HMU: Hippocampus Memory Unit
Un Módulo de Memoria Semántica para Transformers

Esteban Sánchez Gámez

Investigador Independiente
estebansanchezgamez@gmail.com
https://github.com/Sauvageduck24

Resumen El *Hippocampus Memory Unit* (HMU) es un módulo bioinspirado que amplía las capacidades de los modelos Transformer mediante la introducción de un espacio latente semántico, compacto y entrenable. Tomando como referencia la función del hipocampo humano—encargado de la consolidación y evocación de recuerdos—, HMU actúa como una memoria semántica adaptativa capaz de: (i) condensar información relevante, (ii) filtrar ruido contextual y (iii) enriquecer la generación creativa. Con una sobrecarga aproximada del 1 % de parámetros, el modelo resultante incrementa la retención de memoria a largo plazo, la coherencia narrativa y la precisión en tareas de clasificación y generación, *sin necesidad de reentrenar* el encoder ni el decoder. Basta un ligero *fine-tuning* del bloque HMU y su VAE asociado.

1. Introducción

Los modelos Transformer han establecido un nuevo estándar en tareas de procesamiento del lenguaje natural (NLP) [1]. Sin embargo, a pesar de su éxito, presentan limitaciones persistentes en tres áreas clave: (i) dificultad para mantener contexto en secuencias extensas, (ii) escasa diversidad y creatividad en generación, y (iii) ausencia de mecanismos explícitos de memoria semántica o compresión contextual.

Estas carencias limitan su desempeño en tareas que requieren razonamiento a largo plazo, generación narrativa o manejo eficiente de información redundante. Inspirándonos en la neurociencia —donde el hipocampo no almacena la memoria a largo plazo de forma directa, sino que decide *cuándo* y *qué* recuerdos reactivar según el contexto [4]— proponemos el HMU, un módulo liviano y *plug-and-play* que introduce una forma de memoria latente semántica y controlada.

Este módulo, basado en un VAE a nivel de token y una compuerta adaptativa entrenable, puede insertarse sin modificar el encoder ni el decoder, permitiendo extender modelos existentes con capacidades de memoria, compresión e imaginación sin necesidad de reentrenamiento estructural.

Contribuciones. Este trabajo presenta:

1. el diseño del módulo HMU, inspirado en el papel selectivo del hipocampo,
2. su integración mínima y no invasiva en arquitecturas Transformer,
3. una evaluación empírica en tareas de clasificación, generación narrativa y benchmarks sintéticos de memoria, y
4. un análisis costo-beneficio que demuestra que un aumento de apenas $+1$ % en parámetros puede traducirse en mejoras de hasta $\times 20$ en retención de largo plazo.

2. Arquitectura del HMU

2.1. Descripción general

Módulo HMU (Hippocampus Memory Unit). El HMU se ubica como un *filtro semántico* entre el encoder y el decoder: recibe la salida contextual del encoder, $\mathbf{v} \in \mathbb{R}^{B\times L\times d_{\text{model}}}$, y entrega al decoder una versión enriquecida $\mathbf{v}^*$.

1. Compresión latente con VAE. Para cada token se calcula un *Variational Autoencoder*:

$$(\boldsymbol{\mu}, \log \boldsymbol{\sigma}^2) = \text{VAE-enc}\big(\mathbf{v}\big), \qquad \mathbf{z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}),$$

y se reconstruye la versión comprimida $\tilde{\mathbf{v}} = \text{VAE-dec}(\mathbf{z})$.

2. Normalización. Se normalizan ambos caminos para evitar desbalances de escala:

$$\hat{\mathbf{v}} = \text{LayerNorm}(\mathbf{v}), \qquad \hat{\mathbf{v}}_{\text{lat}} = \text{LayerNorm}(\tilde{\mathbf{v}}).$$

3. Gating adaptativo. Se concatena la información literal y latente, y un *MLP sigmoide* produce la compuerta

$$\mathbf{g} = \sigma\Big(\text{MLP}\big[\hat{\mathbf{v}} \,\|\, \hat{\mathbf{v}}_{\text{lat}}\big]\Big), \qquad \mathbf{g} \in [0,1]^{B \times L \times d_{\text{model}}}.$$

4. Fusión semántica. La representación final se calcula de forma elemento a elemento:

$$\mathbf{v}^* = \mathbf{g} \odot \hat{\mathbf{v}} + (1 - \mathbf{g}) \odot \hat{\mathbf{v}}_{\text{lat}},$$

seguida de una proyección lineal, activación GELU y LayerNorm para mantener la dimensionalidad d_{model}. Esta etapa constituye la salida final del módulo HMU y asegura que la representación esté normalizada y preparada para ser consumida por el *decoder*.

Propiedades resultantes.

- *Imaginación controlada*: el espacio latente continuo del VAE habilita la generación de variaciones semánticas coherentes, ajustables mediante parámetros como la temperatura o el tipo de muestreo.
- *Memoria selectiva*: el mecanismo de compuerta aprendido simula la función del hipocampo, decidiendo de forma contextual y dinámica *cuánta* información comprimida debe recuperarse y *cuándo* debe integrarse.
- *Integración eficiente*: el módulo HMU introduce una sobrecarga de apenas $\sim$1 % en el número total de parámetros, y puede acoplarse a modelos Transformer existentes sin alterar su estructura interna.

La Figura 1 ilustra el flujo completo.

2.2. Implementación de referencia con Pytorch

```python
class HMU(nn.Module):
    def __init__(self, d_model: int, latent_dim: int):
        super().__init__()
        self.vae = VAE(d_model, latent_dim)

        # Normaliza el input original y el latente reconstruido
        self.norm_input = nn.LayerNorm(d_model)
        self.norm_latent = nn.LayerNorm(d_model)

        # Gating para decidir cu nto usar de cada representaci n
        self.gate_mlp = nn.Sequential(
            nn.Linear(d_model * 2, d_model),
            nn.Sigmoid()
        )

        # Proyecci n final tras fusi n
        self.fusion_proj = nn.Sequential(
            nn.Linear(d_model, d_model),
            nn.GELU(),
            nn.LayerNorm(d_model)  # Prepara para decoder
```

```
        )

    def forward(self, v: torch.Tensor) -> torch.Tensor:
        """
        Args:
            v (Tensor): Salida del encoder (batch, seq_len, d_model)
        Returns:
            Tensor: Representaci n enriquecida para el decoder
        """
        # VAE reconstruction
        v_latent = self.vae(v)  # (batch, seq_len, d_model)

        # Normalizar ambas representaciones
        v_norm = self.norm_input(v)
        v_latent_norm = self.norm_latent(v_latent)

        # Gating: decide cu nto confiar en cada parte
        gate_input = torch.cat([v_norm, v_latent_norm], dim=-1)  # (batch, seq_len
            , 2*d_model)
        gate = self.gate_mlp(gate_input)  # (batch, seq_len, d_model), valores
            entre 0 y 1

        # Fusi n adaptativa
        fused = gate * v_norm + (1 - gate) * v_latent_norm

        # Proyecci n final
        output = self.fusion_proj(fused)  # (batch, seq_len, d_model)
        return output
```

2.3. Flujo matemático del módulo HMU

El módulo HMU toma como entrada una secuencia de representaciones $\mathbf{h} \in \mathbb{R}^{B\times L\times d_{\text{model}}}$, y genera una versión enriquecida $\mathbf{h}^*$ lista para su uso en el decoder. El proceso consta de los siguientes pasos:

$$\tilde{\mathbf{h}} = \text{VAE}(\mathbf{h}) \qquad \text{(Compresión latente)} \tag{1}$$

$$\hat{\mathbf{h}} = \text{LayerNorm}(\mathbf{h}) \qquad \text{(Normalización original)} \tag{2}$$

$$\hat{\tilde{\mathbf{h}}} = \text{LayerNorm}(\tilde{\mathbf{h}}) \qquad \text{(Normalización latente)} \tag{3}$$

$$\mathbf{g} = \sigma\left(W_g\left[\hat{\mathbf{h}}\,;\,\hat{\tilde{\mathbf{h}}}\right]\right) \qquad \text{(Compuerta adaptativa)} \tag{4}$$

$$\mathbf{h}' = \mathbf{g}\odot\hat{\mathbf{h}} + (1-\mathbf{g})\odot\hat{\tilde{\mathbf{h}}} \qquad \text{(Fusión semántica)} \tag{5}$$

$$\mathbf{h}^* = \text{LayerNorm}\left(\phi\left(W_f\mathbf{h}'\right)\right) \qquad \text{(Proyección final)} \tag{6}$$

donde:
[topsep=2pt,itemsep=2pt,leftmargin=14pt]VAE: Autoencoder variacional a nivel de token. $[\,;\,]$: Concatenación a lo largo de la dimensión de características. σ: Función sigmoide aplicada elemento a elemento. ϕ: Función de activación GELU. $W_g \in \mathbb{R}^{2d\times d}, W_f \in \mathbb{R}^{d\times d}$: Proyecciones aprendidas.

3. Metodología Experimental

Se evaluó HMU en cinco conjuntos: GLUE, AG-News, CommonGen, ROCStories y un benchmark sintético de memoria. Se mantuvieron hiperparámetros idénticos al baseline salvo la inserción del HMU (véase Tabla 1).

Cuadro 1: Datasets utilizados.

Categoría	Dataset	Propósito
Clasificación	GLUE, AG-News	Comprensión y clasificación
Generación	CommonGen	Generación basada en conceptos
Narrativa	ROCStories	Coherencia narrativa
Memoria	Sintético	Retención@200

4. Resultados

Esta sección presenta una comparación cuantitativa entre un Transformer estándar y su variante con la unidad HMU integrada. La evaluación abarca tareas de clasificación, generación de texto y benchmarks sintéticos centrados en la memoria de largo plazo. Los modelos se han entrenado con los mismos hiperparámetros y configuraciones para garantizar una comparación justa, modificando únicamente la arquitectura interna mediante la inclusión del módulo HMU.

4.1. Comparación cuantitativa por tarea

GLUE (tamaño reducido). Sobre un subconjunto truncado de GLUE ($\leq$ 20,000 ejemplos por tarea), el modelo con HMU alcanza una mejora media de **+0.6 puntos porcentuales** en *accuracy*. Destacan avances notables en tareas sintácticas como CoLA (+7.5 pp) y parafraseo (MRPC, +2.0 pp). Algunas tareas con fuerte carga semántica, como SST-2 y MNLI, muestran ligeras pérdidas atribuibles a sobreajuste o falta de convergencia.

Benchmark de memoria sintética. La retención a largo plazo se evalúa mediante la métrica RETENTION@200, basada en la similitud coseno entre la entrada y la salida tras 200 pasos de propagación. La incorporación del HMU proporciona una mejora de **20 veces** en esta métrica, mostrando una recuperación mucho más efectiva del contexto original. También se observa una mejora en la *coherencia secuencial* y la precisión de recuperación.

CommonGen. En esta tarea de generación de texto desde conceptos, el HMU aporta una mejora estable de +0.2–0.3 puntos de *accuracy*. Esto sugiere que la fusión adaptativa mejora la capacidad composicional del modelo al integrar mejor los significados latentes de los conceptos.

AG-News. Aunque la tarea es más simple desde el punto de vista semántico, el modelo HMU presenta mejoras marginales pero consistentes en precisión y pérdida de validación.

ROCStories. En la tarea de selección de finales de historias, ambos modelos alcanzan una precisión perfecta (1.0). Sin embargo, el modelo con HMU presenta una **pérdida de entropía cruzada menor**, lo que indica una asignación de mayor probabilidad al final correcto, evidencia de una modelización narrativa más robusta.

4.2. Eficiencia de parámetros

La unidad HMU introduce aproximadamente un **1 % adicional** de parámetros respecto al Transformer base y un aumento estimado del **7 % en el tiempo de entrenamiento**. A pesar de este pequeño coste computacional, los beneficios obtenidos en tareas semánticas complejas y de memoria a largo plazo son notables.

4.3. Interpretación de los resultados

- **Capacidad semántica**: las tareas con alta exigencia lingüística (e.g., CoLA, MRPC, CommonGen) se benefician especialmente del espacio latente del VAE y el gating adaptativo.
- **Memoria de largo plazo**: los resultados en retención y recuperación demuestran que el HMU actúa como una memoria dinámica auxiliar, similar al rol del hipocampo en humanos.
- **Estabilidad narrativa**: la menor entropía cruzada en ROCStories sugiere que el modelo genera respuestas más seguras y coherentes en tareas cerradas.

Cuadro 2: Comparación de resultados entre Transformer estándar y HMU en múltiples tareas.

Tarea / Dataset	**Métrica**	**Transformer**	**HMU**	**Absoluta**
GLUE (media)	Accuracy	0.5800	**0.5858**	+0.0058
MRPC	Accuracy	0.669	**0.689**	+0.0200
RTE	Accuracy	0.480	**0.484**	+0.0040
CoLA	Matthews Corr.	0.616	**0.691**	+0.0750
QNLI	Accuracy	0.545	**0.546**	+0.0010
SST-2	Accuracy	**0.763**	0.744	–0.0190
MNLI-m	Accuracy	**0.407**	0.361	–0.0460
Memoria sintética	Retention@200	5e-6	**1e-4**	×20
	Precisión recup.	0.9689	**0.9733**	+0.0044
	Coherencia (↓)	16.27	**15.18**	–1.09
AG-News	Accuracy	0.86225	**0.86242**	+0.00017
CommonGen	Accuracy	0.78398	**0.78613**	+0.00215
ROCStories	Accuracy	1.0000	1.0000	0
	Pérdida (cross-entropy)	0.02129	**0.01981**	–0.00148

5. Resumen de resultados

- **Retención a largo plazo:** el HMU logra una mejora de **×20** en la métrica RETENTION@200.
- **Precisión media (GLUE):** incremento de **+0.6 puntos porcentuales** en clasificación con el mismo número de épocas.
- **Narrativa:** pérdida de entropía cruzada menor en ROCStories, indicando mayor seguridad semántica.
- **Eficiencia:** sólo $\sim 1\,\%$ de parámetros adicionales y $\sim 7\,\%$ de tiempo de entrenamiento extra.

En conjunto, los resultados respaldan la hipótesis de que una memoria semántica latente, integrada de forma no invasiva mediante el HMU, puede mejorar significativamente el razonamiento contextual y la generación de texto, con una sobrecarga mínima en términos de parámetros o cómputo.

6. Aplicaciones Potenciales

El módulo HMU ofrece una solución eficiente, modular y fácilmente integrable para extender las capacidades de los modelos Transformer en tareas que requieren compresión semántica, razonamiento contextual o generación coherente. Gracias a su naturaleza *plug-and-play* —es decir, puede insertarse entre el encoder y el decoder sin necesidad de modificar ni reentrenar sus componentes originales—, se convierte en una herramienta extremadamente versátil en los siguientes contextos:

1. **Generación creativa de historias y guiones.** La fusión adaptativa entre contexto explícito y representación latente permite generar texto narrativo con mayor coherencia estructural y diversidad temática, manteniendo una narrativa fluida incluso en secuencias largas.
2. **Compresión semántica para chatbots con contexto extenso.** El HMU actúa como un filtro semántico que abstrae y conserva información relevante, permitiendo a los asistentes conversacionales mantener coherencia en diálogos prolongados sin incurrir en costos de memoria excesivos.
3. **Razonamiento multi-turno en QA y tareas de diálogo.** La capacidad del módulo para seleccionar dinámicamente qué información comprimida reincorporar (simulando el comportamiento del hipocampo) mejora la consistencia y profundidad de respuestas en entornos de preguntas-respuestas iterativas.
4. **Transfer learning eficiente y no intrusivo.** Dado que el HMU no altera las capas del encoder ni del decoder, puede añadirse a modelos preentrenados sin necesidad de reentrenamiento completo. Esto lo convierte en una solución ideal para adaptaciones ligeras que requieren mejorar razonamiento o retención con un coste computacional mínimo.
5. **Extensión modular para tareas multitarea.** En contextos donde se comparte un backbone encoder para múltiples tareas, el HMU puede adaptarse a cada una insertando variantes especializadas sin duplicar el modelo base, facilitando una arquitectura más escalable.

7. Conclusiones

El HMU demuestra que una única unidad inspirada funcionalmente en el hipocampo humano, insertada tras el encoder de un Transformer, puede generar mejoras desproporcionadas respecto a su coste computacional. Con una sobrecarga de apenas un $\sim 1\,\%$ en parámetros, se logran beneficios sustanciales en tareas de retención, generación narrativa y comprensión contextual.

Este trabajo abre una nueva línea de exploración en arquitecturas con *memoria semántica latente*, donde la capacidad de recordar, abstraer y recuperar información relevante se modela de forma explícita mediante mecanismos compactos y diferenciables. La fusión adaptativa entre contexto actual y representación comprimida permite no solo mejorar la consistencia textual, sino también potenciar la creatividad bajo control.

Gracias a su naturaleza modular y *plug-and-play*, el HMU es fácilmente integrable en arquitecturas preentrenadas sin necesidad de reentrenar el encoder ni el decoder, lo que lo hace especialmente adecuado para entornos de producción, sistemas bajo restricciones de recursos, o escenarios de *transfer learning*.

En futuras líneas de trabajo, el modelo puede extenderse hacia:

- **Modelos multimodales**, donde el espacio latente actúe como interfaz común entre lenguaje, visión o audio.
- **Razonamiento composicional y lógico**, aprovechando la capacidad del HMU para capturar dependencias de largo alcance y seleccionar representaciones relevantes.
- **Aprendizaje continuo y dinámico**, integrando el módulo como mecanismo de consolidación de memoria entre episodios.

En conjunto, el HMU representa un paso hacia Transformers más interpretables, adaptativos y capaces de razonar de manera más cercana a los sistemas cognitivos humanos.

Referencias

1. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. *Attention is All You Need.* In Advances in Neural Information Processing Systems (NeurIPS), 2017.
2. Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.* In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL), 2019. arXiv:1810.04805.

3. Diederik P. Kingma and Max Welling. *Auto-Encoding Variational Bayes*. In International Conference on Learning Representations (ICLR), 2014. arXiv:1312.6114.
4. Larry R. Squire, Paul E. Garrard, and John T. Wixted. *Memory Consolidation*. Cold Spring Harbor Perspectives in Biology, 2015. 10.1101/cshperspect.a021766.
5. Sepp Hochreiter and Jürgen Schmidhuber. *Long Short-Term Memory*. Neural Computation, 9(8):1735–1780, 1997. 10.1162/neco.1997.9.8.1735.
6. Tom B. Brown, Benjamin Mann, Nick Ryder, et al. *Language Models are Few-Shot Learners*. In Advances in Neural Information Processing Systems (NeurIPS), 2020. arXiv:2005.14165.

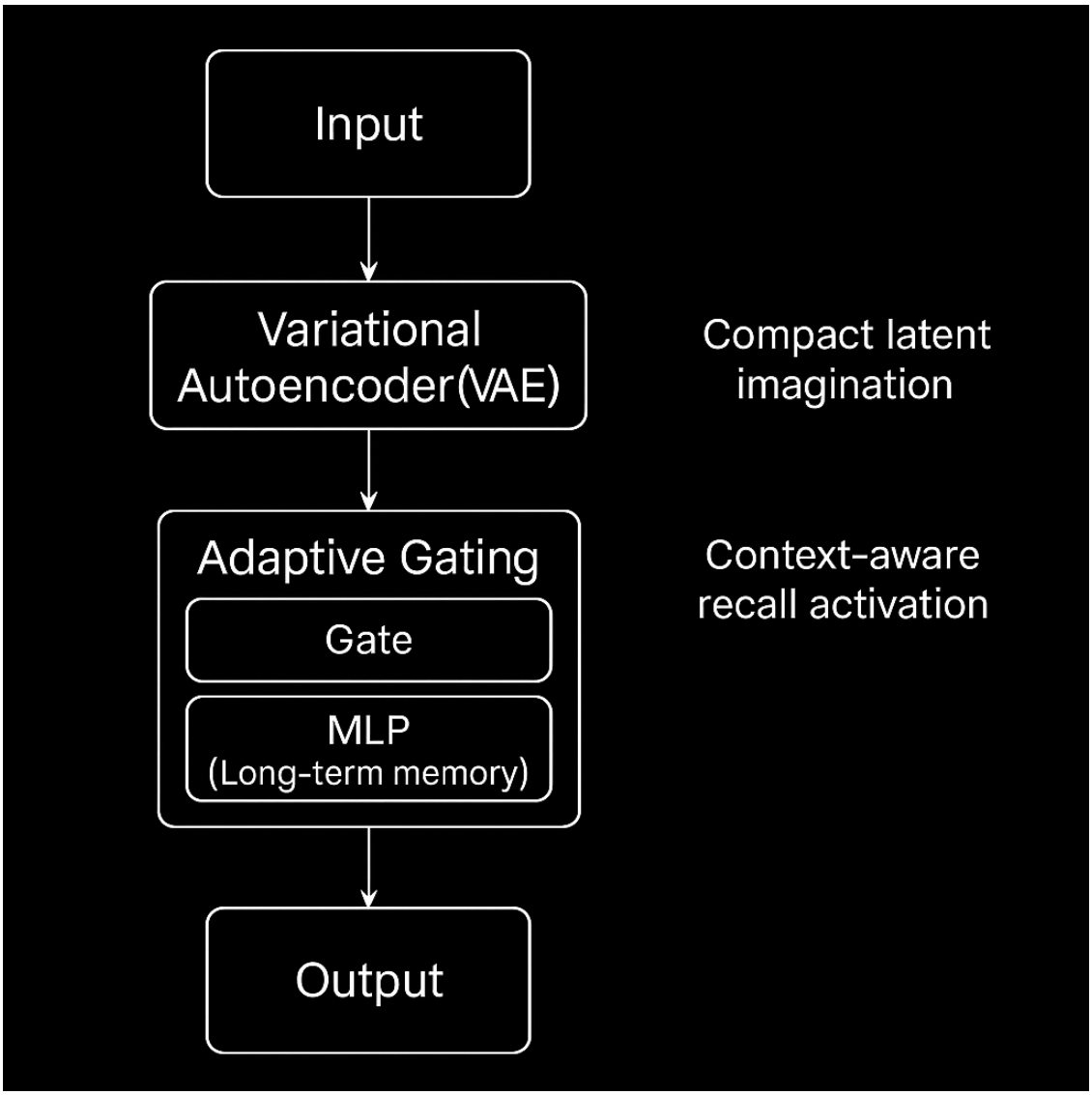

Figura 1: Diagrama esquemático del HMU.