque	Modelos	Entrada	Métricas
LLMs multilingües	5 LLMs	Problemas de razonamiento en 13 idiomas	Rendimiento, energía y <i>AI Energy Score</i>
VLMs multilingües	5 VLMs	Imágenes BDD100K y preguntas VQA	Rendimiento y energía
<i>Grounding</i> visual multilingüe	2 CLIP multilingües	Imágenes, conceptos visuales e idiomas	Alineamiento espacial y activación máxima

3. Resultados

Dado el número de experimentos realizados y la variedad de características evaluadas, se ha obtenido un conjunto amplio de datos particularmente ilustrativos. Se han podido estudiar diversas dimensiones, como el rendimiento en distintas tareas, el consumo de recursos computacionales y energéticos, la consistencia de las respuestas y la latencia de los procesos de inferencia. Más allá de esto, la variedad de experimentos y de tecnologías abordadas permite realizar un análisis global del comportamiento de los modelos de lenguaje y visión-lenguaje en contextos multilingües.

En los experimentos realizados, cambiar el idioma de la consulta modifica la respuesta generada, alterando su longitud o su contenido; modifica el proceso de inferencia, afectando a la latencia o al consumo energético; y modifica la interpretación visual,



Rendimiento y consumo energético en la evaluación multilingüe de LLMs.

condicionando el alineamiento semántico o los niveles de activación. Los resultados obtenidos reflejan la existencia de una dependencia lingüística en los sistemas de inteligencia artificial evaluados.

4. Conclusiones

El idioma como variable técnica relevante

El idioma en el que se formula una consulta afecta a la calidad y al contenido de la respuesta producida. Este comportamiento se ha observado tanto en modelos de lenguaje como en modelos visión-lenguaje, donde el rendimiento en distintas tareas de razonamiento y VQA tiende a degradarse a medida que aumenta la distancia léxica respecto al inglés. En particular, destacan los idiomas de bajos recursos como el euskera, el luxemburgués o el árabe, como aquellos que generalmente presentan los peores resultados [3, 4].

El coste computacional de la variación lingüística

En los experimentos realizados, el consumo energético asociado a los procesos de inferencia se ve afectado por el idioma de la consulta. No obstante, no se observa una relación específica a nivel global, ya que la dependencia lingüística del consumo varía en función de la tarea realizada. Más que el idioma, el factor determinante en la varia-

ción del coste computacional parece ser el modelo y la arquitectura empleados [3, 4].

El grounding visual ante la variabilidad lingüística

En tareas de *grounding* visual, el idioma no parece afectar de forma visible a la identificación —o activación— global de los conceptos presentes en una imagen. No obstante, sí puede afectar a la localización espacial de dichos conceptos dentro de la imagen, lo que sugiere la existencia de un desalineamiento espacial dependiente del idioma. Estos resultados se mantienen en modelos de distinto tamaño, medido por el número de parámetros visuales [5].

5. Referencias

- [1] Finn Lukas Busch, Timon Homberger, Jesús Ortega-Peimbert, Quantao Yang y Olov Andersson. *One Map to Find Them All: Real-time Open-Vocabulary Mapping for Zero-shot Multi-Object Navigation*. 3 de mar. de 2025. DOI: [10.48550/arXiv.2409.11764](https://doi.org/10.48550/arXiv.2409.11764). arXiv: [2409.11764\[cs.R0\]](https://arxiv.org/abs/2409.11764). URL: <http://arxiv.org/abs/2409.11764> (visitado 29-06-2026).
- [2] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan y Trevor Darrell. *BDD100K: A Diverse Driving Dataset for Heterogeneous Multitask Learning*. 8 de abr. de 2020. DOI: [10.48550/arXiv.1805.04687](https://doi.org/10.48550/arXiv.1805.04687). arXiv: [1805.04687\[cs.CV\]](https://arxiv.org/abs/1805.04687). URL: <http://arxiv.org/abs/1805.04687> (visitado 03-07-2026).
- [3] I. de Zarzà, Mauro Liz, J. de Curtò y Carlos T. Calafate. «Energy-Aware Multilingual Evaluation of Large Language Models». En: *Electronics* 15.7 (2026). ISSN: 2079-9292. DOI: [10.3390/electronics15071395](https://doi.org/10.3390/electronics15071395). URL: <https://www.mdpi.com/2079-9292/15/7/1395>.
- [4] J. de Curtò, Mauro Liz, I. de Zarzà y Carlos T. Calafate. «Energy-Aware Multilingual Vision-Language Models for Drone Smart Sensing». En: *Drones* 10.5 (2026). ISSN: 2504-446X. DOI: [10.3390/drones10050361](https://doi.org/10.3390/drones10050361). URL: <https://www.mdpi.com/2504-446X/10/5/361>.
- [5] J. de Curtò, Mauro Liz e I. de Zarzà. *Language-Conditioned Visual Grounding with CLIP Multilingual*. 9 de mayo de 2026. DOI: [10.48550/arXiv.2605.09060](https://doi.org/10.48550/arXiv.2605.09060). arXiv: [2605.09060\[cs.CL\]](https://arxiv.org/abs/2605.09060). URL: <http://arxiv.org/abs/2605.09060> (visitado 07-07-2026).

ONEMAP+: EVALUATION, EXTENSIONS AND EFFICIENT DEPLOYMENT OF OPEN-VOCABULARY SEMANTIC MAPPING FOR REAL-TIME MULTI-OBJECT NAVIGATION

Author: Liz Soto, Mauro.

Supervisor: Dr. de Curtò i Díaz, Joaquim.

Collaborating Entity: ICAI – Universidad Pontificia Comillas / BARCELONA Supercomputing Center.

ABSTRACT

1. Introduction

Modern autonomous navigation systems do not simply need to detect obstacles; in many cases, they must also understand scenes and objectives expressed in natural language. This need requires a transition from a purely geometric representation of the environment to one capable of interpreting the meaning of different objects within the context of a given scene. OneMap [1] addresses precisely this issue by employing a methodology aimed at providing an autonomous agent with the capacity to build a joint spatial-semantic representation of its surroundings, with the purpose of guiding object search.

In contexts similar to OneMap, Vision-Language Models (VLMs) emerge as powerful tools for relating spatial information to queries formulated through text. Nevertheless, in these types of applications, the language in which such queries are formulated can no longer be regarded merely as a means of communication with the user. Instead, it becomes a factor that influences the system's interpretation. This creates a situation in which language can directly affect the outcome of autonomous system tasks, with possible implications for performance, response consistency, and the energy cost associated with processing the queries.

In addition to this linguistic dependence, another relevant dimension must be considered: the energy cost of the models employed. This analysis addresses a research gap in the current literature, since most modern approaches focus their evaluations on benchmarks based on performance or accuracy metrics for specific tasks. Although this enables comparisons between the capabilities of different systems, it has often relegated

other important dimensions, such as energy consumption, to a secondary role.

Considering this, the present study highlights the need to evaluate language and vision-language models not only through traditional performance metrics, but also by incorporating aspects such as multilingual behaviour and the power consumption associated with inference. As such, the focus of this work can be encapsulated in the following research question:

To what extent does language condition the performance and energy efficiency of language and vision-language models employed as perceptual components in autonomous semantic navigation tasks?

2. Methodology

To address the research question, a series of experiments are conducted. These must not be thought of as independent studies, but rather as complementary approaches to a common problem. Initially, the research question is addressed from a purely textual perspective; next, through a multimodal approach applied to navigation scenes; and finally, from a framework similar to the semantic methodology formulated in OneMap: visual grounding. Each section builds on the previous one, such that the development of this work presents a progressive evolution while also reflecting the chronological organization of the experiments.

1. *Multilingual evaluation of language models*

An experimental framework is developed to jointly evaluate the performance and energy consumption of LLMs in reasoning tasks. In this context, five language models are analyzed in thirteen different languages, using a dataset of 79 reasoning problems. The analysis is conducted on a fixed hardware configuration, employing NVIDIA H200 GPUs. The AI Energy Score is introduced and proposed as a benchmark to evaluate different model-language combinations. This metric combines, under a single formulation, the performance scored through a series of reasoning tasks, as well as the energy consumption during the inference process, allowing for a bidimensional comparison of LLMs.

2. *Multilingual evaluation of vision-language models*

An experimental framework is developed to evaluate multimodal systems in visual perception tasks applied to autonomous navigation. The study analyzes whether the linguistic tendencies observed in language models carry over to visual question answering tasks. In this case, instead of formulating purely textual problems,

the queries introduce a visual component. As such, the experiment evaluates the capacity of different models to respond to questions formulated from an input image. The dataset employed consists of a collection of images of driving scenarios from BDD100K [2]. In this way, the evaluation is framed within an autonomous navigation context, employing OneMap as a reference.

3. Language-conditioned visual grounding

This experiment studies the influence of language on visual grounding in CLIP architectures. This approach evaluates the ability of different models and languages to relate spatial and semantic information. The same set of languages and images is employed in the evaluation, assessing whether identical queries formulated in different languages can not only identify a visual concept within an image, but also position it in similar regions of the scene. In this study, the concepts addressed in previous experiments are brought into the specific context of OneMap through the analysis of a semantic similarity mapping process.

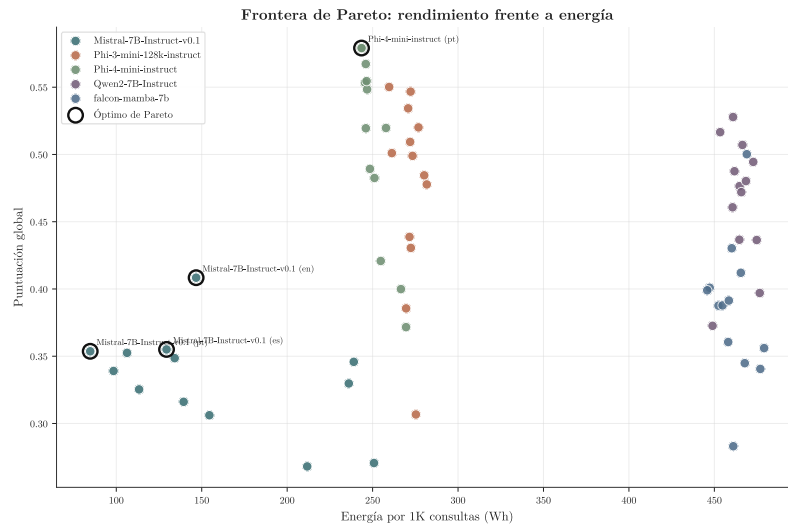
Summary of the experimental blocks developed.

Block	Models	Input	Metrics
Multilingual LLMs	5 LLMs	Reasoning problems in 13 languages	Performance, energy, and <i>AI Energy Score</i>
Multilingual VLMs	5 VLMs	BDD100K images and VQA questions	Performance and energy
Multilingual visual grounding	2 multilingual CLIP models	Images, visual concepts, and languages	Spatial alignment and peak activation

3. Results

Owing to the number of experiments conducted and the variety of metrics evaluated, this study has produced a broad set of results that is particularly illustrative. The experiments have evaluated a wide range of dimensions, such as performance across different task types, the use of computational resources, energy efficiency, response consistency, and the latency of the inference processes. Beyond this, the variety of technologies studied allows for a global comparison of model behavior across languages and multimodal settings.

The experiments show that changing the language of a query modifies the generated response, altering both its length and content; modifies the inference process, affecting latency and power consumption; and modifies visual interpretation, conditioning both



Performance and energy cost in the multilingual evaluation of LLMs.

semantic feature alignment and activation levels. The obtained results reflect the existence of a linguistic dependence in the AI architectures examined.

4. Conclusions

Language as a technically relevant variable

The language in which a query is formulated affects both the quality and the content of the generated response. This behaviour has been observed in both language and vision-language models, where performance in reasoning tasks and visual question answering tends to degrade with lexical distance from English. In particular, low-resource languages such as Basque, Luxembourgish, and Arabic stand out for obtaining the poorest results [3, 4].

The computational cost of linguistic variance

Throughout the experiments, the power consumption associated with the inference processes is seen to be affected by the language in which the query is formulated. Nevertheless, there does not seem to be a specific global tendency, as linguistic dependence varies depending on the task performed. Rather than language, the determining factor in the variation of energy cost seems to be the model and architecture employed [3, 4].

Visual grounding under linguistic variability

In visual grounding tasks, language does not seem to visibly affect the global identification—or activation—of the different concepts present in an image. Instead, it influences the spatial localization of those concepts within the image, suggesting the existence of a language-conditioned spatial misalignment. These results are consistent across architectures of different sizes, measured by the number of visual parameters [5].

5. References

- [1] Finn Lukas Busch, Timon Homberger, Jesús Ortega-Peimbert, Quantao Yang y Olov Andersson. *One Map to Find Them All: Real-time Open-Vocabulary Mapping for Zero-shot Multi-Object Navigation*. 3 de mar. de 2025. DOI: [10.48550/arXiv.2409.11764](https://doi.org/10.48550/arXiv.2409.11764), arXiv: [2409.11764\[cs.R0\]](https://arxiv.org/abs/2409.11764). URL: <http://arxiv.org/abs/2409.11764> (visitado 29-06-2026).
- [2] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan y Trevor Darrell. *BDD100K: A Diverse Driving Dataset for Heterogeneous Multitask Learning*. 8 de abr. de 2020. DOI: [10.48550/arXiv.1805.04687](https://doi.org/10.48550/arXiv.1805.04687), arXiv: [1805.04687\[cs.CV\]](https://arxiv.org/abs/1805.04687). URL: <http://arxiv.org/abs/1805.04687> (visitado 03-07-2026).
- [3] I. de Zarzà, Mauro Liz, J. de Curtò y Carlos T. Calafate. «Energy-Aware Multilingual Evaluation of Large Language Models». En: *Electronics* 15.7 (2026). ISSN: 2079-9292. DOI: [10.3390/electronics15071395](https://doi.org/10.3390/electronics15071395). URL: <https://www.mdpi.com/2079-9292/15/7/1395>.
- [4] J. de Curtò, Mauro Liz, I. de Zarzà y Carlos T. Calafate. «Energy-Aware Multilingual Vision–Language Models for Drone Smart Sensing». En: *Drones* 10.5 (2026). ISSN: 2504-446X. DOI: [10.3390/drones10050361](https://doi.org/10.3390/drones10050361). URL: <https://www.mdpi.com/2504-446X/10/5/361>.
- [5] J. de Curtò, Mauro Liz e I. de Zarzà. *Language-Conditioned Visual Grounding with CLIP Multilingual*. 9 de mayo de 2026. DOI: [10.48550/arXiv.2605.09060](https://doi.org/10.48550/arXiv.2605.09060), arXiv: [2605.09060\[cs.CL\]](https://arxiv.org/abs/2605.09060). URL: <http://arxiv.org/abs/2605.09060> (visitado 07-07-2026).

Agradecimientos

Aprovecho estas líneas para expresar mi más sincero agradecimiento al Dr. Joaquín de Curtò i Díaz. Durante el desarrollo de este trabajo, he podido contar con su valiosa orientación, su dedicación, su esfuerzo constante y su ayuda a lo largo de todo el año. Sin lugar a duda, el presente Trabajo de Fin de Grado no habría sido posible sin su apoyo.

Me gustaría también expresar mi agradecimiento a mi universidad, la Escuela Técnica Superior de Ingeniería ICAI. Los conocimientos adquiridos durante el grado, los profesores que me han acompañado en este camino y los compañeros con los que he compartido estos años han contribuido de manera decisiva a mi desarrollo personal y profesional. El presente trabajo es también fruto de todo ese aprendizaje.

Quisiera agradecer además a Boston University y, en particular, al Departamento de Ingeniería Eléctrica e Informática, por haberme acogido como estudiante de intercambio durante este año. El presente trabajo tampoco habría sido posible sin su colaboración.

Finalmente, quiero dar las gracias a mi familia: a mis padres y a mis tres hermanos. Gracias por su apoyo incondicional, por su esfuerzo incansable y por haberme acompañado en cada paso de este camino. La persona que soy hoy refleja los valores, el cariño y la confianza que me han transmitido. Gracias por darme la oportunidad de estudiar y de crecer.

Índice general

1. Introducción y planteamiento	1
1.1. Contextualización del problema	1
1.1.1. Búsqueda de objetos en entornos desorganizados	1
1.1.2. La evolución de los sistemas autónomos	1
1.1.3. Percepción geométrica y percepción semántica	3
1.1.4. Modelos de lenguaje y visión-lenguaje	5
1.1.5. Percepción visual en sistemas autónomos	6
1.1.6. Retos	8
1.2. Motivación	9
1.3. Planteamiento del problema	9
1.4. Objetivos del trabajo	10
1.5. Contribuciones principales	11
1.6. Organización de la memoria	12
2. Descripción de las tecnologías	13
2.1. Navegación autónoma clásica	13
2.2. Modelos de lenguaje de gran tamaño	18
2.2.1. Tokenización y representación del texto	18
2.2.2. Inferencia autoregresiva y ventana de contexto	20
2.2.3. Diferencias entre modelos	22
2.2.4. Multilingüismo en modelos de lenguaje	23
2.2.5. La arquitectura Transformer	26
2.2.6. Evaluación de modelos de lenguaje: benchmarks	28
2.3. Modelos visión-lenguaje	30
2.4. OneMap	35
2.4.1. Descripción conceptual de OneMap	36
2.4.2. Metodología	37
2.4.3. Búsqueda de objetos	45
2.4.4. Resultados	47

3. Descripción de los modelos desarrollados	51
3.1. Evaluación multilingüe de rendimiento y energía en LLMs	51
3.1.1. Introducción	51
3.1.2. Configuración experimental	53
3.2. Evaluación multilingüe de rendimiento y energía en VLMs	57
3.2.1. Introducción	57
3.2.2. Configuración experimental	58
3.3. Evaluación completamente multilingüe de rendimiento y energía en	
VLMs	65
3.3.1. Introducción	65
3.3.2. Cambios realizados	66
3.4. Evaluación del grounding visual multilingüe en el contexto de OneMap	69
3.4.1. Configuración experimental	70
4. Resultados y discusión	75
4.1. Evaluación multilingüe de rendimiento y energía en LLMs	76
4.2. Evaluación multilingüe de rendimiento y energía en VLMs	83
4.3. Evaluación del grounding visual multilingüe en el contexto de OneMap	90
4.4. Discusión global de resultados	95
5. Conclusiones	99
5.1. Hallazgos principales	99
5.2. Contribuciones	100
5.3. Limitaciones	101
5.4. Líneas de investigación futuras	103
5.5. Publicaciones derivadas	105
A. Alineación con los ODS	107
B. Resultados detallados de la evaluación multilingüe de LLMs	109
C. Resultados detallados de la evaluación multilingüe de VLMs	117
D. Resultados detallados de la evaluación del grounding visual mul-	
tilingüe en el contexto de OneMap	133
Bibliografía	137

Índice de figuras

1.1. George Devol y Unimate [1]	2
2.1. Ciclo general de navegación de un agente autónomo.	14
2.2. Comparación entre una lectura LiDAR ideal y una lectura ruidosa.	14
2.3. Ejemplo esquemático del proceso de localización mediante filtro de partículas.	16
2.4. Representación distintos <i>tokens</i> en un espacio multidimensional. Los <i>tokens</i> con significados similares aparecen próximos entre sí.	27
2.5. Ejemplos de pares imagen-texto	30
2.6. Esquema de preentrenamiento contrastivo de CLIP. Fuente: [31].	31
2.7. Representación esquemática de las combinaciones entre vectores de imagen y vectores de texto. Cada entrada de la matriz contiene el vector asociado a la imagen de la fila y el vector asociado a la descripción textual de la columna.	32
2.8. Representación geométrica del ángulo entre dos embeddings.	33
2.9. Ejemplo de imagen representada como una cuadrícula de 5×5 píxeles, donde cada celda contiene su valor RGB.	38
2.10. Imagen RGB y mapa de profundidad correspondiente del dataset NYU [34].	39
2.11. Imagen dividida en una malla regular de <i>patches</i> .	40
2.12. Representación de un plano de vivienda antes y después de su discretización espacial.	41
2.13. Esquema general del proceso de construcción del mapa semántico en <i>OneMap</i> . Reproducido de [32].	44
2.14. Mapas observado $\mathcal{O}$, semánticamente explorado $\mathcal{E}$ y no navegable $\sim \mathcal{N}$, junto con la frontera δ entre $\mathcal{E}$ y $\mathcal{O}$. Reproducido de [32].	46
2.15. Comparación del SPL obtenido por VLFM y <i>OneMap</i> en función del número de objetos.	49
3.1. Ejemplo de imagen de conducción extraída del conjunto de datos BDD100K [41].	59

4.1. Comparación conjunta de rendimiento y consumo energético por modelo e idioma en la evaluación multilingüe de LLMs.	78
4.2. Frontera de Pareto entre rendimiento y consumo energético en la evaluación multilingüe de LLMs.	79
4.3. Relación entre la longitud media de salida y el consumo energético normalizado por mil consultas.	81
4.4. Relación entre la latencia media por consulta y el consumo energético normalizado por mil consultas.	82
4.5. Comparación del rendimiento por modelo e idioma entre la evaluación multilingüe inicial y la evaluación completamente multilingüe.	87
4.6. Comparación del consumo energético por modelo e idioma entre la evaluación multilingüe inicial y la evaluación completamente multilingüe.	88
4.7. Comparación del compromiso entre rendimiento y consumo energético entre la evaluación multilingüe inicial y la evaluación completamente multilingüe.	89
4.8. Comparación del IoU de máscara de clúster por idioma para los dos modelos CLIP multilingües evaluados.	92
4.9. Relación entre la activación máxima y el IoU de máscara de clúster por idioma y modelo.	93
4.10. IoU de máscara de clúster por idioma y tipo de consulta para los dos modelos CLIP multilingües evaluados.	94
B.1. Distribución general del rendimiento.	111
B.2. Rendimiento y consistencia por modelo e idioma.	112
B.3. Métricas de generación y coste computacional por modelo e idioma.	113
B.4. Rendimiento según la dificultad de las preguntas.	114
B.5. Rendimiento desagregado por tipo de pregunta.	115
C.1. Comparación de la distribución del rendimiento por idioma.	121
C.2. Comparación de la distribución del rendimiento por modelo.	122
C.3. Comparación de la distribución del consumo energético por idioma.	123
C.4. Comparación de la distribución del consumo energético por modelo.	124
C.5. Comparación de la diferencia de rendimiento respecto al inglés.	125
C.6. Comparación de la diferencia de consumo energético respecto al inglés.	126
C.7. Comparación de la distribución del rendimiento por familia lingüística.	127
C.8. Comparación de la distribución del consumo energético por familia lingüística.	128
C.9. Comparación de la distribución del rendimiento por tipo de tarea.	129
C.10. Comparación de la brecha máxima-mínima entre idiomas por tipo de tarea y modelo.	130

C.11.Comparación del ranking medio de eficiencia por modelo.	131
C.12.Comparación del perfil de rendimiento por tipo de tarea y modelo.	132
D.1. Consumo energético normalizado por modelo.	133
D.2. Activación máxima por idioma y modelo.	134
D.3. Activación máxima por concepto visual e idioma para las variantes	
XLM-R base y XLM-R large.	135
D.4. Análisis del grounding visual según disponibilidad de recursos y	
familia lingüística.	136

Índice de cuadros

1.1. Comparación de percepción geométrica y percepción semántica. . .	4
2.1. Número de artículos disponibles en distintas ediciones lingüísticas de Wikipedia. Elaboración propia a partir de las estadísticas de Meta-Wiki [27].	24
2.2. Distribución reducida de datos por idioma en el corpus CC-100. Elaboración a partir de Conneau et al. [28].	25
2.3. Resultados de navegación de un único objeto.	48
2.4. Resultados de navegación multiobjeto.	48
3.1. Especificaciones principales de los modelos evaluados.	53
3.2. Especificaciones principales de los modelos visión-lenguaje evaluados.	58
3.3. Preguntas formuladas para la evaluación VQA.	60
3.4. Especificaciones principales de los modelos evaluados.	70
4.1. Resumen conjunto de rendimiento y consumo energético para los LLMs evaluados. Los valores se muestran como media $\pm$ desviación estándar a través de los idiomas considerados y de las iteraciones realizadas.	76
4.2. Resumen conjunto de rendimiento y consumo energético para los idiomas evaluados. Los valores se muestran como media $\pm$ desviación estándar a través de los modelos considerados y de las iteraciones realizadas.	77
4.3. Resumen conjunto de rendimiento y consumo energético para los LLMs evaluados. Los valores se muestran como media $\pm$ desviación estándar a través de los idiomas considerados y de las iteraciones realizadas.	81
4.4. Resumen global de rendimiento y consumo energético por modelo VLM en la evaluación multilingüe inicial; en negrita se indica el mejor valor de cada columna.	83

4.5. Resumen global de rendimiento y consumo energético por modelo VLM en la evaluación completamente multilingüe; en negrita se indica el mejor valor de cada columna.	83
4.6. Resumen global de rendimiento y consumo energético por idioma en las dos evaluaciones multilingües; en negrita se indica el mejor valor de cada columna.	85
4.7. Resumen global de métricas de grounding visual por modelo. Las métricas se promedian sobre los idiomas distintos del inglés, conceptos e imágenes evaluados; en negrita se indica el mejor valor de cada columna.	90
4.8. Resumen global de métricas de grounding visual por idioma, excluyendo el inglés como idioma de referencia; en negrita se indica el mejor valor de cada columna.	91
B.1. Resultados detallados por modelo e idioma.	109
C.1. Consumo energético por modelo e idioma para la versión original del experimento VLM.	117
C.2. Consumo energético por modelo e idioma para la versión actualizada del experimento VLM.	117
C.3. Puntuación media por modelo e idioma para la versión original del experimento VLM.	118
C.4. Puntuación media por modelo e idioma para la versión actualizada del experimento VLM.	118
C.5. Puntuación media por tipo de tarea e idioma, promediada entre modelos, para la versión original del experimento VLM.	119
C.6. Puntuación media por tipo de tarea e idioma, promediada entre modelos, para la versión actualizada del experimento VLM.	119
C.7. Puntuación media por tipo de tarea y modelo para la versión original del experimento VLM.	120
C.8. Puntuación media por tipo de tarea y modelo para la versión actualizada del experimento VLM.	120

Capítulo 1

Introducción y planteamiento

1.1. Contextualización del problema

1.1.1. Búsqueda de objetos en entornos desorganizados

“Busca la cocina y recoge la taza roja que está sobre la encimera”. Una instrucción como esta resulta muy natural para una persona, que puede interpretarla casi instintivamente y transformarla, con poco esfuerzo, en una secuencia ordenada de acciones. A modo de ejemplo, y a grandes rasgos, la tarea se podría descomponer en los siguientes pasos: encontrar, en primer lugar, la cocina, situar después la encimera y recoger finalmente la taza roja.

Considérese ahora el caso en que, en lugar de una persona, fuese un robot o agente autónomo el responsable de realizar la misma tarea. Incluso dejando de lado la actuación necesaria por parte del agente para poder manipular físicamente la taza (que a su vez ya presenta un reto de ingeniería importante), ¿cómo debe el robot encontrar la cocina?, ¿cómo debe desplazarse de manera eficiente hasta ella?, ¿cómo debe situar la encimera al llegar a la cocina?, y finalmente ¿cómo debe identificar la taza a partir de la descripción que se le ha proporcionado en la instrucción?

Tareas como la búsqueda de objetos en un entorno desorganizado, aparentemente triviales para un ser humano, adquieren un grado de complejidad adicional al ser planteadas para un sistema autónomo.

1.1.2. La evolución de los sistemas autónomos

Quizás el lector reconozca que la adopción de sistemas autónomos ha aumentado de manera pronunciada esta última década. Ciertamente, a día de hoy, estos

CAPÍTULO 1. INTRODUCCIÓN Y PLANTEAMIENTO

sistemas están desplegados en una gran variedad de entornos y abarcan una inmensidad de tareas con distintos niveles de complejidad. No obstante, cabe recalcar que aunque ciertas aplicaciones recientes pueden considerarse emergentes, la robótica industrial no es precisamente una tecnología novedosa. De hecho, los sistemas autónomos se han consolidado en la industria moderna después de varias décadas de investigación y desarrollo. En este ámbito, se considera que la implantación de robots en entornos industriales data de 1961, con *Unimate* (Figura 1.1).

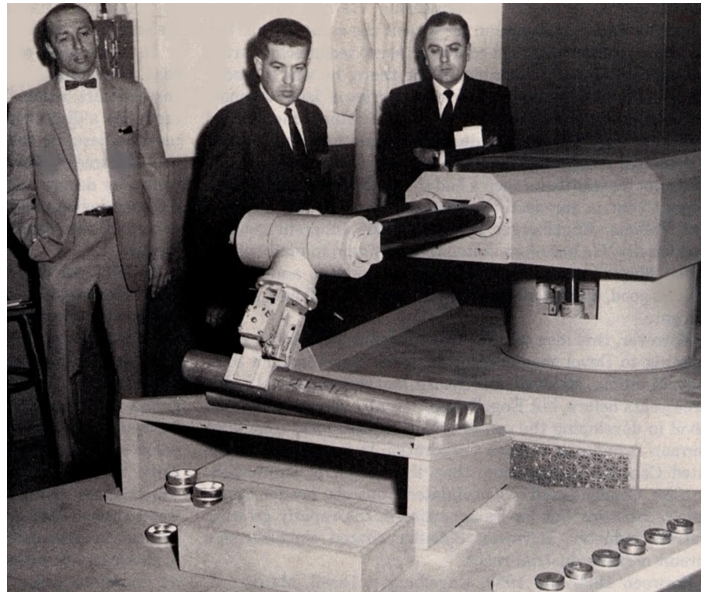


Figura 1.1: George Devol y Unimate [1]

Unimate fue diseñado por George Devol en 1953 y entró en operación en una línea de montaje de General Motors en la fábrica Inland Fisher Guide. *Unimate* se suele citar como uno de los primeros robots industriales y su función consistía en manipular piezas metálicas calientes procedentes de procesos de fundición a presión [2]. En este caso, la manipulación de tales piezas representaba una tarea repetitiva, muy estructurada y realizada en un entorno controlado. Se trataba de una tarea quizás un tanto peligrosa para un operario, pero que, dado su carácter reiterativo y parametrizable, resultaba adecuada para un agente robótico.

Si bien es cierto que las tecnologías robóticas y los sistemas autónomos han evolucionado considerablemente desde entonces, también lo ha hecho la complejidad de las tareas exigidas a estos sistemas. A diferencia de agentes tempranos como *Unimate*, a día de hoy, muchas aplicaciones de la robótica se desarrollan en

entornos desorganizados, abiertos y dinámicos. Es frecuente, además, encontrar procesos en los que agentes modernos deban interpretar instrucciones expresadas en lenguaje natural y, en muchos casos, también realizar cierto grado de razonamiento autónomo. Un robot aspirador doméstico o un vehículo con funciones de conducción autónoma surgen como ejemplos claros de casos de uso modernos de estas tecnologías.

1.1.3. Percepción geométrica y percepción semántica

A medida que aumenta la complejidad de las tareas exigidas a ciertos agentes autónomos, surge la necesidad de adaptar los algoritmos que rigen el comportamiento de estos sistemas. Los primeros robots industriales como *Unimate* no tenían una percepción del entorno en el que se desenvolvían. Estos sistemas estaban diseñados para trabajar en contextos previamente definidos y muy restringidos, en los cuales el agente se limitaba a repetir una secuencia de acciones de manera sucesiva. Este enfoque suele ser adecuado en aplicaciones suficientemente estructuradas, dando lugar a resultados predecibles y fiables.

La necesidad de percepción en sistemas autónomos se entiende fácilmente, considerando, una vez más, el ejemplo de un robot aspirador doméstico. Se parte de la razonable suposición de que el objetivo de este agente es recorrer una habitación a la vez que succiona polvo. Supóngase, además, que el agente no tiene previamente una reconstrucción digital de la habitación, es decir, que se encuentra en un entorno desconocido. En este contexto, el aspirador ha de recorrer la habitación entera mientras aspira y, en la medida de lo posible, evitar los obstáculos que se encuentre por el camino. Nótese la diferencia respecto a tareas tempranas como las que le correspondían a *Unimate*. En el caso de la aspiradora, se requiere que el agente sea capaz de tomar decisiones a partir de la información disponible y no simplemente recrear una secuencia de pasos predefinida. Para que el aspirador se pueda desenvolver en un entorno desorganizado y desconocido, tiene que ser capaz de percibir la escena en la que se encuentra. Para ello, en muchos casos, la información a partir de la cual el robot puede planificar y llevar a cabo su tarea consiste, principalmente, en distancias, bordes y obstáculos. En este sentido, se puede hablar de una percepción geométrica del entorno.

La percepción geométrica abarca ciertos tipos de características espaciales que pueden ser recopiladas por el agente autónomo. En particular, se centra en distancias, ángulos, posiciones y movimientos. Información a partir de la cual el agente puede generar progresivamente un mapa del entorno que le rodea y a través del cual formular una trayectoria. Este mapa se desarrolla a medida que el sistema, a

través de los sensores correspondientes, infiere la posición de obstáculos presentes en la escena. A grandes rasgos, la percepción geométrica trata de responder a la pregunta: ¿por dónde puedo desplazarme? Ya luego la cuestión: ¿por dónde debo desplazarme?, comprende otra tarea ligeramente distinta en la que ya entrarían los algoritmos de planificación de trayectoria.

La percepción geométrica es un planteamiento suficientemente robusto para muchas de las aplicaciones estructuradas de agentes autónomos. Existen, sin embargo, ciertas tareas en las que, debido a su complejidad o naturaleza, estos métodos de interpretación del entorno no resultan adecuados. En el caso de sistemas de conducción autónoma, es frecuente que la identificación de distancias, ángulos, posiciones y movimientos sea útil, pero no suficiente para planificar maniobras seguras de conducción en entornos reales[3]. Aquí, lo verdaderamente práctico sería la capacidad de, a partir de estas mediciones, poder identificar vehículos, señales de tráfico, vías de circulación, peatones, etc. El problema ya no es simplemente que el robot sepa dónde están los bordes (u obstáculos) para decidir por dónde efectuar sus maniobras, sino que al mismo tiempo comprenda qué representa el entorno en el que se está desenvolviendo con el fin de así tomar decisiones orientadas a completar un objetivo de manera eficiente y segura. En este caso, se puede hablar de una percepción semántica del entorno, donde el agente trabaja ya con objetos, escenas, relaciones y objetivos, tratando de responder a “¿qué hay en la escena?”.

A modo de comparación, se recogen en el Cuadro 1.1 las principales diferencias entre la percepción geométrica y la percepción semántica. Cabe destacar que estas técnicas de percepción del entorno no son excluyentes, y en muchos casos coexisten para proporcionar al agente una visión más general de la escena.

Cuadro 1.1: Comparación de percepción geométrica y percepción semántica.

Percepción geométrica	Percepción semántica
Trabaja con distancias, bordes, ángulos	Trabaja con objetos y relaciones
Responde a “¿dónde están los obstáculos y por dónde puedo moverme?”	Responde a “¿qué hay en la escena y que significado tiene para la tarea?”
Útil para evitar un obstáculo o recorrer una habitación	Útil para identificar y categorizar instancias de objetos

1.1.4. Modelos de lenguaje y visión-lenguaje

Dado que a lo largo de este trabajo se hará referencia de forma recurrente a modelos de lenguaje y modelos visión-lenguaje, conviene introducir brevemente ambos conceptos. A fecha de redacción de esta memoria, los modelos de lenguaje están tan integrados en la sociedad que resulta razonable asumir que el lector ha interactuado con ellos o, al menos, ha escuchado hablar de ellos. No obstante, sin entrar en detalles, se destaca el principio de funcionamiento de los mismos.

Considérese el guión de una película describiendo la interacción entre una persona y su asistente de inteligencia artificial. El guión contiene el diálogo de la persona, pero lamentablemente, el fragmento de la respuesta del asistente es ilegible. Supóngase además que se dispone de una máquina capaz de procesar cualquier texto y proporcionar una predicción razonable de la palabra siguiente. Uno podría continuar el guión proporcionándoselo a la máquina y anotando su predicción de la primera palabra de la respuesta del asistente. Sucesivamente, uno podría volver a introducir el guión hasta completarlo, palabra por palabra. [4]

Un modelo de lenguaje se puede entender como una función matemática que, al igual que la máquina del ejemplo anterior, es capaz de predecir la palabra que le sucede a un fragmento de texto [5]. Los modelos aprenden a hacer estas predicciones procesando cantidades inmensas de texto, en un proceso conocido como entrenamiento. Si se considera el caso de un *chatbot* como ejemplo posiblemente familiar para el lector, lo que más o menos está ocurriendo a la hora de producir una respuesta es lo siguiente. Una plataforma prepara un cuerpo de texto que indica que los contenidos describen la interacción entre una persona y su asistente de inteligencia artificial hipotético. A esto se le suma la petición (o *prompt*) del usuario como primera parte de dicha interacción y se le pide al modelo predecir repetidamente la palabra que dicho asistente hipotético respondería. Esta secuencia de predicciones se le presenta al usuario como respuesta. Conviene matizar que estos modelos, en lugar de predecir una única palabra con certeza, le asignan una probabilidad a todas las palabras candidatas a ser la siguiente. En este caso, la respuesta tiende a ser más natural si se le permite al modelo seleccionar de vez en cuando palabras menos probables. Esto significa que, aunque el modelo tenga pesos deterministas, el proceso de muestreo sobre la distribución de probabilidad de la siguiente palabra puede hacer que un mismo *prompt* dé lugar a respuestas distintas. [4]

De manera similar a como un modelo de lenguaje se puede emplear para un *chatbot*, también puede utilizarse para analizar un documento de texto. En este caso, el documento se procesa junto con la consulta del usuario y se introduce

como entrada al modelo de lenguaje. Cabe mencionar que el proceso real es más complejo que esta descripción simplificada, en el cual intervienen matices como la preparación, *tokenización* y codificación de datos en un formato que el modelo pueda procesar. Pero esto excede el alcance de la sección y se tratará debidamente en capítulos posteriores.

¿Qué sucedería si, en lugar de únicamente texto, el documento incluyese también imágenes? Un modelo de lenguaje textual no puede procesar imágenes de la misma manera que puede procesar texto y, por lo tanto, no podría acceder a toda la información del documento. En este caso, se precisaría de un modelo multimodal capaz de interpretar tanto texto como imágenes. Los modelos de visión-lenguaje (*Vision-Language Models* o VLMs) son herramientas diseñadas para interpretar entradas de texto e imagen y redactar una respuesta en lenguaje natural. Los VLMs permiten llevar a cabo un amplio abanico de tareas. Una de ellas es la respuesta visual a preguntas (*Visual Question Answering*, VQA), en la que, ante una imagen y una pregunta, el modelo logra generar una respuesta coherente en lenguaje natural. En estos contextos, los VLMs no se limitan a procesar la imagen como una matriz de píxeles, sino que son capaces de distinguir objetos, formas y relaciones espaciales. [6]

1.1.5. Percepción visual en sistemas autónomos

Aunque a lo largo de la introducción se han propuesto ya algunas aplicaciones de percepción visual en sistemas autónomos, esta sección recoge una serie de casos de uso representativos con el fin de evidenciar su relevancia práctica en aplicaciones actuales.

Waymo Driver

Waymo Driver es un sistema de conducción autónoma elaborado por la empresa estadounidense *Waymo*. Esta herramienta, mediante una multitud de sensores, permite a sus vehículos recopilar datos complejos del entorno. Dichos datos son procesados por algoritmos de inteligencia artificial para interpretar la escena que rodea al vehículo, identificando peatones, ciclistas, zonas de obras y otros vehículos. *Waymo Driver* también es capaz de reaccionar a señalizaciones de tráfico, como semáforos o señales de stop. [7]

Zipline

Zipline, otra empresa estadounidense, se centra en el diseño y fabricación de drones de reparto. Los drones cumplen funciones de transporte de alimentos, suministros médicos y otros bienes de forma energéticamente eficiente y a largas

distancias. La tecnología de navegación de estos sistemas les permite tomar la ruta más rápida a su destino evitando obstáculos a lo largo del camino. [8]

Proteus

Proteus es un robot móvil completamente autónomo desarrollado por *Amazon Robotics*. Estas unidades están desplegadas en determinados centros logísticos de *Amazon* en Estados Unidos y participan en el transporte de carros con paquetes hacia muelles de carga. La arquitectura de sensores de *Proteus* facilita la percepción e interpretación de su entorno, lo que les permite navegar libremente por su espacio de trabajo, detectando y evitando obstáculos a su alrededor, ya sean otros robots, carretillas o incluso personas. [9]

Perseverance Rover

El *Perseverance* es un vehículo explorador de la NASA en Marte cuya misión es buscar señales de vida microbiana. El rover almacena muestras de rocas marcianas para que puedan ser recuperadas en futuras misiones y traídas a la Tierra. El *Perseverance* está equipado con un conjunto de sensores y cámaras, esenciales para su navegación y control. Un ejemplo claro es el sistema de cámaras de aterrizaje, que tomó imágenes de la superficie marciana mientras el rover descendía en paracaídas. Estas imágenes fueron utilizadas por el agente para escoger, de forma autónoma, la zona de toma de contacto dentro del área de aterrizaje especificada. Otro ejemplo es el conjunto de cámaras de detección de peligros, denominadas *Hazard Detection Cameras* o *HazCams*, que permiten detectar obstáculos como rocas, grietas o dunas de arena. Estas cámaras le otorgan una visión tridimensional al rover, permitiéndole tomar sus propias decisiones acerca de la ruta sin necesidad de consultar cada movimiento con el equipo en la Tierra. [10, 11]

Como se ha mostrado, existen numerosas aplicaciones actuales de agentes autónomos que requieren un sistema de visión para percibir su entorno y desenvolverse en él. En particular, esta sección se ha centrado en ejemplos en los que los sistemas de visión se emplean para informar tareas de navegación. Este ámbito constituye una línea de investigación especialmente relevante en los últimos años, al situarse en la intersección entre la inteligencia artificial y la visión artificial. En este contexto, los modelos de visión-lenguaje constituyen un campo de estudio prometedor, ya que pueden facilitar la integración de información visual en los procesos de interpretación, planificación y toma de decisiones autónoma [12, 13].

1.1.6. Retos

Los sistemas autónomos modernos no solo necesitan detectar obstáculos, sino entender escenas y objetivos expresados en lenguaje natural. Esta es la idea principal que se ha tratado de enmarcar a lo largo de la introducción. Y, a medida que surgen técnicas como la percepción semántica y tecnologías como los modelos de visión-lenguaje para afrontar la complejidad creciente de las tareas de navegación, aparecen con ellas nuevos retos.

Se ha argumentado que la percepción semántica tiende un puente entre la recopilación de datos espaciales y la interpretación de lo que representan dentro de una escena. Se puede decir, que esta estrategia busca asignar significado y contextualizar la información visual de un entorno. Este enfoque es, a bajo nivel, independiente del idioma. A pesar de esto, si se consideran tareas de VQA, donde las consultas sobre una imagen se generan en lenguaje natural, la percepción semántica sí puede verse condicionada por el idioma. En estos casos, el idioma pasa de ser una variable externa, a formar parte del proceso de interpretación. Es decir, una imagen podría ser asociada a conceptos distintos, con un grado de confianza variable según el idioma en el que se expresase la consulta. Los idiomas en los que estos modelos de visión-lenguaje han sido entrenados, así como el idioma en el que se formulan las consultas en lenguaje natural, representan por tanto un reto a la hora de producir resultados fiables y consistentes. Esto es particularmente relevante en aplicaciones de sistemas autónomos (como puede ser la navegación) ya que dependen de dichas salidas para informar sus decisiones.

Otra cuestión, también importante en este tipo de aplicaciones, es el consumo energético de los modelos de visión-lenguaje [14]. Si se consideran los casos de uso detallados en la sección anterior, puede observarse que muchos de ellos consisten en plataformas móviles de tamaño limitado sin conexión externa por cable. Por tanto, su alimentación suele depender de una batería integrada en el propio agente. Esto implica que el consumo de dichos modelos no puede ser elevado. De lo contrario, se tendrían que considerar mayores requisitos de batería, un aumento en su tamaño y peso o incluso una reducción significativa del tiempo de operación. Así, el consumo energético de los modelos pasa también a constituir un reto importante en aplicaciones basadas en agentes autónomos.

1.2. Motivación

Aunque los avances recientes han ampliado notablemente las capacidades de los sistemas autónomos, especialmente en tareas de navegación orientadas a la búsqueda de objetos, aún existen limitaciones relevantes en su evaluación y en su aplicabilidad práctica. En particular, la mayoría de los enfoques modernos centran su análisis en *benchmarks* basados en métricas de rendimiento orientadas a tareas específicas; es decir, el grado de acierto con el que un agente completa correctamente la tarea planteada. Esto permite comparar capacidades entre sistemas, pero ha relegado dimensiones clave como la eficiencia energética.

Por otra parte, las tareas de percepción visual, especialmente en aplicaciones de navegación autónoma, se apoyan cada vez más en técnicas de inteligencia artificial. Estos agentes suelen emplear cámaras para captar imágenes, procesarlas y poder entender su entorno. En estos contextos, los modelos de visión-lenguaje emergen como herramientas eficaces para relacionar información visual del entorno con consultas semánticas expresadas en lenguaje natural [15]. Debido a esto, el idioma en el que se formulan dichas consultas deja de ser un simple medio de comunicación con el usuario para convertirse en un factor que interviene en la interpretación del sistema. Se plantea, por tanto, una situación en la que el idioma afecta a los resultados de las tareas desarrolladas por agentes autónomos, con posibles implicaciones sobre el rendimiento, la consistencia de las respuestas y el coste energético asociado al procesamiento de las consultas.

Es precisamente en este contexto donde surge la motivación de este trabajo, que puede plantearse desde una doble vertiente. Por un lado, incorporar el consumo energético como una métrica adicional de evaluación; por otro, analizar el efecto del idioma en el rendimiento de los modelos.

1.3. Planteamiento del problema

El presente trabajo toma como referencia principal la metodología de *OneMap*¹, que ha actuado como marco orientador a lo largo de toda la investigación. De esta referencia surge el título de la presente memoria. Más que reproducir directamente el desarrollo experimental del artículo original, el trabajo se ha apo-

¹Probablemente, el concepto de *OneMap* se ha introducido de manera algo repentina. *OneMap* es una metodología presentada en un artículo que se ha empleado como referencia principal en este trabajo. En dicho artículo, los autores introducen una propuesta de mapeo espacial-semántico para la búsqueda de objetos. Esta metodología se presentará y abordará con mayor detalle en el Capítulo 2.

yado en *OneMap* como hilo conductor para delimitar el problema y motivar sus contribuciones. En particular, este marco orienta el estudio de los VLMs como componentes perceptivos en tareas de navegación semántica y, más adelante, el análisis del *grounding* visual condicionado por el idioma, procesos ambos centrales en el *pipeline* de *OneMap*.

A partir de este planteamiento, surgieron como líneas de trabajo especialmente prometedoras tanto el análisis del consumo energético como el estudio de la variabilidad lingüística, al representar brechas de investigación en la literatura existente. Se decidió profundizar en ellas, manteniendo *OneMap* como referencia constante y recuperando dicho vínculo de forma explícita en el último experimento.

Considerando esto, y en base a la motivación anterior, se plantea la necesidad de evaluar modelos de lenguaje y visión-lenguaje no solo a través de métricas tradicionales de rendimiento, sino incorporando también elementos como el multilingüismo y el coste energético y computacional asociado a su ejecución. Por lo tanto, el enfoque de este trabajo se puede encapsular en la siguiente pregunta de investigación:

¿En qué medida el idioma condiciona el rendimiento y la eficiencia energética de modelos de lenguaje y visión-lenguaje utilizados como componentes perceptivos en tareas de navegación semántica autónoma?

1.4. Objetivos del trabajo

Para abordar la pregunta de investigación, se establecen los siguientes objetivos:

1. Evaluar agentes basados en modelos de lenguaje (LLMs) en tareas multilingües, analizando el impacto del idioma tanto en su rendimiento como en su consumo energético.
2. Evaluar agentes basados en modelos de visión-lenguaje (VLMs) en tareas de interpretación de escenas e imágenes relacionadas con navegación, estudiando el efecto del idioma y su relación con métricas de rendimiento y eficiencia energética.
3. Analizar la aplicabilidad de este marco de evaluación — basado en rendimiento, variabilidad lingüística y eficiencia energética — a sistemas de razonamiento y percepción multimodal, tomando *OneMap* como caso de referencia de navegación semántica.

1.5. Contribuciones principales

1. *Evaluación multilingüe de modelos de lenguaje*

Se desarrolla un marco experimental para evaluar a los modelos de lenguaje de gran tamaño en tareas de razonamiento. Se puntúan distintas combinaciones de idioma y modelo según métricas de rendimiento y de coste computacional asociadas al proceso de inferencia.

2. *Incorporación del consumo energético como métrica de evaluación*

Se propone el *AI Energy Score* como *benchmark* de evaluación de modelos de lenguaje. Esta métrica integra tanto el rendimiento, puntuado a partir de una serie de tareas de razonamiento, como el consumo energético bajo una misma formulación, permitiendo comparaciones bidimensionales entre LLMs.

3. *Evaluación multilingüe de modelos de visión-lenguaje*

Se desarrolla un marco experimental para evaluar a sistemas multimodales en tareas de percepción visual aplicadas a la navegación autónoma. Se analiza si las tendencias lingüísticas observadas en modelos de lenguaje se extienden a tareas de VQA.

4. *Estudio del grounding visual condicionado por el idioma*

Se estudia la influencia del idioma en los resultados del *grounding* visual mediante arquitecturas CLIP. Esta contribución permite evaluar la capacidad de distintos modelos e idiomas para relacionar información espacial y semántica.

5. *Implicaciones para navegación semántica y OneMap*

Se traslada el marco de evaluación multimodal a procesos relacionados con la metodología de *OneMap*. En particular, los resultados obtenidos se vinculan con tareas de navegación semántica y extracción de características, permitiendo analizar cómo el idioma puede influir en el desempeño de estos procesos.

1.6. Organización de la memoria

El resto de esta memoria se organiza de la siguiente manera.

Capítulo 2. Descripción de las tecnologías: Se lleva a cabo una revisión del estado actual de la tecnología. Para ello, se abordan con mayor profundidad algunos de los conceptos tratados en el primer capítulo, como la navegación autónoma, los modelos de lenguaje y los modelos visión-lenguaje. Asimismo, se presenta la metodología *OneMap* como referencia principal de este trabajo y como ejemplo representativo de navegación semántica.

Capítulo 3. Descripción de los modelos desarrollados: Se describe la metodología empleada para abordar la pregunta de investigación. Este capítulo se organiza en cuatro secciones, correspondientes a distintas etapas del desarrollo experimental. Cada sección se apoya en la anterior, de manera que el capítulo presenta una evolución progresiva del trabajo. Asimismo, las secciones están ordenadas de forma cronológica según su implementación.

Capítulo 4. Resultados y discusión: Se presentan e interpretan los resultados obtenidos en cada uno de los experimentos realizados. El análisis permite valorar en qué medida dichos resultados responden a la pregunta de investigación planteada, así como discutir sus implicaciones en el contexto de la navegación autónoma.

Capítulo 5. Conclusiones: Se recogen los principales hallazgos de la investigación. Por otra parte, se incluyen recomendaciones para futuros estudios, posibles modificaciones de la experimentación y líneas de investigación derivadas de este trabajo.

Capítulo 2

Descripción de las tecnologías

2.1. Navegación autónoma clásica

En el primer capítulo de esta memoria, se ha insistido más de una vez en la relevancia de los sensores como medio de interacción entre el sistema autónomo y el mundo físico. Sin embargo, no se ha llegado a explicar el funcionamiento de los mismos, ni el tipo de información que proporcionan. Esta sección tampoco pretende hacerlo por completo: pues un análisis exhaustivo de los sensores empleados en tareas de navegación excedería el alcance de este trabajo. No obstante, con el objetivo de aportar mayor claridad al lector, se introduce brevemente un caso concreto: el sensor LiDAR. En este contexto, el LiDAR no se analiza como una solución absoluta, sino como un ejemplo representativo de las tecnologías empleadas habitualmente en navegación robótica.

LiDAR es el acrónimo de *Light Detection and Ranging*. Como indica su nombre, esta tecnología emplea pulsos de luz, normalmente emitidos por un láser, para estimar distancias a partir de la interacción de dichos pulsos con el entorno. [\[16\]](#)

Sensores como el LiDAR, integrados en la plataforma robótica, proporcionan al sistema una vía de acceso al entorno físico en forma de medidas. No obstante, en este punto, las mediciones obtenidas no dejan de ser precisamente eso: medidas. El siguiente paso consiste en extraer información útil a partir de ellas. Los datos procedentes de los sensores pueden emplearse para estimar la posición de objetos o incluso para construir un mapa del entorno. Más importantemente, también permiten que el robot se sitúe dentro de dicho mapa, es decir, que determine su propia posición y orientación. Una vez disponible esta información, el agente podría planificar una trayectoria desde su posición hasta una meta especificada. Para ello, se apoya en algoritmos de planificación de trayectorias, cuyo estudio

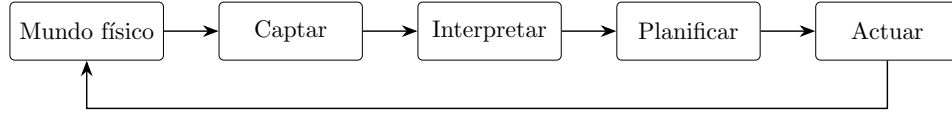


Figura 2.1: Ciclo general de navegación de un agente autónomo.

ha sido ampliamente tratado y discutido en la literatura de robótica. Como paso final, el agente ejecuta la trayectoria mediante el accionamiento de motores y actuadores, desplazándose físicamente por el entorno. A medida que se modifica la posición del robot, también cambia su percepción de la escena, dando lugar a nuevas medidas sensoriales [17]. El ciclo se repite, como se ilustra en la Figura 2.1.

Ahora que el lector dispone de una idea general del proceso de navegación autónoma, se introduce, a modo de ejemplo, uno de sus desafíos más importantes: la autolocalización. Este problema permite introducir algunas de las técnicas y tecnologías más relevantes en el ámbito de la robótica móvil. En primer lugar, se considera el caso en el que el agente se encuentra en una habitación de la cual tiene conocimiento previo. Es decir, cuenta con un mapa que representa la disposición de las paredes, los muebles y otros elementos relevantes del entorno.

Se considera que el robot dispone de un sensor LiDAR que proporciona medidas de la distancia a objetos dentro de su campo de visión. Estas distancias se pueden representar como una nube de puntos en el espacio. Por ejemplo, si el robot está situado frente a una esquina, la nube de puntos obtenida podría asemejarse a la mostrada en la Figura 2.2-i. Cabe recalcar, no obstante, que las medidas obtenidas de un sensor son susceptibles a ruido y otros errores de medición. El LiDAR no es una excepción. Por ello, ante esta misma esquina, las lecturas reales podrían mostrar cierta dispersión, como se ilustra en la Figura 2.2-ii.

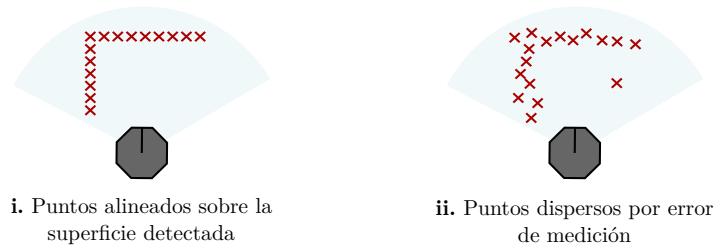


Figura 2.2: Comparación entre una lectura LiDAR ideal y una lectura ruidosa.

La nube de puntos obtenida puede compararse con distintas regiones del mapa de la habitación, restringiendo así el conjunto de posiciones en las que podría en-

contrarse el agente. Considerando de nuevo el ejemplo de la Figura 2.2, si el sensor detecta una esquina, la posición del robot deberá ser compatible con una zona del mapa en la que exista una geometría similar. Esta es la intuición general del proceso de autolocalización. No obstante, conviene profundizar algo más en cómo se lleva a cabo este procedimiento.

Para ello, se introduce el concepto de navegación por estima, también conocido como *dead reckoning*. A grandes rasgos, este método consiste en estimar la posición futura del robot a partir de una posición conocida previamente y de medidas relativas de movimiento (medidas odométricas), como la velocidad lineal y la velocidad angular. Estas magnitudes pueden estimarse a partir de sensores internos del robot, como los *encoders* de las ruedas, que hacen posible medir el giro de estas. Los datos de odometría le permiten al agente actualizar su posición estimada a partir del movimiento realizado. Por ejemplo, si el robot se encuentra orientado hacia el este y avanza a una velocidad constante de 1 m/s, tras 10 segundos su posición estimada será 10 metros al este de la posición inicial. Este tipo de estimación, basada en datos de odometría, puede resultar muy útil a corto plazo. No obstante, los errores asociados a estas medidas se acumulan con el tiempo, provocando que la posición estimada se desvíe progresivamente de la posición real del robot. [18, 19]

Se tienen, por lo tanto, dos maneras de estimar la localización del robot. Por una parte, se dispone de datos procedentes del sensor LiDAR que, aunque pueden estar afectados por ruido, permiten comparar la geometría observada con el mapa conocido de la habitación. Por otra parte, se dispone de datos de odometría, que permiten estimar el movimiento del robot a partir de sus sensores internos. Existen algoritmos, como el filtro de Kalman [20], que combinan estas dos fuentes de información imperfectas con el objetivo de obtener una estimación más robusta de la posición y orientación del robot. Sin embargo, en esta sección se describe otra técnica distinta, denominada filtro de partículas [18].

Considérese un entorno muy simple: una habitación vacía, delimitada por cuatro paredes y con una puerta en una de ellas. Este escenario se muestra en la parte superior de la Figura 2.3. El robot se encuentra dentro de esta habitación y su tarea consiste en determinar su propia posición dentro de la misma. Si en un principio no se contase con ningún tipo de información sensorial (Figura 2.3-i), el robot no tendría la capacidad de inferir su estado. Se podría decir, por lo tanto, que es igual de probable que esté situado en cualquier posición del entorno. Si, por el contrario, el sensor LiDAR indica la presencia de una superficie recta frente al robot (Figura 2.3-ii), el agente puede reducir el conjunto de posiciones posibles a aquellas desde las cuales se esperaría observar un muro delante de él. Además, la medida propor-

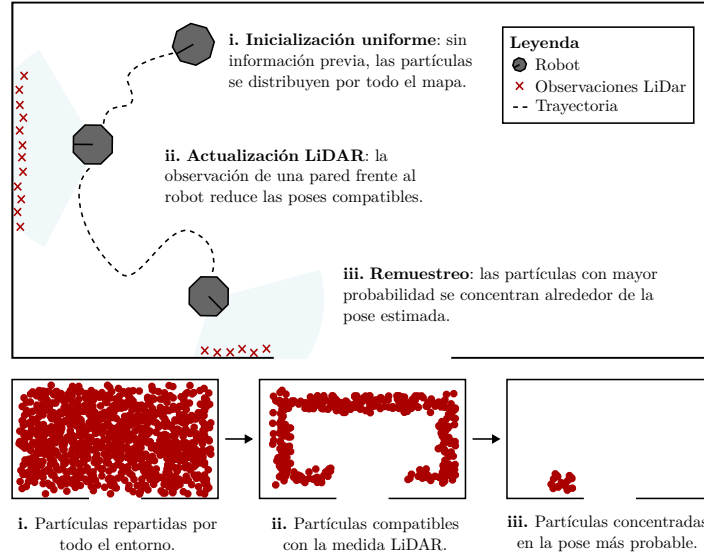


Figura 2.3: Ejemplo esquemático del proceso de localización mediante filtro de partículas.

cionada por el LiDAR aporta información sobre la distancia aproximada hasta la pared observada, la cual podría emplearse para restringir aún más las posiciones posibles. A medida que el robot recorre la habitación, puede comenzar a inferir su posición en el mapa, basándose tanto en las observaciones realizadas como en la estimación de cómo se ha desplazado por el entorno. A partir de esta intuición, el proceso puede expresarse formalmente mediante un algoritmo.

Como se ha indicado, al principio, todos los puntos de la habitación tienen la misma probabilidad de corresponder a la pose real del robot. En vez de aproximar esta probabilidad a través de una función matemática, el filtro de partículas genera un número de poses discretas aleatorias que siguen esta distribución. Cada pose (o partícula) se guarda en el filtro como posible estimación del estado. A continuación, se trata de “descartar” aquellas partículas que no representen con fidelidad el estado real del robot. Para ello se emplean los datos proporcionados por el sensor LiDAR. El filtro de partículas recorre cada una de las poses generadas y estima las mediciones que hubiese generado su LiDAR. Estas son comparadas con las lecturas reales del agente y, a cada partícula se le asigna un peso que refleja su compatibilidad con el estado real del sistema. Una vez obtenida una distribución de probabilidad actualizada a partir de estos pesos, las partículas originales se reemplazan mediante un proceso de remuestreo. Aquí, una vez más, se generan poses discretas aleatorias de acuerdo con la nueva distribución. Este proceso tiene

el efecto de concentrar el nuevo grupo de partículas en las zonas más compatibles con las observaciones del robot, mientras que aquellas menos compatibles tienden a desaparecer. [19]

Durante todo este proceso, el robot se desplaza por la habitación y, mediante los datos de odometría, es posible estimar cómo se ha producido dicho desplazamiento. Esta estimación se aplica a las partículas nuevamente generadas para estimar adónde se desplazarían si se moviesen de la misma forma que el robot. Llegados a este punto, se vuelve a observar el entorno mediante el sensor LiDAR y se repite el proceso. Con cada iteración, la distribución de partículas se actualiza y tiende a converger hacia la pose real del robot. Este método, mostrado en la sección inferior de la Figura 2.3, se conoce también como *Monte Carlo Localization* [18].

El problema adquiere una dimensión adicional si se considera un escenario en el cual el agente no conoce previamente el mapa de su entorno. Se puede hablar, en este caso, de una navegación en un entorno desconocido, donde el agente ha de cartografiar dicho espacio a medida que lo explora. Este tipo de tareas requiere el empleo de algoritmos de localización y mapeo simultáneos (SLAM).

SLAM (*Simultaneous Localization and Mapping*) es un método utilizado en vehículos autónomos que permite crear un mapa y localizar el vehículo en el mapa al mismo tiempo. Los algoritmos de SLAM permiten que el vehículo cree mapas de entornos desconocidos [21, 22].

Al igual que el filtro de partículas, estos algoritmos se nutren de los datos captados por los sensores del robot, aunque introducen una forma distinta de procesar e interpretarlos. Un análisis detallado de las técnicas SLAM, aunque interesante, queda fuera del ámbito de este trabajo. Se considera que el ejemplo del filtro de partículas proporciona al lector una idea suficiente de los principios básicos de la navegación autónoma, permitiéndole contextualizar el marco de este trabajo.

2.2. Modelos de lenguaje de gran tamaño

Los modelos de lenguaje se han descrito en el primer capítulo de esta memoria como funciones matemáticas capaces de predecir una secuencia de palabras a partir de una entrada textual. Para poder llevar a cabo esta función, los modelos de lenguaje son entrenados con un corpus masivo de texto, típicamente extraído de Internet. Durante este entrenamiento, los modelos ajustan progresivamente sus parámetros internos, que pueden entenderse de forma intuitiva como una serie de “diales” que determinan comportamiento del modelo. Es decir, el ajuste de los parámetros internos hace que, ante una misma entrada, el modelo pueda realizar predicciones distintas. Cabe mencionar que los modelos de lenguaje pueden tener miles de millones de parámetros, también llamados pesos. Por esta razón, se suelen denominar modelos de lenguaje de gran tamaño, o *Large Language Models* (LLMs) en inglés.

En el proceso de entrenamiento de un LLM, los pesos se inicializan de forma aleatoria y se refinan progresivamente durante el aprendizaje. El corpus o conjunto de datos de entrenamiento se divide en distintos ejemplos. En el caso de un LLM, estos ejemplos pueden corresponder a frases, párrafos o incluso documentos completos. Cuando se entrena el modelo con uno de ellos, se le proporciona el ejemplo completo salvo la última palabra, y se le pide que la prediga. Después, un algoritmo de retropropagación ajusta los pesos del modelo para favorecer la predicción de la verdadera última palabra. Este proceso se lleva a cabo con miles de millones de ejemplos, de tal manera que el modelo aprende progresivamente a predecir no solo la última palabra de los ejemplos de entrenamiento, sino también a generalizar este comportamiento a ejemplos que nunca ha visto antes.

Entrar en mayor detalle sobre la arquitectura, el funcionamiento y el entrenamiento de los LLMs implicaría abordar conceptos como los *Transformers*, los mecanismos de atención, los *embeddings*, el *RLHF* o el *fine-tuning*. Dado que podrían dedicarse libros enteros a cada uno de estos temas y que no resultan esenciales para la comprensión de este trabajo, no se profundizará en ellos en esta memoria. Se profundiza, no obstante, en ciertos conceptos puntuales que resultan esenciales para comprender el papel de los LLMs en el marco de este trabajo.

2.2.1. Tokenización y representación del texto

La tokenización es el proceso de descomponer una secuencia de texto en unidades más pequeñas, denominadas tokens. Las personas realizan de forma natural un proceso análogo al de la tokenización, descomponiendo intuitivamente un texto

en palabras, sílabas o incluso letras. Sin embargo, de la misma forma que la descomposición de una frase en distintas palabras permite la comprensión por parte de una persona, no necesariamente constituye la representación más eficiente para un modelo de lenguaje. Antes de profundizar en este concepto, conviene introducir brevemente la motivación de la tokenización.

Como se ha aludido, un LLM es una función matemática que opera con lenguaje natural expresado en forma de texto. Sin embargo, para poder realizar operaciones matemáticas sobre un fragmento de texto, este debe representarse previamente de forma numérica. La manera en la que los modelos de lenguaje realizan esta transformación consiste en dividir el texto en distintos *tokens* y asociar a cada uno de ellos un identificador numérico único dentro de su vocabulario. Por lo tanto, de la misma forma que la lengua española dispone de un alfabeto compuesto por 27 letras, un modelo de lenguaje dispone de un vocabulario finito de *tokens*, cada uno asociado a un identificador numérico único. Cabe recalcar que el modelo no opera directamente con estos identificadores numéricos. En su lugar, cada identificador corresponde a una entrada de una tabla de correspondencias, denominada matriz de *embeddings*, en la que cada *token* se asocia a un vector numérico. Son estos vectores, también conocidos como *embeddings*, los que constituyen la representación numérica sobre la que opera el modelo.

Ahora que el lector conoce la relevancia de la tokenización en los modelos de lenguaje, se presenta un ejemplo. Considérese la frase:

En un lugar de la Mancha

Un tokenizador podría descomponerla de la siguiente manera:

En un lugar de la Mancha

Conviene señalar que el proceso de tokenización no es universal y que existen distintos tokenizadores, cada uno definido por su propio vocabulario y reglas de segmentación [23]. Así, esta misma frase podría expresarse mediante diferentes secuencias de *tokens*:

En un lugar de la Mancha

En un lugar de la Mancha

En un lugar de la Mancha

Independientemente del tokenizador empleado, cada token está asociado a un identificador numérico único:

En	un	lugar	de	la	Man	cha
↓	↓	↓	↓	↓	↓	↓
345	912	2487	3456	478	12453	6789

Este identificador se utiliza como índice de una matriz de embeddings, desde la que se obtiene la representación vectorial del token. Nótese que estos vectores suelen tener cientos de dimensiones y pueden emplearse para representar la proximidad entre conceptos similares. No obstante, no se entrará en mayor detalle sobre este aspecto de momento.

ID	Token	Dim 1	Dim 2	...	Dim d
345	En	0,152	-0,231	...	0,612
⋮	⋮	⋮	⋮	⋮	⋮

Se ha introducido la idea de que la tokenización no es universal y de que existen distintos tokenizadores, cada uno de los cuales divide el texto en unidades diferentes. Sin embargo, incluso utilizando un mismo tokenizador, el idioma influye directamente en el proceso de tokenización. Este aspecto es especialmente relevante en el marco de este trabajo. Es decir, en función del idioma y del tokenizador empleado, una misma idea podría ocupar, por ejemplo, 6 *tokens* en español, 9 en inglés y 20 en chino. Resulta razonable plantear, por lo tanto, que, a través del proceso de tokenización, el idioma puede tener un efecto considerable sobre el tiempo de inferencia, el consumo de memoria, el rendimiento y el coste computacional de los modelos.

Como nota final, conviene señalar que la tokenización se ha abordado desde la perspectiva de la lengua castellana. El español, al igual que el inglés o el árabe emplea un sistema de escritura alfabético basado en un conjunto reducido y finito de letras. Para apreciar esta diferencia, basta con considerar sistemas de escritura no alfabéticos, como el chino, donde existen miles de caracteres diferentes.

2.2.2. Inferencia autoregresiva y ventana de contexto

Hasta este punto, se ha descrito de forma simplificada un LLM como un modelo capaz de predecir la siguiente palabra a partir de un segmento de texto. Formalmente, este proceso se conoce como inferencia autorregresiva y se desarrolla con mayor detalle a lo largo de esta sección. Este concepto permite, a su vez, comprender mejor el impacto del idioma sobre el proceso de inferencia, tal como se apuntó

brevemente al final de la sección anterior.

La inferencia autorregresiva sugiere que un modelo de lenguaje genera sus respuestas prediciendo estrictamente un *token* cada vez, y no palabras, como se ha dado a entender previamente. Este *token* se añade entonces al texto original, y la secuencia resultante se utiliza como nueva entrada del modelo. El proceso se repite hasta que se cumple una condición de salida que marca el final de la respuesta.

[Entrada al modelo] $\rightarrow$ Siguiendo token predicho

[En un lugar de la] $\rightarrow$ **Man**

[En un lugar de la Man] $\rightarrow$ **cha**

[En un lugar de la Mancha] $\rightarrow$,

[En un lugar de la Mancha,] $\rightarrow$ **de**

[En un lugar de la Mancha, de] $\rightarrow$...

El mecanismo descrito permite entender que el tiempo de inferencia —es decir, el tiempo necesario para generar una respuesta— crece con el número de *tokens* generados. Además, un mayor número de *tokens* implica que el modelo debe recorrer este ciclo de generación más veces, lo que da lugar a un mayor número de operaciones y, por lo tanto aumenta el consumo energético y la ocupación de memoria. El proceso de generación concluye cuando se cumple una condición de parada. Por lo general esto corresponde a uno de dos casos. En primer lugar, que el modelo considere que la respuesta es suficiente y no es necesario generar más *tokens*. O, al contrario, que se alcance un número máximo de *tokens* que el modelo puede procesar en una misma consulta. Este último caso introduce el concepto de ventana de contexto.

La ventana de contexto, o *context window* de un modelo de lenguaje hace referencia al número máximo de *tokens* que este puede procesar simultáneamente. Aquí, se consideran tanto los *tokens* de entrada como los *tokens* generados durante la respuesta. En un chatbot, por ejemplo, la entrada podría consistir en un *prompt* textual, documentos adjuntos e incluso el historial de conversaciones previas. Independientemente, cuanto mayor sea la cantidad de información proporcionada a la entrada, menor será el número de tokens disponibles para generar la respuesta. Considérese un modelo con una ventana de contexto de 4.000 tokens, aproximadamente la capacidad que ofrecen agentes como GPT-3.5. A modo de referencia, esta cantidad equivale aproximadamente a unas 6–10 páginas de texto continuo

(dependiendo de factores como el idioma y el tokenizador empleado). Si se le proporcionase al modelo una consulta con una longitud de 3.000 tokens, únicamente quedarían 1.000 disponibles para producir la respuesta. Existirán casos, naturalmente, en los que la salida deba superar este límite. Para ello, parte del contexto debe descartarse o resumirse con el fin de poder ser procesado por el modelo adecuadamente.

Finalmente, se plantean dos cuestiones que permiten considerar el papel del idioma desde una perspectiva más amplia:

1. Si el número de tokens empleados para expresar una misma idea varía de forma significativa entre idiomas, también lo hará el número de operaciones necesarias durante el proceso de inferencia y, potencialmente, el coste computacional asociado.
2. A su vez, en tareas grandes, existirán casos en los que una idea pueda expresarse dentro de la ventana de contexto en un idioma pero no en otro. Esto puede limitar la cantidad de contexto aportado al modelo según el idioma, afectando a la calidad de la respuesta generada.

2.2.3. Diferencias entre modelos

A día de hoy existe una amplia variedad de modelos de lenguaje, cada uno con características particulares que lo distinguen del resto. En esta sección se describen, a nivel de especificaciones, algunos de los parámetros que caracterizan a los LLMs. Esto permitirá, más adelante, evaluar su desempeño y compararlos de forma adecuada.

- **Número de parámetros:** los parámetros son los ajustes internos que controlan el comportamiento de un LLM y que se aprenden durante el proceso de entrenamiento. Estos ajustes corresponden a valores numéricos, normalmente representados en forma de pesos o sesgos. En los modelos de lenguaje de gran tamaño, el número de parámetros puede alcanzar cientos de miles de millones. En estos casos, cuanto mayor es el número de parámetros, mayor suele ser el consumo de recursos computacionales del modelo, aunque también puede aumentar su capacidad para discernir patrones y relaciones complejas. [\[24\]](#)
- **Arquitectura:** Se refiere a los principios y estructuras internas que definen como funcionan los modelos de lenguaje. La mayoría de los LLMs modernos se basan en la arquitectura *Transformer*, pero existen variantes y optimizaciones que difieren de un modelo a otro. Entre estas diferencias pueden

encontrarse el número de capas del modelo, la dimensión de los *embeddings* o los mecanismos de atención empleados.

- **Corpus de entrenamiento:** Representa el conjunto de datos empleado durante el proceso de aprendizaje para ajustar los parámetros del modelo. Distintos modelos pueden entrenarse con diferentes cantidades y tipos de datos, en función del propósito para el que han sido diseñados y del tipo de información que se espera que procesen. A grandes rasgos, los modelos entrenados con corpus más amplios y diversos tienden a presentar una mayor capacidad de generalización.
- **Idiomas cubiertos:** El corpus empleado en el entrenamiento de modelos de lenguaje de gran tamaño rara vez es uniforme entre idiomas. Por ello, la capacidad de generalización de estos modelos suele depender del idioma en el que se realizan las consultas. Este tema se aborda con mayor detalle en la siguiente sección.
- **Longitud de ventana de contexto:** Los LLMs pueden diferir en el número máximo de *tokens* que pueden procesar simultáneamente. Los modelos más tempranos tenían ventanas de contexto de alrededor de 1.000 *tokens*, mientras que, en la fecha de redacción de esta memoria, algunos modelos comerciales alcanzan ventanas del orden de los 100.000 *tokens*, existiendo incluso modelos que llegan al millón de *tokens*.
- **Fine-tuning:** El *fine-tuning* es el proceso de adaptar un modelo pre-entrenado a una tarea o caso de uso específico mediante un entrenamiento adicional; reajustando sus parámetros. Dos modelos pueden compartir el mismo pre-entrenamiento y, sin embargo, haber sido ajustados mediante *fine-tuning* para tareas diferentes, dando lugar a modelos con capacidades y comportamientos distintos. [\[25\]](#)

2.2.4. Multilingüismo en modelos de lenguaje

Muchas fuentes en la literatura apuntan a que existe una disparidad lingüística en el ámbito del procesamiento del lenguaje natural [\[26\]](#). Aquí, se hace referencia a la distribución desigual de idiomas en los recursos de datos disponibles, en la producción científica y en el desarrollo de agentes multimodales. Con el fin de que los avances tecnológicos puedan ser empleados por el mayor número posible de personas en todo el mundo, independientemente del idioma que utilicen, resulta evidente la necesidad de entrenar estos modelos con datos multilingües.

Cuadro 2.1: Número de artículos disponibles en distintas ediciones lingüísticas de Wikipedia. Elaboración propia a partir de las estadísticas de Meta-Wiki [27].

Código	Idioma	Artículos
en	Inglés	7.201.970
de	Alemán	3.132.060
fr	Francés	2.766.665
nl	Neerlandés	2.221.851
es	Español	2.121.887
ru	Ruso	2.107.174
it	Italiano	1.975.608
pl	Polaco	1.699.668
zh	Chino	1.541.554
ja	Japonés	1.507.613
ar	Árabe	1.321.672
vi	Vietnamita	1.302.698
pt	Portugués	1.176.183
ca	Catalán	796.188
ko	Coreano	751.856
tr	Turco	687.783
eu	Euskera	491.331
gl	Gallego	190.322
hi	Hindi	170.296
lb	Luxemburgués	67.404

A medida que se desarrollan estos modelos de lenguaje y se extiende su acceso, ha aumentado el interés también por lo que la literatura denomina *Cross-Lingual Understanding* (XLU). XLU hace referencia a la capacidad de un modelo para entender y relacionar significados entre distintos idiomas. Es decir, ante entradas semánticamente equivalentes expresadas en distintos idiomas, un modelo debería ser capaz de generar salidas equivalentes, con independencia del idioma utilizado.

Quizás el factor más determinante en este ámbito sea la distribución de idiomas en el corpus de entrenamiento de los modelos de lenguaje, que a su vez se ve condicionada por la cantidad de recursos digitales disponibles para cada lengua. Como ejemplo familiar para el lector, puede considerarse el caso de Wikipedia, una fuente textual ampliamente utilizada en la construcción de corpus de pre-entrenamiento para modelos de lenguaje de gran tamaño. El Cuadro 2.1 recoge el número de artículos disponibles en distintas ediciones lingüísticas de Wikipedia en el momento de redacción de esta memoria. Aunque estas cifras no representan

Cuadro 2.2: Distribución reducida de datos por idioma en el corpus CC-100. Elaboración a partir de Conneau et al. [28].

Código ISO	Idioma	Tokens (M)	Tamaño (GiB)
en	Inglés	55.608	300.8
ru	Ruso	23.408	278.0
vi	Vietnamita	24.757	137.3
de	Alemán	10.297	66.6
ja	Japonés	530	69.3
pt	Portugués	8.405	69.1
fr	Francés	9.780	56.8
ko	Coreano	5.644	54.2
es	Español	9.374	53.3
nl	Neerlandés	5.025	49.3
zh	Chino simplificado	259	46.9
pl	Polaco	6.490	44.6
ar	Árabe	2.869	28.0
tr	Turco	2.736	20.9
hi	Hindi	1.715	20.2
zh	Chino tradicional	176	16.6
ca	Catalán	1.752	10.1
gl	Gallego	495	2.9
eu	Euskera	270	2.0
lb	Luxemburgués	—	—

perfectamente la disponibilidad real de datos lingüísticos, sí muestran la gran disparidad existente entre idiomas, con una disponibilidad mucho mayor para lenguas como el inglés o el alemán frente a otras como el gallego.

Con el objetivo de reducir esta disparidad, han surgido conjuntos de datos de entrenamiento orientados a incorporar una mayor diversidad lingüística. La literatura indica que los modelos de lenguaje multilingües que emplean este tipo de corpus de entrenamiento tienden a obtener mejores resultados en tareas de XLU [28]. A modo de ejemplo, se presenta una versión reducida de la distribución de datos por idioma en el corpus CC-100 (Figura 2.2). Nótese que incluso datasets como CC-100 pueden presentar disparidades en la representación de distintos idiomas, lo que recalca una vez más la dificultad de construir conjuntos de datos multilingües equilibrados. Este concepto resulta especialmente relevante a la hora de interpretar el rendimiento de los LLMs en distintos idiomas, y se retomará en capítulos posteriores.

2.2.5. La arquitectura Transformer

Inicialmente, no se contemplaba incluir esta sección en la memoria, ya que resumir la arquitectura *Transformer* en unos pocos párrafos constituye una tarea especialmente compleja. Sin embargo, al revisar los contenidos tratados en capítulos posteriores, resulta evidente que contar con una visión general de su funcionamiento es esencial para comprender los modelos de visión-lenguaje, la metodología planteada por *OneMap* y otros conceptos relevantes para esta memoria. Por lo tanto, a continuación se expone, aunque de forma breve, una descripción a alto nivel de la arquitectura *Transformer* [5].

Como se ha indicado en secciones anteriores, la tokenización descompone la entrada a un LLM en una serie de *tokens*. La tokenización se ha explicado únicamente en el contexto textual. No obstante, de forma análoga, una imagen también puede descomponerse en unidades más pequeñas, habitualmente denominadas *patches*. Por ahora, sin embargo, conviene centrar la explicación exclusivamente en el caso textual. También se ha introducido la idea de *embeddings*, entendidos como vectores de alta dimensionalidad que codifican el “significado” cada *token*. Estos *embeddings* se almacenan en una *lookup table*, que permite asociar a cada *token* un vector a partir de su identificador, denominado *token ID*.

En	un	lugar	de	la	Man	cha
↓	↓	↓	↓	↓	↓	↓
345	912	2487	3456	478	12453	6789
↓	↓	↓	↓	↓	↓	↓
$\begin{bmatrix} 0,152 \\ -0,231 \\ \vdots \\ 0,612 \end{bmatrix}$	$\begin{bmatrix} -0,084 \\ 0,317 \\ \vdots \\ -0,205 \end{bmatrix}$	$\begin{bmatrix} 0,441 \\ -0,092 \\ \vdots \\ 0,738 \end{bmatrix}$	$\begin{bmatrix} -0,118 \\ 0,054 \\ \vdots \\ 0,093 \end{bmatrix}$	$\begin{bmatrix} 0,206 \\ -0,410 \\ \vdots \\ 0,481 \end{bmatrix}$	$\begin{bmatrix} -0,206 \\ 0,154 \\ \vdots \\ -0,372 \end{bmatrix}$	$\begin{bmatrix} 0,332 \\ -0,047 \\ \vdots \\ 0,219 \end{bmatrix}$

Una forma de interpretar este proceso es considerar que cada palabra, o más precisamente cada *token*, ocupa una posición determinada en un espacio multidimensional. Además, los *tokens* con significados similares tienden a quedar representados por vectores próximos en dicho espacio. A modo de ilustración, la Figura 2.4 muestra una interpretación gráfica simplificada de distintos *embeddings* en un espacio de tres dimensiones. Como puede observarse, conceptos similares, como animales o vehículos, aparecen próximos entre sí. Cabe recalcar que los *embeddings* reales suelen tener cientos o incluso miles de dimensiones, por lo que no pueden representarse gráficamente de forma directa.

Dentro del *transformer*, una vez obtenida una representación vectorial de los *tokens* de entrada, la secuencia de vectores se introduce en un módulo conocido como bloque de atención. Sin entrar en detalle, este bloque de atención puede entenderse como una función que permite a los vectores “hablar entre sí”, inter-

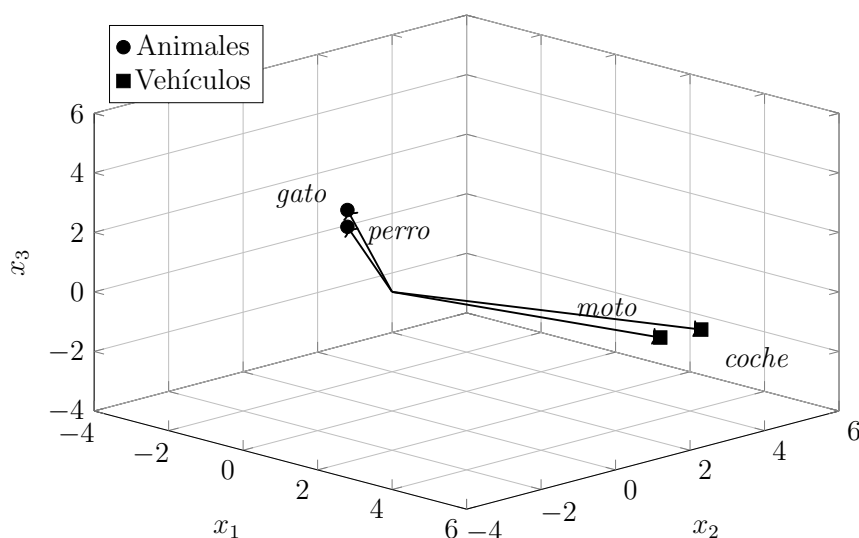


Figura 2.4: Representación distintos *tokens* en un espacio multidimensional. Los *tokens* con significados similares aparecen próximos entre sí.

cambiar información y actualizar sus valores en función del contexto. Esto resulta especialmente interesante, ya que una misma palabra, o *token*, puede adquirir significados distintos según el contexto. De la misma forma en la que un humano entiende la diferencia de significado de la palabra “copa” en “se tomó una copa de vino” y “España va a ganar la Copa del Mundo”, los mecanismos de atención permiten que el modelo ajuste la representación de cada *token* en función del contexto en el que aparece. Para ello, estos mecanismos evalúan la secuencia de *tokens* de entrada y determinan cuáles pueden ser relevantes para el significado de otros, permitiendo así reformar su significado. Se reitera que, al hablar de “significado”, se está haciendo referencia a las coordenadas vectoriales que representan al *token*.

Los mecanismos de atención representan tan solo una de las muchas operaciones que se realizan en los *transformers*, los cuales están compuestos por distintas capas que van modificando progresivamente los vectores de entrada. Estas capas no son, en esencia, más que funciones matemáticas u operaciones matriciales. A la salida de las mismas, la información contenida en la entrada queda representada en una representación vectorial final.

2.2.6. Evaluación de modelos de lenguaje: benchmarks

A continuación, se detalla cómo se evalúan los modelos de lenguaje de gran tamaño, es decir, cómo se comparan entre sí para determinar cuál resulta más adecuado para una tarea específica. Estos modelos pueden ser valorados según sus capacidades en distintas tareas, ya sea la generación y procesamiento de texto, la programación o incluso la traducción. En este caso, los *benchmarks* actúan como procedimientos estandarizados para evaluar y comparar el desempeño de los LLMs.

El diseño de un *benchmark* comienza con la selección de un conjunto de muestras representativas, sobre las que posteriormente se evaluará el desempeño de los modelos. Estas muestras pueden consistir en documentos de texto, problemas matemáticos, ejercicios de programación u otros tipos de tareas, en función de la capacidad que se quiera evaluar en el modelo. Las muestras empleadas suelen estar etiquetadas con respuestas de referencia, indicando, ante una pregunta, la respuesta ideal que tendría que proporcionar el modelo.

Existen varios métodos en lo que respecta la propia evaluación del modelo. Aquí entra en juego principalmente el número de ejemplos incluidos en el enunciado de la tarea, que se proporcionan al modelo antes de solicitarle la respuesta evaluada. [29]

- **Few-shot:** se presentan al modelo varios ejemplos que indican cómo completar la tarea antes de evaluarlo con una muestra específica. Este enfoque permite analizar la capacidad del modelo para generalizar a partir de un número reducido de ejemplos proporcionados en el propio prompt.
- **Zero-shot:** se evalúa al modelo con una muestra específica sin haberle mostrado previamente ningún ejemplo de la tarea. Este planteamiento señala la capacidad del modelo para comprender conceptos nuevos y adaptarse a nuevos escenarios.
- **Fine-tuned:** el modelo se entrena previamente con un conjunto de datos semejante a las muestras seleccionadas para la evaluación. Aquí la meta es mejorar el rendimiento de un modelo en una tarea específica.

Cabe recalcar que, a diferencia del *fine-tuning*, el método *few-shot* no implica de un proceso de entrenamiento o aprendizaje. En este caso no se están modificando los parámetros o pesos del modelo, por lo que no aprende de forma permanentemente. En su lugar, estos ejemplos permiten al modelo contextualizar la tarea que debe realizar. Esta idea se entiende mejor considerando un ejemplo. Para una muestra de evaluación, el enfoque *zero-shot* puede plantear el siguiente *prompt* al modelo:

Zero-shot

Traduce al inglés: azul

Donde la respuesta esperada sería *blue*. Por otra parte, un método *few-shot* introduciría, junto a la muestra, una serie de ejemplos previos.

Few-shot

Español: rojo

Inglés: red

Español: amarillo

Inglés: yellow

Español: azul

Inglés:

Finalmente, el proceso de benchmarking puntúa los resultados proporcionados por el modelo para evaluar cuánto se aproximan a la respuesta esperada. Para ello se suelen emplear una combinación de métricas cuantitativas. Algunas de ellas son:

- **Accuracy:** porcentaje de predicciones correctas.
- **Precision:** proporción de predicciones positivas que son realmente correctas.
- **Recall:** proporción de verdaderos positivos identificados correctamente.
- **F1 score:** media armónica entre *precision* y *recall*.
- **Exact match:** proporción de predicciones que coinciden exactamente con la respuesta de referencia.
- **Perplexity:** mide la capacidad del modelo para predecir texto. Cuanto menor es la *perplexity*, mejor es su capacidad predictiva.
- **BLEU:** métrica empleada en tareas de traducción automática.
- **ROUGE:** métrica empleada en tareas de resumen de texto.



(a) Imagen de un perro (b) Imagen de una taza roja (c) Imagen de una silla

Figura 2.5: Ejemplos de pares imagen-texto

2.3. Modelos visión-lenguaje

En el primer capítulo se ha aludido a que el propósito de un agente multimodal es trabajar con distintos tipos de datos de forma conjunta. Se hizo especial hincapié en los modelos de visión-lenguaje, capaces de acoplar información textual y visual [30, 13]. El objetivo de estas arquitecturas se podría resumir como la capacidad de relacionar significado entre distintas representaciones. Esta sección centra su análisis en CLIP como ejemplo representativo de modelo de visión-lenguaje. CLIP es el acrónimo de *Contrastive Language-Image Pre-training* (preentrenamiento contrastivo de lenguaje e imagen), y hace referencia a un modelo propuesto por OpenAI en febrero de 2021, entrenado con un corpus de 400 millones de pares imagen-texto procedentes de Internet [31].

La idea o motivación de CLIP es establecer una representación común de conceptos similares, independientemente de la modalidad de procedencia. Es decir, tanto una imagen de un perro como una frase en la que aparezca la palabra “perro” deberían poder asociarse a un mismo concepto. En la sección correspondiente a los modelos de lenguaje de gran tamaño se estudió una forma de representar ideas expresadas en lenguaje natural mediante estructuras con las que el modelo podía operar: los *embeddings*. Recuerdese además que, en este caso, los *tokens* con significado similar tendían a agruparse de forma natural en regiones próximas del espacio vectorial, al interpretar los vectores como posiciones en dicho espacio. CLIP propone un enfoque análogo a este, con la diferencia de que, en este caso, se busca ampliar la funcionalidad para agrupar representaciones procedentes tanto de texto como de imágenes. La metodología planteada por CLIP, por lo tanto, puede entenderse como un intento de situar próximos entre sí, dentro de un mismo

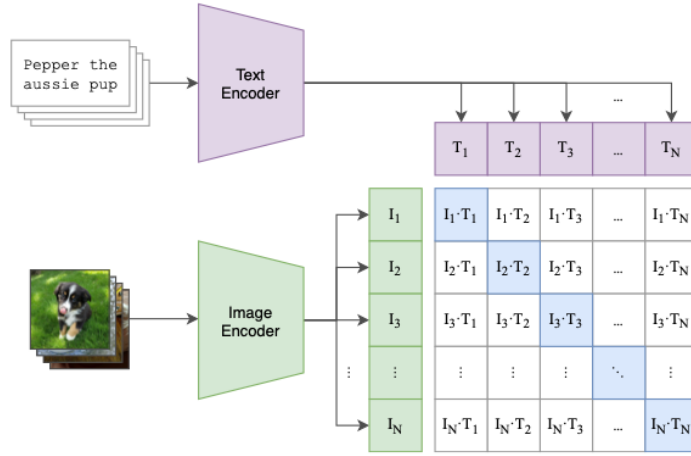


Figura 2.6: Esquema de preentrenamiento contrastivo de CLIP. Fuente: [31].

espacio de representación, aquellos vectores que representan conceptos similares, ya procedan de imágenes o de texto.

Para comprender esta idea, resulta conveniente presentar de forma conceptual la metodología de entrenamiento empleada por CLIP. Sin embargo, en lugar de emplear 400 millones de imágenes anotadas, se parte de tres ejemplos sencillos. Se propone, por lo tanto, una imagen de un perro, una imagen de una taza roja y una imagen de una silla, cada una acompañada de su descripción textual correspondiente. Esto quiere decir que, junto a la imagen del perro, se proporciona una frase en lenguaje natural que indica “imagen de un perro”, y de forma análoga para el resto de ejemplos (Figura 2.5).

Estos datos de entrenamiento serán introducidos en el modelo, que por ahora conviene entender como dos modelos —o, más precisamente, dos codificadores— independientes: un codificador visual y un codificador textual (Figura 2.6). El codificador de texto, de forma similar a un LLM, genera a partir de una entrada textual su representación vectorial de alta dimensionalidad. Análogamente, el codificador de imagen realiza el mismo proceso, pero a partir de imágenes. Por lo tanto, en el ejemplo artificial aquí presentado, las imágenes se introducirían en el codificador visual y sus descripciones textuales en el codificador textual.

Se obtendría, consecuentemente, tres vectores de correspondientes a las imagen y tres vectores correspondientes a las etiquetas de texto. Cada combinación posible entre un vector de imagen y un vector de texto puede colocarse en una entrada de

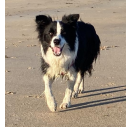
	“imagen de un perro”	“imagen de una taza roja”	“imagen de una silla”
	$\begin{bmatrix} 0,42, & -0,18, & \dots, & 0,73 \\ 0,39, & -0,21, & \dots, & 0,70 \end{bmatrix}$	$\begin{bmatrix} 0,42, & -0,18, & \dots, & 0,73 \\ -0,31, & 0,64, & \dots, & 0,12 \end{bmatrix}$	$\begin{bmatrix} 0,42, & -0,18, & \dots, & 0,73 \\ 0,08, & 0,51, & \dots, & -0,46 \end{bmatrix}$
	$\begin{bmatrix} -0,28, & 0,67, & \dots, & 0,16 \\ 0,39, & -0,21, & \dots, & 0,70 \end{bmatrix}$	$\begin{bmatrix} -0,28, & 0,67, & \dots, & 0,16 \\ -0,31, & 0,64, & \dots, & 0,12 \end{bmatrix}$	$\begin{bmatrix} -0,28, & 0,67, & \dots, & 0,16 \\ 0,08, & 0,51, & \dots, & -0,46 \end{bmatrix}$
	$\begin{bmatrix} 0,11, & 0,49, & \dots, & -0,52 \\ 0,39, & -0,21, & \dots, & 0,70 \end{bmatrix}$	$\begin{bmatrix} 0,11, & 0,49, & \dots, & -0,52 \\ -0,31, & 0,64, & \dots, & 0,12 \end{bmatrix}$	$\begin{bmatrix} 0,11, & 0,49, & \dots, & -0,52 \\ 0,08, & 0,51, & \dots, & -0,46 \end{bmatrix}$

Figura 2.7: Representación esquemática de las combinaciones entre vectores de imagen y vectores de texto. Cada entrada de la matriz contiene el vector asociado a la imagen de la fila y el vector asociado a la descripción textual de la columna.

una matriz, obteniendo así una representación conjunta de todas las combinaciones imagen-texto. Esto se muestra a continuación, donde las imágenes se distribuyen a lo largo de las filas de la matriz, y las descripciones textuales a lo largo de las columnas. Como se ha expresado, se pretende que los vectores que representan el mismo concepto sean semejantes entre sí. Es decir, que los pares situados en la diagonal de la matriz presenten una mayor similitud. Además, de manera igualmente importante, se precisa que aquellos pares de vectores que no están situados en la diagonal no se parezcan entre sí. Esto equivale a asociar conceptos distintos con representaciones vectoriales distintas. Por lo tanto, el entrenamiento de CLIP busca maximizar la similitud entre entradas correspondientes y, simultáneamente, minimizar la similitud entre entradas no correspondientes [31]. Precisamente por este motivo, CLIP se considera un modelo contrastivo.

Para evaluar la proximidad entre dos vectores, $\mathbf{e}_1$ y $\mathbf{e}_2$, se emplean la similitud de coseno,

$$\text{sim}_{\cos}(\mathbf{e}_1, \mathbf{e}_2) = \frac{\mathbf{e}_1 \cdot \mathbf{e}_2}{\|\mathbf{e}_1\| \|\mathbf{e}_2\|} = \cos(\theta)$$

Ya que cada vector o *embedding* puede interpretarse como una dirección en un espacio de alta dimensión, la similitud de coseno mide la proximidad entre vectores a partir del coseno del ángulo que forman entre sí (Figura 2.8). Si estos

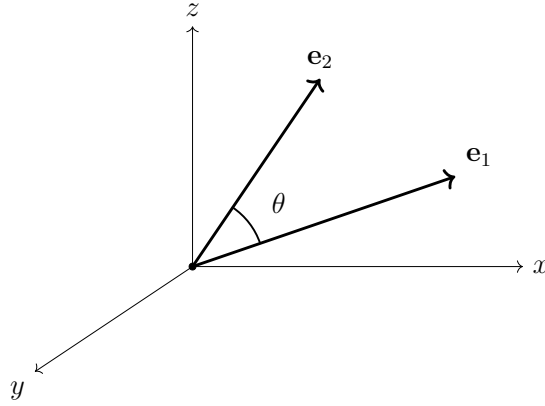


Figura 2.8: Representación geométrica del ángulo entre dos embeddings.

vectores son exactamente iguales, el coseno del ángulo entre ellos tendrá un valor igual a uno; si son perpendiculares, el coseno será nulo; y si apuntan en direcciones opuestas, tomará un valor igual a -1. Entre estos casos extremos existe un continuo de valores intermedios. Entonces, expresado ya matemáticamente, puede decirse que la función objetivo empleada durante el entrenamiento trata de maximizar la similitud de coseno (o alineamiento) entre imágenes y descripciones textuales correspondientes, y de minimizar la existente entre aquellas que no se corresponden entre sí.

Este espacio vectorial, en el que coexisten las representaciones de imágenes y texto, se conoce en la literatura como *embedding space*. El *embedding space* se ajusta progresivamente durante el proceso de aprendizaje, de forma análoga a como los LLMs aprenden representaciones vectoriales a partir de sus datos de entrenamiento. Con la diferencia de que, en este caso, se actualizan los parámetros tanto del codificador textual como del codificador visual a partir de un corpus de millones de muestras. Como resultado de este entrenamiento, las representaciones de conceptos similares, ya procedan de imágenes o de texto, tienden a agruparse en un espacio común.

El concepto de *embedding space* constituye una herramienta poderosa dentro del campo de la inteligencia artificial. La geometría aprendida por modelos como CLIP permite operar matemáticamente con ideas y conceptos presentes en las imágenes y en el texto. Para ilustrar la utilidad de este tipo de representación, se presenta el siguiente ejemplo. Considérense dos imágenes: una que representa un hombre y otra que representa un hombre con un sombrero. Naturalmente, estas

imágenes presentan cierta similitud entre sí, por lo que los vectores que recogen su significado podrían estar relativamente próximos. Se representa el vector asociado al concepto “hombre” como $\mathbf{e}_{\text{hombre}}$ y el vector asociado al concepto “hombre con sombrero” como $\mathbf{e}_{\text{hombre+sombrero}}$ tal que,

$$\mathbf{e}_{\text{hombre}}, \mathbf{e}_{\text{hombre+sombrero}} \in \mathbb{R}^f$$

Donde f representa la dimensionalidad del *embedding space* empleado por CLIP. Como estos *embeddings* no dejan de ser vectores, es posible realizar operaciones matemáticas entre ellos, como la resta:

$$\mathbf{e}_{\text{operación}} = \mathbf{e}_{\text{hombre+sombrero}} - \mathbf{e}_{\text{hombre}}$$

El resultado de la operación es otro vector, $\mathbf{e}_{\text{operación}}$, dentro del mismo espacio vectorial. Por lo tanto, dado que este nuevo vector también pertenece al *embedding space*, cabe esperar que pueda asociarse a algún tipo de significado. Para poder interpretarlo en lenguaje natural, lo lógico sería introducir este *embedding* en un decodificador, ya sea visual o textual, para obtener una imagen o una frase que describa el vector $\mathbf{e}_{\text{operación}}$. Sin embargo, los codificadores de CLIP solo funcionan en una dirección; es decir, no son funciones invertibles. Por tanto, permiten pasar de una imagen o un texto a un *embedding*, pero no realizar directamente el proceso inverso.

No obstante, existen métodos que permiten superar parcialmente esta limitación. Considérese que se introducen todas las palabras de un diccionario de lengua castellana en el codificador textual de CLIP. Se obtendría, como resultado, una representación en el *embedding space* para cada una de ellas. A continuación, se podría recorrer la lista de vectores generados y compararlos uno a uno con el vector $\mathbf{e}_{\text{operación}}$ mediante la similitud de coseno. Se podría interpretar que $\mathbf{e}_{\text{operación}}$ tiene un significado próximo a aquellas entradas del diccionario con las que presenta una similitud de coseno cercana a la unidad. En el ejemplo planteado, cabría esperar que la resta entre los conceptos “hombre con sombrero” y “hombre” produjera algo relacionado con “sombrero”. Lo verdaderamente llamativo, y una de las razones por las que CLIP resulta especialmente útil, es que, si se realizase este procedimiento, los conceptos con mayor similitud serían precisamente “sombrero”, “gorra”, “casco”, etc. Cabe señalar que este ejemplo constituye una idealización ilustrativa del comportamiento del espacio de *embeddings*.

2.4. OneMap

OneMap constituye el artículo de referencia principal sobre el que se articula este trabajo. Se introduce, quizás, en una etapa relativamente avanzada de la memoria. El motivo de esto es que *OneMap* aborda un marco teórico especialmente complejo. Se ha considerado necesario, por tanto, reforzar una serie de conceptos ya introducidos en los primeros capítulos antes de presentarlo.

OneMap es el nombre que los autores de *One Map to Find Them All: Real-time Open-Vocabulary Mapping for Zero-shot Multi-Object Navigation* [32] dan a la herramienta propuesta en el artículo. No obstante, más que una herramienta, *OneMap* se podría interpretar como una metodología para llevar a cabo la búsqueda de objetos en entornos complejos mediante agentes autónomos. Como se ha insistido hasta este punto, esto comprende una tarea fundamental en muchas aplicaciones reales de navegación robótica.

Esta memoria se ha iniciado con el ejemplo de la búsqueda de una taza roja. Nuevamente, en esta sección se plantea la misma instrucción:

“Busca la cocina y recoge la taza roja que está sobre la encimera”

En los primeros capítulos se argumentó que un humano, aun sin disponer de información previa del entorno, sería capaz de ejecutar dicha instrucción con relativa soltura. Se explicó que esto se debía a la capacidad humana de aprovechar la información semántica y espacial de la escena para, en lugar de realizar una búsqueda aleatoria, guiar su exploración de forma estructurada. Así, una persona tiene la capacidad de razonar, por ejemplo:

“Puede que no sepa dónde están la taza roja o la encimera, pero antes he visto un horno. Los hornos suelen estar en la cocina; por tanto, debería dirigirme hacia allí.”

A medida que un humano explora un entorno desconocido, forma mentalmente una representación semántico-espacial de la escena en la que se desenvuelve. Dicha representación se actualiza a medida que surgen nuevas direcciones de búsqueda y se construyen nuevas relaciones semánticas entre elementos ya observados. Además, con cierto nivel de detalle, un humano es capaz de recordar este “mapa mental” y apoyarse en la distribución general de los objetos para guiar futuras consultas. De este modo, si una vez encontrada la taza roja se le pidiese llevarla al salón, un humano que durante la búsqueda hubiese visto un sofá podría razonar que este probablemente se encuentra en el salón y utilizar esa información para

dirigirse hasta allí.

Este ejemplo introduce, de forma intuitiva, el funcionamiento general de *OneMap*. En el artículo, los autores proponen dotar a un agente robótico de la capacidad de formar una representación conjunta semántica y espacial del entorno, con el fin de llevar a cabo la búsqueda de objetos de manera eficiente, de forma similar a como lo haría un humano en el ejemplo de la taza roja.

2.4.1. Descripción conceptual de OneMap

Una vez introducido *OneMap* de forma intuitiva, se formula con mayor precisión el problema que aborda. Para ello, quizás la mejor manera sea analizar el propio título.

One Map to Find Them All: Real-time Open-Vocabulary Mapping for Zero-shot Multi-Object Navigation

Un único mapa para encontrarlos todos: mapeo en tiempo real de vocabulario abierto para navegación multiobjeto zero-shot.

Más allá de la aparente referencia a El Señor de los Anillos, el título resulta especialmente denso en significado. Conviene entonces, descomponer y analizar los conceptos clave para asentar las ideas principales que los autores pretenden transmitir.

- **Mapeo (*mapping*)**: se hace referencia a la función principal de *OneMap*: elaborar un mapa del entorno. En particular, se refiere a una representación semántico-espacial de la escena, en la que se busca asignar significado a cada punto del espacio.
- **Tiempo real (*real-time*)**: indica que el mapa no se construye únicamente al final del recorrido, sino que se actualiza durante el proceso de exploración a partir de las observaciones sucesivas del agente. Más allá de esto, también implica la existencia de restricciones de tiempo real, como el uso de memoria y el consumo de recursos computacionales a las que está sujeto el robot.
- **Vocabulario abierto (*open-vocabulary*)**: el sistema no está limitado a la búsqueda de un conjunto cerrado de objetos predefinidos. Este sería el caso de un enfoque de vocabulario cerrado, en el que, por ejemplo, un agente solo está capacitado para buscar categorías predefinidas, como tazas rojas o platos azules. En cambio, en un enfoque de vocabulario abierto se admiten consultas más flexibles y abiertas en lenguaje natural.

- **Navegación multiobjeto (*multi-object navigation*)**: a diferencia de métodos anteriores, en los que el mapa semántico-espacial se formaba desde cero para cada consulta, *OneMap* busca mantener dicha representación entre consultas sucesivas. Es decir, se plantea un mapa reutilizable capaz de guiar búsquedas posteriores de objetos.
- ***Zero-shot***: otros métodos requieren entrenamiento específico o demostraciones humanas para realizar una tarea específica en un entorno determinado, por ejemplo mediante *fine-tuning* o aprendizaje a partir de demostraciones. *OneMap*, en cambio, lo hace *zero-shot*, lo que significa que puede realizar estas tareas sin recibir contexto o ejemplos específicos, ni haber sido entrenado específicamente para ese concepto y entorno.

Más allá de la metodología propuesta por los autores para construir el mapa semántico-espacial, las principales contribuciones del artículo incluyen la validación del sistema tanto en simulación¹ como en experimentos reales², así como la propuesta de un *benchmark* para la evaluación de tareas de navegación multiobjeto.

2.4.2. Metodología

En esta sección se describe el *pipeline* empleado por la metodología *OneMap*. Aunque no se aborda desde un nivel matemático especialmente profundo, sí se trata con detalle su funcionamiento general, con el objetivo de proporcionar al lector una idea clara de en qué consiste la herramienta.

Se parte de una secuencia continua de imágenes RGB y de profundidad, procedentes de la cámara situada, en este caso, en el agente robótico. En términos prácticos, esto equivale a capturar un flujo de vídeo de la escena y transmitirlo a la siguiente etapa del proceso en forma de imágenes sucesivas. Quizás el lector no esté familiarizado con la forma en que se almacena una imagen digital en memoria. Para facilitar la comprensión de esta sección, se describe brevemente a continuación.

Una imagen puede interpretarse como una cuadrícula en la que cada celda, denominada píxel, representa un punto de la escena y almacena la información de su color. El número de celdas utilizadas para representar la imagen se denomina

¹Las simulaciones realizadas por los autores se llevan a cabo sobre el dataset HM3D [33], un conjunto compuesto por 1.000 reconstrucciones reales de interiores.

²Los experimentos realizados por los autores se llevan a cabo en un robot equipado con una NVIDIA Jetson AGX Orin.

(220, 220, 220)	(170, 170, 170)	(110, 110, 110)	(60, 60, 60)	(0, 0, 0)
(255, 255, 128)	(128, 255, 255)	(200, 128, 255)	(200, 255, 128)	(255, 128, 200)
(128, 128, 128)	(255, 128, 128)	(128, 255, 128)	(128, 128, 255)	(255, 200, 128)
(128, 0, 128)	(0, 255, 255)	(0, 255, 0)	(255, 192, 203)	(165, 42, 42)
(255, 0, 0)	(255, 165, 0)	(255, 255, 0)	(0, 128, 0)	(0, 0, 255)

Figura 2.9: Ejemplo de imagen representada como una cuadrícula de 5×5 píxeles, donde cada celda contiene su valor RGB.

resolución y esta relacionado con el nivel de detalle que puede apreciarse en ella. Los colores almacenados en cada uno de los píxeles pueden codificarse en distintos formatos, siendo el formato RGB uno de los más empleados. Este formato cuantifica, para un mismo píxel, la intensidad de luz roja (Red), verde (Green) y azul (Blue) percibida, asignando a cada canal un valor entre 0 y 255. De esta forma, es posible representar un gran número de colores mediante la superposición de los tres canales. El ejemplo de la Figura 2.9 ilustra una imagen con una resolución de 5×5 píxeles.

La representación gráfica de una imagen resulta útil para que las personas interpreten su contenido. Sin embargo, los ordenadores la representan mediante una secuencia de números, habitualmente organizada en forma de tensor tridimensional. Aquí, las dos primeras dimensiones corresponden a la resolución espacial de la imagen, mientras que la tercera almacena los tres canales de color RGB. Por lo tanto, una imagen puede expresarse como:

$$\mathcal{I} \in \mathbb{R}^{H \times W \times 3}$$

donde H y W representan, respectivamente, la altura y la anchura de la imagen en píxeles. Cada elemento del tensor corresponde, por lo tanto, a un píxel y almacena un vector con las intensidades de los canales rojo, verde y azul. Por ejemplo,

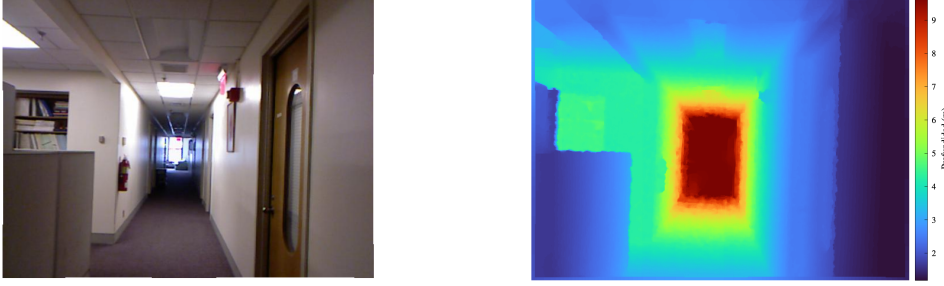


Figura 2.10: Imagen RGB y mapa de profundidad correspondiente del dataset NYU[34].

$$\mathcal{I} = \begin{bmatrix} [220, 220, 220] & [170, 170, 170] & \dots \\ [225, 225, 128] & [128, 255, 255] & \dots \\ \vdots & \vdots & \ddots \end{bmatrix}$$

Considérese ahora el caso en que la representación se lleva a cabo mediante cuatro canales: tres de ellos correspondientes al color RGB y un cuarto que almacena, para cada punto de la imagen, la profundidad registrada. Esto es posible gracias a la existencia de cámaras RGB-D, capaces de estimar, para cada píxel, la distancia entre la cámara y el punto de la escena representado por dicho píxel (Figura 2.10). Sin embargo, en lugar de representar esta profundidad como un cuarto canal de la imagen $\mathcal{I}$, resulta más cómodo para la explicación considerarla como una imagen distinta, $\mathcal{D}$, correspondiente a un tensor bidimensional.

$$\mathcal{D} \in \mathbb{R}^{H \times W}$$

Retomando la explicación anterior, se parte de imágenes RGB y de profundidad. Junto a ellas se almacena también la pose de la cámara, es decir, su posición y orientación en el momento en que se captura cada par de imágenes. Esta información permitirá, más adelante, ubicar las observaciones en el mapa construido.

Para extraer la mayor cantidad posible de información de las imágenes capturadas, resulta conveniente dividir cada imagen en distintas regiones, en lugar de procesarla como una única entidad. Esto permite al sistema atribuir un significado independiente a cada región de la imagen, pudiendo así asociar distintos objetos o conceptos a puntos concretos de la escena [35]. Considérese una imagen de una habitación. En lugar de describirla únicamente a nivel global, como una habitación, resulta más informativo identificar y caracterizar sus distintas regiones, como la

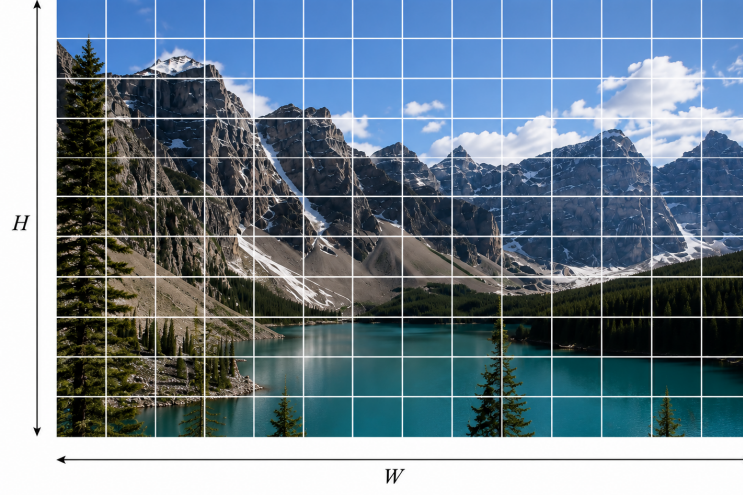


Figura 2.11: Imagen dividida en una malla regular de *patches*.

zona correspondiente a la cama, la mesilla de noche o el armario. De este modo, la información semántica queda asociada a posiciones concretas dentro de la imagen. Para llevar a cabo este proceso, la imagen se divide en una malla regular de pequeñas regiones, conocidas en la literatura como *patches*, como muestra la Figura 2.11. Cada imagen $\mathcal{I}$ queda, por lo tanto, dividida en $H_{\mathcal{F}} \times W_{\mathcal{F}}$ *patches*.

A continuación, se pretende extraer de cada *patch* información representativa sobre su contenido visual. Este proceso se conoce como *feature extraction* (o extracción de características en castellano). Para ello, *OneMap* se apoya en SED, un codificador visual [36]. Esta herramienta, de forma similar a un modelo visión-lenguaje, es capaz de procesar una imagen y generar un vector descriptivo de su contenido visual dentro del espacio de *embeddings* de CLIP [36, 31]. Por lo tanto, a partir los $H_{\mathcal{F}} \times W_{\mathcal{F}}$ *patches* de la imagen $\mathcal{I}$, que no dejan de ser imágenes más pequeñas que la original, se obtiene un vector de dimensión f correspondiente a cada región. Los autores de *OneMap* denominan a este conjunto de vectores, o *embeddings*, campo de características discretizado $\mathcal{F}$,

$$\mathcal{F} \in \mathbb{R}^{H_{\mathcal{F}} \times W_{\mathcal{F}} \times f}$$

Representar las regiones de la imagen como vectores permitirá que, en etapas posteriores, estas puedan compararse con consultas textuales en lenguaje natural mediante modelos visión-lenguaje como CLIP. Cabe recalcar que SED trabaja únicamente con imágenes RGB; por tanto, las imágenes de profundidad aún no

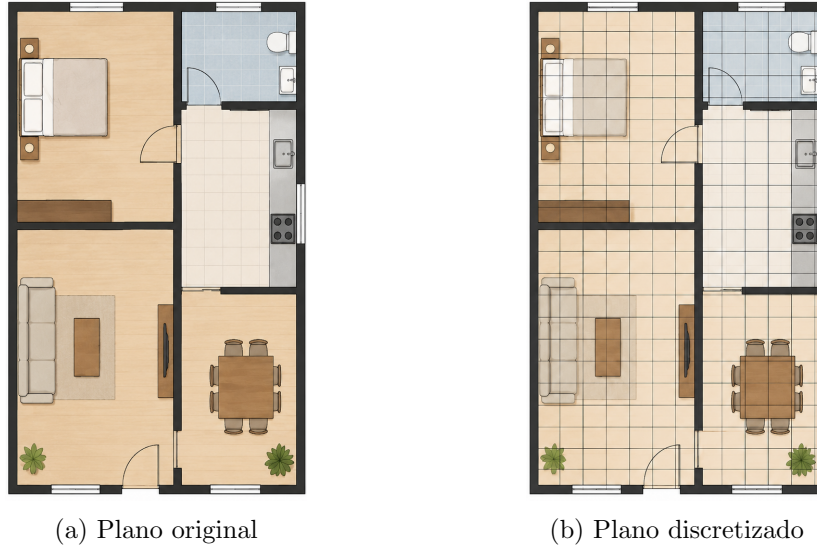


Figura 2.12: Representación de un plano de vivienda antes y después de su discretización espacial.

intervienen en este punto del *pipeline*. Lo harán a continuación.

El siguiente paso en la metodología de *OneMap*, consiste en proyectar el campo de características $\mathcal{F}$ en un espacio tridimensional empleando la imagen de profundidad correspondiente $\mathcal{D}$. El acto de proyectar el campo, permite situar espacialmente las características visuales observadas en un entorno tridimensional. Este proceso es necesario porque la imagen RGB solo muestra la escena en dos dimensiones. Sin embargo, en el entorno real, los objetos no están todos a la misma distancia de la cámara. Al usar la imagen de profundidad, se puede estimar dónde se encuentra cada parte de la imagen en el espacio y se pueden colocar las características visuales acordemente. Se recalca que esta representación espacial se obtiene a partir de un único conjunto de imágenes $\mathcal{I}$ y $\mathcal{D}$.

La cuestión que surge ahora es cómo actualizar y ampliar esta representación a partir de nuevas imágenes capturadas desde distintas poses. Para abordarla, *OneMap* introduce un mapa probabilístico $\mathcal{M}$. La idea es tratar el entorno de exploración no como un espacio continuo, sino como una región discretizable (Figura 2.12), en la que a cada casilla se le asocia información sobre una zona concreta del entorno. Por lo tanto, $\mathcal{M}$ no es más que un mallado bidimensional del entorno donde cada celda queda indexada mediante dos coordenadas:

$$x \in \{0, \dots, n_x\}, \quad y \in \{0, \dots, n_y\}$$

Aquí, n_x y n_y representan las dimensiones del mapa en celdas. Cabe recalcar que al comienzo de la exploración, el mapa $\mathcal{M}$ no contiene todavía información semántica ni geométrica del entorno. Lo único que se proporciona inicialmente son sus dimensiones, n_x y n_y , según las cuales se discretiza el entorno. $\mathcal{M}$, por lo tanto, se aproxima más a una matriz de elementos vacíos. La tarea consiste entonces en ir rellenando progresivamente el mapa con información semántica. Esta información se almacena, como cabría esperar en este contexto, en forma de *embeddings*. De este modo, a cada celda del mapa se le asocia un vector descriptivo de las características visuales observadas en la región correspondiente del entorno. Más específicamente, *OneMap* asocia a cada celda de $\mathcal{M}$ dos elementos: un *embedding* y una varianza. Por tanto, el mapa $\mathcal{M}$ puede expresarse como el par:

$$\mathcal{M} = (F, \sigma^2),$$

donde

$$F \in \mathbb{R}^{n_x \times n_y \times f}, \quad \sigma^2 \in \mathbb{R}^{n_x \times n_y}.$$

De este modo, cada celda del mapa queda definida como:

$$\mathcal{M}(x, y) = (F(x, y), \sigma^2(x, y)),$$

con

$$F(x, y) \in \mathbb{R}^f, \quad \sigma^2(x, y) \in \mathbb{R}.$$

Conviene distinguir entre $\mathcal{F}$, que representa el campo de características extraído de una imagen concreta, y F , que representa el campo global de características almacenado en el mapa.

Ahora que se ha explicado en qué consiste el mapa probabilístico, se va a explicar cómo se pasa de una colección de campos de características locales asociadas a distintas imágenes a un único campo global. Considérese el instante inicial de la exploración, en el que el mapa $\mathcal{M}$ se encuentra vacío. En un instante posterior, el sistema recibe el primer conjunto de imágenes RGB ($\mathcal{I}$) y de profundidad ($\mathcal{D}$), junto con la pose de cámara desde la que fueron capturadas. La imagen RGB se introduce en el codificador SED para obtener una descripción a nivel de *patch*, representada como una matriz de vectores y denominada campo de características local $\mathcal{F}$. Seguidamente, $\mathcal{F}$ se proyecta en un espacio tridimensional a partir de la imagen de profundidad, tomando como sistema de coordenadas la pose del robot.

Esta proyección tridimensional puede interpretarse como una nube de puntos en la que cada punto tiene asociado un *embedding*. Además de esto, a cada punto también se le puede asociar una medida de incertidumbre, que refleja la fiabilidad de la observación realizada. Este aspecto se describe con mayor detalle más adelante; por ahora, basta con considerar que cada punto tiene asociado un *embedding* y una medida de incertidumbre.

A continuación, la representación local asociada a la imagen $\mathcal{I}$ se debe incorporar al mapa probabilístico global ($\mathcal{M}$). Para ello, la nube de puntos se proyecta nuevamente, esta vez sobre la malla bidimensional del mapa $\mathcal{M}$. Esto es posible porque se conoce la pose del robot, lo que permite situar el sistema de coordenadas local de la cámara dentro del sistema de coordenadas global del mapa. De forma intuitiva, este proceso puede imaginarse como si los puntos de la nube, una vez situados en el sistema de coordenadas global, ‘cayeran’ sobre la malla bidimensional del mapa, de manera similar a gotas de lluvia. Posteriormente, se determina en qué celda cae cada punto, para transferir su información a la celda correspondiente. Pueden darse tres situaciones. En primer lugar, si sobre una celda se proyecta un único punto, se copia en ella el *embedding* asociado, junto con su incertidumbre. En segundo lugar, si sobre una celda no se proyecta ningún punto, su contenido permanece sin modificar. Este caso se dará en aquellas regiones que no hayan sido observadas desde la pose de la cámara. Por último, si varios puntos se proyectan sobre una misma celda, *OneMap* combina sus características mediante una suma ponderada, teniendo en cuenta la incertidumbre asociada a cada observación.

En un instante posterior, el sistema recibe otro conjunto de imágenes, correspondientes a una nueva pose de la cámara. El proceso descrito anteriormente se repite, con la diferencia de que ahora el mapa está parcialmente poblado. En este caso, *OneMap* fusiona las nuevas observaciones con la información ya almacenada. Es decir, las observaciones anteriores no se descartan ni se sustituyen directamente por las nuevas. En su lugar, las nuevas observaciones se emplean para refinar la estimación almacenada en $\mathcal{M}$, ya sea reduciendo la incertidumbre asociada a observaciones previas o modificando el vector de características de una celda cuando la nueva evidencia así lo justifica.

Por ejemplo, un *embedding* obtenido a partir de una primera imagen podría indicar que una determinada celda del mapa contiene una silla, pero con una incertidumbre alta. Si a medida que se explora el entorno y se adquieren nuevas imágenes desde distintas poses, los *embeddings* correspondientes a la misma celda del mapa siguen indicando la presencia de una silla, entonces, la incertidumbre asociada tendería a bajar. En cambio, si imágenes posteriores producen interpre-

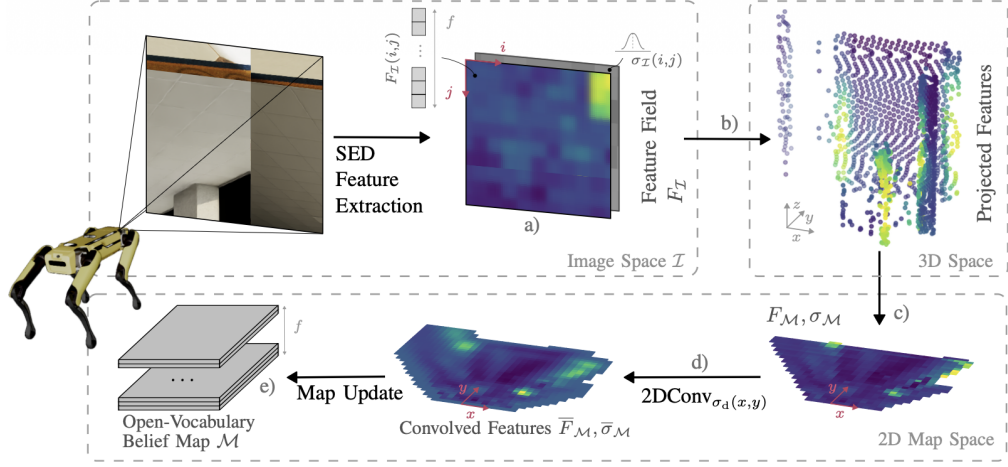


Figura 2.13: Esquema general del proceso de construcción del mapa semántico en *OneMap*. Reproducido de [32].

taciones contradictorias como una planta, una mesa, un sofá, etc., la incertidumbre de la celda posiblemente aumentaría. Finalmente, si en lugar de una silla las observaciones posteriores indican repetidamente la presencia de una taza roja, la representación vectorial de la celda en cuestión podría modificarse progresivamente conforme se incorporan nuevas observaciones. Este proceso se realiza de forma probabilística, actualizándose los valores de $\mathcal{M}$ en función de la incertidumbre asociada al vector de características. La metodología completa de *OneMap* se resume en la Figura 2.13.

Como se ha mencionado, cada punto del mapa tiene asociado una varianza. *OneMap* modela tres fuentes independientes de incertidumbre procedentes de las mediciones y operaciones realizadas. A partir de ellas, se estima una varianza para cada vector de la nube de puntos, que representa la confianza del sistema en la observación correspondiente. Las fuentes de incertidumbre consideradas son:

1. **Extractor de características:** las características correspondientes a puntos muy lejanos a la cámara pueden no contar con resolución suficiente para ser interpretadas correctamente. Del mismo modo, en las características correspondientes a puntos muy cercanos a la cámara puede faltar contexto visual.
2. **Fuga proyectiva de características entre *patches*:** un alto gradiente de profundidad indica un cambio repentino en la geometría de la escena, por ejemplo, al pasar de un objeto próximo a la cámara a una región situada en el

fondo. En estas zonas de transición, la información proyectada resulta menos fiable, ya que las características de los objetos cercanos pueden fugarse hacia el fondo, y viceversa.

3. **Incertidumbre en la localización de las características:** las cámaras de profundidad presentan menos exactitud a mayores distancias, dando lugar a proyecciones menos fiables.

2.4.3. Búsqueda de objetos

El apartado anterior ha descrito el proceso mediante el cual las observaciones realizadas por el robot se procesan e integran en un mapa semántico. A continuación, se especifica cómo la información recopilada guía el proceso de exploración.

En el contexto de una búsqueda de objetos, se parte de una consulta en lenguaje natural introducida por el usuario, como podría ser «taza roja». Para la consulta dada, se emplea el codificador de texto de CLIP para generar una representación equivalente en forma de *embedding*. Este *embedding* se encuentra dentro del espacio de *embeddings* de CLIP, por lo que la consulta puede representarse como un vector Q , tal que

$$Q \in \mathbb{R}^f$$

A continuación, este vector se compara con los *embeddings* almacenados en el mapa $\mathcal{M}$ mediante la similitud de coseno. Se obtiene, por tanto una puntuación asociada a cada celda del mapa, que indica la similitud entre la consulta y la información observada en ese punto del espacio. Esto se puede interpretar como un mapa de similitud dependiente de la consulta, denotado como

$$\mathcal{S}_Q \in \mathbb{R}^{n_x \times n_y}$$

donde cada valor,

$$\mathcal{S}_Q(x, y) \in \mathbb{R}$$

representa la similitud entre la consulta Q y la información semántica almacenada en la celda (x, y) . Además, con el propósito de guiar el proceso de navegación, *OneMap* introduce el parámetro $\sigma_E(x, y)$, que representa una estimación de varianza análoga a $\sigma(x, y)$. La única diferencia es que este parámetro puede reiniciarse durante el proceso de exploración.

A partir de esta base, *OneMap* genera cuatro mapas binarios. Un mapa binario puede entenderse como un mallado en el que cada celda toma un valor de cero o

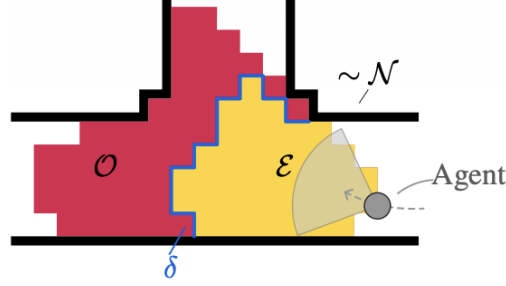


Figura 2.14: Mapas observado $\mathcal{O}$, semánticamente explorado $\mathcal{E}$ y no navegable $\sim \mathcal{N}$, junto con la frontera δ entre $\mathcal{E}$ y $\mathcal{O}$. Reproducido de [32].

uno en función de un criterio específico. Dicho criterio varía para cada uno de los mapas binarios considerados. Estos se muestran también en la Figura 2.14.

- **Mapa observado, $\mathcal{O}$:** corresponde a aquellas celdas que hayan sido observadas por cualquier actualización.
- **Mapa semánticamente explorado, $\mathcal{E}$:** corresponde a aquellas celdas que tenga una varianza $\sigma(x, y)$ suficientemente baja como para considerarse precisas y descriptivas del entorno. Es un subconjunto de $\mathcal{O}$.
- **Mapa de regiones ya buscadas, $\mathcal{C}$:** Es equivalente a $\mathcal{E}$, pero aplicado a una consulta específica. Es decir, este mapa representa las zonas que ya han sido exploradas durante la búsqueda asociada a la consulta concreta y se reinicia antes de procesar la siguiente consulta. Esto corresponde celdas que tenga una varianza $\sigma_E(x, y)$ suficientemente baja.
- **Mapa navegable, $\sim \mathcal{N}$:** corresponde a aquellas celdas libres de obstáculos.

A partir de estos mapas binarios se calculan dos objetivos de navegación:

1. Zonas observadas, pero no semánticamente exploradas, correspondientes a la frontera δ dada por $\mathcal{O} - \mathcal{E}$. Es decir, celdas que ya han sido vistas, pero cuyo contenido semántico todavía no se considera fiable.
2. Zonas semánticamente exploradas en consultas previas, pero no buscadas durante la tarea actual, $\mathcal{E} - \mathcal{C}$. Es decir, zonas de las que ya se dispone de contenido semántico fiable procedente de exploraciones anteriores, pero hacia las que todavía no se ha navegado en el contexto de la consulta actual. Esto permite seleccionar como objetivo de búsqueda regiones previamente observadas que presentan una alta similitud con la consulta.

Las celdas correspondientes a ambos objetivos representan regiones de posible interés. Para determinar cuál de ellas resulta más prometedora en el contexto de la consulta, se puntúan en función del mapa de similitud $\mathcal{S}_Q$. Se selecciona aquella con mayor puntuación y se planifica una trayectoria libre de colisiones hasta ella mediante el algoritmo A* [37]. Este proceso se lleva a cabo a medida que el agente cartografía el mapa semántico $\mathcal{M}$, hasta detectar el objeto buscado. Para ello, el agente emplea un identificador de objetos. Un objeto se considera detectado cuando el identificador confirma su presencia y cuando el mapa de similitud $\mathcal{S}_Q$ presenta valores por encima del percentil 95 para dicha región. Una vez detectado el objeto, el agente puede navegar hacia él para finalizar la exploración.

2.4.4. Resultados

Los autores de *OneMap* han evaluado su metodología en dos tipos de tareas de navegación. La primera de ellas, la navegación de un único objeto, requiere que el agente se desplace desde una posición inicial hasta una instancia del objeto buscado. Por otra parte, en tareas de navegación multiobjeto, tras navegar hasta una instancia del objeto buscado, se le proporciona al agente una nueva consulta. En ambos casos, la navegación se considera exitosa si el agente termina a menos de 1,5 m de una instancia del objeto buscado.

Para evaluar el rendimiento del agente *OneMap* propone los siguientes *benchmarks* para la navegación de un único objeto,

1. **Success Rate (SR)**: el porcentaje de ejecuciones en las que el agente encuentra satisfactoriamente el objeto especificado.
2. **Success weighted by Path Length (SPL)**: si la búsqueda es satisfactoria, el SPL se calcula como la razón entre la longitud de la trayectoria óptima y la longitud de la trayectoria recorrida por el agente. En caso contrario, el SPL toma valor cero.

Del mismo modo, para la navegación multi-objeto, se evalúa

1. **Success Rate (SR)**: el porcentaje de ejecuciones en los que el agente encuentra satisfactoriamente todos los objetos especificados.
2. **Success weighted by Path Length (SPL)**: es similar al SPL definido para navegación de un único objeto. En este caso, solo toma un valor distinto de cero si se encuentran todos los objetos. La trayectoria óptima se calcula como la suma de las trayectorias más cortas hacia cada uno de los objetos sucesivos.

3. **Progress (PR)**: la fracción de objetos encontrados en un mismo intento.
4. **Progress weighted by Path Length (PPL)**: una versión del PR que también tiene en cuenta la eficiencia de la trayectoria.

Los resultados obtenidos por *OneMap* tanto en tareas de navegación de un único objeto como en navegación multiobjeto se presentan en los Cuadros 2.3 y 2.4. Concretamente, estos resultados se obtienen en simulación, empleando el entorno de evaluación HM3D. En ellos se evalúa el desempeño de *OneMap* a partir de las métricas establecidas y se compara con otros modelos.

Método	Loc.	Sem.	SPL	SR
PIRLNav	✓	✓	27.1	64.1
ZSON	✓	✓	12.6	25.5
ESC	✗	✗	22.3	39.2
VLFM	✓	✗	30.4	52.5
<i>OneMap</i>	✗	✗	37.4	55.8

Cuadro 2.3: Resultados de navegación de un único objeto.

Método	Loc.	Sem.	SPL	SR	PPL	PR
VLFM	✓	✗	19.90	44.92	24.87	57.34
<i>OneMap</i>	✗	✗	27.77	54.24	32.68	65.54

Cuadro 2.4: Resultados de navegación multiobjeto.

En las tareas de un único objeto, *OneMap* alcanza un SPL superior a otros métodos *zero-shot* como VLFM y ESC, indicando una mejora en la eficiencia de la trayectoria frente a estos métodos. En navegación multiobjeto, tan solo se compara con VLFM, obteniendo también mejores resultados en todas las métricas. Los resultados señalan que el modelo PIRLNav supera a *OneMap* en tasa de éxito (SR) en tareas de navegación de un único objeto. Sin embargo, cabe recalcar que PIRLNav es un enfoque entrenado específicamente para la tarea. Este requiere datos de entrenamiento adicionales y demostraciones humanas, equivalentes a aproximadamente 2378 horas de anotación. En cambio, *OneMap* logra resultados competitivos en condiciones *zero-shot*, sin requerir entrenamiento específico para la tarea ni para el entorno evaluado.

Finalmente, la Figura 2.15 resume la métrica SPL en función del número de objetos buscados. La brecha de rendimiento entre *OneMap* y VLFM aumenta a medida que crece el número de objetos de la tarea, indicando que *OneMap* aprovecha la información recopilada durante consultas anteriores.

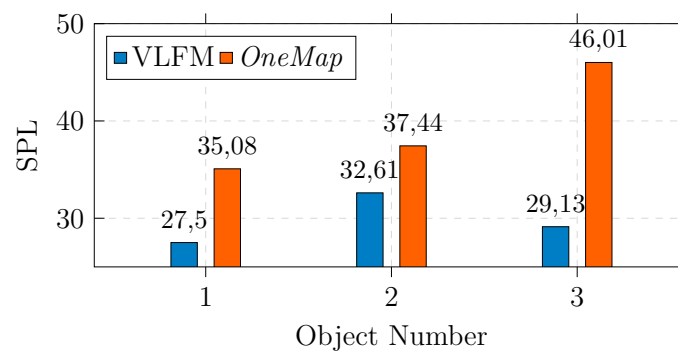


Figura 2.15: Comparación del SPL obtenido por VLFM y *OneMap* en función del número de objetos.

Capítulo 3

Descripción de los modelos desarrollados

En este capítulo, se describe la metodología empleada para abordar la pregunta de investigación, que se recuerda nuevamente a continuación:

¿En qué medida el idioma condiciona el rendimiento y la eficiencia energética de modelos de lenguaje y visión-lenguaje utilizados como componentes perceptivos en tareas de navegación semántica autónoma?

El capítulo se organiza en cuatro secciones, correspondientes a distintas líneas de análisis experimental desarrolladas durante el trabajo. Esta forma de estructurar la metodología, permite mostrar el desarrollo del trabajo como una evolución progresiva, en la que cada bloque experimental se apoya en los resultados y motivaciones del anterior. De este modo, la organización refleja también el orden cronológico en que se han llevado a cabo dichos experimentos.

3.1. Evaluación multilingüe de rendimiento y energía en LLMs

3.1.1. Introducción

Esta sección presenta un estudio multilingüe del rendimiento y del consumo energético de los modelos de lenguaje de gran tamaño. Se prescinde de momento de la componente visual introducida en secciones anteriores, centrando el análisis en la dimensión textual. Esto permite abordar de forma independiente el impacto del idioma en distintos aspectos de los LLMs, dejando fuera inicialmente la complejidad asociada a la multimodalidad. De este modo, la presente sección

introduce de forma gradual el enfoque metodológico que guía el desarrollo del trabajo, estableciendo una base para las secciones posteriores. En ellas, se incorporan progresivamente nuevos elementos de análisis hasta llegar al estudio de modelos visión-lenguaje en contextos relacionados con la navegación semántica.

La evaluación de LLMs en la literatura se ha abordado tradicionalmente, y continúa abordándose en gran medida, mediante *benchmarks* centrados en métricas de precisión. Dichos enfoques ponen el foco en la tasa de acierto o en la capacidad de razonamiento de un modelo, mientras que el consumo energético asociado a los procesos de inferencia queda relegado a un segundo plano. Se podría argumentar que este tipo de evaluación ofrece una visión limitada, pues el crecimiento de los sistemas de inteligencia artificial ha situado la eficiencia energética como una de las principales consideraciones en su evaluación y despliegue. A modo de contextualización, se estima que el consumo energético de los centros de datos representará aproximadamente el 3 % del consumo eléctrico global en 2030 [38].

Para tratar de cerrar la brecha entre la evaluación del rendimiento y la evaluación del consumo energético, han surgido iniciativas como *AI Energy Score* [39], que proponen métricas estandarizadas para medir la eficiencia energética de distintos modelos y tipos de tareas. Hasta el momento, estos marcos de evaluación se han empleado principalmente en tareas simples, como la generación de texto o la clasificación, y no contemplan, por ejemplo, aplicaciones multilingües. Esta limitación resulta especialmente relevante: no es suficiente evaluar un modelo únicamente en inglés y asumir que los resultados obtenidos representan su comportamiento en todos los idiomas, ya que los perfiles de consumo energético podrían variar de forma significativa entre lenguas.

Para abordar esta cuestión, se presenta una evaluación conjunta de energía y rendimiento de cinco modelos de lenguaje en trece idiomas distintos. Se emplea un conjunto de 79 problemas mediante el cual se evalúan dichos modelos. El análisis se realiza bajo una configuración de *hardware* controlada basada en GPUs NVIDIA H200, registrando tanto métricas de rendimiento, como la *accuracy*, como el consumo energético asociado al proceso de inferencia.

3.1.2. Configuración experimental

Modelos evaluados

Se evalúan cinco modelos distintos, seleccionados para representar una variedad de arquitecturas, tamaños (número de parámetros) y longitudes de ventana de contexto. En concreto, se analizan: Falcon-Mamba-7B, Mistral-7B-Instruct-v0.1, Phi-3-mini-128k-instruct, Phi-4-mini-instruct y Qwen2-7B-Instruct. El Cuadro 3.1 resume algunas de sus especificaciones.

Cuadro 3.1: Especificaciones principales de los modelos evaluados.

Modelo	Params. (B)	Arquitectura	Contexto (K)	FP16 (GB)
Falcon-Mamba-7B	7.27	SSM (Mamba)	8	~14
Mistral-7B-Instruct-v0.1	7.24	Transformer+GQA	8	~14
Phi-3-mini-128k-instruct	3.82	Transformer	128	~8
Phi-4-mini-instruct	3.84	Transformer+GQA	128	~8
Qwen2-7B-Instruct	7.07	Transformer+GQA	32	~14

Nota. Params. (B): número aproximado de parámetros del modelo, expresado en *billions* o miles de millones. Contexto: número máximo de *tokens* de la ventana de contexto. FP16 (GB): memoria aproximada necesaria para almacenar los pesos del modelo en precisión FP16.

Conjunto de datos empleado

Para evaluar los modelos, se solicita a cada uno de ellos que responda a una serie de 79 preguntas, procedentes de [40] y que constituyen el conjunto de datos empleado. Las preguntas planteadas no son inmediatas y requieren una serie de pasos de razonamiento intermedio para producir una respuesta determinada, también incluida en el conjunto de datos. A modo de ejemplo, se muestra una de las entradas:

Ejemplo de entrada del conjunto de datos

Problema	Determine los puntos de inflexión de $f(x) = x^4 - 4x^3 + 6x^2$.
Solución	No hay puntos de inflexión.
Pasos	<ol style="list-style-type: none"> 1. Calcular $f''(x) = 12x^2 - 24x + 12 = 12(x - 1)^2 \geq 0$ para todo x. 2. La concavidad nunca es negativa y no cambia de signo; $f''(1) = 0$, pero sin cambio de signo. 3. Por tanto, f no tiene puntos de inflexión.
Categoría	Calculus
Dificultad	Hard

CAPÍTULO 3. DESCRIPCIÓN DE LOS MODELOS DESARROLLADOS

De todos estos campos, al modelo tan solo se le proporciona el correspondiente al enunciado del problema. La respuesta generada, así como los pasos intermedios elaborados por el sistema, se comparan posteriormente con la solución y los pasos de referencia para puntuar la respuesta producida. Como se muestra, las preguntas también están etiquetadas según su categoría –por ejemplo, física, química o estadística– y según su nivel de dificultad, que puede ser fácil, medio o difícil. Estas etiquetas permiten estudiar los resultados en función del tipo de problema y de su complejidad.

El conjunto de datos, originalmente en inglés, fue traducido a las 13 lenguas estudiados: Árabe (ar), catalán (ca), alemán (de), inglés (en), español (es), euskera (eu), francés (fr), italiano (it), luxemburgués (lb), portugués (pt), ruso (ru), chino simplificado (zh-cn) y chino tradicional (zh-tw). Esta colección de idiomas cubre una variedad amplia de familias lingüísticas, sistemas de escritura y niveles de disponibilidad de recursos digitales.

Cabe destacar además que las traducciones se realizaron procurando preservar la estructura semántica y la dificultad de los problemas originales, con el fin de mantener la comparabilidad entre idiomas.

Hardware empleado

Todas las evaluaciones desarrolladas se han llevado a cabo en una tarjeta gráfica NVIDIA H200. De este modo, se elimina la variabilidad introducida por el hardware en las distintas combinaciones de idioma y modelo. Finalmente, al emplear hardware idéntico en cada evaluación se asegura que las mediciones de consumo energético reflejan el verdadero coste computacional de cada modelo en el proceso de inferencia.

Proceso de evaluación

A partir de lo expuesto hasta ahora, el procedimiento de evaluación no presenta especial complejidad. Se tiene un total de 65 combinaciones de modelo e idioma, resultantes de evaluar cinco modelos en trece idiomas. En cada caso, se proporciona al modelo, de forma individual, el enunciado de cada una de las 79 preguntas. Con el objetivo de aumentar la robustez de la evaluación y reducir el impacto de posibles anomalías, cada inferencia se repite tres veces. Es decir, cada combinación de modelo e idioma resuelve cada una de las preguntas un total de tres veces. Se obtiene, por tanto, un total de

$5 \text{ modelos} \times 13 \text{ idiomas} \times 79 \text{ preguntas} \times 3 \text{ repeticiones} = 15\,405$
ejecuciones de inferencia independientes. En cada caso, el modelo trata de res-

ponder al enunciado mediante un fragmento de texto en lenguaje natural, posiblemente incluyendo una serie de pasos de razonamiento seguidos de una solución final. Para puntuar la calidad de estas respuestas, se comparan tanto los pasos intermedios como la solución final generados por el modelo con los pasos y la solución de referencia incluidos en el conjunto de datos. Este proceso se realiza de forma automática, transformando ambas respuestas en *embeddings* y comparándolas posteriormente mediante su similitud de coseno. Los valores obtenidos como resultado de esta computación permiten calcular distintas métricas de desempeño, que se describen con mayor detalle en apartados siguientes. Durante el proceso de inferencia de cada ejecución, también se registran medidas correspondientes al consumo energético.

Métricas de energía

El consumo de potencia de la GPU se supervisa mediante NVML (*NVIDIA Management Library*), que permite acceder a las lecturas de potencia instantánea de la tarjeta gráfica. Las lecturas se muestrean en intervalos de 100 ms y se almacenan junto con su marca temporal. Con ello, se estima la energía total consumida en Wh durante un periodo de tiempo determinado mediante integración numérica:

$$E_{GPU} = \frac{1}{3600} \int_{t_0}^{t_1} P_{GPU}(t) dt \approx \frac{1}{3600} \sum_o P_o \cdot \Delta t_o$$

donde P_o representa la potencia instantánea registrada en el intervalo o y Δt_o el intervalo de muestreo correspondiente.

Para cada combinación modelo-idioma se calculan también, a partir de las estimaciones energéticas, las siguientes métricas:

- E_{total} : energía total consumida durante la evaluación del conjunto de 79 problemas; expresada en Wh.
- E_{1K} : energía consumida por cada 1000 consultas, $E_{1K} = E_{\text{total}} \times (1000/N_{\text{prompts}})$, donde $N_{\text{prompts}} = 79 \times 3 = 237$.
- $\bar{P}$: potencia media registrada durante la evaluación.
- $P_{\text{máx}}$: potencia máxima registrada durante la evaluación.
- T_{wall} : tiempo total transcurrido durante la evaluación.
- E_{prompt} : energía media consumida por consulta, $E_{\text{prompt}} = E_{\text{total}}/N_{\text{prompts}}$.

Métricas de rendimiento

Para evaluar el rendimiento de cada combinación modelo-idioma y capturar dimensiones complementarias de calidad de razonamiento, se proponen las siguientes métricas:

- Puntuación global, $S_{overall} \in [0, 1]$: fracción de problemas para los cuales el modelo produce una respuesta correcta, tomando como referencia las soluciones incluidas en el conjunto de datos. Esta métrica refleja la capacidad del modelo para resolver los problemas planteados y se emplea como métrica principal de rendimiento.
- Precisión por pasos, $S_{step} \in [0, 1]$: mide hasta qué punto el razonamiento intermedio del modelo es correcto, independientemente de la respuesta final. Para ello, los pasos intermedios generados se comparan con los pasos de referencia. Esta métrica permite diferenciar los modelos que alcanzan la solución correcta mediante un razonamiento lógico de aquellos que lo hacen a pesar de presentar errores intermedios.
- Consistencia, $C \in [0, 1]$: desviación estándar de las puntuaciones obtenidas por problema en las tres ejecuciones de inferencia. Valores bajos de esta métrica indican que la combinación modelo-idioma proporciona salidas similares en distintos intentos ante una misma pregunta. Valores altos, en cambio, sugieren una mayor variabilidad en las respuestas generadas.

AI Energy Score

Con el objetivo de integrar el rendimiento y el consumo energético de un modelo en un único indicador, se introduce el concepto de *AI Energy Score*. Se define esta métrica como la razón entre la puntuación global del modelo y el consumo energético asociado a 1000 consultas. Se aplica, además, un factor de escala de 10^3 con el propósito de obtener valores en un rango numérico más conveniente, ya que la razón original suele presentar valores del orden de 10^{-3} o 10^{-4} :

$$\text{AI Energy Score} = \frac{S_{overall}}{E_{1K}} \times 10^3$$

Esta formulación está alineada con la iniciativa *AI Energy Score* [39], que busca establecer clasificaciones estandarizadas de eficiencia energética para modelos de IA.

3.2. Evaluación multilingüe de rendimiento y energía en VLMs

3.2.1. Introducción

Como continuación natural de la metodología anterior, este bloque extiende el marco de evaluación conjunta rendimiento-energía al análisis de modelos de visión-lenguaje. Se considera que el estudio de dichos aspectos en LLMs aporta una primera visión del comportamiento de estos sistemas en tareas multilingües; por esa razón, se ha tomado como punto de partida. No obstante, las conclusiones obtenidas no deberían extrapolarse necesariamente a modelos de visión-lenguaje. De aquí surge la necesidad de abordar un estudio de forma independiente. Asimismo, la extensión a modelos multimodales permite acercar este marco de evaluación al tipo de aplicación planteada en el artículo de *OneMap*.

El presente estudio tiene una motivación muy similar a la detallada en la sección anterior. Como se ha reiterado a lo largo de la memoria, el coste computacional y energético asociado a la inferencia en modelos de inteligencia artificial supone un porcentaje significativo y creciente del consumo eléctrico global. A pesar de ello, como también se ha insistido, las métricas de energía están infrarrepresentadas en la evaluación de dichos modelos. Esta limitación es especialmente relevante, ya que, en muchas aplicaciones basadas en sistemas de IA, la viabilidad práctica depende en gran medida del consumo energético. La ausencia de análisis energético adquiere aún mayor importancia en los denominados despliegues *edge*, donde la inferencia se ejecuta localmente en dispositivos como drones, robots o plataformas embebidas; todos ellos sujetos a estrictas restricciones operativas de energía, memoria y capacidad computacional. El despliegue *edge* es especialmente relevante en el contexto de esta memoria, considerando que los VLMs están pasando de ejecutarse únicamente en centros de datos centralizados a desplegarse de forma distribuida.

La necesidad de incorporar métricas de energía en la evaluación de VLMs resulta más evidente en tareas multilingües, donde el consumo energético por consulta no se puede asumir independiente del idioma. Es decir, dada la disparidad lingüística de estos modelos, cabría esperar que consultas formuladas en inglés requieran menos recursos computacionales que consultas equivalentes en árabe, por ejemplo. En la literatura, este último se considera un idioma de bajos recursos (*low-resource language*), al igual que el euskera o el luxemburgués. El término hace referencia a lenguas con una disponibilidad limitada de recursos lingüísticos digitales, lo que dificulta el entrenamiento de modelos de inteligencia artificial para tareas de procesamiento del lenguaje natural en dichas lenguas. Precisamente por esta razón, el

desempeño de los modelos visión-lenguaje en tareas multilingües no debe evaluarse únicamente mediante *benchmarks* centrados en el inglés, pues estos pueden ocultar o subestimar la disparidad lingüística existente entre idiomas. Ciertamente, habrá modelos capaces de realizar tareas multilingües mejor que otros; por ello, en este estudio se evalúan cinco modelos con distintas arquitecturas, diseños de codificador visual, corpus de entrenamiento y estrategias de integración visión-lenguaje. Las prácticas de evaluación tradicionales tienden a resumir estas diferencias en una única métrica de *accuracy*, ocultando otros parámetros relevantes para aplicaciones reales, como puede ser el consumo energético.

Para abordar lo anterior, se propone una evaluación multilingüe del rendimiento y del consumo energético de cinco VLMs. El experimento se realiza para el mismo conjunto de trece idiomas presentados en la sección anterior, dando lugar a 65 pares modelo-idioma. Sin embargo, en lugar de plantear problemas puramente textuales, las consultas incorporan una componente visual. Particularmente, se evalúa la capacidad del modelo para responder a distintas preguntas formuladas a partir de una imagen de entrada. El conjunto de datos empleado para dichas imágenes, consiste de un conjunto de fotografías de escenarios de conducción, procedentes del *dataset* BDD100K [41]. De este modo, la evaluación se plantea desde un contexto de navegación autónoma, tomando como referencia el enfoque de *OneMap*.

3.2.2. Configuración experimental

Modelos evaluados

En este experimento se evalúan cinco modelos de visión-lenguaje: InternVL2-8B, Qwen2-VL-7B-Instruct, LLaVA-1.5-7B, LLaVA-1.6-Mistral-7B y Phi-3-Vision. Estos modelos representan una variedad de arquitecturas, diseños de codificador visual y corpus de entrenamiento. Se ha tratado de seleccionar modelos con un número de parámetros relativamente similar, situado aproximadamente entre 4,2 y 8B, con el fin de excluir modelos de mayor tamaño cuya implementación local en agentes autónomos resultaría poco razonable. El Cuadro 3.2 resume algunas de sus especificaciones.

Cuadro 3.2: Especificaciones principales de los modelos visión-lenguaje evaluados.

Modelo	Params. (B)	Modelo base	Contexto (K)	FP16 (GB)
InternVL2-8B	~8.0	InternLM2.5-7B / InternVL	8	~16
Qwen2-VL-7B-Instruct	~7.0	Qwen2-7B	32	~14
LLaVA-1.5-7B	~7.0	Vicuna-7B	4	~14
LLaVA-1.6-Mistral-7B	~7.0	Mistral-7B-Instruct	32	~14
Phi-3-Vision-128k-Instruct	~4.2	Phi-3-mini	128	~8



Figura 3.1: Ejemplo de imagen de conducción extraída del conjunto de datos BDD100K [41].

Conjunto de datos empleado

El conjunto de datos empleado en esta evaluación se ha adaptado a partir de BDD100K [41]. BDD100K es un *dataset* de conducción autónoma compuesto por 100.000 imágenes capturadas desde vehículos en distintas áreas geográficas de Nueva York, San Francisco y otras regiones. Además, cubre distintos tipos de escenarios de conducción, incluyendo calles urbanas, zonas residenciales y autopistas. En particular, se ha realizado un muestreo de 210 imágenes individuales, seleccionadas con el fin de cubrir una diversidad geográfica, medioambiental y temporal suficiente para evaluar los modelos en distintos entornos de conducción. Más allá de esto, el conjunto presenta una variedad de condiciones de circulación, incluyendo congestión de tráfico, presencia de peatones, distinta calidad de las vías y escenas capturadas tanto de día como de noche. Las imágenes han sido obtenidas con una resolución de 720p; suficiente para que su contenido sea, en principio, interpretable por los modelos evaluados. A modo de ejemplo, la Figura 3.1 muestra una imagen procedente del conjunto BDD100K.

Debido al considerable tiempo de ejecución asociado a la inferencia, no se empleó el conjunto completo de 100.000 muestras. En su lugar, se seleccionó de forma aleatoria un subconjunto fijo de 210 muestras, utilizado de manera constante para cada combinación modelo-idioma. Para garantizar la reproducibilidad de los experimentos, tanto el subconjunto empleado como la lista de identificadores correspondientes a las imágenes fueron congelados y almacenados de forma independiente.

Hardware empleado

Todas las evaluaciones desarrolladas se han llevado a cabo en una tarjeta gráfica NVIDIA H200. De este modo, se elimina la variabilidad introducida por el *hard-*

CAPÍTULO 3. DESCRIPCIÓN DE LOS MODELOS DESARROLLADOS

ware en las distintas combinaciones de idioma y modelo. Finalmente, al emplear *hardware* idéntico en cada evaluación se asegura que las mediciones de consumo energético reflejan el verdadero coste computacional de cada modelo en el proceso de inferencia.

Proceso de evaluación

La colección de muestras representa tan solo una parte de la información de entrada proporcionada a los modelos. Las imágenes constituyen la base sobre la que se formula el problema de evaluación, pero, para establecer un criterio de puntuación uniforme, se asocia a cada una de ellas una serie de preguntas. De este modo, la evaluación se plantea como una tarea de *Visual Question Answering* (VQA), en la que, dada una imagen, el modelo debe responder a consultas textuales formuladas en lenguaje natural.

Las preguntas se han diseñado con el objetivo de cubrir un espectro amplio de dificultad, desde tareas de descripción general de la escena hasta tareas de razonamiento orientadas a la navegación. Específicamente, se formulan cuatro preguntas, puntuadas de manera independiente y expresadas en los trece idiomas evaluados: Árabe (ar), catalán (ca), alemán (de), inglés (en), español (es), euskera (eu), francés (fr), italiano (it), luxemburgués (lb), portugués (pt), ruso (ru), chino simplificado (zh-cn) y chino tradicional (zh-tw). Este formato de evaluación permite analizar la comprensión multimodal del sistema bajo diferentes condiciones lingüísticas. Las preguntas empleadas se muestran a continuación:

Cuadro 3.3: Preguntas formuladas para la evaluación VQA.

Tarea	Dificultad	Prompt
Comprensión de escena	Fácil	Describe la escena de conducción en esta imagen. ¿Qué puedes ver en la carretera y a su alrededor?
Detección de vehículos	Media	¿Qué vehículos, peatones u objetos importantes puedes identificar en esta escena de conducción?
Condición de la vía	Media	Describe las condiciones de la carretera y el entorno. ¿Qué tipo de carretera es y cuáles son las condiciones climáticas y de iluminación?
Razonamiento de navegación	Difícil	Desde una perspectiva de conducción autónoma, ¿hay obstáculos o peligros que requieran atención? ¿Dónde están ubicados en relación con el vehículo?

Al igual que en el primer experimento, se tiene un total de 65 de pares modelo-idioma, resultantes de evaluar cinco modelos en trece idiomas. Para cada uno de ellos, cada combinación imagen-pregunta se proporciona al modelo de forma individual, generando una respuesta independiente en cada caso. Con el objetivo de aumentar la robustez de la evaluación y reducir el impacto de posibles anomalías, cada inferencia se repite tres veces. Es decir, cada combinación de modelo e idioma resuelve cada una de las preguntas un total de tres veces para cada imagen. Por tanto, se obtiene un total de

$$5 \text{ modelos} \times 13 \text{ idiomas} \times 210 \text{ imágenes} \times 4 \text{ preguntas} \times 3 \text{ repeticiones} = 163\,800$$

ejecuciones de inferencia independientes. En cada caso, el modelo genera una respuesta en forma de texto en lenguaje natural. Para poder comparar la calidad de las respuestas entre modelos e idiomas, se asigna una puntuación a cada una de ellas. Dado que cada pregunta se centra en un aspecto distinto de la interpretación de imágenes, las respuestas correspondientes se evalúan mediante criterios específicos que se describen en las secciones siguientes.

Al igual que en la evaluación de LLMs, durante el proceso de inferencia se registran también las mediciones de consumo energético asociadas.

Puntuación de las respuestas – comprensión de escena

El conjunto global del *dataset* está compuesto, en esencia, por imágenes de carreteras y escenarios de conducción. La tarea de comprensión de escena busca evaluar, precisamente, si el modelo es capaz de reconocer correctamente este contexto visual. Para ello, se produce un único *embedding* vectorial que describe de forma general la totalidad del corpus de evaluación. En particular, se representa en forma de *embedding* de la frase “A driving scene showing road, vehicles, surroundings, and traffic conditions”¹; empleándose esta como referencia fija. El *embedding* se obtiene mediante el modelo `all-MiniLM-L6-v2`, entrenado principalmente con datos de similitud textual en inglés.

En el proceso de inferencia, cada combinación idioma-modelo genera una respuesta textual para cada imagen (tres veces independientes). Se quiere averiguar, en cada caso, en qué medida el modelo logra identificar que la imagen corresponde a una escena de conducción. Esto supone comparar la respuesta producida por el modelo con la descripción genérica de referencia en inglés. Para realizar la comparación de forma matemática, se genera asimismo un *embedding* de la respuesta

¹En español: “Una escena de conducción que muestra la carretera, los vehículos, el entorno y las condiciones de tráfico”.

producida, empleando también el *encoder* `all-MiniLM-L6-v2`. Por tanto, la evaluación de similitud consiste en calcular la similitud de coseno entre el *embedding* de la respuesta, $\mathbf{e}(r)$, y el *embedding* de referencia, $\mathbf{e}(\hat{r})$,

$$s_{\text{scene}}(r, \hat{r}) = \frac{\mathbf{e}(r) \cdot \mathbf{e}(\hat{r})}{\|\mathbf{e}(r)\| \|\mathbf{e}(\hat{r})\|}$$

Como resultado, se obtiene una puntuación de similitud s_{scene} que permite medir la proximidad semántica entre la respuesta generada y la referencia².

Puntuación de las respuestas – detección de vehículos

Se parte también de la premisa de que las imágenes del corpus contienen elementos relacionados con la conducción, como coches, carreteras, señales de tráfico o peatones. En particular, se ha diseñado un conjunto fijo de siete términos expresados en inglés (*car, vehicle, road, lane, traffic, sign, pedestrian*)³ que representan las categorías más relevantes habitualmente presentes en las imágenes de BDD100K. Esta tarea evalúa la capacidad del modelo para identificar dichas categorías visuales en una imagen.

En el proceso de inferencia, cada combinación idioma-modelo genera una respuesta textual para cada imagen (tres veces independientes). Para puntuar la respuesta del modelo, se contabiliza el número de términos del conjunto fijo que aparecen en la respuesta generada. Esta cantidad se normaliza dividiéndola entre el número total de términos definidos. Por ejemplo, si la respuesta del modelo es:

“The image shows a car on a road with traffic signs.”

Al pasarla a minúsculas, aparecen las siguientes palabras del conjunto predefinido: *car, road, traffic* y *sign*. Por tanto, se le otorgaría al modelo una puntuación de $s_{\text{vehicle}} = \frac{4}{7} \approx 0,57$ en esta tarea. La fórmula de puntuación se define formalmente como:

$$s_{\text{vehicle}}(r) = \frac{|\{k \in \mathcal{K} : k \in \text{lower}(r)\}|}{|\mathcal{K}|}$$

²La ecuación de similitud de coseno ya se ha introducido en capítulos previos, donde se indicó que su rango teórico se encuentra entre $[-1, 1]$. Sin embargo, en el ámbito de comparación de *embeddings* textuales y con el *encoder* empleado, la gran mayoría de los valores observados se sitúan entre 0 y 1. Esto se debe a que, en muchos casos, se comparan conceptos semánticamente relacionados, cuyas representaciones vectoriales presentan direcciones similares. Los valores negativos aparecen únicamente en casos excepcionales, por ejemplo cuando el modelo no produce una respuesta válida o genera una salida fuera de contexto.

³En español: (coche, vehículo, carretera, carril, tráfico, señal, peatón).

Esta métrica asume que algunas respuestas en otros idiomas pueden incluir palabras en inglés o términos similares a aquellos presentes en el conjunto de palabras empleado.

Puntuación de las respuestas – condición de la vía

Esta métrica es idéntica a la empleada para la detección de vehículos, salvo por el conjunto léxico utilizado. En este caso, se define una serie de categorías que reflejan las condiciones de la superficie, la iluminación y ciertos atributos geométricos presentes en el conjunto de datos BDD100K. El conjunto fijo de términos está expresado en inglés e incluye las categorías *paved*, *asphalt*, *lane*, *clear*, *day*, *straight* y *urban* ⁴.

La fórmula empleada para calcular la puntuación es análoga a la utilizada en la tarea de detección de vehículos:

$$s_{\text{road}}(r) = \frac{|\{f \in \mathcal{F} : f \in \text{lower}(r)\}|}{|\mathcal{F}|}$$

Donde $\mathcal{F}$ representa el conjunto fijo de términos definidos para esta tarea.

Puntuación de las respuestas – razonamiento de navegación

El razonamiento de navegación se ha considerado la tarea más compleja de las cuatro descritas. En ella, el modelo no debe limitarse a identificar obstáculos o peligros, sino que también debe situarlos espacialmente en relación con el vehículo. Se valora, por tanto, que la respuesta incorpore indicadores espaciales sobre la posición de obstáculos u otros elementos relevantes. Para evaluar las respuestas generadas, se ha optado por una función de puntuación híbrida:

$$s_{\text{nav}}(r, \hat{r}) = \frac{1}{2} \left(\frac{|\{t \in \mathcal{T}_{\text{nav}} : t \in \text{lower}(r)\}|}{|\mathcal{T}_{\text{nav}}|} + s_{\text{scene}}(r, \hat{r}) \right)$$

donde $\mathcal{T}_{\text{nav}}$ representa un conjunto fijo de indicadores espaciales expresados en múltiples idiomas: *left*, *right*, *ahead*, *behind*, *lane*, *road*, *path*, *straight*, *turn*, *intersection*, *near*, *far* y *side* (inglés); *izquierda*, *derecha* y *adelante* (español); *gauche*, *droite* y *devant* (francés); *links*, *rechts* y *vorne* (alemán); *sinistra*, *destra* y *davanti* (italiano); y términos equivalentes de orientación espacial en chino.

La puntuación, se descompone en dos partes: una que calcula el porcentaje de términos espaciales empleados en la respuesta del modelo a partir de un conjunto multilingüe fijo; y otra que compara el *embedding* de la respuesta con una

⁴En español: (pavimentado, asfalto, carril, claro, día, recto, urbano)

referencia también fija. Concretamente, la respuesta de referencia empleada está formulada en inglés y enfatiza la localización espacial de obstáculos y peligros respecto a la trayectoria del vehículo: “*Spatial description of obstacles, hazards, and their positions relative to the vehicle path*”⁵

Puntuación global de rendimiento

Para cada imagen, cada combinación idioma-modelo genera tres respuestas independientes para cada pregunta planteada. A partir de las puntuaciones por pregunta definidas anteriormente, se calculan las siguientes métricas globales de rendimiento :

- Puntuación media por idioma, S_ℓ : se define, para cada par modelo-idioma, como la puntuación media obtenida entre todas las tareas e imágenes.
- Puntuación media por modelo, S_m : Se define, para cada modelo, como la puntuación media obtenida entre todas las tareas, imágenes e idiomas.
- Consistencia, σ_m : se define como la desviación estándar media de cada tarea, considerando todas las ejecuciones, idiomas e imágenes. Valores bajos de esta métrica indican que las salidas son más reproducibles y presentan menos variabilidad.

Comentario respecto a métricas de rendimiento empleadas

Las métricas empleadas se han seleccionado con el objetivo de cubrir un abanico razonable de aspectos relevantes para la navegación autónoma de vehículos. Para evaluar las respuestas, se ha optado por un enfoque de puntuación automático, en el que no interviene un evaluador humano. Una evaluación manual podría resultar útil, ya que permitiría considerar matices específicos y admitir respuestas más flexibles al no ceñirse a un conjunto léxico fijo. Sin embargo, incluso sin entrar en cuestiones de imparcialidad, el elevado número de consultas, 163.800 en total, hace que este tipo de evaluación no sea viable desde un punto de vista práctico.

Métricas de energía

Al igual que en el experimento anterior, se emplea un proceso auxiliar que registra las métricas de consumo de potencia de la tarjeta gráfica con una frecuencia de muestreo de 10 Hz. Dado que las evaluaciones se ejecutaron por modelo, la energía total consumida por cada uno se aproxima integrando la potencia instantánea media registrada durante cada ejecución.

⁵En español: Descripción espacial de obstáculos, peligros y sus posiciones relativas a la trayectoria del vehículo.

3.3. Evaluación completamente multilingüe de rendimiento y energía en VLMs

$$E_{\text{total, model}} = \bar{P} \cdot \frac{T}{3600} \quad [\text{Wh}]$$

donde T representa el tiempo total de ejecución de cada modelo, y $\bar{P}$ la potencia media calculada a partir de las muestras de potencia instantánea registradas⁶

Se calcula como métrica principal de energía, la energía por 1000 consultas. Para cada modelo, este resultado se obtiene como

$$E_{1K, \text{model}} = \frac{E_{\text{total, model}}}{N_{\text{queries}}} \times 10^3 \quad \left[\frac{\text{Wh}}{10^3 \text{ consultas}} \right]$$

donde el número de consultas realizadas por cada modelo corresponde a

$$13 \text{ idiomas} \times 210 \text{ imágenes} \times 4 \text{ preguntas} \times 3 \text{ repeticiones} = 32\,760$$

Finalmente, para estimar la energía consumida por idioma, el consumo total se distribuye proporcionalmente en función del tiempo de inferencia asociado a cada idioma, $T^{(l)}$. Esta descomposición asume un uso homogéneo de los recursos computacionales entre idiomas, lo cual resulta razonable dado que las consultas, imágenes y parámetros de inferencia mantienen el mismo formato y configuración experimental. Se tiene, por tanto:

$$E_{1K, \text{model}}^{(l)} = \frac{E_{\text{total, model}} \cdot (T^{(l)}/T)}{N_{\text{queries}}^{(l)}} \times 10^3 \quad \left[\frac{\text{Wh}}{10^3 \text{ consultas}} \right]$$

3.3. Evaluación completamente multilingüe de rendimiento y energía en VLMs

3.3.1. Introducción

La evaluación multilingüe de rendimiento y energía en VLMs, presentada en la sección anterior, se planteó inicialmente con el objetivo de analizar el efecto de la variabilidad lingüística en las respuestas de agentes multimodales ante tareas de VQA orientadas a contextos de navegación. No obstante, los resultados obtenidos, como se comentará en capítulos posteriores, mostraron diferencias entre idiomas mucho más pronunciadas de lo inicialmente esperado. Esta variabilidad se atribuyó a unas condiciones experimentales sesgadas, que favorecerían los resultados

⁶Esta formulación difiere de la empleada en el experimento anterior. No obstante, ambas son equivalentes en el límite de muestreo continuo.

ante consultas formuladas en inglés. Más que un experimento independiente planteado desde cero, esta sección corrige algunas de las limitaciones de la anterior, con el objetivo de obtener resultados más imparciales entre idiomas. Por este motivo, esta sección se ha denominado evaluación **completamente** multilingüe del rendimiento y del consumo energético en VLMs.

Aunque el experimento correspondiente a la sección anterior introducía cierto sesgo en la evaluación, se ha decidido incluirlo en esta memoria para mostrar también el impacto que pueden tener las evaluaciones no plenamente multilingües en el ámbito del procesamiento del lenguaje natural. Además, esto permite comparar los resultados procedentes de ambas metodologías.

Puesto que este experimento no presenta un cambio metodológico significativo respecto al anterior, la organización de esta sección es algo distinta a la que el lector habrá podido observar en secciones anteriores. En concreto, se limita a resaltar las debilidades de la metodología previa y los cambios propuestos para mitigarlas. El nuevo experimento se basa en dichos cambios.

3.3.2. Cambios realizados

Modelo de embeddings

Tanto las tareas de comprensión de escena como las de razonamiento de navegación se apoyan en un codificador de texto que transforma las respuestas producidas por los modelos, expresadas en lenguaje natural, en *embeddings*. De este modo, se pueden comparar con un *embedding* fijo de referencia. Originalmente el codificador empleado era **all-MiniLM-L6-v2**, entrenado principalmente con datos de similitud textual en inglés. Aunque el codificador presenta un comportamiento razonable en idiomas latinos de altos recursos, su rendimiento tiende a empeorar conforme aumenta la distancia léxica respecto al inglés; lo cual resultaba especialmente notable en idiomas como árabe, ruso y chino.

Esta revisión del experimento, sustituye el *encoder* **all-MiniLM-L6-v2** por **paraphrase-multilingual-MiniLM-L12-v2**, entrenado en un corpus que cubre más de 50 idiomas, incluyendo todos aquellos evaluados en el estudio. El codificador produce *embeddings* de la misma dimensión que el modelo anterior, por lo que supone un cambio mínimo en la metodología empleada. Se espera que, con este cambio, se generen representaciones vectoriales más significativas en los distintos idiomas.

Descripciones de referencia

3.3. Evaluación completamente multilingüe de rendimiento y energía en VLMs

Este cambio también hace referencia al tratamiento de los *embeddings* en las tareas de comprensión de escena y razonamiento de navegación. En la versión anterior, los *embeddings* resultantes de transformar cada respuesta en lenguaje natural se comparaban con un único *embedding* obtenido a partir de una referencia fija en inglés. En concreto, se empleaban las siguientes referencias:

“A driving scene showing road, vehicles, surroundings, and traffic conditions”

“Spatial description of obstacles, hazards, and their positions relative to the vehicle path”

para las tareas de comprensión de escena y razonamiento de navegación respectivamente. Por tanto, la similitud de coseno se calculaba siempre entre una respuesta potencialmente no inglesa y una referencia fija en inglés. Cabe esperar que esta configuración introdujera una penalización para las respuestas generadas en idiomas distintos del inglés.

Para mitigar este sesgo, se genera una traducción de cada frase de referencia para cada uno de los idiomas evaluados, de modo que la similitud de coseno se calcule siempre entre *embeddings* procedentes de la misma lengua. Por otra parte, en esta revisión se ha ajustado ligeramente la formulación de la respuesta de referencia correspondiente a la tarea de razonamiento y navegación:

“Spatial description of obstacles and hazards: position to the left, right, ahead or behind the vehicle along the lane or road”

Esta nueva expresión incorpora términos espaciales de forma más específica, dando lugar a comparaciones más adecuadas para la tarea. Como se ha indicado, la referencia se traduce a todos los idiomas evaluados.

Conjuntos de palabras

De forma similar, las tareas de detección de vehículos y condiciones de la vía empleaban conjuntos de palabras clave expresados únicamente en inglés.

$\{car, vehicle, road, lane, traffic, sign, pedestrian\}$
 $\{paved, asphalt, lane, clear, day, straight, urban\}$

Una vez más, esta revisión incorpora traducciones de dichos términos en todos los idiomas evaluados, de modo que las respuestas puedan puntuarse en el idioma correspondiente.

CAPÍTULO 3. DESCRIPCIÓN DE LOS MODELOS DESARROLLADOS

Puntuación híbrida

Este cambio es bastante relevante porque corrige un error crítico en la formulación de la puntuación híbrida, concretamente en la parte encargada de comprobar cuántas palabras de un conjunto de referencia aparecen en la respuesta proporcionada por el modelo.

Originalmente se disponía de una lista multilingüe de indicadores espaciales, formada por términos expresados en distintos idiomas: *left, right, ahead, behind, lane, road, path, straight, turn, intersection, near, far* y *side* (inglés); *izquierda, derecha* y *adelante* (español); *gauche, droite* y *devant* (francés); *links, rechts* y *vorne* (alemán); *sinistra, destra* y *davanti* (italiano); y términos equivalentes de orientación espacial en chino.

En lugar de identificar estos términos en la respuesta proporcionada por el modelo, un error en la formulación del código hacía que se identificaran en la respuesta de referencia: “*Spatial description of obstacles, hazards, and their positions relative to the vehicle path*”. Como resultado, el único término del léxico espacial detectado era *path*. Posteriormente, la métrica comprobaba si dicho término aparecía en la respuesta generada por el modelo, de modo que la componente léxica de la puntuación tomaba valor 1 si la respuesta contenía *path*, y valor 0 en caso contrario. Es decir, aunque la lista incluía términos como *izquierda, derecha, gauche* o *droite*, estos nunca llegaban a buscarse realmente en la respuesta del modelo.

Evidentemente este no es el comportamiento deseado; por lo que se ha corregido en esta versión. Particularmente, se ha aprovechado para introducir una serie de cambios. En primer lugar, la comparación pasa a realizarse directamente entre la respuesta del modelo y el conjunto de términos espaciales, sin considerar la respuesta de referencia en esta componente de la puntuación híbrida. En segundo lugar, se ha sustituido el conjunto léxico original (que mezclaba términos multilingües en una única lista) por diccionarios independientes definidos para cada uno de los idiomas evaluados.

Por ejemplo, supóngase que el conjunto original estuviera formado únicamente por los términos *derecha, droite*. Si el modelo generaba una respuesta en español e incluía correctamente la palabra *derecha*, pero no *droite*, podía verse penalizado por no contener el término en francés. Con el nuevo enfoque, esto no ocurre, ya que cada respuesta se evalúa exclusivamente con el diccionario correspondiente al idioma en el que se formula la consulta.

3.4. Evaluación del grounding visual multilingüe en el contexto de OneMap

La evaluación experimental concluye con este último estudio, donde se trata de aterrizar los conceptos abordados en los experimentos anteriores en una aplicación similar a la de *OneMap*. Quizá, llegados a este punto, no sea necesario reiterar el papel del consumo energético y del multilingüismo dentro del marco de evaluación de los sistemas multimodales. En su lugar, se dedicará este breve apartado a explicar cómo el presente experimento encaja dentro de la metodología de *OneMap*.

Como el artículo de *OneMap* abarca un contenido bastante diverso, esta sección se centrará especialmente en uno de los componentes de su *pipeline*, al que se le ha dado ya bastante relevancia a lo largo de la memoria. Con ello se hace referencia no a la construcción del mapa ni a los algoritmos empleados para que el robot se desplace sobre él, sino a aquellos procesos en los que intervienen los agentes multimodales para, a partir de imágenes, representar e interpretar el significado semántico. Dentro de esta descripción podemos situar el proceso de extracción de características, un proceso en el que *OneMap* se apoyaba en el codificador visual SED; pero también podemos situar el proceso de elaboración de un mapa de similitud a partir de una consulta en lenguaje natural.

En la sección anterior, se ha analizado la capacidad de distintos VLMs para interpretar imágenes procedentes de un entorno de conducción y responder a consultas sobre el contenido global de la escena. Esto resulta muy útil de por sí, pero, como se recordará, el *pipeline* de *OneMap* abordaba la extracción de características a nivel de *patches*. Este enfoque permitía situar espacialmente los distintos elementos observados, ayudando así a definir su localización dentro del mapa de la escena. Tras una serie de operaciones y proyecciones, se obtenía una representación discretizada del entorno en la que las características visuales asociadas a cada casilla se resumían mediante un *embedding*. Así, la selección de direcciones de exploración para la búsqueda de un objeto se reducía a identificar la casilla del mapa con mayor similitud respecto al *embedding* de una consulta expresada en lenguaje natural. En este caso se plantea un enfoque similar, evaluando si el proceso de generación de un mapa de similitud se ve condicionado por el idioma en el que se formula la consulta.

La estructuración del *pipeline* de *OneMap* responde a un esquema habitual en aplicaciones multimodales, basado en la combinación de un *encoder* visual y una rama de procesamiento textual. *OneMap*, por ejemplo, empleaba el codificador visual SED para generar un campo de *embeddings* a partir de una imagen; mien-

tras que la consulta en lenguaje natural se proyectaba al mismo espacio vectorial mediante la rama textual de un modelo de visión-lenguaje. En esta sección se propone una versión reducida del *pipeline* empleado por *OneMap*, con el objetivo de evaluar la interacción entre ambos componentes sin entrar en las operaciones y proyecciones más complejas de la metodología original.

El procedimiento general del experimento se describe más adelante en esta sección. No obstante, con el objetivo de contextualizar el problema, se introduce previamente el siguiente escenario: Imagínese una imagen en la que aparece un coche situado en una carretera. Tras extraer su características, la imagen queda representada mediante un conjunto de vectores asociados a los distintos *patches*. Se desea localizar el coche comparando dichos vectores visuales con el *embedding* textual de una consulta como *coche*, *car* o *voiture*. La cuestión es si todas estas consultas localizan el objeto en la misma región de la imagen. Este proceso de localización dentro de una imagen se conoce como *grounding* visual.

3.4.1. Configuración experimental

Modelos evaluados

En este experimento se evalúan dos configuraciones CLIP multilingües. La primera combina una rama textual XLM-RoBERTa **base** con un codificador visual ViT-B/32, mientras que la segunda combina una rama textual XLM-RoBERTa **large** con un codificador visual ViT-H/14.

Cuadro 3.4: Especificaciones principales de los modelos evaluados.

Modelo	Identificador OpenCLIP	Encoder visual	Params. visuales
XLM-R base + ViT-B/32	xlm-roberta-base-ViT-B-32	ViT-B/32	~87M
XLM-R large + ViT-H/14	xlm-roberta-large-ViT-H-14	ViT-H/14	~632M

Existe, por tanto, una diferencia clara de escala entre las arquitecturas evaluadas. En la rama textual, por ejemplo, XLM-RoBERTa **base** constituye una versión más ligera que XLM-RoBERTa **large**. Esta diferencia existe también en la rama visual, donde ViT-B representa una configuración base, mientras que ViT-H corresponde a una configuración de mayor tamaño. Como se recoge en el Cuadro 3.4, ambas variantes difieren de forma significativa en el número de parámetros visuales. Además, las notaciones /32 y /14 hacen referencia al tamaño del *patch* empleado. Así, ViT-H/14 trabaja con *patches* de menor tamaño que ViT-B/32, lo que proporciona una discretización espacial más fina y permite realizar un *groun-*

ding visual más denso.

Esta diferencia de escala entre las arquitecturas empleadas es intencional, ya que permite analizar no solo el impacto del idioma en el proceso de *grounding* visual, sino también cómo dicho impacto varía según el tamaño y la capacidad representacional del modelo.

Conjunto de datos empleado

El conjunto de datos utilizado en esta evaluación es exactamente el mismo que el empleado en los experimentos de evaluación de VLMs. Es decir, se parte de un subconjunto fijo de 210 muestras procedentes de BDD100K [41]. Por tanto, las imágenes cubren distintos tipos de escenarios de conducción, incluyendo calles urbanas, zonas residenciales y autopistas.

Hardware empleado

Todas las evaluaciones desarrolladas se han llevado a cabo en una tarjeta gráfica NVIDIA H200. De este modo, se elimina la variabilidad introducida por el *hardware* en las distintas combinaciones de idioma y modelo. Finalmente, al emplear *hardware* idéntico en cada evaluación se asegura que las mediciones de consumo energético reflejan el verdadero coste computacional de cada modelo en el proceso de inferencia.

Proceso de evaluación

Se parte de un conjunto de imágenes $\mathcal{O} = \{O_1, \dots, O_N\}$, donde N corresponde al número de muestras. En el caso de este experimento, $N = 210$. Cada imagen se introduce en un codificador visual para obtener así una representación en forma de campo de *embeddings*, $V(O)$. En esta representación, cada *patch* queda asociado a un vector de dimensión d , de modo que

$$V(O) \in \mathbb{R}^{H \times W \times d}$$

Cabe destacar que la codificación visual es independiente del idioma. Por tanto, para una misma imagen, la representación vectorial $V(O)$ que se compara con cada consulta es la misma, independientemente del idioma en el que esta se formule.

La evaluación se realiza para un conjunto $\mathcal{L}$ de idiomas coherente con el empleado en los experimentos anteriores: árabe (ar), catalán (ca), alemán (de), inglés (en), español (es), euskera (eu), francés (fr), italiano (it), luxemburgués (lb), portugués (pt), ruso (ru), chino simplificado (zh-cn) y chino tradicional (zh-tw). De todos ellos, se toma el inglés como idioma de referencia, ℓ_0 .

Se dispone también de un conjunto finito $\mathcal{C}$ de etiquetas representativas de elementos visuales comunes en el conjunto de datos. Este conjunto se traduce a cada uno de los idiomas evaluados. El conjunto diseñado, en español, es:

{coche, camión, autobús, persona, peatón, semáforo, señal de tráfico, bicicleta, motocicleta, carretera, edificio}

Cada concepto, formulado en un idioma determinado, se codifica mediante una transformación T_ℓ , que lo proyecta a una representación vectorial de dimensión d :

$$T_\ell : \mathcal{C} \rightarrow \mathbb{R}^d$$

Se cuenta, por tanto, con un conjunto de 210 imágenes, 11 conceptos y 13 idiomas, dando lugar a 30.030 combinaciones imagen-concepto-idioma⁷. Para cada una de ellas, se obtiene un mapa de similitud $S_c^{(\ell)}$ mediante la comparación de los *embeddings* visuales asociados a los *patches* de la imagen con el *embedding* textual generado para la consulta correspondiente:

$$S_c^{(\ell)}(O)[u, z] = \frac{V(O)[u, z] \cdot T_\ell(c)}{\|V(O)[u, z]\| \|T_\ell(c)\|} \in [-1, 1]^{H \times W}$$

Aquí, u y z representan las distintas posiciones espaciales dentro del mapa de *patches* generado a partir de cada imagen.

Se reitera nuevamente que $V(O)$ se mantiene constante en todas las combinaciones correspondientes a una misma imagen. Por tanto, las variaciones observadas en el mapa de similitud dependen de la representación textual generada por $T_\ell(c)$.

Concordancia entre idiomas

Para valorar el rendimiento de los modelos, se opta por una evaluación relativa. Esto quiere decir que, en lugar de puntuar cada iteración en una escala absoluta, los resultados obtenidos en cada idioma se comparan con los resultados correspondientes en inglés. Por tanto, para cada combinación (O, c, ℓ) tal que $\ell \neq \ell_0$, se calcula la concordancia entre idiomas comparando el mapa de similitud obtenido en el idioma ℓ , $S_c^{(\ell)}(O)$, con el mapa de referencia en inglés, $S_c^{(\ell_0)}(O)$. A raíz de esto, se proponen cuatro métricas complementarias.

Métricas de rendimiento – IoU basada en máscaras de clústeres

Se quiere evaluar si las regiones del mapa que presentan una alta similitud semántica con respecto a una consulta en un idioma determinado son coherentes

⁷Se ha incluido en este cálculo la comparación del inglés consigo mismo.

con las regiones de alta similitud obtenidas en el mapa de referencia en inglés. Para ello, tanto en el mapa evaluado como en el mapa de referencia se aplica el mismo procedimiento. En primer lugar, se identifican los *patches* correspondientes a máximos locales de similitud. A partir de estos máximos, las regiones se expanden hacia *patches* vecinos que también presentan una similitud elevada. Para determinar si un *patch* vecino debe incorporarse a la región, se exige que su valor de similitud sea al menos el 80 % del valor del máximo local correspondiente. A continuación, se descartan los grupos formados por menos de tres *patches* contiguos. El resultado de este proceso es una máscara binaria para cada mapa, que indica qué zonas se consideran regiones de alta similitud semántica. Finalmente, se comparan las máscaras binarias obtenidas para cada uno de los dos mapas mediante la razón entre su intersección y su unión,

$$\text{IoU} = \frac{|M_c^{(\ell)}(O) \cap M_c^{(\ell_0)}(O)|}{|M_c^{(\ell)}(O) \cup M_c^{(\ell_0)}(O)|}$$

Esta métrica se conoce como *Intersection over Union* (IoU) y permite determinar si ambos idiomas identifican la misma región espacial, independientemente de la escala absoluta de los valores de similitud. Un valor de IoU igual a 1 indica que las máscaras coinciden perfectamente, mientras que un valor igual a 0 indica que no existe solapamiento entre ellas.

Métricas de rendimiento – IoU por percentil superior

Consiste en un análisis similar IoU basado en máscaras de clústeres. Sin embargo, en lugar de identificar máximos locales y expandir regiones a partir de ellos, la comparación se realiza para distintos percentiles $p \in \{90, 95, 99\}$. Por tanto, para $p=90$ se comparan aproximadamente el 10 % de los *patches* con mayor similitud, para $p=95$ el 5 % superior, y para $p=99$ el 1 % superior.

Métricas de rendimiento – Coeficiente de correlación de Spearman

El coeficiente de correlación de Spearman, $\rho^{(\ell)}(O, c)$ es una medida de correlación entre dos variables independientes. En el contexto empleado, se puede interpretar de la siguiente manera:

Tanto para el mapa de similitud del idioma evaluado $S_c^{(\ell)}(O)$ como para el mapa de similitud de referencia $S_c^{(\ell_0)}(O)$, se ordenan todos los *patches* según su valor de similitud con la consulta; de mayor a menor por ejemplo. Seguidamente, se comparan los órdenes obtenidos en ambos mapas, y el coeficiente de correlación de Spearman les atribuye un valor entre -1 y 1 según cuánto se parezcan. Un valor cercano a 1 indica que el orden obtenido es similar en ambos; un valor cercano a

0 indica que no existe una relación clara entre ambos órdenes; y un valor cercano a -1 indica que el orden es prácticamente inverso. En el contexto empleado, valores altos significan que ambos idiomas asocian la relevancia semántica a regiones espaciales parecidas.

Métricas de rendimiento – Razón de similitud máxima

Se obtiene como la razón entre el valor máximo de similitud del mapa del idioma evaluado, $S_c^{(\ell)}(O)$, y el valor máximo de similitud del mapa de referencia, $S_c^{(\ell_0)}(O)$,

$$\eta^{(\ell)}(O, c) = \frac{\max S_c^{(\ell)}(O)}{\max S_c^{(\ell_0)}(O)}$$

La razón de similitud máxima permite evaluar si el mapa del idioma evaluado alcanza niveles de similitud, o activación, comparables a los del mapa de referencia. Más allá de esto, esta métrica tiene una utilidad adicional cuando se interpreta de forma conjunta con la IoU. En aquellos casos en los que se observen resultados muy distintos entre idiomas, la razón de similitud máxima permite identificar y distinguir dos posibles situaciones:

1. Colapso de señal ($\eta \ll 1$, IoU baja): el mapa del idioma evaluado no alcanza valores de similitud comparables a los del mapa de referencia. Es decir, no es que no coincidan las regiones de alta similitud, sino más bien que dichas regiones no llegan a activarse.
2. Desalineamiento espacial ($\eta \approx 1$, IoU baja): ambos mapas presentan una activación semántica comparable, pero dicha activación se localiza en regiones distintas de la imagen.

Métricas de energía

Al igual que en experimentos anteriores, se emplea NVML (*NVIDIA Management Library*) para registrar las métricas de consumo de potencia de la tarjeta gráfica con una frecuencia de muestreo de 10 Hz. Se estima la energía total consumida, expresada en Wh, a partir de la potencia media registrada y del tiempo total de ejecución, siguiendo la misma formulación⁸ empleada en la Sección 3.2. A partir de ello, se obtiene el consumo energético por mil consultas, E_{1K} . Esta es la principal métrica energética empleada en el estudio y normaliza la energía total consumida en función del número de inferencias ejecutadas, entendiendo cada inferencia como una combinación de imagen, concepto e idioma.

⁸Esta formulación, basada en potencia media por tiempo de ejecución, difiere de la empleada en el primer experimento, basada en integración por muestra (Sección 3.1). No obstante, ambas aproximaciones son equivalentes en el límite de muestreo continuo.

Capítulo 4

Resultados y discusión

En este capítulo se presentan y analizan algunos de los resultados obtenidos a partir de las metodologías expuestas en el capítulo anterior. Dado el número de experimentos realizados y la variedad de características evaluadas, se ha obtenido un conjunto amplio de datos particularmente ilustrativos, a partir de los cuales se han elaborado distintas representaciones gráficas que permiten analizar de forma visual algunos de los aspectos más relevantes del estudio. A su vez, el número de gráficas elaboradas es elevado. Precisamente por esta razón, conviene puntualizar que no todas ellas serán mostradas en esta sección. En su lugar, se ha seleccionado cuidadosamente un conjunto reducido que permite exponer los principales resultados obtenidos de los distintos estudios. Esta decisión se ha tomado con el fin de no abrumar al lector con una cantidad excesiva de gráficas distintas y sobrecargar el contenido del capítulo. Para compensar esta selección, se adjunta en los anexos finales de la memoria una colección extendida de visualizaciones adicionales correspondientes a cada uno de los experimentos.

También con el objetivo de no sobrecargar este capítulo, los comentarios correspondientes a cada figura se mantendrán lo más breves posible. Se considera que, en muchos casos, las propias figuras son suficientemente ilustrativas, por lo que extenderse en comentarios descriptivos sobre cada gráfica podría resultar redundante. Esto no significa que las gráficas se presenten sin ningún tipo de explicación. Cada una irá acompañada de un comentario interpretativo breve, evitando descripciones excesivamente densas. La idea es agilizar la lectura del capítulo, de manera que el lector pueda obtener una visión suficientemente clara de los resultados obtenidos sin que la exposición resulte pesada. Posteriormente, al final de este capítulo, se contextualizan los principales resultados y se profundiza en sus implicaciones dentro del marco del trabajo abordado.

4.1. Evaluación multilingüe de rendimiento y energía en LLMs

El cuadro 4.1 se presenta como resumen global de los resultados obtenidos en la evaluación multilingüe de rendimiento y energía en LLMs. Las métricas se muestran en función del modelo y engloban todas las inferencias realizadas por cada uno de ellos (considerando todas las preguntas, idiomas e iteraciones). De izquierda a derecha, las columnas representan el consumo energético por mil consultas, la puntuación global, la precisión por pasos, la consistencia y el *AI Energy Score*.

Cuadro 4.1: Resumen conjunto de rendimiento y consumo energético para los LLMs evaluados. Los valores se muestran como media $\pm$ desviación estándar a través de los idiomas considerados y de las iteraciones realizadas.

Modelo	E_{1K}	S_{overall}	S_{step}	C	AI En. Score ($\times 10^3$)
Mistral-7B-Instruct-v0.1	157,3 $\pm$ 57,4	0,332 $\pm$ 0,037	0,389 $\pm$ 0,029	0,046 $\pm$ 0,007	2,11
Phi-4-mini-instruct	251,3 $\pm$ 8,6	0,507 $\pm$ 0,070	0,660 $\pm$ 0,077	0,065 $\pm$ 0,015	2,02
Phi-3-mini-128k-instruct	272,1 $\pm$ 6,3	0,476 $\pm$ 0,070	0,652 $\pm$ 0,073	0,058 $\pm$ 0,008	1,75
Qwen2-7B-Instruct	464,6 $\pm$ 7,9	0,467 $\pm$ 0,046	0,639 $\pm$ 0,057	0,049 $\pm$ 0,009	1,00
Falcon-Mamba-7B	461,3 $\pm$ 10,2	0,384 $\pm$ 0,052	0,514 $\pm$ 0,048	0,057 $\pm$ 0,006	0,83

Los resultados no indican una correspondencia directa entre rendimiento y consumo energético. Es decir, los modelos que consumen más energía no son necesariamente los que producen mejores resultados. El mejor rendimiento obtenido por los modelos estudiados corresponde a **Phi-4-mini-instruct**, que alcanza los valores más altos tanto en S_{overall} como en S_{step} , manteniendo además un consumo energético moderado. El modelo **Phi-3-mini-128k-instruct** presenta puntuaciones similares (aunque ligeramente inferiores) en las métricas de energía y rendimiento. **Mistral-7B-Instruct-v0.1** se posiciona como el modelo más eficiente según la métrica *AI Energy Score*, al obtener los mejores resultados en relación con la energía empleada. Esto no se debe tanto a que produzca resultados particularmente buenos, ya que presenta las puntuaciones de rendimiento más bajas, sino a que su consumo energético es notablemente inferior al del resto de modelos. También se observa que este modelo es el que presenta mayor variabilidad en el consumo energético. Finalmente, se puede observar que la consistencia de las respuestas es sorprendentemente buena, lo que indica que los modelos generan respuestas reproducibles entre distintas ejecuciones.

El cuadro 4.2 es el complemento directo de la tabla mostrada previamente. Aquí se recogen exactamente los mismos resultados, pero organizados según la

4.1. Evaluación multilingüe de rendimiento y energía en LLMs

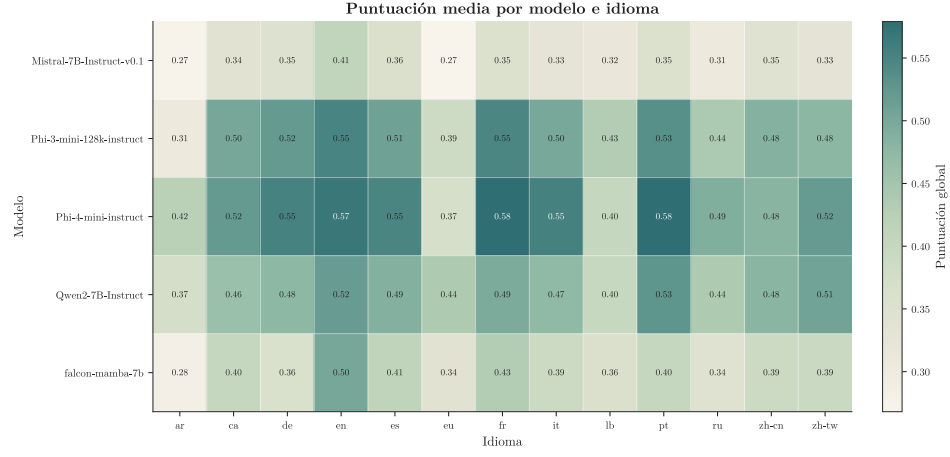
dimensión del idioma en lugar de la del modelo. Las métricas evaluadas corresponden nuevamente, de izquierda a derecha, al consumo energético por mil consultas, la puntuación global, la precisión por pasos y la consistencia. No se incluye en este caso el *AI Energy Score*, puesto que se definió inicialmente como una métrica asociada a cada modelo.

Cuadro 4.2: Resumen conjunto de rendimiento y consumo energético para los idiomas evaluados. Los valores se muestran como media $\pm$ desviación estándar a través de los modelos considerados y de las iteraciones realizadas.

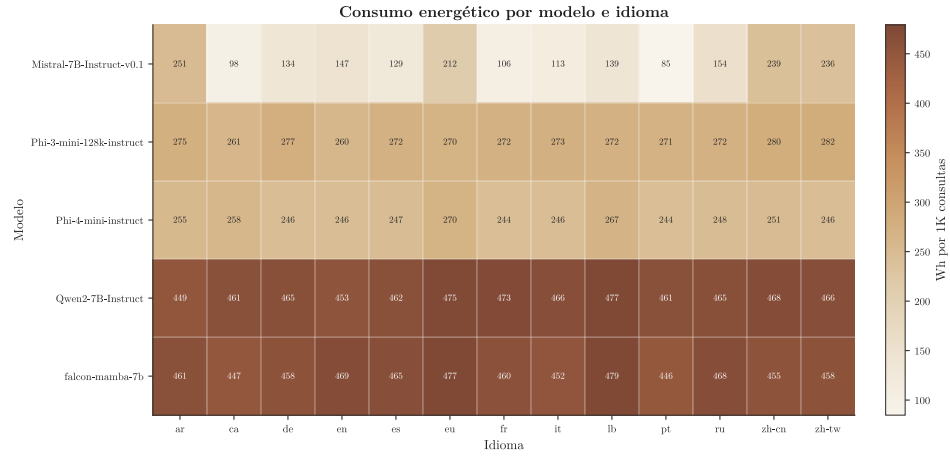
Idioma	E_{1K}	S_{overall}	S_{step}	C
pt	301,2 $\pm$ 156,2	0,479 $\pm$ 0,097	0,607 $\pm$ 0,145	0,051 $\pm$ 0,010
ca	305,1 $\pm$ 151,1	0,444 $\pm$ 0,074	0,575 $\pm$ 0,116	0,053 $\pm$ 0,012
it	310,3 $\pm$ 148,8	0,448 $\pm$ 0,091	0,599 $\pm$ 0,144	0,048 $\pm$ 0,010
fr	311,0 $\pm$ 155,2	0,481 $\pm$ 0,091	0,592 $\pm$ 0,151	0,059 $\pm$ 0,002
en	315,0 $\pm$ 140,5	0,509 $\pm$ 0,062	0,612 $\pm$ 0,114	0,047 $\pm$ 0,007
es	315,1 $\pm$ 145,9	0,462 $\pm$ 0,078	0,610 $\pm$ 0,135	0,049 $\pm$ 0,010
de	315,9 $\pm$ 143,1	0,452 $\pm$ 0,093	0,590 $\pm$ 0,132	0,053 $\pm$ 0,006
ru	321,4 $\pm$ 139,3	0,403 $\pm$ 0,075	0,527 $\pm$ 0,114	0,055 $\pm$ 0,010
lb	326,8 $\pm$ 147,8	0,380 $\pm$ 0,044	0,503 $\pm$ 0,099	0,063 $\pm$ 0,017
zh-tw	337,7 $\pm$ 115,1	0,445 $\pm$ 0,082	0,610 $\pm$ 0,124	0,059 $\pm$ 0,007
ar	338,2 $\pm$ 107,1	0,331 $\pm$ 0,064	0,504 $\pm$ 0,090	0,065 $\pm$ 0,013
zh-cn	338,7 $\pm$ 113,3	0,436 $\pm$ 0,065	0,614 $\pm$ 0,115	0,052 $\pm$ 0,010
eu	340,5 $\pm$ 125,8	0,360 $\pm$ 0,062	0,475 $\pm$ 0,093	0,064 $\pm$ 0,019

La tabla muestra que el consumo energético medio por idioma se mantiene en un rango relativamente acotado, [301,2, 340,5] Wh/1000 consultas. Dentro de este rango, **pt**, **ca**, **it** y **fr** surgen como los idiomas menos energéticamente demandantes, mientras que los mayores consumos aparecen en **eu**, **zh-cn**, **ar** y **zh-tw**. Como cabría esperar, la mejor puntuación global se obtiene en inglés. Por el contrario, **ar**, **eu** y **lb** presentan las puntuaciones más bajas en S_{overall} . Esto es coherente considerando que, dentro del conjunto evaluado, estos tres idiomas son precisamente los únicos denominados por la literatura como *low-resource*. El razonamiento intermedio por pasos presenta puntuaciones muy similares en todos los idiomas, aunque resulta interesante señalar que **zh-cn** supera ligeramente al inglés en esta métrica. Al igual que a nivel de modelo, la consistencia de las respuestas es bastante buena para todos los idiomas estudiados, indicando nuevamente que se generan respuestas reproducibles entre distintas ejecuciones. Quizás lo más importante que se puede extraer de la tabla es que el idioma afecta de forma más visible al rendimiento que al consumo energético medio; indicado por la mayor dispersión de las métricas de rendimiento frente al rango más acotado de valores de E_{1K} .

CAPÍTULO 4. RESULTADOS Y DISCUSIÓN



(a) Puntuación global media, S_{overall} , por modelo e idioma.



(b) Consumo energético normalizado, E_{1K} , por modelo e idioma.

Figura 4.1: Comparación conjunta de rendimiento y consumo energético por modelo e idioma en la evaluación multilingüe de LLMs.

4.1. Evaluación multilingüe de rendimiento y energía en LLMs

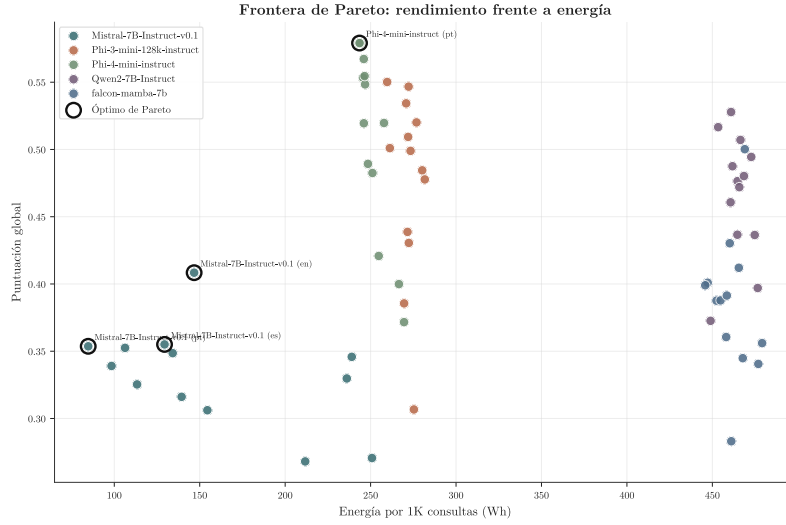


Figura 4.2: Frontera de Pareto entre rendimiento y consumo energético en la evaluación multilingüe de LLMs.

Para complementar la información proporcionada en las dos tablas anteriores, se presentan dos mapas de calor. La Figura 4.1a muestra el rendimiento, basándose en la puntuación media como métrica principal de cada par modelo-idioma. Resulta evidente que los modelos con mejores resultados son **Phi-3-mini-128k-instruct**, **Phi-4-mini-instruct** y **Qwen2-7B-Instruct**. También se observan de forma más clara las peores puntuaciones en los idiomas *low-resource*: **ar**, **eu** y **lb**. Del mismo modo, la Figura 4.1b recoge los valores de E_{1K} como métrica principal de energía para cada combinación evaluada. Aquí, se aprecia que el consumo energético está fuertemente ligado al modelo, más que al idioma.

Puede resultar interesante combinar ambos mapas de calor para representar cada par idioma-modelo en un espacio bidimensional, tomando la energía y el rendimiento como ejes de comparación. Cada par, por tanto, se puede representar como un punto en dicho espacio, como ilustra la Figura 4.2. En esta representación gráfica, también se han señalado aquellos puntos que son Pareto-eficientes.

La eficiencia de Pareto permite analizar el compromiso entre rendimiento y consumo energético. En el presente contexto de evaluación, un par modelo-idioma se considera Pareto-eficiente cuando no existe otra combinación que obtenga simultáneamente una mayor puntuación de rendimiento y un menor consumo energético. Como muestra la figura, la mayoría de estos puntos corresponden al mode-

lo **Mistral-7B-Instruct-v0.1**, pues este presenta el menor consumo energético entre todos los modelos evaluados. No obstante, entre los puntos resultantes como Pareto-eficientes destaca **Phi-4-mini-instruct** en portugués, mostrando un rendimiento notablemente superior al de las combinaciones asociadas a **Mistral-7B-Instruct-v0.1**. También cabe mencionar los resultados del modelo **Phi-4-mini-instruct** en francés, que prácticamente coinciden espacialmente con los obtenidos en portugués y, por ello, no se aprecian con claridad en el mapa. Este punto es casi Pareto-eficiente. Se destaca también que, dentro del conjunto de puntos resaltados, el portugués aparece más de una vez, siendo esta lengua casi siempre asociada al menor consumo energético dentro de cada modelo evaluado.

En todos los apartados previos, las métricas de energía han evaluado los modelos considerando la energía consumida por mil consultas. Cabría esperar que mejores respuestas empleasen más recursos computacionales y, por tanto, más energía. El primer cuadro ha demostrado que este no es el caso, pues no se observaba una correlación clara entre rendimiento y consumo energético. Sin embargo, hasta ahora el rendimiento se ha medido en función de cómo de bien se resolvía un problema. Otro enfoque igual de válido sería considerar como mejor respuesta aquella que proporcionase más información. Es decir, quizá los modelos consumen más energía no por producir respuestas más correctas, sino por generar respuestas con mayor contenido. Para abordar esta cuestión, se puede comparar la energía por mil consultas con los tokens medios generados en cada combinación modelo-idioma. Esta comparativa se ilustra en la Figura [4.3](#).

Por lo general, se puede apreciar una correlación positiva entre ambas variables. Es decir, las consultas que generan más tokens suelen requerir más energía. Esto es coherente con el proceso de inferencia autorregresiva mediante el cual funcionan los LLMs. No obstante, la relación no es uniforme para todos los modelos, ya que el consumo, además del tamaño de la salida, parece depender de la arquitectura de los mismos. Esta representación permite entender por qué el modelo denominado más eficiente es **Mistral-7B-Instruct-v0.1**: genera respuestas más cortas y por ello presenta los consumos más bajos. Asimismo, tanto **Qwen2-7B-Instruct** como **Falcon-Mamba-7B** se sitúan en la zona de mayor consumo, incluso con longitudes de salida similares a otros modelos.

Se puede desarrollar aún más este análisis formulando las métricas de eficiencia no en función de la energía por 1000 consultas, sino de la energía por mil tokens generados. Esta nueva métrica se muestra en el Cuadro [4.3](#). En él, en la última columna se indica la posición de estos modelos al ser reordenados según el consumo por tokens generados. Se destaca que, bajo esta métrica, **Phi-4-mini-instruct**

4.1. Evaluación multilingüe de rendimiento y energía en LLMs

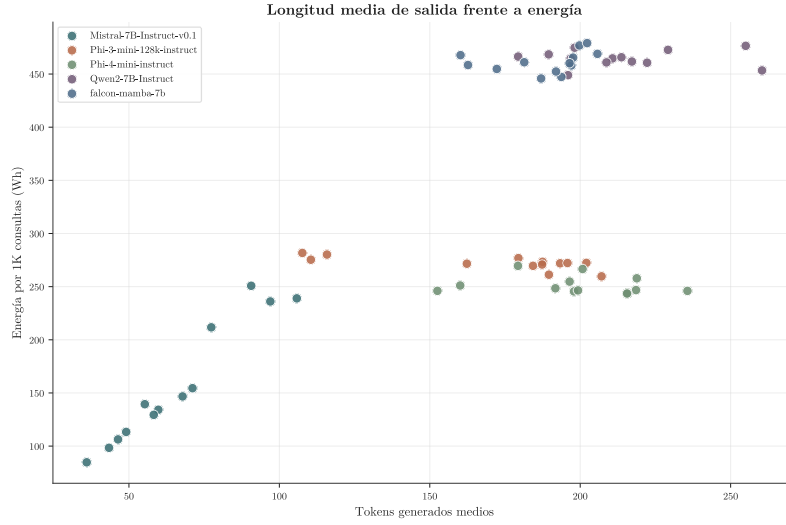


Figura 4.3: Relación entre la longitud media de salida y el consumo energético normalizado por mil consultas.

pasa a ser el modelo más eficiente, mientras que **Mistral-7B-Instruct-v0.1** cae del primer al cuarto puesto.

Cuadro 4.3: Resumen conjunto de rendimiento y consumo energético para los LLMs evaluados. Los valores se muestran como media $\pm$ desviación estándar a través de los idiomas considerados y de las iteraciones realizadas.

Modelo	Tok./resp.	Wh/1000 cons.	Wh/1000 tok.	Rank cons. $\rightarrow$ tok.
Mistral-7B-Instruct-v0.1	66 $\pm$ 22	157,3 $\pm$ 57,4	2,37 $\pm$ 0,20	1 $\rightarrow$ 4
Phi-4-mini-instruct	199 $\pm$ 24	251,3 $\pm$ 8,6	1,28 $\pm$ 0,18	2 $\rightarrow$ 1
Phi-3-mini-128k-instruct	171 $\pm$ 36	272,1 $\pm$ 6,3	1,68 $\pm$ 0,48	3 $\rightarrow$ 2
Falcon-Mamba-7B	188 $\pm$ 15	461,3 $\pm$ 10,2	2,46 $\pm$ 0,21	4 $\rightarrow$ 5
Qwen2-7B-Instruct	214 $\pm$ 24	464,6 $\pm$ 7,9	2,20 $\pm$ 0,24	5 $\rightarrow$ 3

Como análisis final, se incluye una representación gráfica de la relación entre el consumo energético y la latencia asociada al proceso de inferencia de cada consulta. A partir de la gráfica, se observa una elevada dependencia lineal entre ambas variables, de modo que un mayor consumo energético se corresponde, en general, con una mayor latencia de inferencia. Aunque esta relación pueda parecer intuitiva, su implicación es especialmente relevante. En la mayoría de aplicaciones de sistemas de inferencia, se tiene un consumo energético prácticamente constante, por lo que las consultas que requieren más energía suelen estar asociadas directamente a un mayor tiempo de procesamiento. Este comportamiento puede resultar crítico en

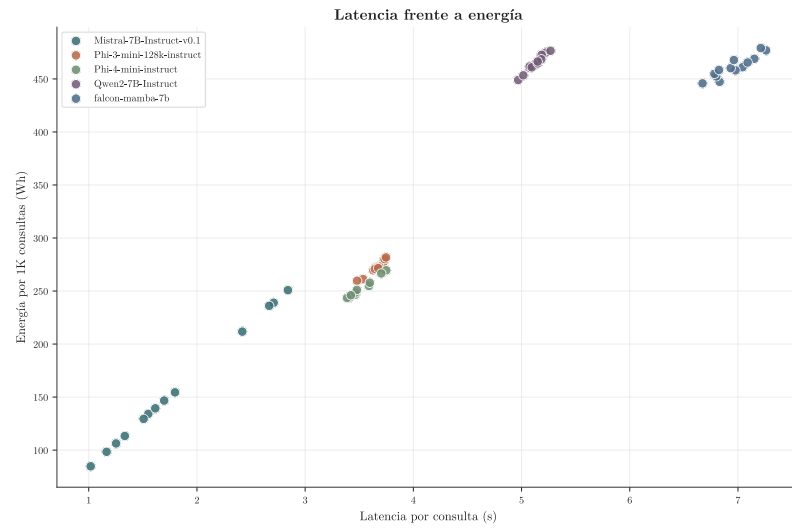


Figura 4.4: Relación entre la latencia media por consulta y el consumo energético normalizado por mil consultas.

aplicaciones con restricciones estrictas de tiempo de respuesta, como puede ser la navegación autónoma.

4.2. Evaluación multilingüe de rendimiento y energía en VLMs

En esta sección, con el fin de facilitar la comparación, se incluyen resultados relevantes de la evaluación multilingüe de rendimiento y energía en VLMs y de la denominada evaluación completamente multilingüe de rendimiento y energía en VLMs. Se recuerda que el desarrollo experimental de ambos estudios es prácticamente idéntico, salvo por la corrección de una serie de limitaciones metodológicas detectadas en la primera evaluación. Al presentar los datos de esta forma, se puede apreciar con mayor claridad la evolución entre ambos enfoques.

Se muestran en primer lugar las tablas de métricas agregadas de rendimiento y consumo energético en función del modelo empleado. El Cuadro 4.4 corresponde a la evaluación inicial, mientras que el Cuadro 4.5 corresponde a la revisión completamente multilingüe.

Cuadro 4.4: Resumen global de rendimiento y consumo energético por modelo VLM en la evaluación multilingüe inicial; en negrita se indica el mejor valor de cada columna.

Modelo	S_m	σ_m	E_{1K} (Wh/1K)	Scene	Veh.	Road	Nav.
Qwen2-VL	0.126	0.005	111.6	0.192	0.123	0.075	0.115
InternVL2	0.137	0.026	164.9	0.183	0.164	0.072	0.127
LLaVA-1.6	0.160	0.032	130.0	0.229	0.215	0.083	0.113
LLaVA-1.5	0.137	0.028	82.5	0.242	0.146	0.036	0.124
Phi-3-V	0.125	0.030	66.3	0.217	0.109	0.072	0.101

Cuadro 4.5: Resumen global de rendimiento y consumo energético por modelo VLM en la evaluación completamente multilingüe; en negrita se indica el mejor valor de cada columna.

Modelo	S_m	σ_m	E_{1K} (Wh/1K)	Scene	Veh.	Road	Nav.
Qwen2-VL	0.424	0.013	104.7	0.643	0.409	0.197	0.445
InternVL2	0.413	0.050	174.9	0.638	0.391	0.156	0.469
LLaVA-1.6	0.368	0.054	121.1	0.603	0.329	0.175	0.362
LLaVA-1.5	0.351	0.045	75.0	0.652	0.347	0.083	0.323
Phi-3-V	0.338	0.049	66.0	0.630	0.298	0.115	0.308

Las tablas indican una mejora considerable de las puntuaciones obtenidas en el estudio revisado, lo que sugiere que la primera metodología no estaba del todo

bien planteada. Al corregir el procedimiento para hacerlo más inclusivo con idiomas distintos del inglés, las puntuaciones generales aumentan de forma clara. Este efecto podrá observarse con mayor detalle al evaluar estas mismas métricas desde la dimensión del idioma, como se hará más adelante. Se destaca, además, que las métricas de energía también varían entre ambas metodologías, aunque no lo hacen en la misma medida que el rendimiento.

Para comentar los resultados entre modelos e idiomas se tomará como referencia, salvo que se indique lo contrario, la metodología corregida. Esto se debe a que resulta más relevante interpretar esta evaluación experimental que la metodología inicial, en la que algunos errores introducían un sesgo en favor del inglés.

Las tablas indican que no existe un modelo claramente superior al resto, ya que las puntuaciones obtenidas en las métricas de rendimiento y energía se concentran en rangos relativamente próximos. Además, el mejor rendimiento en las distintas tareas se distribuye entre tres modelos. Quizá la proximidad de los resultados observados pueda deberse, en parte, a que todos los modelos evaluados tienen un número de parámetros relativamente similar, como se ha mencionado en la metodología. Es curioso observar que la descripción de condiciones de la vía resulta ser la tarea más exigente para los modelos evaluados, y no el razonamiento de navegación, como se había considerado inicialmente en la metodología. Finalmente, la consistencia de las respuestas presenta valores bajos, lo que indica que los modelos producen respuestas relativamente estables entre distintas ejecuciones de una misma inferencia.

Para apreciar con mayor claridad el sesgo introducido por la evaluación inicial, se muestran los cuadros complementarios a los dos anteriores, analizando el mismo conjunto de datos desde la dimensión del idioma. El Cuadro [4.6a](#) corresponde a la evaluación inicial, mientras que el Cuadro [4.6b](#) corresponde a la revisión completamente multilingüe.

4.2. Evaluación multilingüe de rendimiento y energía en VLMs

Cuadro 4.6: Resumen global de rendimiento y consumo energético por idioma en las dos evaluaciones multilingües; en negrita se indica el mejor valor de cada columna.

(a) Evaluación multilingüe inicial.

Idioma	S_ℓ	σ_ℓ	E_{1K} (Wh/1K)	Scene	Veh.	Road	Nav.
ar	-0.007	0.006	142.535	-0.041	0.001	0.000	0.010
ca	0.187	0.029	125.314	0.222	0.254	0.092	0.180
de	0.087	0.022	108.878	0.193	0.018	0.061	0.077
en	0.511	0.049	100.748	0.649	0.703	0.351	0.340
es	0.157	0.019	113.659	0.268	0.150	0.083	0.125
eu	0.041	0.017	67.157	0.081	0.039	0.001	0.042
fr	0.145	0.026	115.300	0.279	0.081	0.010	0.210
it	0.193	0.028	106.907	0.293	0.282	0.099	0.097
lb	0.137	0.041	105.399	0.257	0.185	0.031	0.076
pt	0.145	0.018	106.966	0.244	0.148	0.093	0.096
ru	0.055	0.038	119.964	0.018	0.100	0.057	0.046
zh-cn	0.043	0.012	107.651	0.100	0.005	0.001	0.067
zh-tw	0.088	0.010	123.389	0.204	0.003	0.001	0.143

(b) Evaluación completamente multilingüe.

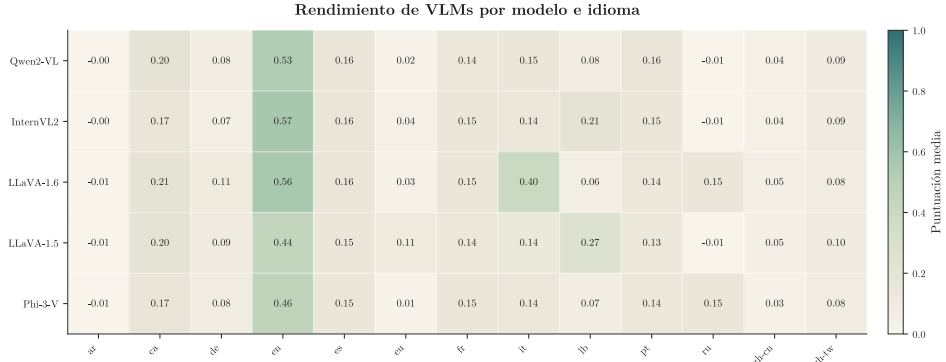
Idioma	S_ℓ	σ_ℓ	E_{1K} (Wh/1K)	Scene	Veh.	Road	Nav.
ar	0.241	0.040	137.472	0.511	0.189	0.008	0.255
ca	0.377	0.045	122.509	0.723	0.264	0.129	0.392
de	0.488	0.050	106.349	0.745	0.565	0.234	0.408
en	0.567	0.044	99.035	0.693	0.701	0.354	0.519
es	0.437	0.045	111.079	0.781	0.411	0.135	0.423
eu	0.155	0.026	65.564	0.232	0.091	0.095	0.203
fr	0.509	0.048	112.599	0.733	0.512	0.324	0.467
it	0.374	0.041	102.953	0.750	0.234	0.085	0.426
lb	0.109	0.032	103.413	0.204	0.085	0.008	0.138
pt	0.453	0.049	105.309	0.757	0.498	0.160	0.396
ru	0.337	0.039	117.033	0.727	0.215	0.049	0.356
zh-cn	0.436	0.043	105.449	0.685	0.418	0.158	0.482
zh-tw	0.441	0.045	119.777	0.692	0.430	0.149	0.494

En la revisión, aunque el inglés sigue liderando, la disparidad entre idiomas se suaviza. Los resultados obtenidos a nivel general son muy superiores a los de la primera evaluación en casi todos los idiomas, incluido el propio inglés. La mejora es especialmente notable en el árabe, el ruso y el chino. Por ejemplo, el árabe pasa de una puntuación media negativa a $S_\ell = 0,241$, mientras que el chino simplificado aumenta de $S_\ell = 0,043$ a $S_\ell = 0,436$. Se puede observar, además, que al igual que en la evaluación de LLMs, el luxemburgués, el árabe y el euskera aparecen como los idiomas con menor rendimiento, lo cual resulta coherente con su consideración habitual como lenguas de bajos recursos.

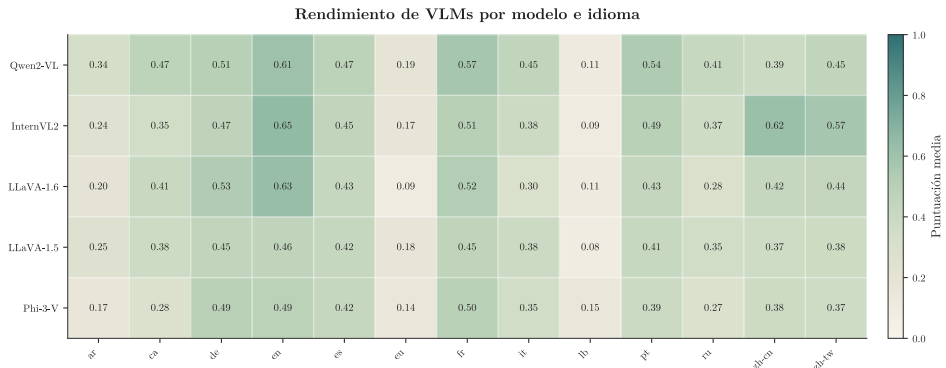
Las métricas de energía son, a su vez, bastante consistentes a lo largo de la dimensión del idioma. En la evaluación completamente multilingüe, E_{1K} se sitúa aproximadamente entre 65.6 y 137.5 Wh/1K consultas, un rango relativamente contenido en comparación con las diferencias observadas en las métricas de rendimiento.

Para ilustrar los hallazgos de esta sección de forma visual, se incluyen una serie de gráficos comparativos entre los dos estudios realizados. Estas representaciones permiten apreciar con mayor claridad los patrones comentados anteriormente.

4.2. Evaluación multilingüe de rendimiento y energía en VLMs

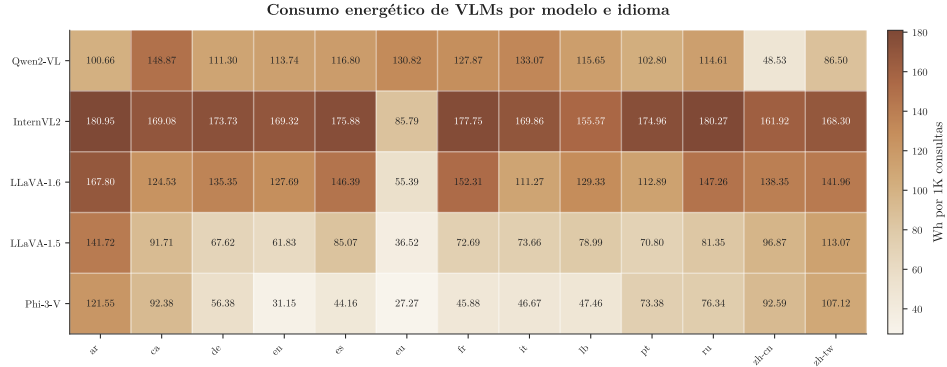


(a) Evaluación multilingüe inicial.

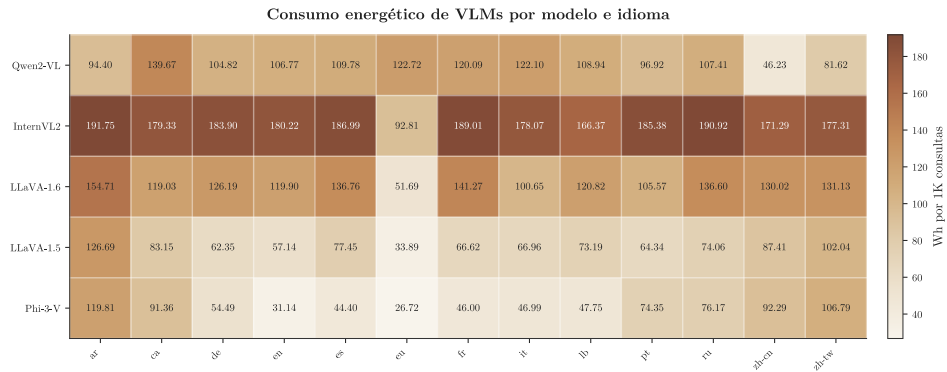


(b) Evaluación completamente multilingüe.

Figura 4.5: Comparación del rendimiento por modelo e idioma entre la evaluación multilingüe inicial y la evaluación completamente multilingüe.



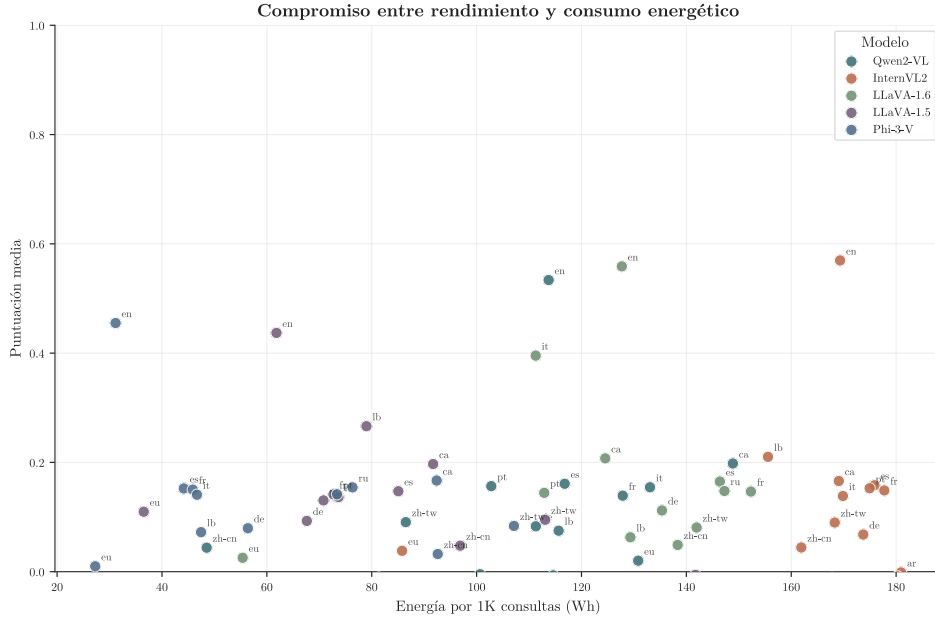
(a) Evaluación multilingüe inicial.



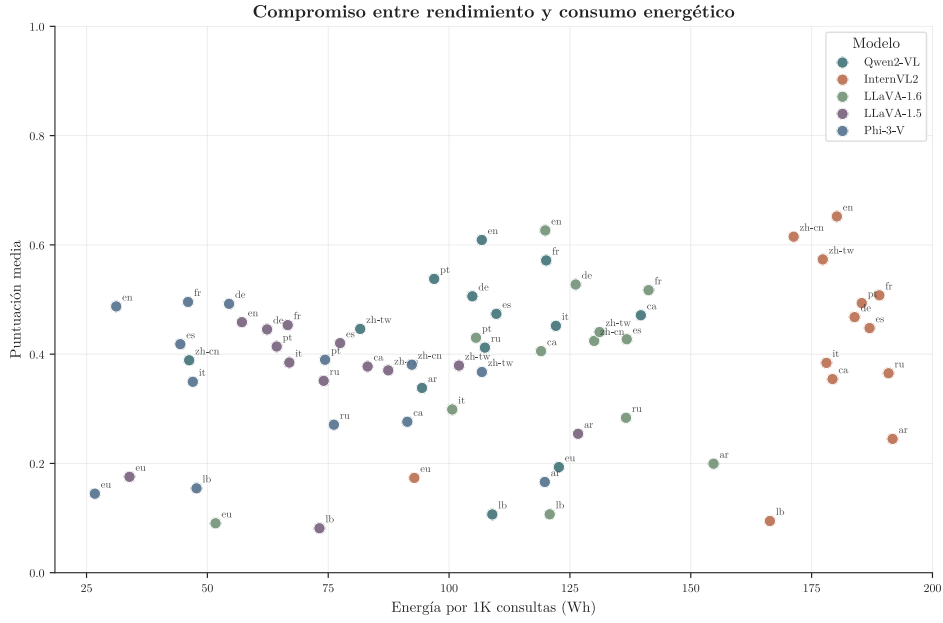
(b) Evaluación completamente multilingüe.

Figura 4.6: Comparación del consumo energético por modelo e idioma entre la evaluación multilingüe inicial y la evaluación completamente multilingüe.

4.2. Evaluación multilingüe de rendimiento y energía en VLMs



(a) Evaluación multilingüe inicial.



(b) Evaluación completamente multilingüe.

Figura 4.7: Comparación del compromiso entre rendimiento y consumo energético entre la evaluación multilingüe inicial y la evaluación completamente multilingüe.

4.3. Evaluación del grounding visual multilingüe en el contexto de OneMap

El Cuadro 4.7 resume las puntuaciones obtenidas por ambos modelos en las métricas de rendimiento y consumo energético detalladas en la metodología. De izquierda a derecha, se presentan el IoU basado en clústeres, los IoU calculados a partir de percentiles, la correlación de Spearman, la razón de activación máxima respecto a la consulta de referencia en inglés y, finalmente el consumo energético normalizado por cada mil consultas.

Cuadro 4.7: Resumen global de métricas de grounding visual por modelo. Las métricas se promedian sobre los idiomas distintos del inglés, conceptos e imágenes evaluados; en negrita se indica el mejor valor de cada columna.

Modelo	IoU cl.	IoU@90	IoU@95	IoU@99	ρ	η	E_{1K} (Wh/1K)
XLM-R base	0.584	0.477	0.411	0.329	0.839	0.981	3.868
XLM-R large	0.572	0.500	0.439	0.315	0.841	0.937	3.358

Se recuerda que los modelos evaluados difieren considerablemente en el número de parámetros del codificador visual: 87M frente a 632M, lo que supone un incremento de casi un orden de magnitud. No obstante, las métricas globales a nivel de modelo indican desempeños muy similares de rendimiento y consumo. Se podría concluir, por tanto, que para los modelos estudiados, el aumento de escala no se traduce en una mejora proporcional del *grounding* visual multilingüe.

Para estudiar el impacto de la dimensión lingüística en el *grounding* visual, se presentan, de forma análoga a los experimentos anteriores, los resultados agregados por idioma (Cuadro 4.8). Se muestran las mismas métricas que en la tabla anterior, con la excepción del consumo energético, que en este estudio solo se obtuvo a nivel de modelo. Esto, sin embargo, resulta coherente con la configuración experimental. Al fin y al cabo, no se está evaluando la capacidad de razonamiento del modelo, sino la variación del *grounding* visual inducida por el idioma de la entrada. De hecho, como se ha explicado en la metodología, el proceso de extracción de características visuales se mantiene constante para una misma imagen, mientras que únicamente se modifica el *embedding* textual asociado a la consulta. En este caso, la rama textual se limita a proyectar cada consulta al espacio de *embeddings* de CLIP, por lo que no se espera una variación energética significativa entre idiomas. La posterior comparación con las características visuales mediante operaciones de similitud se realiza ya como una etapa de postprocesado, donde no interviene de nuevo el modelo.

4.3. Evaluación del grounding visual multilingüe en el contexto de OneMap

Además, si los resultados obtenidos en los experimentos anteriores son indicativos, cabe esperar que el consumo energético esté más fuertemente relacionado con el modelo empleado que con el idioma.

Cuadro 4.8: Resumen global de métricas de grounding visual por idioma, excluyendo el inglés como idioma de referencia; en negrita se indica el mejor valor de cada columna.

Idioma	IoU cl.	IoU@90	IoU@95	IoU@99	ρ	η
ar	0.546	0.444	0.376	0.273	0.826	1.005
ca	0.618	0.541	0.483	0.378	0.887	0.946
de	0.628	0.548	0.486	0.379	0.901	0.946
es	0.652	0.577	0.516	0.410	0.915	0.966
eu	0.472	0.377	0.317	0.224	0.684	0.922
fr	0.634	0.550	0.484	0.371	0.903	0.949
it	0.633	0.549	0.483	0.371	0.902	0.949
lb	0.426	0.314	0.250	0.166	0.622	0.903
pt	0.637	0.550	0.486	0.375	0.905	0.972
ru	0.564	0.467	0.396	0.290	0.843	0.943
zh-cn	0.553	0.465	0.404	0.306	0.843	1.010
zh-tw	0.568	0.482	0.420	0.319	0.850	0.995

Se señala, respecto a este cuadro, que todas las métricas puntúan los idiomas en relación con un idioma de referencia ℓ_0 , que en este caso es el inglés. Este enfoque permite valorar la concordancia lingüística en las tareas de *grounding* visual. Precisamente por esta razón, la puntuación del inglés en todas las métricas es de 1.000 y, por tanto, no se incluye en el cuadro. Esto no debe interpretarse como que el inglés rinde mejor, sino como que constituye la referencia frente a la cual se comparan el resto de idiomas.

Aclarado esto, se pueden comentar varios aspectos. En primer lugar, las métricas de IoU basadas en clústeres son relativamente coherentes entre idiomas, con valores comprendidos en el intervalo $[0,426, 0,652]$ para los idiomas distintos del inglés. Esto indica que, en general, los modelos sitúan las regiones relevantes en posiciones parcialmente similares a las obtenidas con la referencia inglesa. Entre los casos de mayor desalineamiento espacial destacan el euskera y el luxemburgués, que presentan las peores puntuaciones en las distintas métricas de alineamiento. Este resultado refleja el hecho de que ambas lenguas se clasifican como lenguas de bajos recursos.

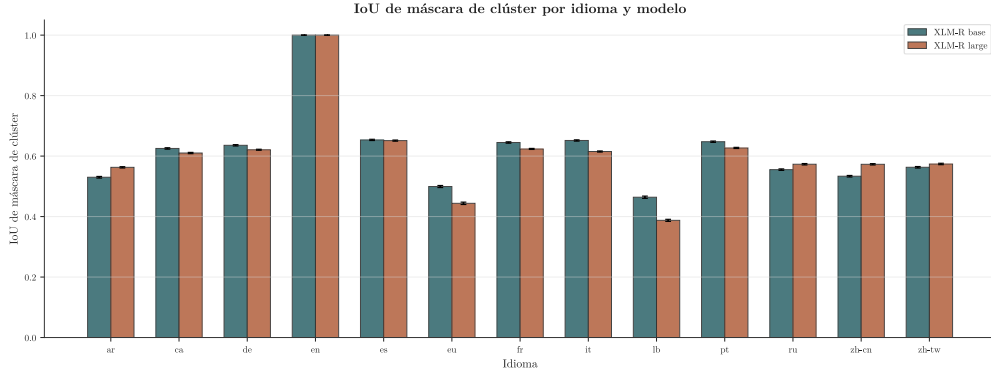


Figura 4.8: Comparación del IoU de máscara de clúster por idioma para los dos modelos CLIP multilingües evaluados.

Nótese también que las puntuaciones tienden a disminuir para percentiles más altos de IoU. Esto se explica porque, al aumentar el percentil, se comparan regiones cada vez más selectivas del mapa de similitud. Por otra parte, las métricas de correlación de Spearman sitúan a idiomas como el alemán, el portugués, el español, el francés y el italiano relativamente próximos al inglés, con valores superiores a 0.90.

Finalmente, el nivel de activación relativa, evaluado mediante la razón de similitud máxima η , es bastante similar entre idiomas y, en general, próximo al inglés, con valores comprendidos en el intervalo $[0,903, 1,010]$. Esto indica que, cuando se obtienen resultados distintos a los del inglés, no se debe necesariamente a un colapso de la señal, sino más bien a un desalineamiento espacial. Es decir, las consultas siguen generando activaciones relevantes, pero estas pueden aparecer en regiones ligeramente distintas de la imagen.

Para mostrar de forma más ilustrativa la influencia de distintos idiomas en el grounding visual, se representan los datos en forma de gráfico de barras. Concretamente, la Figura 4.8 muestra el IoU de máscara de clúster por idioma para los dos modelos CLIP multilingües evaluados. Se podría realizar un análisis similar considerando otras métricas. No obstante, se considera que el IoU de máscara de clúster es suficiente por sí mismo como cuantificador del alineamiento espacial.

La otra dimensión que podría evaluarse, el nivel de activación, presenta, como se ha mencionado, una variación menor entre idiomas. No obstante, puede resultar interesante evaluar la relación conjunta entre estas dos dimensiones; es decir, el alineamiento espacial y el nivel de activación. Estos datos se recogen en la Figura 4.9.

4.3. Evaluación del grounding visual multilingüe en el contexto de OneMap

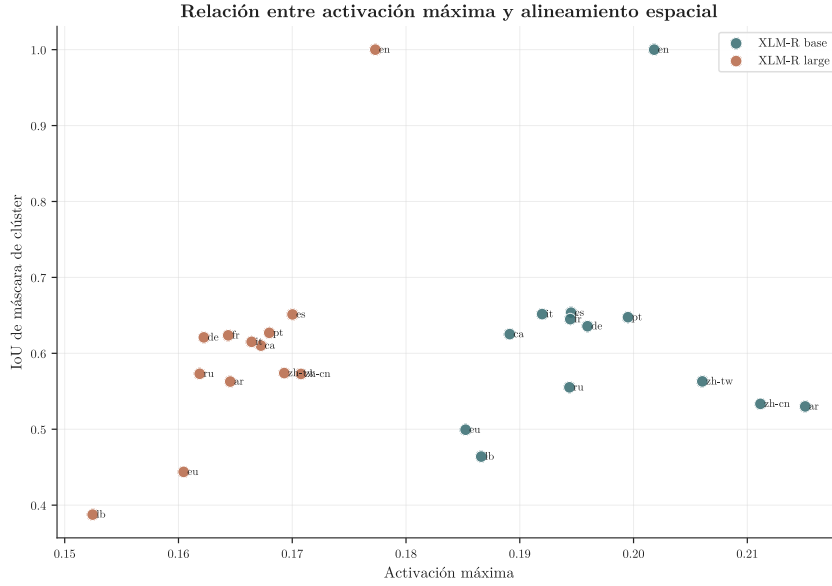
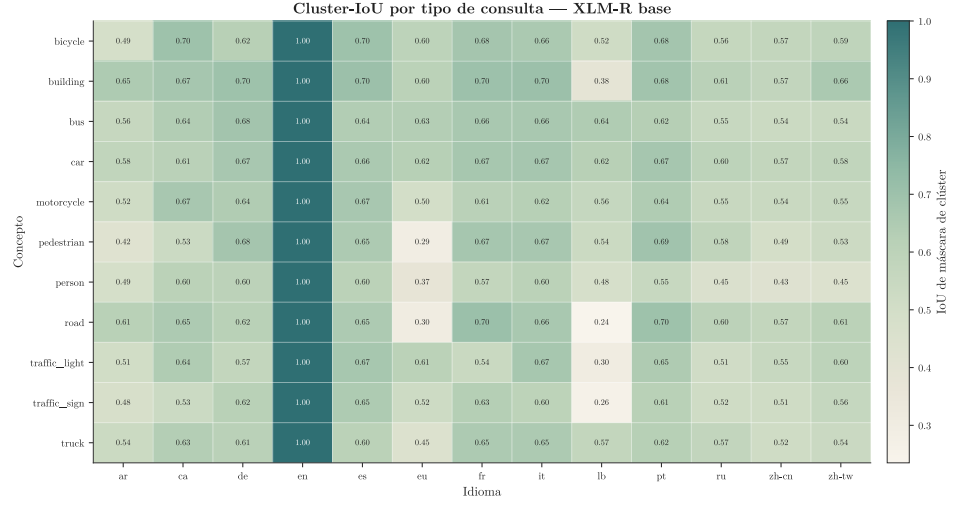


Figura 4.9: Relación entre la activación máxima y el IoU de máscara de clúster por idioma y modelo.

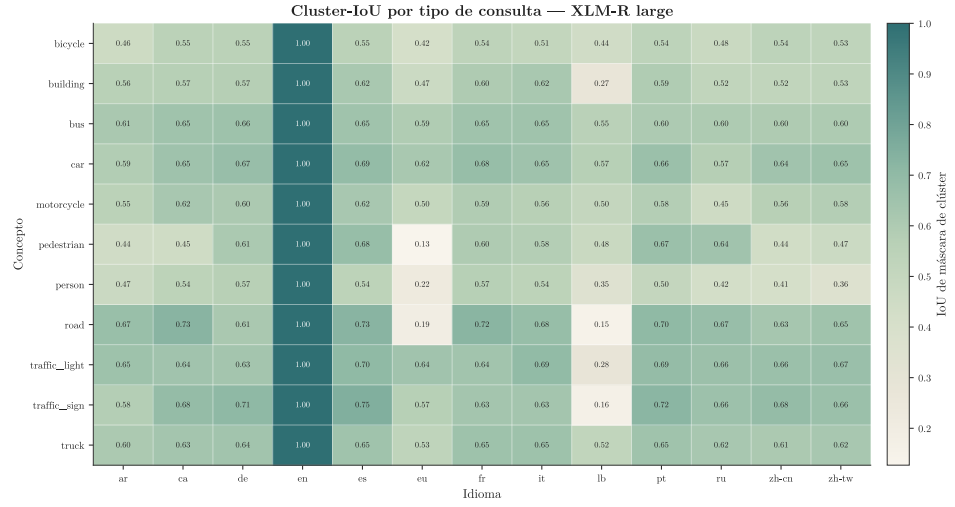
Para ofrecer una perspectiva ligeramente distinta, los niveles de activación se presentan como métricas globales. Es decir, se representa la media del valor máximo de similitud alcanzado por cada idioma y modelo. El alineamiento espacial no parece mostrar una relación directa con la rama textual empleada. Sin embargo, el modelo sí parece desplazar sistemáticamente los puntos en la dimensión de activación. En particular, el **XLM-R base** suele alcanzar máximos de similitud más altos que **XLM-R large**. Esto podría interpretarse como una mayor compatibilidad entre la formulación de los *embeddings* textuales y el codificador visual empleado. Se señala que, aunque se aprecia una diferencia, esta no es particularmente pronunciada, como puede observarse en la escala del eje correspondiente a la activación: todos los puntos mostrados están comprendidos aproximadamente en el intervalo $[0,15, 0,21]$.

Para finalizar esta sección, se introducen dos mapas de calor (Figura 4.10) con el objetivo de mostrar el efecto del idioma sobre el desalineamiento espacial de distintos conceptos visuales. Aunque esta evaluación no pretende analizar específicamente la capacidad de identificar y posicionar objetos concretos, resulta interesante observar si algunos de ellos presentan una mayor sensibilidad lingüística que otros.

CAPÍTULO 4. RESULTADOS Y DISCUSIÓN



(a) XLM-R base + ViT-B/32.



(b) XLM-R large + ViT-H/14.

Figura 4.10: IoU de máscara de clúster por idioma y tipo de consulta para los dos modelos CLIP multilingües evaluados.

4.4. Discusión global de resultados

Antes de pasar a las conclusiones finales del trabajo, se ofrece una interpretación global del conjunto de resultados obtenidos en los distintos experimentos. El objetivo principal de esta memoria ha sido analizar si el idioma condiciona el rendimiento y la eficiencia energética de LLMs y VLMs en el contexto de la navegación semántica autónoma. Para tratar de abordar esta cuestión, se han llevado a cabo una serie de experimentos. Se insiste al lector, que estos no deben entenderse como estudios independientes, sino como aproximaciones complementarias a un problema común. Inicialmente, este problema se ha abordado desde una perspectiva puramente textual; seguidamente, desde un enfoque multimodal aplicado a escenas de navegación; y, finalmente, desde un marco próximo al mecanismo de consulta semántica empleado en OneMap: el *grounding* visual.

A lo largo de la memoria se ha recalcado que el idioma no se debe interpretar simplemente como una manera distinta de articular una pregunta, sino como una dimensión adicional de la inferencia y la percepción de los modelos. Esto se evidencia en que, en los experimentos realizados, cambiar el idioma de la consulta modifica la respuesta generada, alterando su longitud o su contenido; modifica el proceso de inferencia, afectando a la latencia o al consumo energético¹ y modifica la interpretación visual, condicionando el alineamiento semántico o los niveles de activación.

En el primer experimento se ha podido comprobar cómo, en los LLMs, el idioma afecta tanto al rendimiento como al consumo energético. En particular, se observó una dependencia más fuerte con el rendimiento, estando el consumo más estrechamente ligado al modelo empleado. En los casos estudiados, lenguas como el inglés, el portugués o el francés producían resultados consistentemente por encima de la media. Al otro lado del espectro, los peores resultados se asociaron a los denominados idiomas de bajos recursos, como el árabe, el luxemburgués y el euskera. Esta distinción permite plantear que la evaluación del rendimiento mediante *benchmarks* de *accuracy* centrados exclusivamente en el inglés no es representativa en tareas multilingües; pues se observan fluctuaciones considerables entre las salidas producidas en distintos idiomas. Finalmente, al acoplar las métricas de rendimiento y energía en un marco de estudio común, se ha podido observar que aquellos modelos que consumen más no son necesariamente los que obtienen mejor

¹Debe tenerse en cuenta que los valores de E_{1K} se estiman de forma distinta entre experimentos. En la evaluación de LLMs correspondiente al primer experimento, se estiman mediante integración por muestra, y no mediante potencia media por tiempo de ejecución como en el resto de secciones (3.2, 3.3 y 3.4). Aunque ambas aproximaciones son equivalentes en el límite, las cifras absolutas no deben compararse directamente entre experimentos.

rendimiento.

En el segundo experimento, se ha ampliado el problema a la interpretación visual. Aun así, muchos de los hallazgos procedentes de este estudio presentan cierta correspondencia con los obtenidos para los LLMs. Los modelos estudiados presentaron métricas agregadas de rendimiento muy parecidas entre sí, lo cual se atribuyó parcialmente a la similitud en el tamaño de estos. Se destacó, por tanto, que más que la arquitectura del modelo, lo que verdaderamente afectaba a la calidad de las respuestas era el idioma. Una vez más, los mejores resultados se obtienen en inglés, mientras que las lenguas de bajos recursos ocupan las últimas posiciones.

De manera casi accidental, se ha podido apreciar también el impacto que puede tener sobre los resultados el diseño de metodologías de evaluación no equitativas. Esto es especialmente relevante en el contexto de la evaluación de modelos, donde muchas métricas favorecen implícitamente el inglés, como ocurría en la formulación denominada no completamente multilingüe. Lo que sucede en estos casos es que se oculta parte de la realidad, como se ha mostrado al comparar los resultados originales con los obtenidos tras la revisión. Se destaca, por tanto, la importancia de diseñar *benchmarks* equitativos, que permitan a los usuarios tomar decisiones más informadas sobre qué modelos presentan mejor desempeño a nivel lingüístico.

En el experimento final, se ha podido analizar una dimensión mucho más cercana a la navegación semántica y, en particular, al método de navegación semántica planteado por *OneMap*. El *grounding* visual representa una de las etapas clave de *OneMap* y está estrechamente enlazado con el proceso de extracción de características. Este concepto hace referencia no tanto a cómo los modelos razonan o generan respuestas a partir de imágenes, sino a cómo interpretan espacialmente la información visual de forma localizada. Los resultados del experimento indican que todos los idiomas identifican consultas en una imagen razonablemente bien; se ha referido a esta idea como activación. El problema, por tanto, es que, dicho de forma intuitiva, “los idiomas no siempre se ponen de acuerdo a la hora de situar conceptos en una imagen”. A este fenómeno se le ha denominado desalineamiento espacial. Como resultado, ante consultas e imágenes idénticas el idioma puede generar mapas de similitud completamente distintos. Una vez más, se aprecia que el idioma no se debe interpretar simplemente como una manera distinta de articular una pregunta, sino como una dimensión adicional de la inferencia y la percepción de los modelos. Y, en este caso, la parte de percepción se puede interpretar de forma más literal.

A medida que la discusión se orienta hacia contextos de navegación semántica,

se aprovecha para señalar que la dependencia lingüística de los sistemas multimodales puede tener implicaciones relevantes en este ámbito. Atendiendo a los resultados obtenidos, es razonable afirmar que las metodologías basadas en VLMs no deberían asumir que consultas formuladas en distintos idiomas sean equivalentes desde el punto de vista computacional. Se resalta, por tanto, la importancia de evaluar los modelos bajo las condiciones de uso previstas. Considérese, por ejemplo, el caso de OneMap: si el proceso de inferencia se realiza en lenguas de peor rendimiento, podrían identificarse regiones incorrectas o aumentar la incertidumbre de las características extraídas. Además, una elección no fundamentada de modelos o idiomas podría introducir un consumo energético o una latencia excesivos, haciendo que el sistema dejase de ser operativo en condiciones reales.

Se ha querido subrayar, a partir del análisis presentado, que un modelo no debe considerarse mejor que otro simplemente por obtener un mayor rendimiento. En entornos reales, que un modelo produzca buenos resultados es importante; no debe interpretarse lo contrario. Pero, precisamente al tratarse de entornos reales, también se debería dar importancia al consumo energético, a los tiempos de inferencia, a la consistencia y, en contextos multilingües, a la robustez interlingüística. Se propone, por tanto, que el mejor modelo no es necesariamente el más preciso, sino aquel que consigue un compromiso adecuado entre todas estas características.

Capítulo 5

Conclusiones

5.1. Hallazgos principales

El idioma como variable técnica relevante

El idioma en el que se formula una consulta afecta a la calidad y al contenido de la respuesta producida. Este comportamiento se ha observado tanto en modelos de lenguaje como en modelos visión-lenguaje, donde el rendimiento en distintas tareas de razonamiento y VQA tiende a degradarse a medida que aumenta la distancia léxica respecto al inglés. En particular, destacan los idiomas de bajos recursos como el euskera, el luxemburgués o el árabe, como aquellos que generalmente presentan los peores resultados.

El coste computacional de la variación lingüística

En los experimentos realizados, el consumo energético asociado a los procesos de inferencia se ve afectado por el idioma de la consulta. No obstante, no se observa una relación específica a nivel global, ya que la dependencia lingüística del consumo varía en función de la tarea realizada. Más que el idioma, el factor determinante en la variación del coste computacional parece ser el modelo y la arquitectura empleados.

El grounding visual ante la variabilidad lingüística

En tareas de grounding visual, el idioma no parece afectar de forma visible a la identificación global de los conceptos presentes en una imagen. Es decir, los resultados obtenidos no apuntan a un colapso generalizado de la señal de activación en la mayoría de los idiomas evaluados. En su lugar, la lengua afecta principalmente a la localización espacial de dichos conceptos dentro de la imagen, lo que sugiere la existencia de un desalineamiento espacial dependiente del idioma. Estos resul-

tados se mantienen en modelos de distinto tamaño, medido a partir del número de parámetros visuales.

Más allá del rendimiento: energía, latencia y robustez

El enfoque tradicional de evaluación basado en benchmarks unidimensionales de precisión puede ocultar los compromisos existentes entre rendimiento, coste energético y robustez lingüística. En este sentido, se ha observado que los modelos con mayor consumo energético no son necesariamente los que ofrecen un mejor desempeño en distintas tareas. Por ello, resulta esencial, especialmente en despliegues distribuidos, adoptar una evaluación que considere también el consumo energético, los tiempos de inferencia y la consistencia de las respuestas.

Hacia una evaluación multilingüe imparcial

Las diferencias de rendimiento entre idiomas suelen asociarse, en parte, a la distribución desigual de idiomas en los recursos de datos disponibles. No obstante, también se ha observado que los propios marcos de evaluación pueden introducir sesgos adicionales. Por ello, el diseño de *benchmarks* debe considerar la componente lingüística, especialmente en despliegues multilingües o en entornos en los que los modelos interactúen con usuarios en distintos idiomas. Se destaca, en estos casos, la importancia de diseñar e incorporar métricas equitativas que no favorezcan implícitamente o se inclinen hacia un conjunto de idiomas.

5.2. Contribuciones

Evaluación multilingüe de modelos de lenguaje

Se ha desarrollado un marco experimental para evaluar a los modelos de lenguaje de gran tamaño en tareas de razonamiento. Se han puntuado distintas combinaciones de idioma y modelo según métricas de rendimiento y de coste computacional asociadas al proceso de inferencia.

Incorporación del consumo energético como métrica de evaluación

Se ha propuesto el AI Energy Score como benchmark de evaluación de modelos de lenguaje. Esta métrica integra tanto el rendimiento, puntuado a partir de una serie de tareas de razonamiento, como el consumo energético bajo una misma formulación, permitiendo comparaciones bidimensionales entre LLMs.

Evaluación multilingüe de modelos de visión-lenguaje

Se ha desarrollado un marco experimental para evaluar a sistemas multimodales en tareas de percepción visual aplicadas a la navegación autónoma. Se ha analizado si las tendencias lingüísticas observadas en modelos de lenguaje se extienden a tareas de VQA.

Estudio del grounding visual condicionado por el idioma

Se ha estudiado la influencia del idioma en los resultados del grounding visual mediante arquitecturas CLIP. Esta contribución permite evaluar la capacidad de distintos modelos e idiomas para relacionar información espacial y semántica.

Implicaciones para navegación semántica y OneMap

Se ha trasladado el marco de evaluación multimodal a procesos relacionados con la metodología de OneMap. En particular, los resultados obtenidos se han vinculado con tareas de navegación semántica y extracción de características, permitiendo analizar cómo el idioma puede influir en el desempeño de estos procesos.

5.3. Limitaciones

Desde el planteamiento inicial de este trabajo, la metodología de *OneMap* se ha empleado como marco orientador del trabajo, más que como un sistema a reproducir de forma directa. Este enfoque ha actuado como hilo conductor a lo largo de la memoria en un doble sentido. Por una parte lado, ha motivado el estudio de los VLMs como componentes perceptivos en tareas de navegación semántica, extendiendo su evaluación a un contexto próximo al planteado por *OneMap*. Por otra, ha constituido la motivación principal de las contribuciones centradas en el grounding visual condicionado por el idioma, al situar los procesos de extracción de características y de consulta semántica en el centro del análisis. De este modo, la relación con *OneMap* es fundamentalmente conceptual y metodológica: en lugar de reproducir su implementación, se han aislado y evaluado de forma independiente los mecanismos multimodales sobre los que se apoya, incorporando además el multilingüismo y el consumo energético como ejes de estudio propios. Este marco orientador se recupera de forma explícita en el último experimento, dedicado precisamente a los procesos de extracción de características empleados por *OneMap*. No obstante, la ausencia de una integración completa de estos mecanismos dentro de un sistema de mapeo semántico operativo queda planteada como línea de trabajo futura.

Más allá de esto, y redirigiendo la atención hacia los experimentos realizados,

no cabrá duda de que el idioma ha sido uno de los aspectos centrales de este trabajo. En este sentido, y con el fin de mantener cierto nivel de consistencia, todas las etapas experimentales han evaluado el mismo conjunto de trece lenguas. Se considera que dicho conjunto es adecuadamente representativo de los idiomas hablados a escala global, y presenta una variabilidad lingüística suficiente como para producir resultados interpretables. No obstante, otra de las limitaciones de este estudio es no haber empleado un conjunto más diverso o más amplio, que incorporase aún más familias lingüísticas, variantes dialectales, y sistemas de escritura.

Del mismo modo que se podía haber ampliado la muestra de idiomas, también se podría haber incorporado un conjunto mayor de modelos. En este sentido, se quiere resaltar la rápida evolución del campo de la inteligencia artificial, en el que surgen cada vez más y mejores alternativas de modelos de lenguaje y visión-lenguaje. Incluso con una configuración experimental idéntica a la usada, se podrían haber obtenido resultados muy distintos al considerar arquitecturas o versiones más modernas, dando lugar a un análisis más completo y actualizado.

Por otra parte, el conjunto de datos empleado, aunque aporta una visión general de los fenómenos estudiados, podría considerarse reducido. Como se ha comentado en capítulos anteriores, durante el proceso de aprendizaje, los modelos en cuestión pueden llegar a procesar miles de millones de muestras. En este estudio, se han empleado corpus de evaluación varios órdenes de magnitud por debajo de estas cifras. Considérese, por ejemplo, la evaluación de VLMs, donde los modelos se evaluaron a partir de 210 imágenes de escenarios de conducción. Se podrían haber obtenido resultados mucho más robustos e interpretativos incrementando el número de muestras, quizás empleando el conjunto de datos completo de 100.000 imágenes de BDD100K. Esto habría sido lo ideal. Sin embargo, lo ideal no siempre es práctico o factible. El desarrollo experimental abordado contaba con restricciones importantes a nivel de tiempo y recursos computacionales. En este mismo experimento se han tenido que realizar más de 300.000 inferencias independientes. Ciertamente, habría sido ideal emplear un mayor número de muestras, pero no resultaba del todo factible desde un punto de vista práctico. Un análisis similar podría hacerse para el estudio de LLMs con la colección de 79 preguntas.

En el estudio del grounding visual, los *benchmarks* se han formulado en base a un idioma de referencia, siendo en este caso el inglés. Intuitivamente, esto es como decir que se han tomado los resultados de consultas en inglés como una guía de corrección para el resto de idiomas. Por tanto, se ha obviado, o mejor dicho, se ha asumido como válida, la posibilidad de que los resultados planteados por el inglés no fuesen correctos. Por esta razón, este experimento se ha evaluado más desde un

punto de vista de concordancia lingüística que mediante métricas de rendimiento global. Por otra parte, la colección de conceptos que componen las consultas fue diseñada manualmente, eligiendo elementos que aparecían con frecuencia en el dataset BDD100K. Esto supone una limitación, al no realizar consultas personalizadas para cada imagen según los contenidos observados. Como posible solución a estos dos problemas, se podría haber considerado un conjunto de datos con anotaciones por cajas delimitadoras, o *bounding boxes*, permitiendo comparar directamente las activaciones espaciales del modelo con anotaciones de referencia. Finalmente, ya no desde un punto de vista experimental, sino ilustrativo, también habría enriquecido el análisis mostrar en esta memoria distintos ejemplos mapas lingüísticos reales generados por los distintos pares modelo-idioma. De este modo, se podría haber estudiado precisamente dónde surgen las activaciones espaciales o dónde se identifican los conceptos en las imágenes. Sin embargo, estos mapas no fueron guardados durante la experimentación y tan solo se almacenaron las métricas una vez calculadas.

5.4. Líneas de investigación futuras

A raíz de las limitaciones expuestas, surge una serie de posibles líneas de investigación como continuación natural del trabajo abordado. Quizás la más interesante de ellas sea una integración completa de los elementos estudiados a lo largo de esta memoria dentro de la metodología de *OneMap*. Es decir, implementar el marco de evaluación multilingüe y energéticamente consciente en un sistema de mapeo semántico, permitiendo analizar el impacto de estos factores en un proceso de navegación autónoma. Esto podría realizarse tanto en entornos simulados como en plataformas físicas mediante el despliegue de robots. En estas últimas, aspectos como la latencia o la energía cobran más relevancia y se convierten en métricas de mayor interés experimental.

En la presente memoria, los distintos modelos se han evaluado con imágenes o consultas aisladas. Podría tener una mayor aplicabilidad práctica pasar a una evaluación embarcada, en la que el proceso de inferencia lo realizase el propio agente y donde las imágenes, en lugar de ser extraídas de un conjunto de datos, fuesen adquiridas en tiempo real mediante una cámara. Se podría estudiar de forma análoga la capacidad de razonamiento, interpretación visual y grounding, y resultaría interesante analizar si los resultados obtenidos se verían modificados con respecto a este trabajo y en qué medida.

A lo largo de las distintas etapas experimentales, se ha observado que la disponibilidad de recursos afecta directamente a la capacidad de razonamiento de los

modelos. Se ha argumentado que una manera de reducir esta brecha interlingüística sería equilibrar los corpus de entrenamiento. Sin embargo, precisamente por la disponibilidad desigual de datos en distintos idiomas, esto no siempre resulta factible. En este contexto, una línea de investigación relevante sería encontrar estrategias para mitigar o reducir esta disparidad en el rendimiento entre idiomas. Podría ser interesante evaluar algoritmos de traducción automática, de forma que las consultas en idiomas con menor rendimiento observado sean traducidas a un idioma puente, como el inglés, para llevar a cabo una inferencia más favorable. Otra posibilidad sería seleccionar los modelos de forma dinámica según la consulta recibida. Es decir, a partir de un conjunto de modelos, escoger automáticamente, para cada consulta, el más adecuado en función del idioma, la dificultad, el tipo de tarea o incluso los requisitos energéticos o de latencia.

Finalmente, aunque quizá sea la línea de investigación menos original, también resultaría útil ampliar esta misma evaluación a un conjunto de datos más amplio, a modelos más recientes y un mayor número de idiomas.

5.5. Publicaciones derivadas

El desarrollo de este Trabajo de Fin de Grado ha desembocado en una serie de publicaciones científicas centradas en la evaluación multilingüe y energéticamente eficiente de sistemas de inteligencia artificial. Estas publicaciones abordan distintos bloques experimentales presentados en la memoria.

I. de Zarzà, Mauro Liz, J. de Curtò y Carlos T. Calafate. «Energy-Aware Multilingual Evaluation of Large Language Models». En: *Electronics* 15.7 (2026). ISSN: 2079-9292. DOI: [10.3390/electronics15071395](https://doi.org/10.3390/electronics15071395). URL: <https://www.mdpi.com/2079-9292/15/7/1395>

J. de Curtò, Mauro Liz, I. de Zarzà y Carlos T. Calafate. «Energy-Aware Multilingual Vision–Language Models for Drone Smart Sensing». En: *Drones* 10.5 (2026). ISSN: 2504-446X. DOI: [10.3390/drones10050361](https://doi.org/10.3390/drones10050361). URL: <https://www.mdpi.com/2504-446X/10/5/361>

J. de Curtò, Mauro Liz, I. de Zarzà y Carlos T. Calafate. «Multilingual VLM Energy Signatures for Edge Deployment». En: *Proceedings of the 2026 International Conference on Artificial Intelligence Models and Systems (IEEE iAIMS2026)*. Accepted as Full Paper. Barcelona, Spain, nov. de 2026

J. de Curtò, Mauro Liz e I. de Zarzà. *Language-Conditioned Visual Grounding with CLIP Multilingual*. 9 de mayo de 2026. DOI: [10.48550/arXiv.2605.09060](https://doi.org/10.48550/arXiv.2605.09060). arXiv: [2605.09060\[cs.CL\]](https://arxiv.org/abs/2605.09060). URL: <http://arxiv.org/abs/2605.09060> (visitado 07-07-2026)

Apéndice A

Alineación con los ODS

A lo largo de la memoria se menciona de forma repetida el impacto social, medioambiental y energético de las tecnologías abordadas. No obstante, a modo de resumen, en este anexo se realiza una breve reflexión acerca de los Objetivos de Desarrollo Sostenible (ODS) de las Naciones Unidas que mejor encajan en el marco de este trabajo.

Objetivo 10: Reducir la desigualdad en y entre los países

Este proyecto analiza el idioma como dimensión adicional en los procesos de inferencia de los modelos de lenguaje y visión-lenguaje. Este marco de investigación se alinea con el ODS número 10, pues el idioma es una lente mediante la cual personas de distintos países y orígenes perciben e interpretan su entorno; y muchas veces no se tiene en cuenta en el diseño de las tecnologías abordadas. Precisamente, se ha investigado si el idioma condiciona el funcionamiento de los sistemas de inteligencia artificial — que han surgido como unas de las herramientas más relevantes y disruptivas de esta década. Por tanto, mediante el desarrollo experimental realizado, se puede comprobar si existe verdaderamente un acceso desigual a la inteligencia artificial introducido por el idioma en el que habla una persona.

A partir de los resultados obtenidos, se ha demostrado que, efectivamente, el idioma afecta de forma clara al rendimiento de los agentes multimodales en tareas como el razonamiento o la interpretación de imágenes. Es decir, existe una desigualdad a nivel lingüístico. Los modelos estudiados presentan un rendimiento favorable en inglés, que tiende a degradarse a medida que aumenta la distancia léxica respecto a este idioma. En particular, destacan los idiomas de bajos recursos, como el euskera, el luxemburgués o el árabe, como aquellos que generalmente presentan los peores resultados. Se recalca que este comportamiento no es inten-

cionado, ni mucho menos, sino que tiende a estar relacionado con la disponibilidad de datos y el desarrollo de *benchmarks*. En este ámbito, el inglés surge como el idioma dominante de la comunidad científica responsable del desarrollo de estas tecnologías.

Aunque la mitigación de estas desigualdades a nivel tecnológico queda fuera del alcance de este proyecto, sí se ha hecho bastante énfasis en visibilizar la disparidad lingüística existente, y por ello se menciona el ODS 10.

Objetivo 7: Energía asequible y no contaminante

El Objetivo de Desarrollo Sostenible número 7 se centra en la energía asequible y no contaminante. Se trata de un objetivo de carácter transversal y, por tanto, fuertemente relacionado con los Objetivos de Desarrollo Sostenible número 9 —Industria, innovación e infraestructuras— y número 13 —Acción por el clima—. La reflexión que se presenta a continuación podría abordarse desde cualquiera de estas tres perspectivas; no obstante, para evitar redundancias, se mencionan conjuntamente en este apartado.

Si bien el idioma ha sido uno de los principales ejes de análisis del estudio realizado, el consumo energético constituye el otro gran foco del trabajo. Según la ONU, “la energía es central para casi todos los grandes desafíos y oportunidades a los que se enfrenta el mundo en la actualidad” [46]; y, ciertamente, el consumo energético tiene un papel fundamental en el ámbito de la inteligencia artificial. De hecho, como se ha indicado en la memoria, se estima que el consumo energético de los centros de datos representará aproximadamente el 3 % del consumo eléctrico global en 2030 [38].

En el marco experimental abordado, se ha reiterado la importancia de adoptar métricas de evaluación que consideren no únicamente el rendimiento de los modelos, sino también su consumo de recursos computacionales en los procesos de inferencia. En este sentido, se ha insistido en el diseño de *benchmarks* energéticamente conscientes. En particular, se ha propuesto el *AI Energy Score* como métrica de evaluación de modelos de lenguaje, integrando bajo un único marco evaluativo el rendimiento y el consumo energético. Dichos *benchmarks* se han empleado para evaluar el desempeño de distintos modelos en tareas de razonamiento e interpretación visual, mostrando que un mayor consumo energético no siempre se traduce en mejores resultados. Esto tiene implicaciones importantes en lo que respecta al diseño de arquitecturas energéticamente eficientes.

Apéndice B

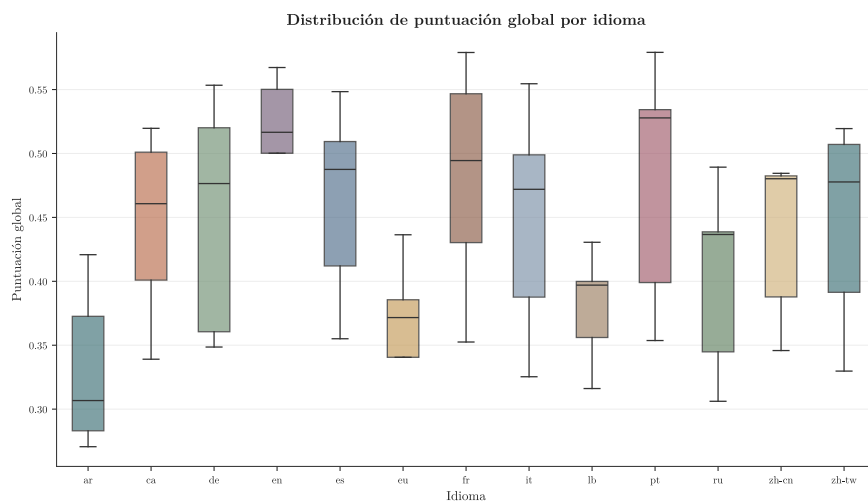
Resultados detallados de la evaluación multilingüe de LLMs

Cuadro B.1: Resultados detallados por modelo e idioma.

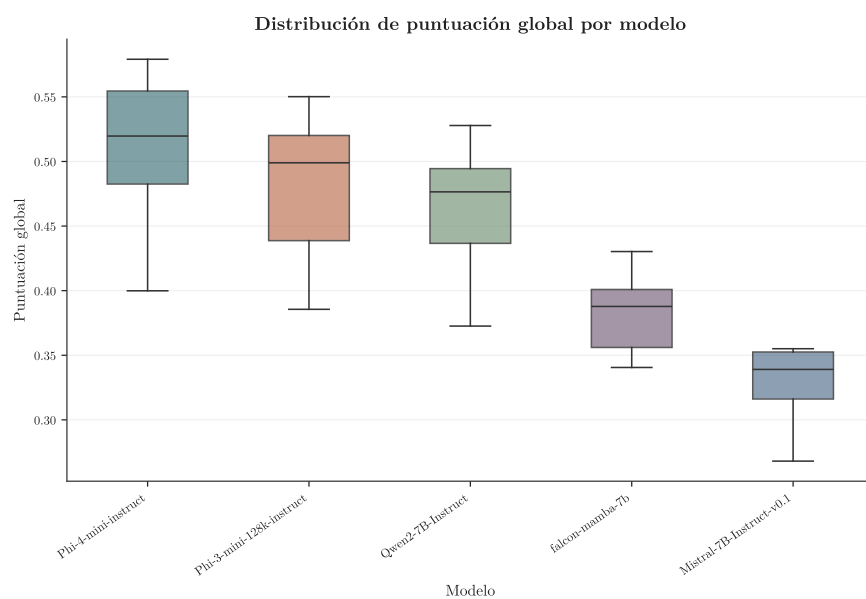
Modelo	Idioma	E_{1000} Wh/1000 cons.	S_{overall}	S_{step}	C	Lat. s/cons.
Falcon-Mamba-7B	ar	461.1	0.283	0.461	0.062	7.04
Falcon-Mamba-7B	ca	447.3	0.401	0.522	0.046	6.83
Falcon-Mamba-7B	de	458.1	0.361	0.526	0.061	6.98
Falcon-Mamba-7B	en	469.0	0.500	0.613	0.053	7.15
Falcon-Mamba-7B	es	465.5	0.412	0.540	0.054	7.09
Falcon-Mamba-7B	eu	476.9	0.341	0.449	0.063	7.26
Falcon-Mamba-7B	fr	460.2	0.430	0.497	0.061	6.93
Falcon-Mamba-7B	it	452.4	0.388	0.527	0.060	6.80
Falcon-Mamba-7B	lb	479.1	0.356	0.467	0.055	7.21
Falcon-Mamba-7B	pt	445.8	0.399	0.544	0.046	6.67
Falcon-Mamba-7B	ru	467.8	0.345	0.449	0.066	6.96
Falcon-Mamba-7B	zh-cn	454.8	0.388	0.565	0.059	6.78
Falcon-Mamba-7B	zh-tw	458.5	0.391	0.524	0.061	6.82
Mistral-7B-Instruct-v0.1	ar	250.8	0.271	0.390	0.047	2.84
Mistral-7B-Instruct-v0.1	ca	98.4	0.339	0.399	0.044	1.17
Mistral-7B-Instruct-v0.1	de	134.2	0.349	0.391	0.045	1.55
Mistral-7B-Instruct-v0.1	en	146.7	0.408	0.416	0.047	1.70
Mistral-7B-Instruct-v0.1	es	129.4	0.355	0.406	0.035	1.51
Mistral-7B-Instruct-v0.1	eu	211.7	0.268	0.335	0.045	2.42
Mistral-7B-Instruct-v0.1	fr	106.3	0.353	0.373	0.057	1.25
Mistral-7B-Instruct-v0.1	it	113.3	0.325	0.378	0.034	1.33
Mistral-7B-Instruct-v0.1	lb	139.4	0.316	0.356	0.057	1.61
Mistral-7B-Instruct-v0.1	pt	84.8	0.354	0.380	0.044	1.02
Mistral-7B-Instruct-v0.1	ru	154.5	0.306	0.371	0.044	1.80
Mistral-7B-Instruct-v0.1	zh-cn	239.0	0.346	0.433	0.049	2.71
Mistral-7B-Instruct-v0.1	zh-tw	236.1	0.330	0.434	0.054	2.67
Phi-3-mini-128k-instruct	ar	275.3	0.307	0.489	0.075	3.71
Phi-3-mini-128k-instruct	ca	261.3	0.501	0.659	0.051	3.54
Phi-3-mini-128k-instruct	de	276.9	0.520	0.705	0.055	3.72
Phi-3-mini-128k-instruct	en	259.7	0.550	0.676	0.047	3.48
Phi-3-mini-128k-instruct	es	272.0	0.509	0.704	0.052	3.67

APÉNDICE B. RESULTADOS DETALLADOS DE LA EVALUACIÓN
MULTILINGÜE DE LLMS

Modelo	Idioma	E_{1000} Wh/1000 cons.	S_{overall}	S_{step}	C	Lat. s/cons.
Phi-3-mini-128k-instruct	eu	269.6	0.386	0.533	0.054	3.63
Phi-3-mini-128k-instruct	fr	272.2	0.547	0.709	0.059	3.65
Phi-3-mini-128k-instruct	it	273.4	0.499	0.717	0.050	3.69
Phi-3-mini-128k-instruct	lb	272.4	0.431	0.621	0.062	3.67
Phi-3-mini-128k-instruct	pt	270.8	0.534	0.706	0.060	3.65
Phi-3-mini-128k-instruct	ru	271.6	0.439	0.592	0.064	3.67
Phi-3-mini-128k-instruct	zh-cn	280.2	0.484	0.671	0.060	3.73
Phi-3-mini-128k-instruct	zh-tw	281.7	0.478	0.693	0.071	3.75
Phi-4-mini-instruct	ar	254.7	0.421	0.625	0.079	3.59
Phi-4-mini-instruct	ca	257.8	0.520	0.682	0.074	3.60
Phi-4-mini-instruct	de	245.5	0.553	0.696	0.055	3.42
Phi-4-mini-instruct	en	246.0	0.567	0.669	0.053	3.43
Phi-4-mini-instruct	es	246.8	0.548	0.722	0.060	3.44
Phi-4-mini-instruct	eu	269.6	0.372	0.474	0.096	3.75
Phi-4-mini-instruct	fr	243.8	0.579	0.708	0.059	3.41
Phi-4-mini-instruct	it	246.5	0.555	0.696	0.053	3.46
Phi-4-mini-instruct	lb	266.6	0.400	0.524	0.092	3.70
Phi-4-mini-instruct	pt	243.5	0.579	0.716	0.063	3.39
Phi-4-mini-instruct	ru	248.4	0.489	0.653	0.050	3.46
Phi-4-mini-instruct	zh-cn	251.1	0.483	0.702	0.056	3.48
Phi-4-mini-instruct	zh-tw	246.0	0.519	0.708	0.060	3.42
Qwen2-7B-Instruct	ar	448.9	0.373	0.558	0.060	4.97
Qwen2-7B-Instruct	ca	460.7	0.461	0.616	0.052	5.07
Qwen2-7B-Instruct	de	464.7	0.476	0.633	0.052	5.14
Qwen2-7B-Instruct	en	453.4	0.517	0.687	0.035	5.02
Qwen2-7B-Instruct	es	461.8	0.488	0.681	0.043	5.08
Qwen2-7B-Instruct	eu	474.8	0.436	0.581	0.060	5.22
Qwen2-7B-Instruct	fr	472.8	0.494	0.674	0.061	5.19
Qwen2-7B-Instruct	it	465.8	0.472	0.677	0.044	5.15
Qwen2-7B-Instruct	lb	476.5	0.397	0.547	0.051	5.27
Qwen2-7B-Instruct	pt	460.9	0.528	0.690	0.042	5.10
Qwen2-7B-Instruct	ru	464.6	0.437	0.570	0.049	5.15
Qwen2-7B-Instruct	zh-cn	468.5	0.480	0.698	0.035	5.18
Qwen2-7B-Instruct	zh-tw	466.4	0.507	0.692	0.052	5.15



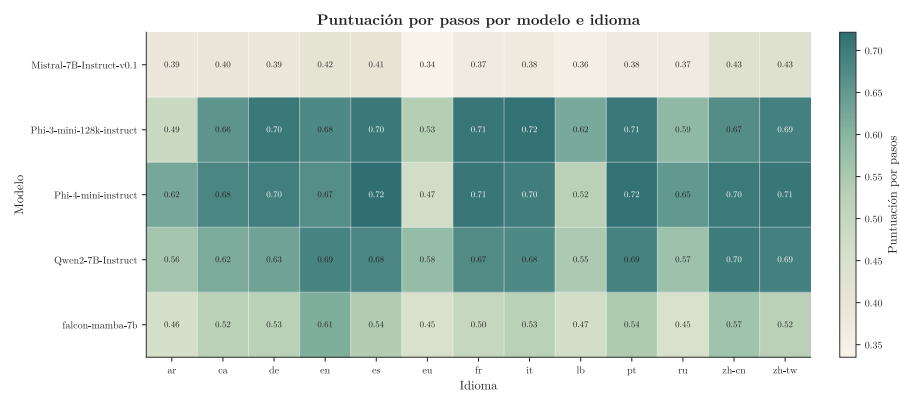
(a) Distribución de la puntuación global por idioma.



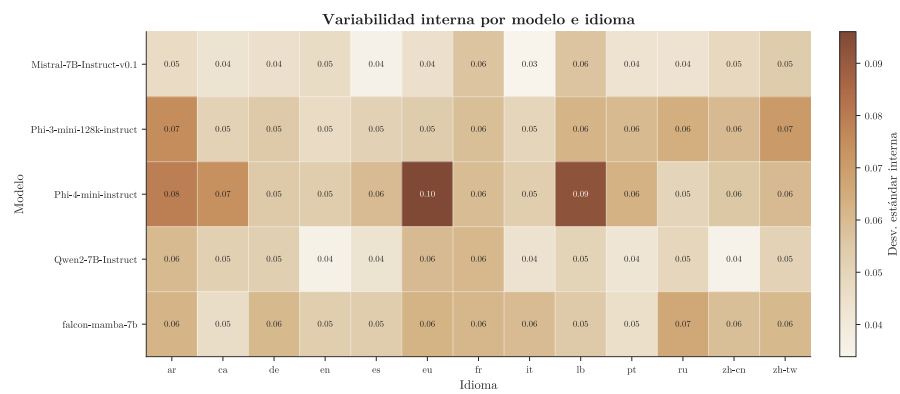
(b) Distribución de la puntuación global por modelo.

Figura B.1: Distribución general del rendimiento.

APÉNDICE B. RESULTADOS DETALLADOS DE LA EVALUACIÓN MULTILINGÜE DE LLMS

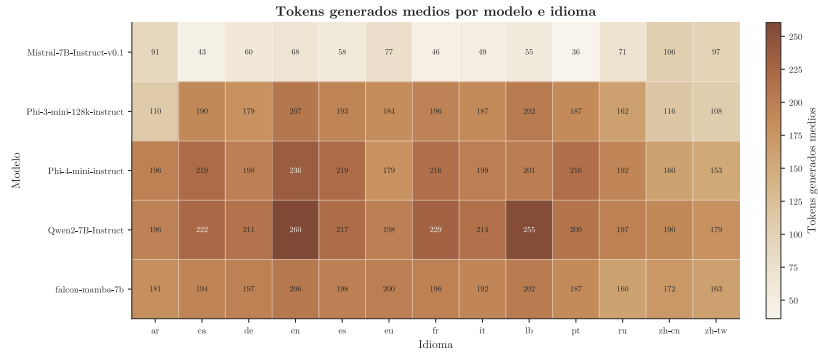


(a) Puntuación por pasos desagregada por modelo e idioma.

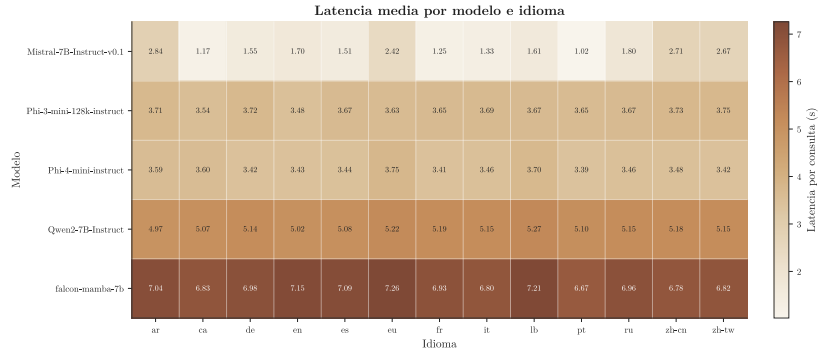


(b) Consistencia de las respuestas por modelo e idioma, medida a través de la desviación estándar.

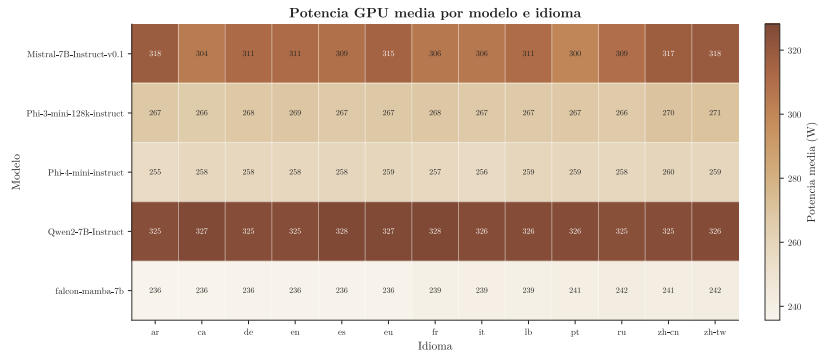
Figura B.2: Rendimiento y consistencia por modelo e idioma.



(a) Número medio de tokens generados por modelo e idioma.



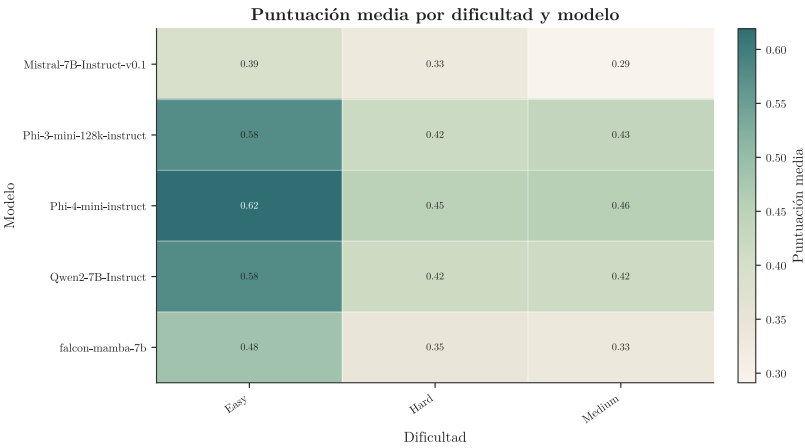
(b) Latencia media por consulta desagregada por modelo e idioma.



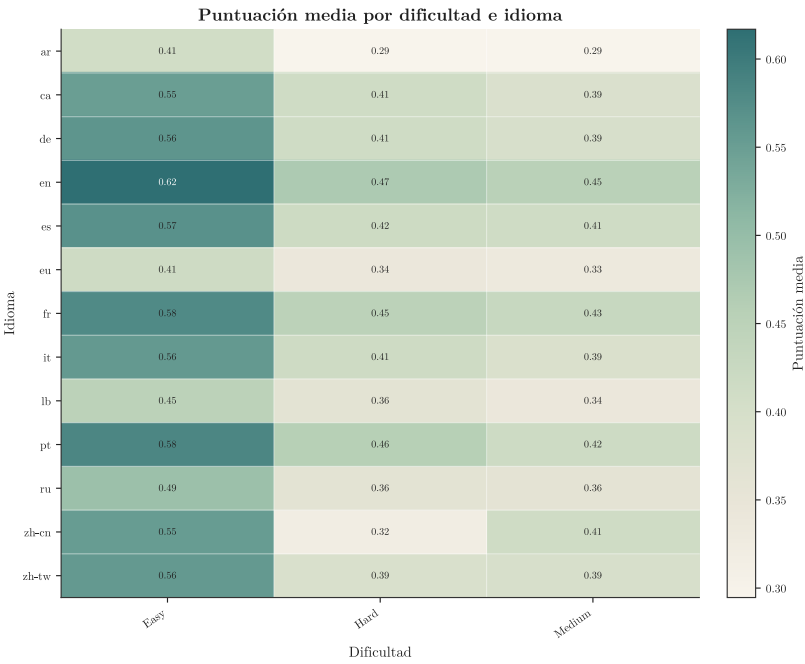
(c) Potencia media de GPU registrada durante la inferencia.

Figura B.3: Métricas de generación y coste computacional por modelo e idioma.

APÉNDICE B. RESULTADOS DETALLADOS DE LA EVALUACIÓN MULTILINGÜE DE LLMS

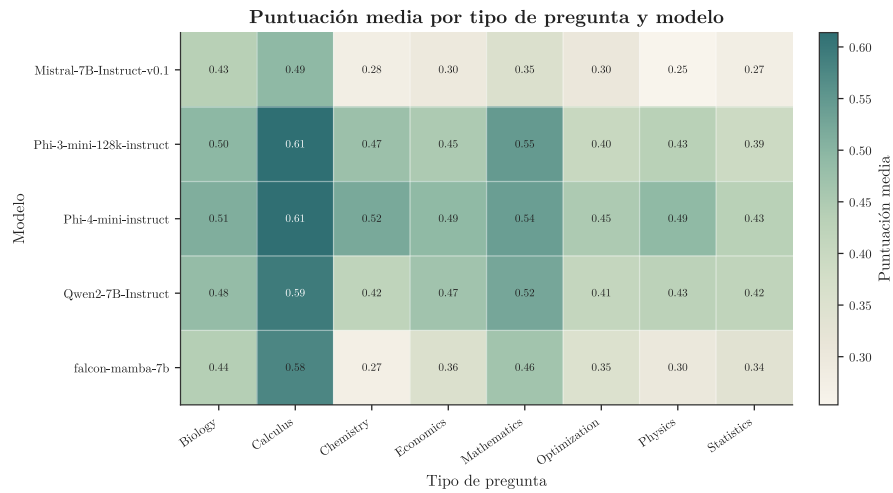


(a) Puntuación media por dificultad y modelo.

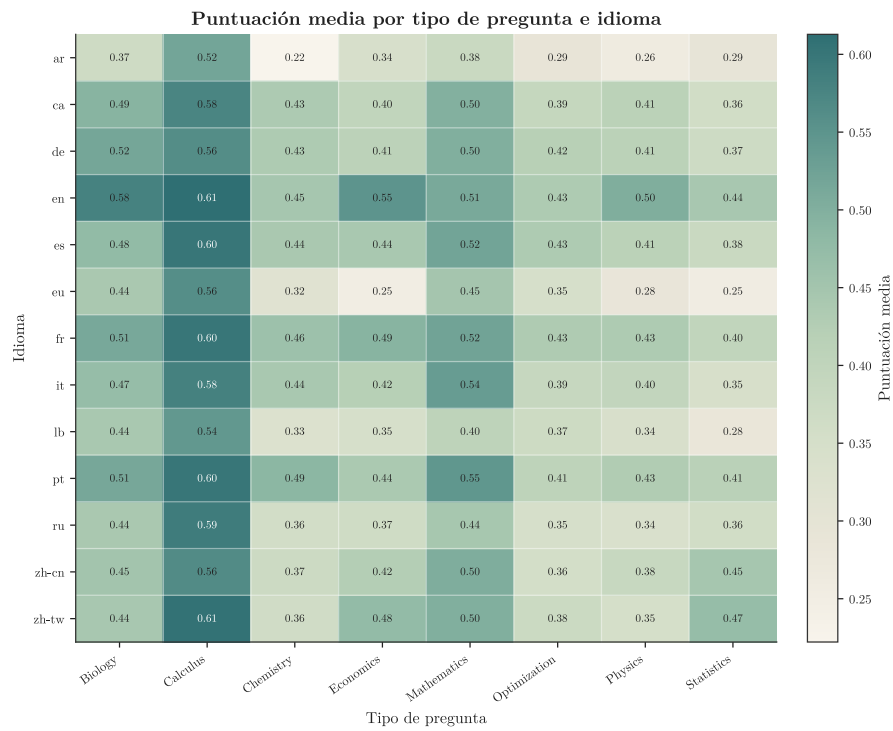


(b) Puntuación media por dificultad e idioma.

Figura B.4: Rendimiento según la dificultad de las preguntas.



(a) Puntuación media por tipo de pregunta y modelo.



(b) Puntuación media por tipo de pregunta e idioma.

Figura B.5: Rendimiento desagregado por tipo de pregunta.

*APÉNDICE B. RESULTADOS DETALLADOS DE LA EVALUACIÓN
MULTILINGÜE DE LLMS*

Apéndice C

Resultados detallados de la evaluación multilingüe de VLMs

Cuadro C.1: Consumo energético por modelo e idioma para la versión original del experimento VLM.

Modelo	ar	ca	de	en	es	eu	fr	it	lb	pt	ru	zh-cn	zh-tw
LLaVA-1.6	167.80	124.53	135.35	127.69	146.39	55.39	152.31	111.27	129.33	112.89	147.26	138.35	141.96
LLaVA-1.5	141.72	91.71	67.62	61.83	85.07	36.52	72.69	73.66	78.99	70.80	81.35	96.87	113.07
InternVL2	180.95	169.08	173.73	169.32	175.88	85.79	177.75	169.86	155.57	174.96	180.27	161.92	168.30
Qwen2-VL	100.66	148.87	111.30	113.74	116.80	130.82	127.87	133.07	115.65	102.80	114.61	48.53	86.50
Phi-3-V	121.55	92.38	56.38	31.15	44.16	27.27	45.88	46.67	47.46	73.38	76.34	92.59	107.12

Cuadro C.2: Consumo energético por modelo e idioma para la versión actualizada del experimento VLM.

Modelo	ar	ca	de	en	es	eu	fr	it	lb	pt	ru	zh-cn	zh-tw
Qwen2-VL	94.40	139.67	104.82	106.77	109.78	122.72	120.09	122.10	108.94	96.92	107.41	46.23	81.62
InternVL2	191.75	179.33	183.90	180.22	186.99	92.81	189.01	178.07	166.37	185.38	190.92	171.29	177.31
LLaVA-1.6	154.71	119.03	126.19	119.90	136.76	51.69	141.27	100.65	120.82	105.57	136.60	130.02	131.13
LLaVA-1.5	126.69	83.15	62.35	57.14	77.45	33.89	66.62	66.96	73.19	64.34	74.06	87.41	102.04
Phi-3-V	119.81	91.36	54.49	31.14	44.40	26.72	46.00	46.99	47.75	74.35	76.17	92.29	106.79

APÉNDICE C. RESULTADOS DETALLADOS DE LA EVALUACIÓN
MULTILINGÜE DE VLMS

Cuadro C.3: Puntuación media por modelo e idioma para la versión original del experimento VLM.

Modelo	ar	ca	de	en	es	eu	fr	it	lb	pt	ru	zh-cn	zh-tw
LLaVA-1.6	-0.012	0.207	0.112	0.559	0.165	0.025	0.147	0.395	0.063	0.144	0.148	0.049	0.081
LLaVA-1.5	-0.006	0.197	0.093	0.437	0.147	0.110	0.141	0.136	0.266	0.130	-0.011	0.048	0.095
InternVL2	-0.001	0.166	0.068	0.570	0.158	0.038	0.149	0.139	0.210	0.153	-0.009	0.044	0.090
Qwen2-VL	-0.005	0.198	0.083	0.534	0.161	0.020	0.139	0.155	0.075	0.157	-0.007	0.044	0.091
Phi-3-V	-0.013	0.167	0.080	0.455	0.152	0.010	0.150	0.141	0.072	0.142	0.154	0.032	0.084

Cuadro C.4: Puntuación media por modelo e idioma para la versión actualizada del experimento VLM.

Modelo	ar	ca	de	en	es	eu	fr	it	lb	pt	ru	zh-cn	zh-tw
Qwen2-VL	0.338	0.471	0.506	0.609	0.474	0.193	0.572	0.452	0.107	0.538	0.412	0.389	0.446
InternVL2	0.245	0.354	0.468	0.652	0.448	0.173	0.508	0.384	0.095	0.494	0.365	0.615	0.573
LLaVA-1.6	0.200	0.406	0.527	0.626	0.427	0.090	0.517	0.299	0.107	0.430	0.284	0.424	0.441
LLaVA-1.5	0.254	0.377	0.445	0.459	0.420	0.176	0.453	0.385	0.081	0.414	0.351	0.370	0.379
Phi-3-V	0.166	0.276	0.492	0.487	0.418	0.145	0.496	0.349	0.154	0.390	0.271	0.381	0.367

Cuadro C.5: Puntuación media por tipo de tarea e idioma, promediada entre modelos, para la versión original del experimento VLM.

Idioma	Scene Understanding	Vehicle Detection	Road Condition	Navigation Reasoning
ar	-0.041	0.001	0.000	0.010
ca	0.222	0.254	0.092	0.180
de	0.193	0.018	0.061	0.077
en	0.649	0.703	0.351	0.340
es	0.268	0.150	0.083	0.125
eu	0.081	0.039	0.001	0.042
fr	0.279	0.081	0.010	0.210
it	0.293	0.282	0.099	0.097
lb	0.257	0.185	0.031	0.076
pt	0.244	0.148	0.093	0.096
ru	0.018	0.100	0.057	0.046
zh-cn	0.100	0.005	0.001	0.067
zh-tw	0.204	0.003	0.001	0.143

Cuadro C.6: Puntuación media por tipo de tarea e idioma, promediada entre modelos, para la versión actualizada del experimento VLM.

Idioma	Scene Understanding	Vehicle Detection	Road Condition	Navigation Reasoning
ar	0.511	0.189	0.008	0.255
ca	0.723	0.264	0.129	0.392
de	0.745	0.565	0.234	0.408
en	0.693	0.701	0.354	0.519
es	0.781	0.411	0.135	0.423
eu	0.232	0.091	0.095	0.203
fr	0.733	0.512	0.324	0.467
it	0.750	0.234	0.085	0.426
lb	0.204	0.085	0.008	0.138
pt	0.757	0.498	0.160	0.396
ru	0.727	0.215	0.049	0.356
zh-cn	0.685	0.418	0.158	0.482
zh-tw	0.692	0.430	0.149	0.494

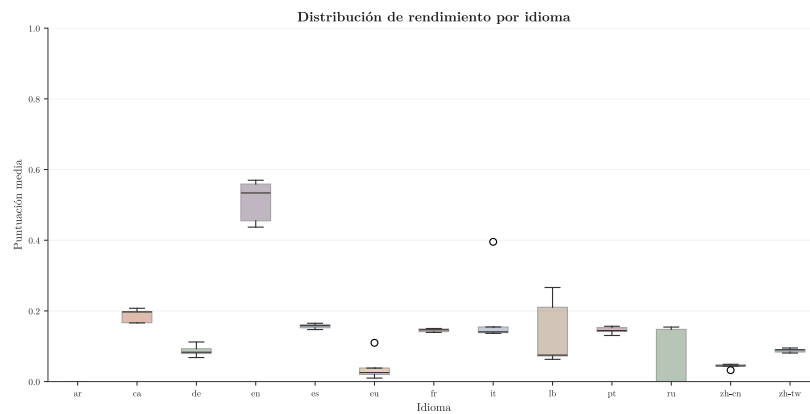
APÉNDICE C. RESULTADOS DETALLADOS DE LA EVALUACIÓN
MULTILINGÜE DE VLMS

Cuadro C.7: Puntuación media por tipo de tarea y modelo para la versión original del experimento VLM.

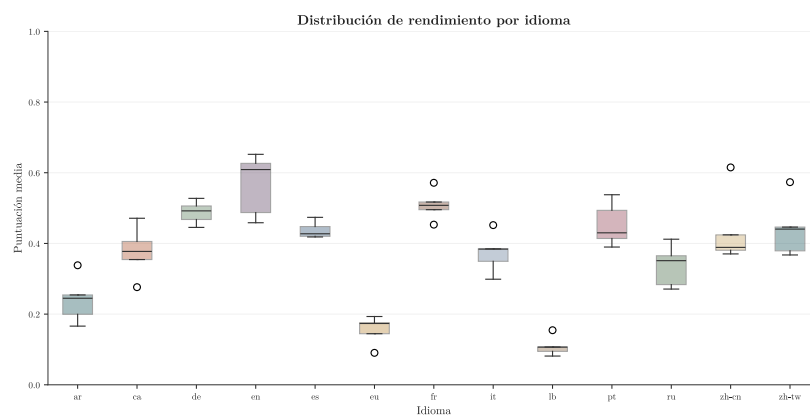
Modelo	Scene Understanding	Vehicle Detection	Road Condition	Navigation Reasoning
LLaVA-1.6	0.229	0.215	0.083	0.113
LLaVA-1.5	0.242	0.146	0.036	0.124
InternVL2	0.183	0.164	0.072	0.127
Qwen2-VL	0.192	0.123	0.075	0.115
Phi-3-V	0.217	0.109	0.072	0.101

Cuadro C.8: Puntuación media por tipo de tarea y modelo para la versión actualizada del experimento VLM.

Modelo	Scene Understanding	Vehicle Detection	Road Condition	Navigation Reasoning
Qwen2-VL	0.643	0.409	0.197	0.445
InternVL2	0.638	0.391	0.156	0.469
LLaVA-1.6	0.603	0.329	0.175	0.362
LLaVA-1.5	0.652	0.347	0.083	0.323
Phi-3-V	0.630	0.298	0.115	0.308



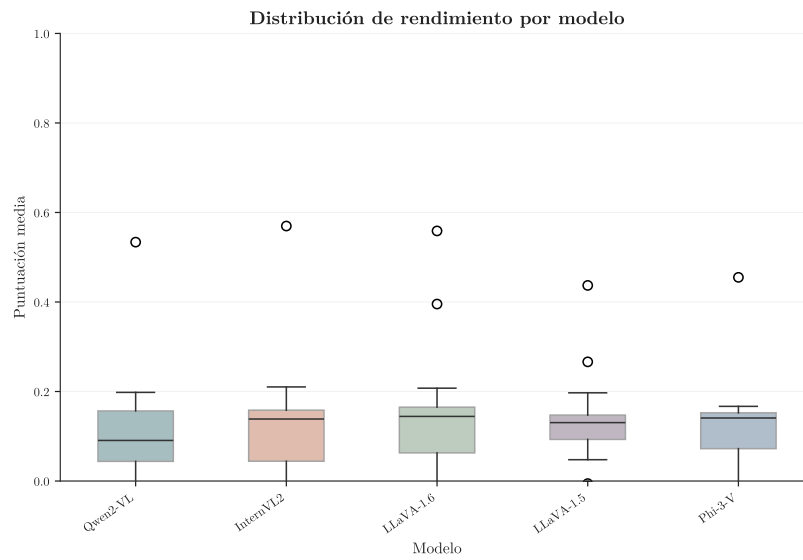
(a) Versión original.



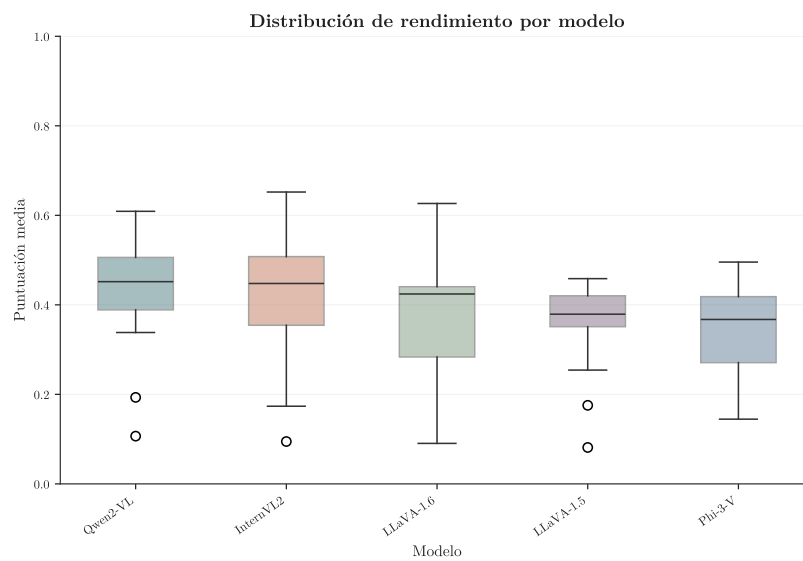
(b) Versión actualizada.

Figura C.1: Comparación de la distribución del rendimiento por idioma.

APÉNDICE C. RESULTADOS DETALLADOS DE LA EVALUACIÓN MULTILINGÜE DE VLMS

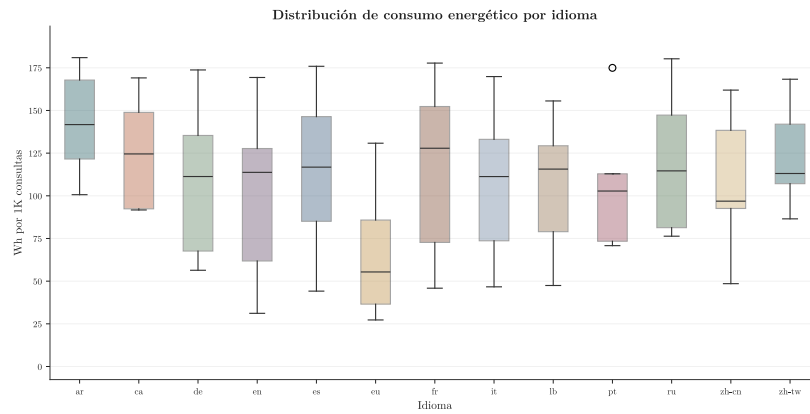


(a) Versión original.

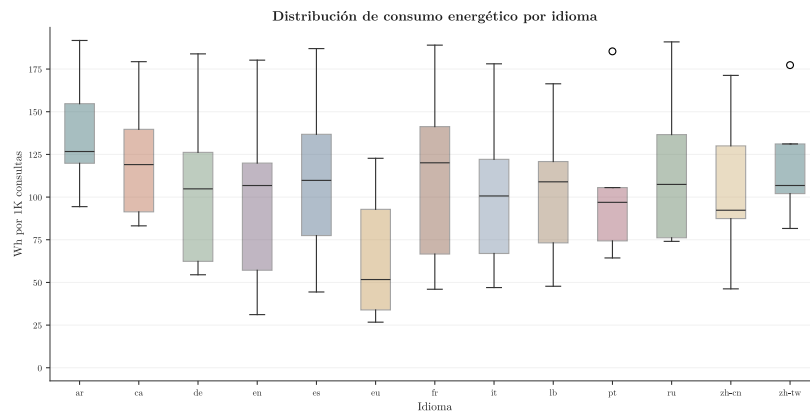


(b) Versión actualizada.

Figura C.2: Comparación de la distribución del rendimiento por modelo.



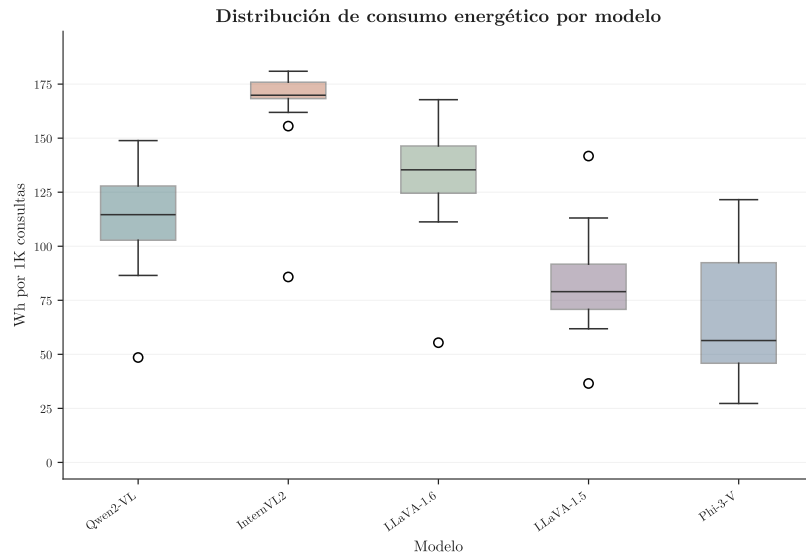
(a) Versión original.



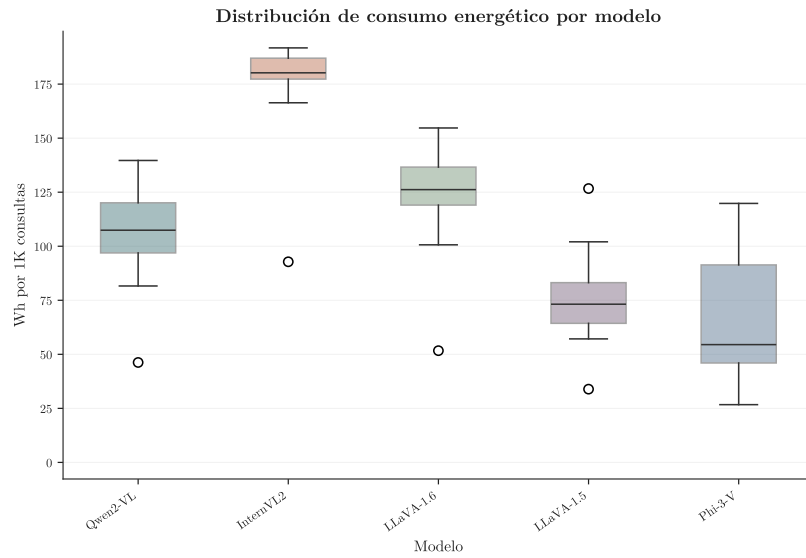
(b) Versión actualizada.

Figura C.3: Comparación de la distribución del consumo energético por idioma.

APÉNDICE C. RESULTADOS DETALLADOS DE LA EVALUACIÓN MULTILINGÜE DE VLMS

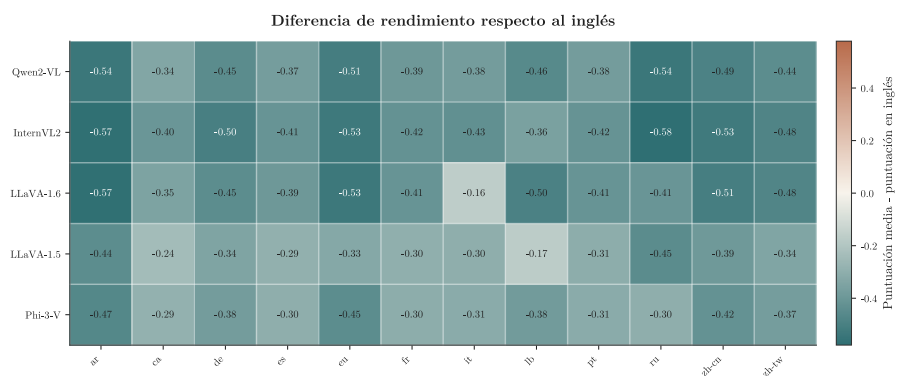


(a) Versión original.

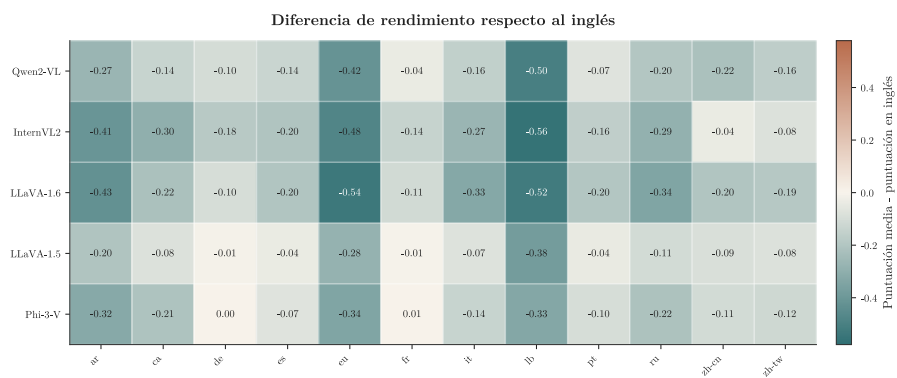


(b) Versión actualizada.

Figura C.4: Comparación de la distribución del consumo energético por modelo.



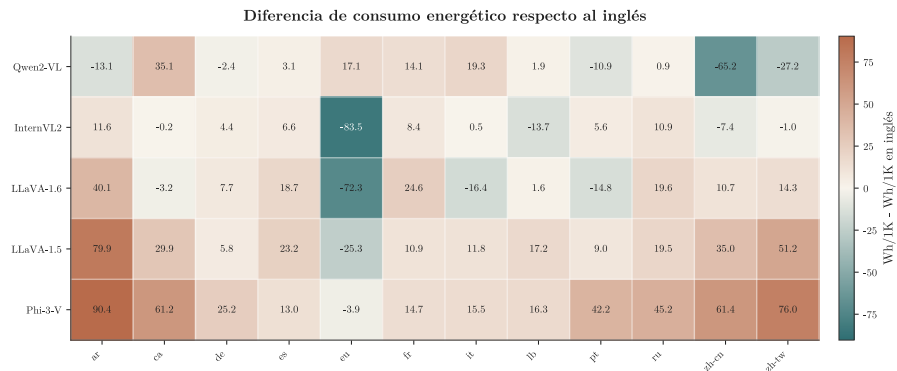
(a) Versión original.



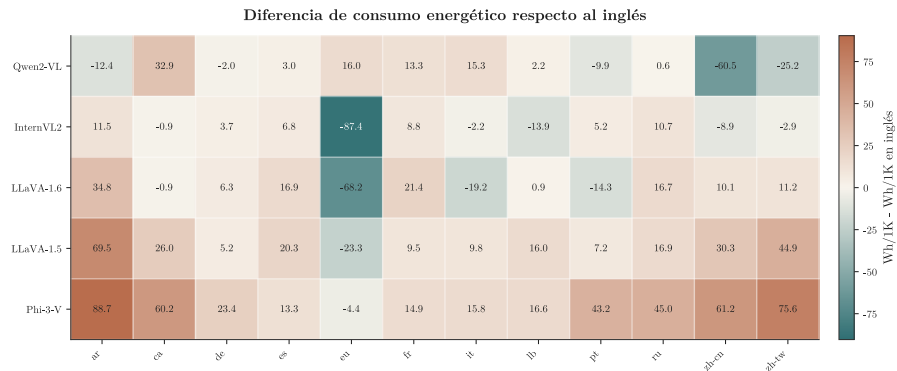
(b) Versión actualizada.

Figura C.5: Comparación de la diferencia de rendimiento respecto al inglés.

APÉNDICE C. RESULTADOS DETALLADOS DE LA EVALUACIÓN MULTILINGÜE DE VLMS

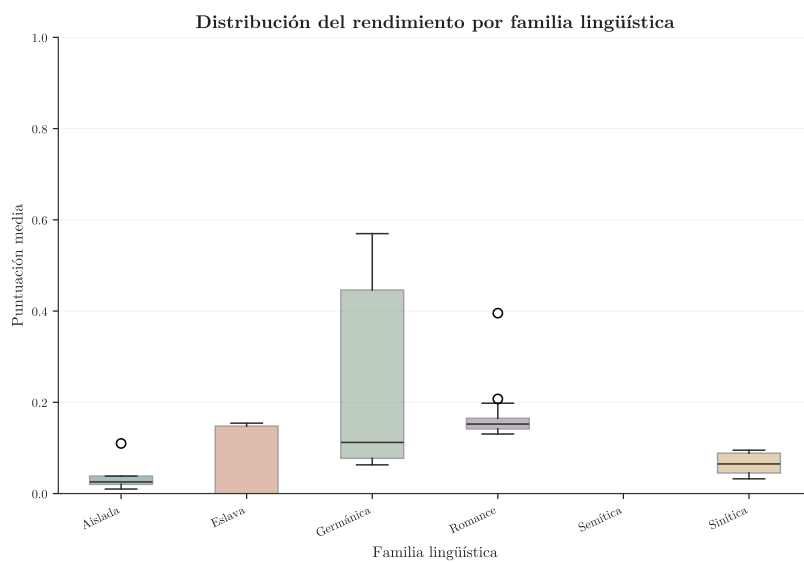


(a) Versión original.

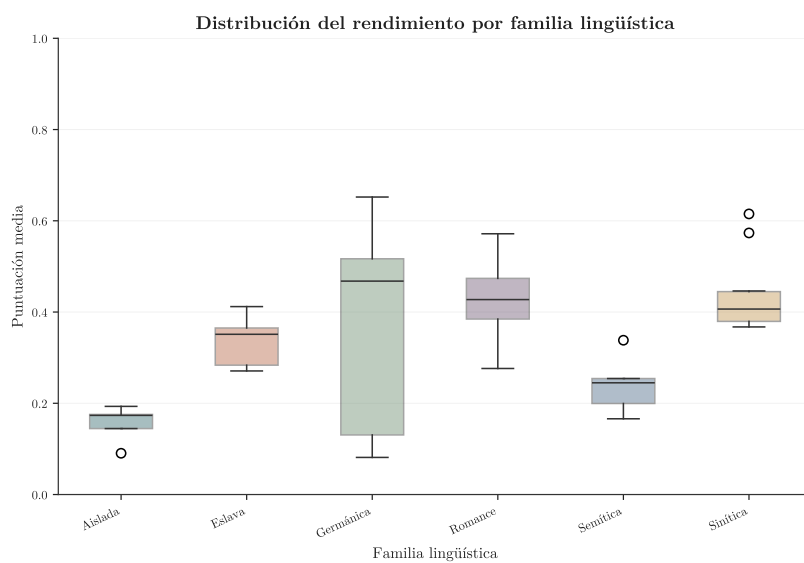


(b) Versión actualizada.

Figura C.6: Comparación de la diferencia de consumo energético respecto al inglés.



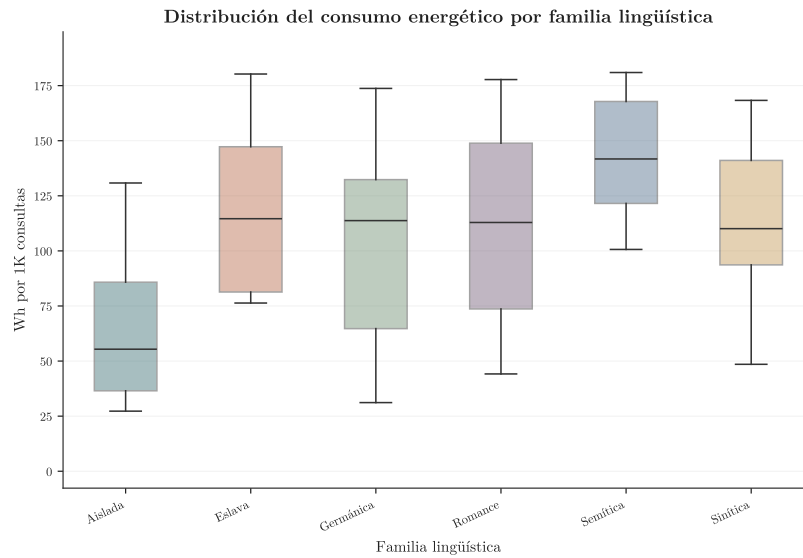
(a) Versión original.



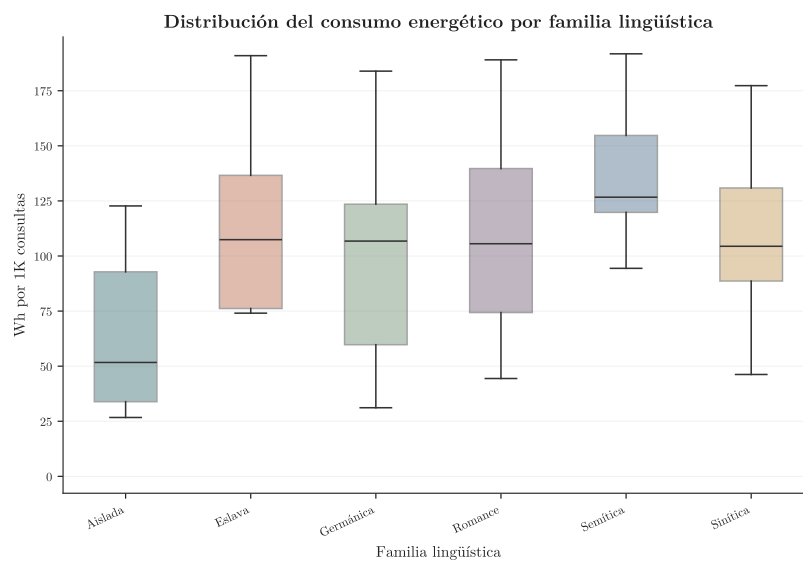
(b) Versión actualizada.

Figura C.7: Comparación de la distribución del rendimiento por familia lingüística.

APÉNDICE C. RESULTADOS DETALLADOS DE LA EVALUACIÓN MULTILINGÜE DE VLMS

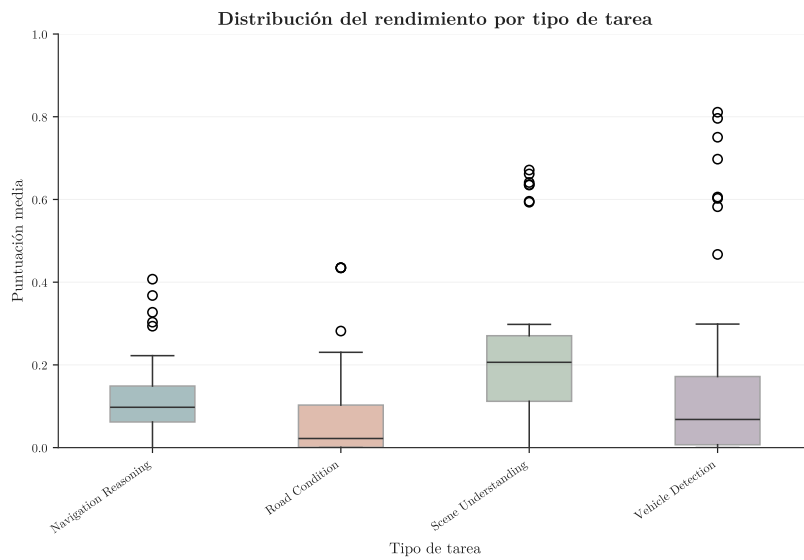


(a) Versión original.

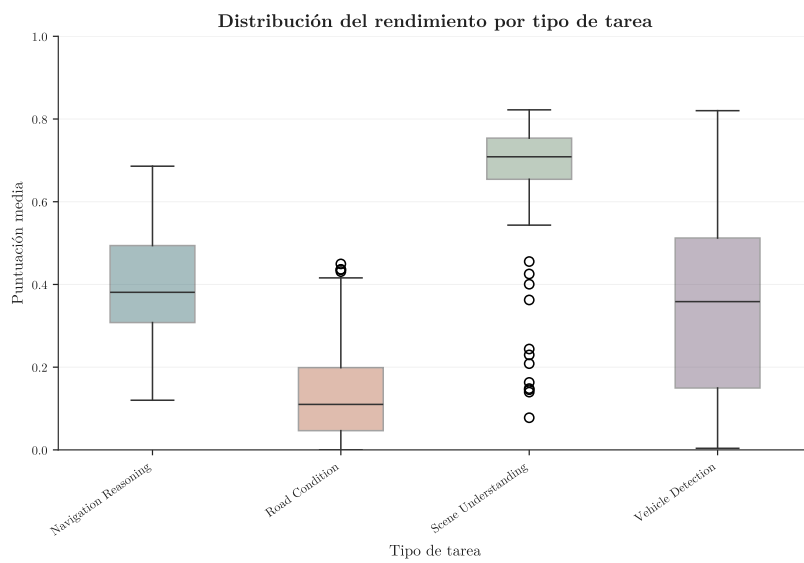


(b) Versión actualizada.

Figura C.8: Comparación de la distribución del consumo energético por familia lingüística.



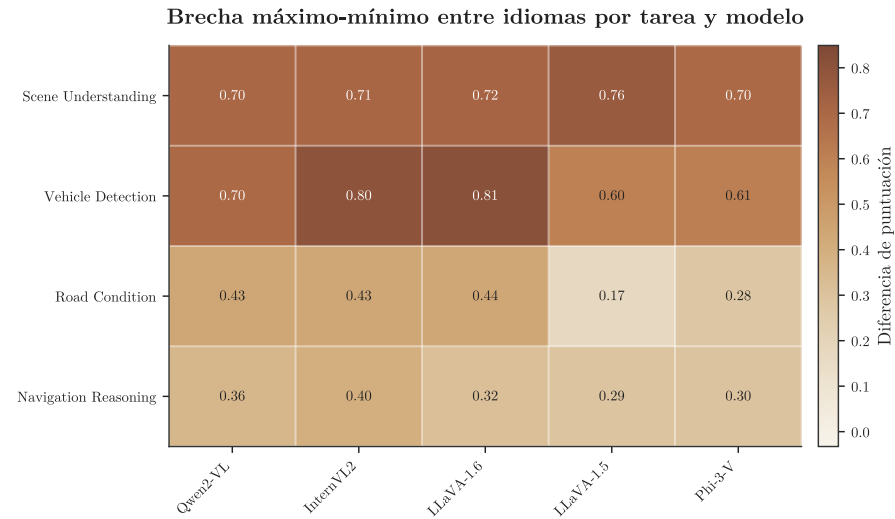
(a) Versión original.



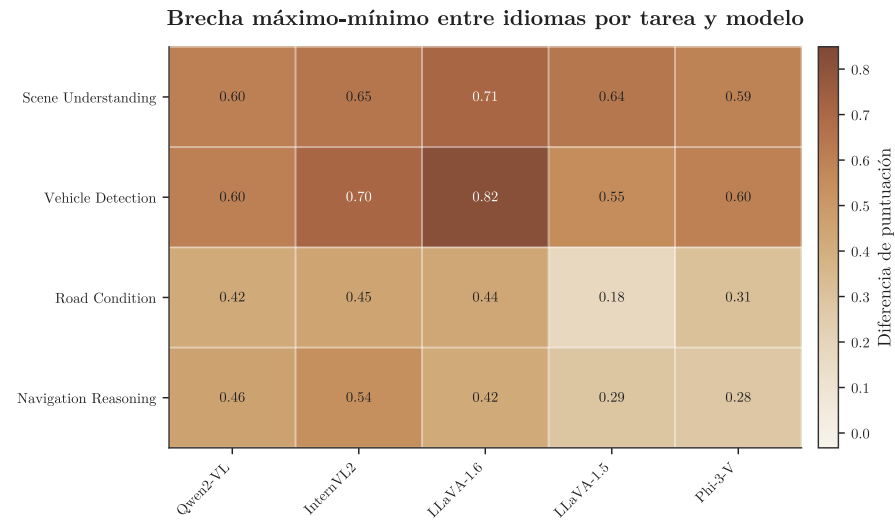
(b) Versión actualizada.

Figura C.9: Comparación de la distribución del rendimiento por tipo de tarea.

APÉNDICE C. RESULTADOS DETALLADOS DE LA EVALUACIÓN
MULTILINGÜE DE VLMS

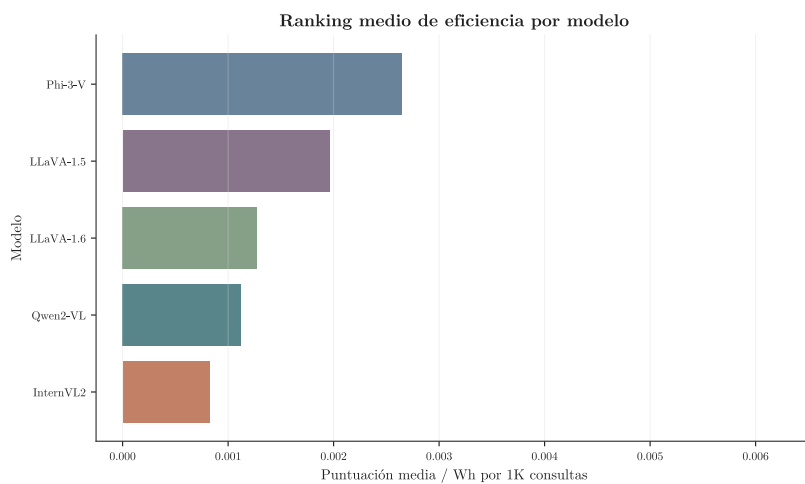


(a) Versión original.

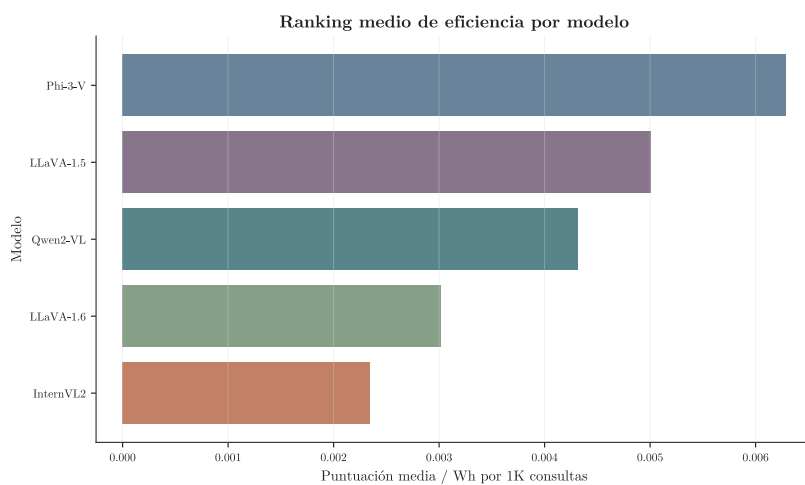


(b) Versión actualizada.

Figura C.10: Comparación de la brecha máxima-mínima entre idiomas por tipo de tarea y modelo.



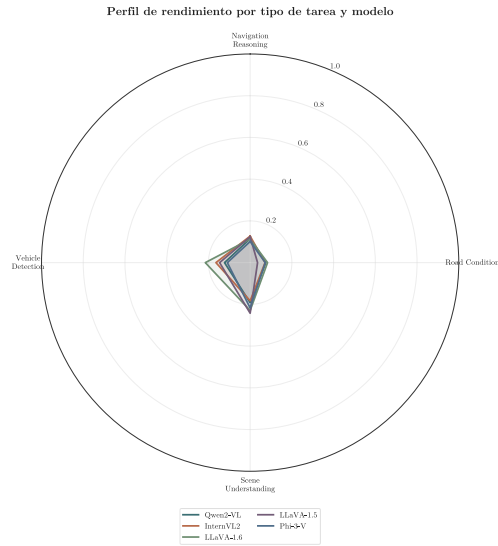
(a) Versión original.



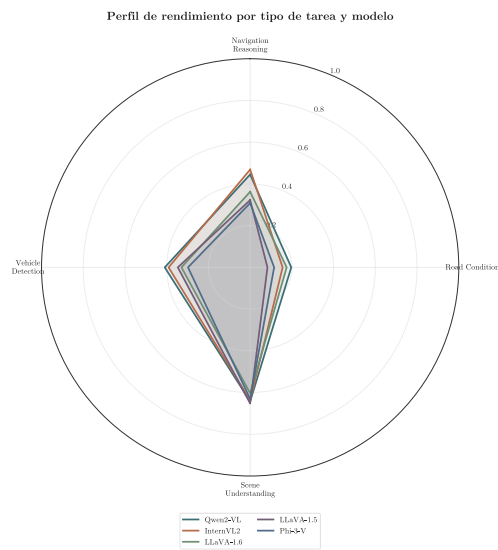
(b) Versión actualizada.

Figura C.11: Comparación del ranking medio de eficiencia por modelo.

APÉNDICE C. RESULTADOS DETALLADOS DE LA EVALUACIÓN MULTILINGÜE DE VLMS



(a) Versión original.



(b) Versión actualizada.

Figura C.12: Comparación del perfil de rendimiento por tipo de tarea y modelo.

Apéndice D

Resultados detallados de la evaluación del grounding visual multilingüe en el contexto de OneMap

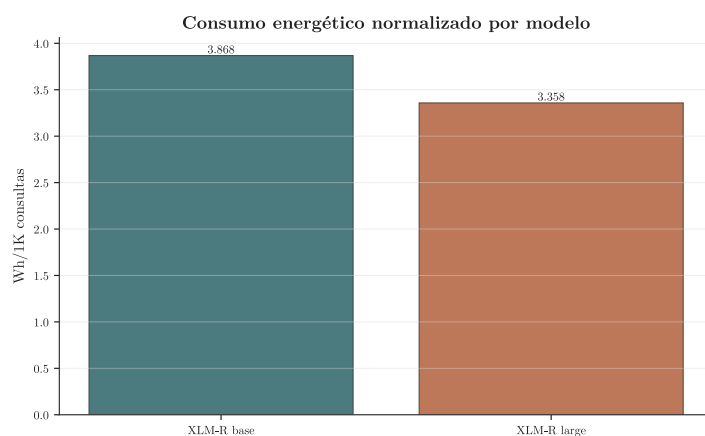


Figura D.1: Consumo energético normalizado por modelo.

*APÉNDICE D. RESULTADOS DETALLADOS DE LA EVALUACIÓN DEL
GROUNDING VISUAL MULTILINGÜE EN EL CONTEXTO DE ONEMAP*

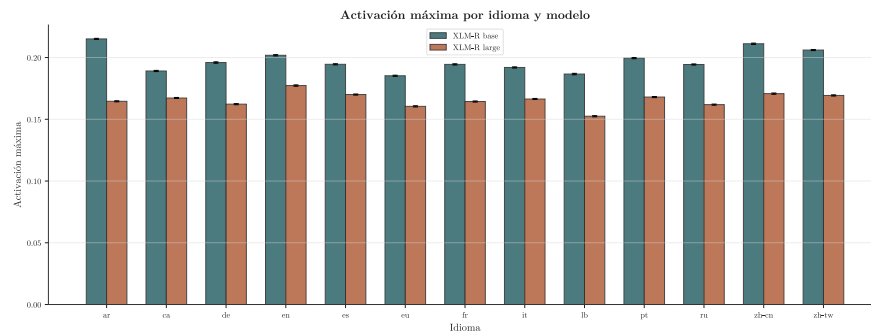
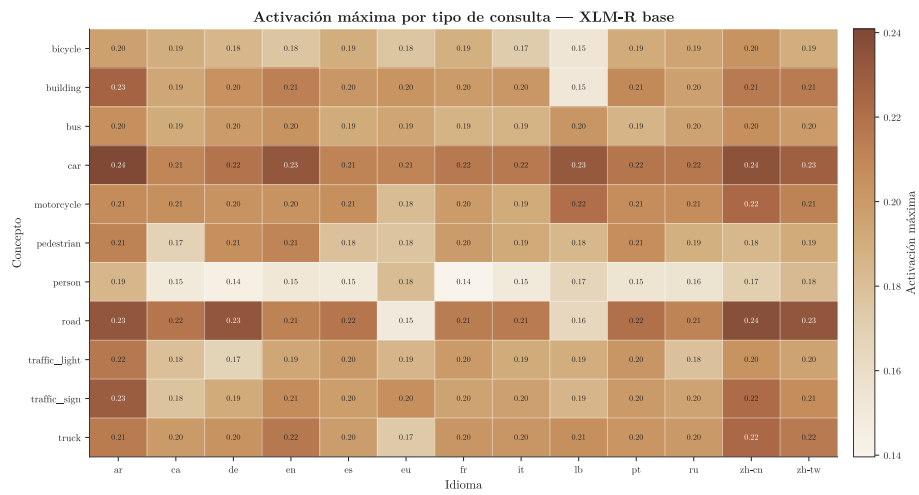
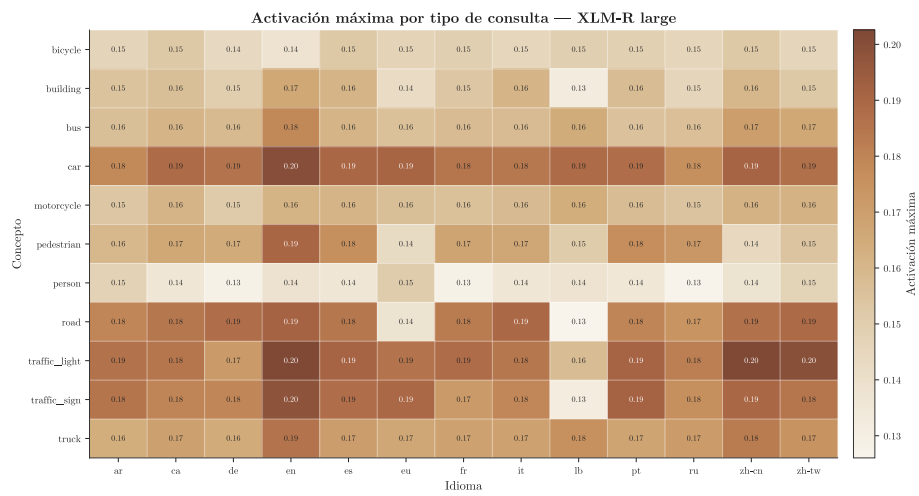


Figura D.2: Activación máxima por idioma y modelo.



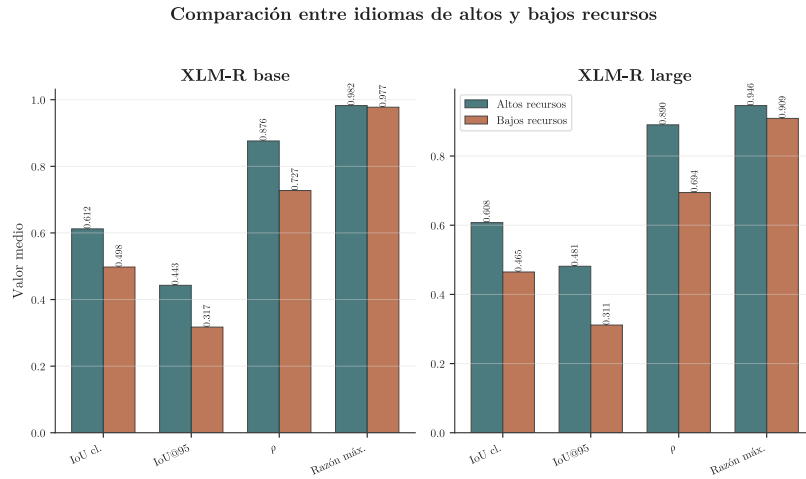
(a) Activación máxima por concepto e idioma para XLM-R base.



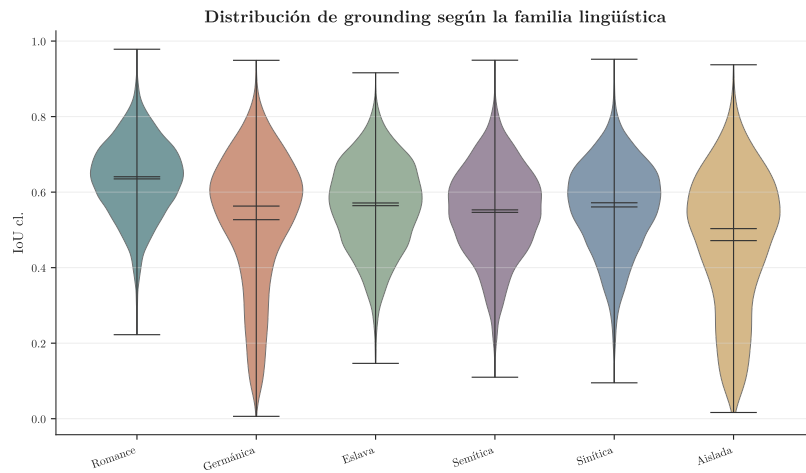
(b) Activación máxima por concepto e idioma para XLM-R large.

Figura D.3: Activación máxima por concepto visual e idioma para las variantes XLM-R base y XLM-R large.

APÉNDICE D. RESULTADOS DETALLADOS DE LA EVALUACIÓN DEL GROUNDING VISUAL MULTILINGÜE EN EL CONTEXTO DE ONEMAP



(a) Comparación entre idiomas de altos y bajos recursos.



(b) Distribución de grounding según la familia lingüística.

Figura D.4: Análisis del grounding visual según disponibilidad de recursos y familia lingüística.

Bibliografía

- [1] Bob Malone. *George Devol: A Life Devoted to Invention, and Robots - IEEE Spectrum*. IEEE Spectrum. 26 de sep. de 2011. URL: <https://spectrum.ieee.org/george-devol-a-life-devoted-to-invention-and-robots> (visitado 21-06-2026).
- [2] Shimon Y. Nof, ed. *Handbook of Industrial Robotics*. en. 1.^a ed. Wiley, feb. de 1999. ISBN: 9780471177838. DOI: [10.1002/9780470172506](https://doi.org/10.1002/9780470172506). URL: <https://onlinelibrary.wiley.com/doi/book/10.1002/9780470172506> (visitado 20-06-2026).
- [3] I. de Zarzà, J. de Curtò, Gemma Roig y Carlos T. Calafate. «LLM Multi-modal Traffic Accident Forecasting». En: *Sensors* 23.22 (2023). ISSN: 1424-8220. DOI: [10.3390/s23229225](https://doi.org/10.3390/s23229225). URL: <https://www.mdpi.com/1424-8220/23/22/9225>.
- [4] 3Blue1Brown. *Large Language Models explained briefly*. 20 de nov. de 2024. URL: <https://www.youtube.com/watch?v=LPZh9B0jkQs> (visitado 22-06-2026).
- [5] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser e Illia Polosukhin. *Attention Is All You Need*. 2 de ago. de 2023. DOI: [10.48550/arXiv.1706.03762](https://doi.org/10.48550/arXiv.1706.03762). arXiv: [1706.03762\[cs.CL\]](https://arxiv.org/abs/1706.03762). URL: <http://arxiv.org/abs/1706.03762> (visitado 08-07-2026).
- [6] IBM Technology. *What Are Vision Language Models? How AI Sees & Understands Images*. 19 de mayo de 2025. URL: https://www.youtube.com/watch?v=10D_EE96jhM (visitado 22-06-2026).
- [7] Waymo. *Self-Driving Car Technology for a Reliable Ride - Waymo Driver*. Waymo. URL: <https://waymo.com/waymo-driver/> (visitado 22-06-2026).
- [8] Zipline. *Autonomous Delivery Technology*. Zipline. URL: <https://www.zipline.com/technology> (visitado 22-06-2026).
- [9] Proteus. *Meet Proteus*. Amazon News. 11 de oct. de 2024. URL: <https://occ-tests.shorthandstories.com/meet-proteus/> (visitado 23-06-2026).

BIBLIOGRAFÍA

- [10] NASA. *Mars 2020: Perseverance Rover - NASA Science*. 5 de dic. de 2017. URL: <https://science.nasa.gov/mission/mars-2020-perseverance/> (visitado 23-06-2026).
- [11] NASA. *Perseverance Rover Components - NASA Science*. Running Time: 69 Section: Perseverance (Rover). 16 de abr. de 2024. URL: <https://science.nasa.gov/mission/mars-2020-perseverance/rover-components/> (visitado 23-06-2026).
- [12] J. de Curtò, I. de Zarzà y Carlos T. Calafate. «Semantic Scene Understanding with Large Language Models on Unmanned Aerial Vehicles». En: *Drones* 7.2 (2023). ISSN: 2504-446X. DOI: [10.3390/drones7020114](https://doi.org/10.3390/drones7020114). URL: <https://www.mdpi.com/2504-446X/7/2/114>.
- [13] I. de Zarzà, J. de Curtò y Carlos T. Calafate. «Socratic Video Understanding on Unmanned Aerial Vehicles». En: *Procedia Computer Science* 225 (2023), págs. 144-154. DOI: [10.1016/j.procs.2023.09.101](https://doi.org/10.1016/j.procs.2023.09.101).
- [14] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang y Ion Stoica. *Efficient Memory Management for Large Language Model Serving with PagedAttention*. 12 de sep. de 2023. DOI: [10.48550/arXiv.2309.06180](https://doi.org/10.48550/arXiv.2309.06180), arXiv: [2309.06180\[cs.LG\]](https://arxiv.org/abs/2309.06180). URL: <http://arxiv.org/abs/2309.06180> (visitado 08-07-2026).
- [15] Álvaro Moraga, J. de Curtò, I. de Zarzà y Carlos T. Calafate. «AI-Driven UAV and IoT Traffic Optimization: Large Language Models for Congestion and Emission Reduction in Smart Cities». En: *Drones* 9.4 (2025). ISSN: 2504-446X. DOI: [10.3390/drones9040248](https://doi.org/10.3390/drones9040248). URL: <https://www.mdpi.com/2504-446X/9/4/248>.
- [16] Tao Yang, You Li, Cheng Zhao, Dexin Yao, Guanyin Chen, Li Sun, Tomas Krajník y Zhi Yan. *3D ToF LiDAR in Mobile Robotics: A Review*. 22 de feb. de 2022. DOI: [10.48550/arXiv.2202.11025](https://doi.org/10.48550/arXiv.2202.11025), arXiv: [2202.11025\[cs.R0\]](https://arxiv.org/abs/2202.11025). URL: <http://arxiv.org/abs/2202.11025> (visitado 25-06-2026).
- [17] *What Is Autonomous Navigation?* Col. de MATLAB. 24 de jun. de 2020. URL: <https://www.youtube.com/watch?v=Fw8JQ5Q-ZwU> (visitado 01-07-2026).
- [18] Sebastian Thrun, Wolfram Burgard y Dieter Fox. *Probabilistic Robotics*. Cambridge, MA: MIT Press, 2005. ISBN: 9780262201629.
- [19] MATLAB. *Understanding the Particle Filter*. 7 de ene. de 2020. URL: https://www.youtube.com/watch?v=NrzmH_yerBU (visitado 26-06-2026).

- [20] R. E. Kalman. «A New Approach to Linear Filtering and Prediction Problems». En: *Journal of Basic Engineering* 82.1 (mar. de 1960). eprint: https://asmedigitalcollection.asme.org/fluidsengineering/article-pdf/82/1/35/5518977/35_1.pdf, págs. 35-45. ISSN: 0021-9223. DOI: [10.1115/1.3662552](https://doi.org/10.1115/1.3662552). URL: <https://doi.org/10.1115/1.3662552>.
- [21] Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, Jose Neira, Ian Reid y John J. Leonard. «Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age». En: *IEEE Transactions on Robotics* 32.6 (2016), págs. 1309-1332. DOI: [10.1109/RO.2016.2624754](https://doi.org/10.1109/RO.2016.2624754).
- [22] Mathworks. *SLAM (localización y mapeo simultáneos) – MATLAB y Simulink*. MathWorks. URL: <https://es.mathworks.com/discovery/slam.html> (visitado 26-06-2026).
- [23] Rico Sennrich, Barry Haddow y Alexandra Birch. «Neural Machine Translation of Rare Words with Subword Units». En: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*. 2016, págs. 1715-1725. DOI: [10.18653/v1/P16-1162](https://doi.org/10.18653/v1/P16-1162).
- [24] Ivan Belcic Stryker Cole. *¿Qué son los parámetros de LLM? — IBM*. 7 de ago. de 2025. URL: <https://www.ibm.com/es-es/think/topics/llm-parameters> (visitado 27-06-2026).
- [25] Dave Bergmann. *What is Fine-Tuning? — IBM*. 15 de mar. de 2024. URL: <https://www.ibm.com/think/topics/fine-tuning> (visitado 27-06-2026).
- [26] Surangika Ranathunga y Nisansa de Silva. *Some Languages are More Equal than Others: Probing Deeper into the Linguistic Disparity in the NLP World*. 20 de oct. de 2022. DOI: [10.48550/arXiv.2210.08523](https://doi.org/10.48550/arXiv.2210.08523). arXiv: [2210.08523\[cs.CL\]](https://arxiv.org/abs/2210.08523). URL: <http://arxiv.org/abs/2210.08523> (visitado 27-06-2026).
- [27] Meta-Wiki. *List of Wikipedias*. https://meta.wikimedia.org/wiki/List_of_Wikipedias. Accessed: 2026-06-27. 2026.
- [28] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer y Veselin Stoyanov. *Unsupervised Cross-lingual Representation Learning at Scale*. 8 de abr. de 2020. DOI: [10.48550/arXiv.1911.02116](https://doi.org/10.48550/arXiv.1911.02116). arXiv: [1911.02116\[cs.CL\]](https://arxiv.org/abs/1911.02116). URL: <http://arxiv.org/abs/1911.02116> (visitado 27-06-2026).
- [29] Rina Diane Caballar Stryker Cole. *What Are LLM Benchmarks? — IBM*. 29 de ene. de 2024. URL: <https://www.ibm.com/think/topics/llm-benchmarks> (visitado 27-06-2026).

- [30] Junnan Li, Dongxu Li, Silvio Savarese y Steven Hoi. «BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models». En: *Proceedings of the 40th International Conference on Machine Learning*. PMLR. 2023, págs. 19730-19742.
- [31] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger e Ilya Sutskever. *Learning Transferable Visual Models From Natural Language Supervision*. 26 de feb. de 2021. DOI: [10.48550/arXiv.2103.00020](https://doi.org/10.48550/arXiv.2103.00020). arXiv: [2103.00020\[cs.CV\]](https://arxiv.org/abs/2103.00020). URL: <http://arxiv.org/abs/2103.00020> (visitado 30-06-2026).
- [32] Finn Lukas Busch, Timon Homberger, Jesús Ortega-Peimbert, Quantao Yang y Olov Andersson. *One Map to Find Them All: Real-time Open-Vocabulary Mapping for Zero-shot Multi-Object Navigation*. 3 de mar. de 2025. DOI: [10.48550/arXiv.2409.11764](https://doi.org/10.48550/arXiv.2409.11764). arXiv: [2409.11764\[cs.R0\]](https://arxiv.org/abs/2409.11764). URL: <http://arxiv.org/abs/2409.11764> (visitado 29-06-2026).
- [33] Santhosh K. Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alex Clegg, John Turner, Eric Undersander, Wojciech Galuba, Andrew Westbury, Angel X. Chang, Manolis Savva, Yili Zhao y Dhruv Batra. *Habitat-Matterport 3D Dataset (HM3D): 1000 Large-scale 3D Environments for Embodied AI*. 2021. arXiv: [2109.08238 \[cs.CV\]](https://arxiv.org/abs/2109.08238). URL: <https://arxiv.org/abs/2109.08238>.
- [34] Nathan Silberman, Derek Hoiem, Pushmeet Kohli y Rob Fergus. «Indoor Segmentation and Support Inference from RGBD Images». En: *Computer Vision – ECCV 2012*. Ed. por Andrew Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato y Cordelia Schmid. Red. de David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell, Moni Naor, Oscar Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Madhu Sudan, Demetri Terzopoulos, Doug Tygar, Moshe Y. Vardi y Gerhard Weikum. Vol. 7576. Series Title: Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, págs. 746-760. ISBN: 978-3-642-33714-7 978-3-642-33715-4. DOI: [10.1007/978-3-642-33715-4_54](https://doi.org/10.1007/978-3-642-33715-4_54). URL: http://link.springer.com/10.1007/978-3-642-33715-4_54 (visitado 01-07-2026).
- [35] Xueyan Zou, Jianwei Yang, Hao Zhang, Feng Li, Linjie Li, Jianfeng Wang, Lijuan Wang, Jianfeng Gao y Yong Jae Lee. *Segment Everything Everywhere All at Once*. 2023. arXiv: [2304.06718 \[cs.CV\]](https://arxiv.org/abs/2304.06718). URL: <https://arxiv.org/abs/2304.06718>.

-
- [36] Bin Xie, Jiale Cao, Jin Xie, Fahad Shahbaz Khan y Yanwei Pang. *SED: A Simple Encoder-Decoder for Open-Vocabulary Semantic Segmentation*. 2024. arXiv: [2311.15537 \[cs.CV\]](https://arxiv.org/abs/2311.15537). URL: <https://arxiv.org/abs/2311.15537>.
 - [37] Peter E. Hart, Nils J. Nilsson y Bertram Raphael. «A Formal Basis for the Heuristic Determination of Minimum Cost Paths». En: *IEEE Transactions on Systems Science and Cybernetics* 4.2 (1968), págs. 100-107. DOI: [10.1109/TSSC.1968.300136](https://doi.org/10.1109/TSSC.1968.300136).
 - [38] International Energy Agency. *Key Questions on Energy and AI*. Licence: CC BY 4.0. Paris: International Energy Agency, 16 de abr. de 2026. URL: <https://www.iea.org/reports/key-questions-on-energy-and-ai> (visitado 02-07-2026).
 - [39] Sasha Luccioni, Yacine Jernite, Régis Pierrard, Ilyas Moutawwakil, Margaret Mitchell, Boris Gamazaychikov, Jin Qu, Bin Bi, Michael Weimann, Hannah Downey et al. *AI Energy Score: Standardized Energy Efficiency Ratings for AI Models*. 2025. URL: <https://huggingface.github.io/AIEnergyScore/> (visitado 01-02-2026).
 - [40] J. De Curtò e I. De Zarzà. «Metamorphic Testing for Semantic Invariance in Large Language Models». En: *IEEE Access* 13 (2025), págs. 214772-214791. ISSN: 2169-3536. DOI: [10.1109/ACCESS.2025.3646270](https://doi.org/10.1109/ACCESS.2025.3646270). URL: <https://ieeexplore.ieee.org/document/11305018/> (visitado 02-07-2026).
 - [41] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan y Trevor Darrell. *BDD100K: A Diverse Driving Dataset for Heterogeneous Multitask Learning*. 8 de abr. de 2020. DOI: [10.48550/arXiv.1805.04687](https://doi.org/10.48550/arXiv.1805.04687). arXiv: [1805.04687\[cs.CV\]](https://arxiv.org/abs/1805.04687). URL: <http://arxiv.org/abs/1805.04687> (visitado 03-07-2026).
 - [42] I. de Zarzà, Mauro Liz, J. de Curtò y Carlos T. Calafate. «Energy-Aware Multilingual Evaluation of Large Language Models». En: *Electronics* 15.7 (2026). ISSN: 2079-9292. DOI: [10.3390/electronics15071395](https://doi.org/10.3390/electronics15071395). URL: <https://www.mdpi.com/2079-9292/15/7/1395>.
 - [43] J. de Curtò, Mauro Liz, I. de Zarzà y Carlos T. Calafate. «Energy-Aware Multilingual Vision–Language Models for Drone Smart Sensing». En: *Drones* 10.5 (2026). ISSN: 2504-446X. DOI: [10.3390/drones10050361](https://doi.org/10.3390/drones10050361). URL: <https://www.mdpi.com/2504-446X/10/5/361>.
 - [44] J. de Curtò, Mauro Liz, I. de Zarzà y Carlos T. Calafate. «Multilingual VLM Energy Signatures for Edge Deployment». En: *Proceedings of the 2026 International Conference on Artificial Intelligence Models and Systems (IEEE iAIMS2026)*. Accepted as Full Paper. Barcelona, Spain, nov. de 2026.

BIBLIOGRAFÍA

- [45] J. de Curtò, Mauro Liz e I. de Zarzà. *Language-Conditioned Visual Grounding with CLIP Multilingual*. 9 de mayo de 2026. DOI: [10.48550/arXiv.2605.09060](https://doi.org/10.48550/arXiv.2605.09060). arXiv: [2605.09060](https://arxiv.org/abs/2605.09060) [cs.CL]. URL: <http://arxiv.org/abs/2605.09060> (visitado 07-07-2026).
- [46] Naciones Unidas. *Objetivos de Desarrollo Sostenible*. <https://www.un.org/sustainabledevelopment/es/sustainable-development-goals/>. Accedido: 8 de julio de 2026. 2026.