LM-NAV

Robotic Navigation with Large Pre-trained Models of Language, Vision and Action

Arroyo Joaquin, Bolzan Francisco y Montoro Emiliano

Introducción

El proyecto seleccionado fue desarrollado por investigadores de:

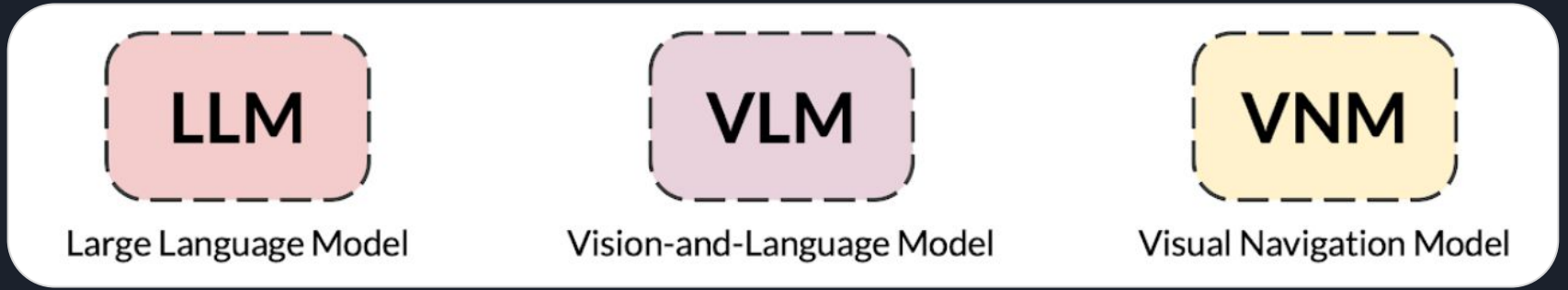
- ❑ *Universidad de Berkeley*
- ❑ *Universidad de Varsovia*
- ❑ *Google Research*

Introducción: Planteo del problema e idea principal

Dada una instrucción en lenguaje natural, para navegar un ambiente del mundo real, ¿cómo puede un robot seguirlas solamente a partir de observaciones visuales egocéntricas?

Para resolver lo anterior, la idea principal del proyecto es utilizar modelos pre entrenados de procesamiento de imágenes y lenguajes, para proveer una interfaz textual a un modelo de navegación.

Los modelos utilizados son:

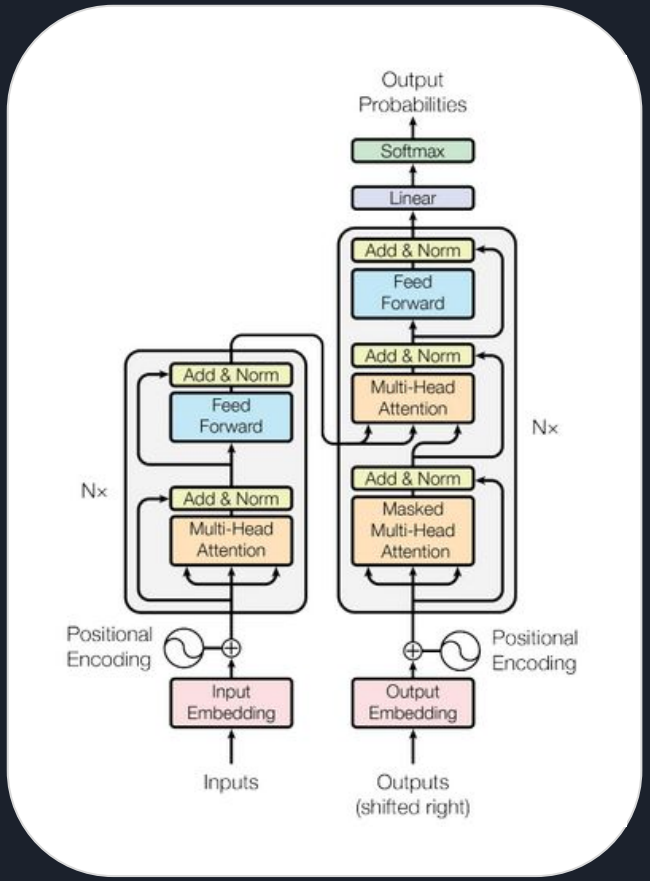


LLM: Modelo de Lenguaje

Los modelos de lenguaje se utilizan para el procesamiento del lenguaje natural (NLP), entrenados en masivos datasets de texto.

Dentro del conjunto de estos modelos, los más avanzados y efectivos están basados en la arquitectura **Transformer**.

La idea detrás de esta arquitectura es que, en lugar de procesar palabras de forma secuencial, se utiliza la **atención múltiple** para enfocarse en diferentes partes de la secuencia de entrada en paralelo.



LLM: Modelo de Lenguaje - Transformers vs otras soluciones

En comparación con otras soluciones, como las redes neuronales recurrentes (RNN) y las redes neuronales convolucionales (CNN), los Transformers ofrecen varias ventajas clave.

- Son capaces de procesar la información de entrada en paralelo.
- Pueden capturar dependencias de largo alcance entre los elementos de la secuencia, lo que les permite modelar relaciones más complejas.
- Son más fáciles de entrenar debido a su estructura modular.

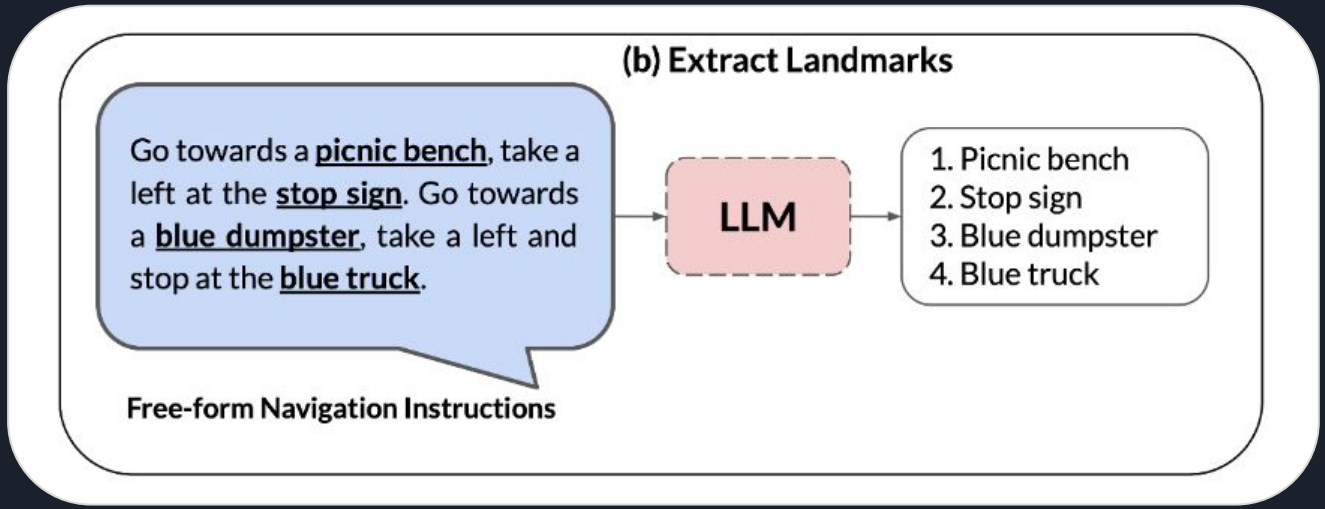
LLM: Modelo de Lenguaje - Transformers vs otras soluciones

En la siguiente tabla podemos observar la comparación entre modelos que utilizan RNN o CNN contra modelos que utilizan Transformers. La comparación se da sobre el problema de traducción automática.

Model	BLEU		Training Cost (FLOPs)	
	EN-DE	EN-FR	EN-DE	EN-FR
ByteNet [18]	23.75			
Deep-Att + PosUnk [39]		39.2		$1.0 \cdot 10^{20}$
GNMT + RL [38]	24.6	39.92	$2.3 \cdot 10^{19}$	$1.4 \cdot 10^{20}$
ConvS2S [9]	25.16	40.46	$9.6 \cdot 10^{18}$	$1.5 \cdot 10^{20}$
MoE [32]	26.03	40.56	$2.0 \cdot 10^{19}$	$1.2 \cdot 10^{20}$
Deep-Att + PosUnk Ensemble [39]		40.4		$8.0 \cdot 10^{20}$
GNMT + RL Ensemble [38]	26.30	41.16	$1.8 \cdot 10^{20}$	$1.1 \cdot 10^{21}$
ConvS2S Ensemble [9]	26.36	**41.29**	$7.7 \cdot 10^{19}$	$1.2 \cdot 10^{21}$
Transformer (base model)	27.3	38.1	**$3.3 \cdot 10^{18}$**	
Transformer (big)	**28.4**	**41.8**	$2.3 \cdot 10^{19}$	

LLM: Modelo de Lenguaje - Función

El modelo de lenguaje utilizado por LM-Nav es **GPT-3**, desarrollado por OpenAI, y su función es parsear instrucciones en lenguaje natural, a una secuencia de puntos de referencia que pueden servir como sub-metas intermediarias para la navegación.



VLM: Modelo Lenguaje-Visión

Los VLM son modelos multimodales que se centran en la relación entre imágenes y lenguaje natural. De estos, el más prominente en la actualidad es **CLIP (Contrastive Language-Image Pre-training)**, que es también el VLM usado por LM-Nav.

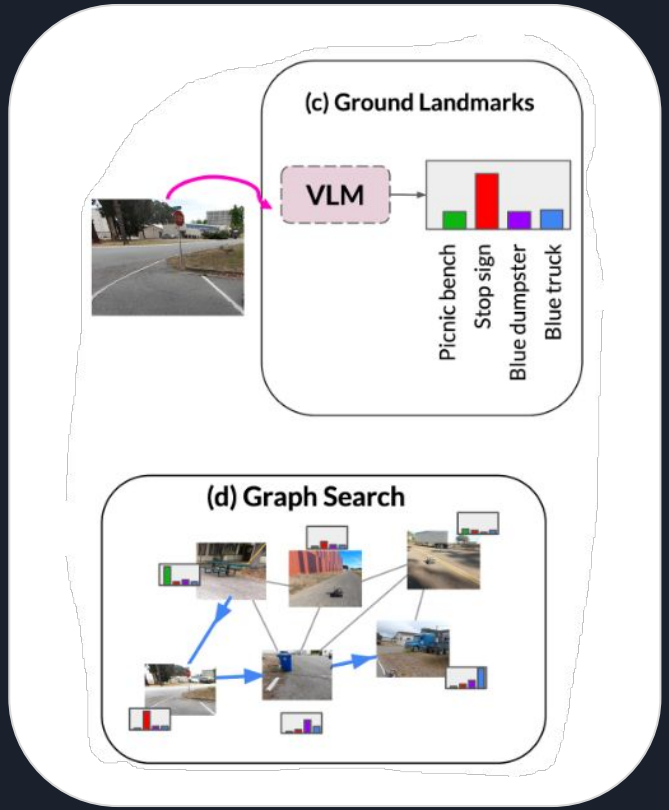
Ventajas de CLIP:

- ❏ Pre-training masivo
- ❏ No requiere labelling del dataset
- ❏ LLM con Transformer architecture
- ❏ Contrastive > Predictive
- ❏ Robusto y zero-shot
- ❏ Downstream fine-tuning
- ❏ Razonamiento real entre lenguaje e imágen

VLM: Modelo Lenguaje-Visión

CLIP en LM-Nav:

Una vez generado el grafo de conectividad de VNM e interpretada la entrada por LLM, el VLM (CLIP) será el encargado de asociar los landmarks interpretados con sus posibles nodos en el grado de conectividad.



VNM: Modelo de Navegación Visual

Los modelos VNM aprenden el comportamiento de navegación a partir de observaciones visuales, asociando imágenes y acciones a través del tiempo.
Se utiliza un modelo ViNG VNM condicionado a objetivos que predice las distancias temporales entre pares de imágenes y acciones correspondientes a ejecutar. El VNM tiene 2 propósitos:

- Dado un conjunto de observaciones en el entorno se crea un grafo topológico representado un mapa mental.
- Dado una secuencia de subobjetivos conectados a un nodo objetivo el VNM puede guiar al robot.

VNM: Modelo de Navegación Visual

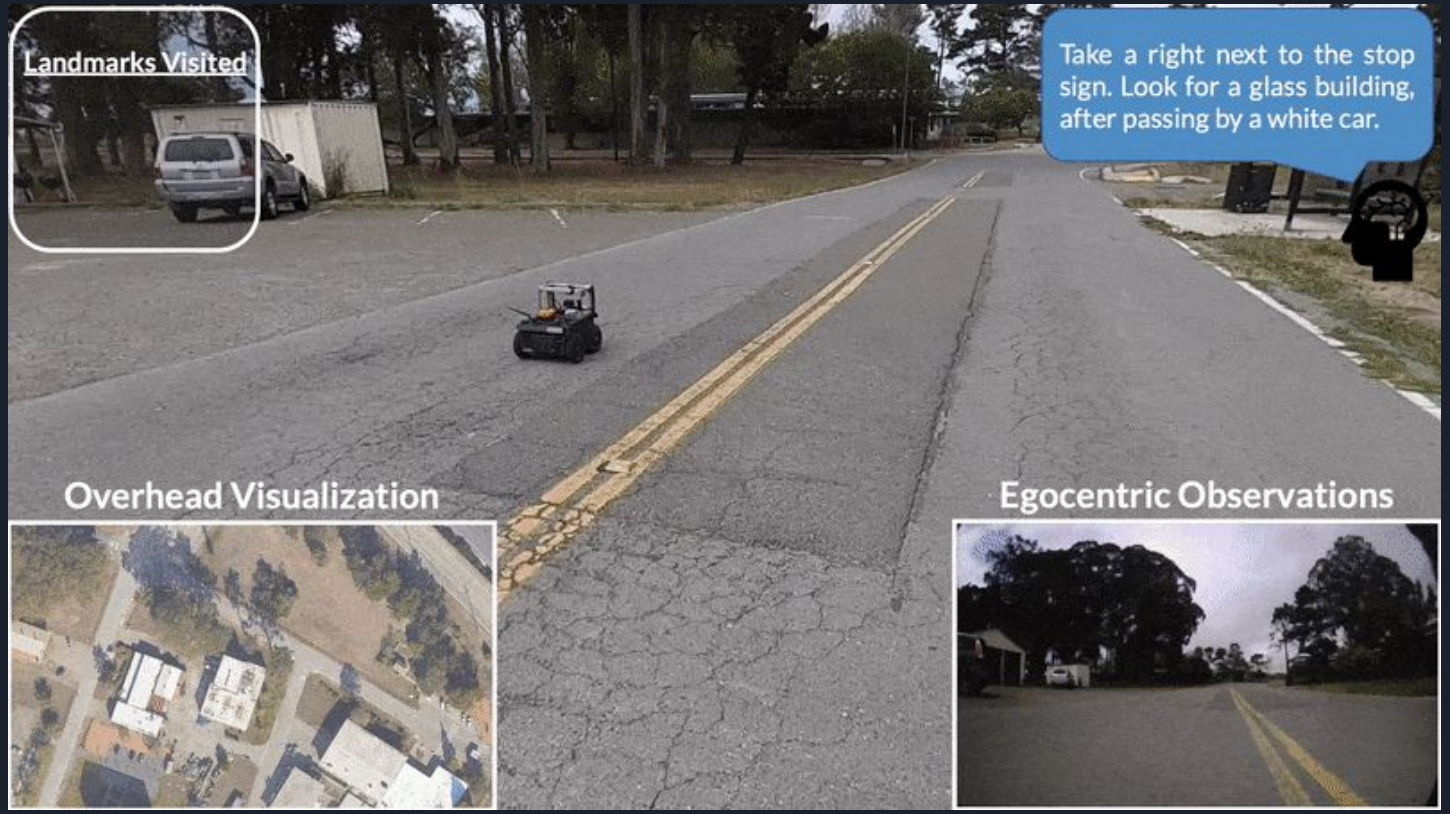
Para la generación del grafo se utiliza la proximidad espacial del GPS, proximidad temporal y estimación de distancias aprendidas para deducir la conectividad de las aristas. Si las estimaciones VNM de las imágenes en dos nodos están cercanas se conectan los nodos correspondientes dando lugar a la conexión de nodos que se generaron en diferentes trayectorias.

Algorithm 2: Graph Building

Input: Nodes $n_i, n_j \in \mathcal{G}$ containing robot observations; **VNM** distance function f_d; hyperparameters $\{\tau, \epsilon, \eta\}$
Output: Boolean e_{ij} corresponding to the existence of edge in $\mathcal{G}$, and its weight
learned distance $D_{ij} = f_d(n_i[\text{'image'}], n_j[\text{'image'}])$
timestamp distance $T_{ij} = |n_i[\text{'timestamp'}] - n_j[\text{'timestamp'}]|$
spatial distance $X_{ij} = \|n_i[\text{'GPS'}] - n_j[\text{'GPS'}]\|$
if ($T_{ij} < \epsilon$) **then** return $\{\textit{True}, D_{ij}\}$
else if ($D_{ij} < \tau$) AND ($X_{ij} < \eta$) **then** return $\{\textit{True}, D_{ij}\}$
else return *False*

Algorithm 1: Graph Search

Input: Landmarks $(\ell_1, \ell_2, \ldots, \ell_n)$.
Input: Graph $\mathcal{G}(V, E)$.
Input: Starting node S.
$\forall_{\substack{i=0,\ldots,n \\ v \in V}} \; Q[i, v] = -\infty$
$Q[0, S] = 0$
Dijkstra_algorithm$(\mathcal{G}, Q[0, *])$
for i in $1, 2, \ldots, n$ **do**
  $\forall v \in V Q[i, v] = Q[i-1, v] + \text{CLIP}(v, \ell_i)$
  Dijkstra_algorithm$(\mathcal{G}, Q[i, *])$
end for
destination $= \arg\max(Q[n, *])$
return backtrack(destination, $Q[n, *]$)


Landmarks Visited
Take a right next to the stop sign. Look for a glass building, after passing by a white car.
Overhead Visualization
Egocentric Observations

Conclusión

- A pesar de ser un modelo de navegación superador a sus predecesores, LM-Nav aún presenta limitaciones considerables.
- La combinación de modelos altamente especializados puede brindar buenos resultados en aplicaciones nuevas.
- Existen muchas formas correctas de atacar un problema, no todas serán óptimas (como fue visto en los distintos modelos y sus predecesores).
- Robotic navigation is complicated.

Referencias

- B. Itcher D. Shah, B. Osinski and S. Levine. Lm-nav: Robotic navigation with large pre-trained models of language, vision, and action. 2022
- N. Ryder M. Subbiah J. D. Kaplan P. Dhariwal T. Brown, B. Mann and A. Amodei. Language models are few-shot learners. in advances in neural information processing systems. 2020
- N. Parmar J. Uszkoreit L. Jones A. N. Gomez L. Kaiser A. Vaswani, N. Shazeer and I. Polosukhin. Attention is all you need. 2017
- Rewon Child David Luan Dario Amodei Alec Radford, Jeffrey Wu and Ilya Sutskever. Language models are unsupervised multitask learners. 2019
- C. Hallacy A. Ramesh G. Goh S. Agarwal G. Sastry A. Askell P. Mishkin J. Clark et al. A. Radford, J. W. Kim. Learning transferable visual models from natural language supervision. 2021