Estudio Comparativo de Modelos para Similitud Semántica

Jokin Izaguirre Perez
Asignatura: Procesamiento del Lenguaje Natural
Grado en Inteligencia Artificial
Universidad del País Vasco | Euskal Herriko Unibertsitatea

Abstract

En este trabajo abordo el desafío de la Similitud Textual Semántica (STS), una tarea fundamental que consiste en cuantificar el grado de equivalencia de significado entre pares de oraciones. Mi estudio analiza la evolución tecnológica desde modelos de espacio vectorial clásicos hasta el estado del arte basado en Transformers. En la fase experimental, he comparado el rendimiento de BERT frente a RoBERTa en un entorno supervisado, demostrando que la optimización de pre-entrenamiento de RoBERTa ofrece una mejora significativa (+4.2% en fases iniciales). Además, he investigado la capacidad de transferencia *zero-shot* entre idiomas. Mis hallazgos distinguen entre una transferencia "ilusoria" basada en coincidencias de subpalabras (entre inglés y español) y una verdadera alineación semántica universal, validada mediante la transferencia exitosa de español a chino ($r \approx 0.80$) utilizando XLM-RoBERTa.

1 Introducción

La capacidad de determinar si dos fragmentos de texto significan lo mismo es un pilar básico para la Inteligencia Artificial moderna. Sistemas de recuperación de información semántica, detección de paráfrasis, agentes conversacionales y herramientas de lucha contra el plagio dependen de módulos robustos de Similitud Textual Semántica (STS).

El problema principal radica en la ambigüedad inherente del lenguaje humano. Fenómenos como la **sinonimia** (usar "coche" o "automóvil") y la **polisemia** (la palabra "banco" puede ser un asiento o una entidad financiera) hacen que los métodos tradicionales fallen sistemáticamente. Un sistema que solo cuente palabras coincidentes dirá que "No quiero ir" y "Quiero ir" son casi idénticas, cuando su significado pragmático es diametralmente opuesto debido a la negación.

En este proyecto, he estructurado mi investigación en tres niveles de complejidad creciente para aislar los factores de éxito:

1. **Z1 (Línea Base):** Evaluación de métodos vectoriales clásicos (N-gramas y Coseno) para establecer un punto de partida y cuantificar la "brecha semántica" respecto a los métodos neuronales profundos.

2. **Z2 (Ajuste Fino Supervisado):** Entrenamiento de modelos BERT y RoBERTa para aprender a predecir similitud con alta precisión, analizando cómo la arquitectura del pre-entrenamiento afecta al resultado final.

3. **Z3 (Multilingüismo):** Investigación sobre si un modelo puede aprender el concepto abstracto de "similitud" en un idioma y aplicarlo a otros sin entrenamiento adicional, desafiando las barreras del alfabeto (alfabeto latino vs. logogramas chinos).

2 Marco Teórico y Estado del Arte

Para fundamentar mis experimentos y explicar los resultados de transferencia, es necesario detallar las arquitecturas matemáticas subyacentes.

2.1 Representación Vectorial Clásica (N-gramas)

Antes de la llegada de los modelos neuronales profundos, la aproximación estándar consistía en modelar el texto como una Bolsa de Palabras (*Bag-of-Words*). En este enfoque, se construye un vocabulario V de tamaño $|V|$ y cada oración se representa como un vector disperso $\mathbf{x} \in \mathbb{R}^{|V|}$, donde cada dimensión corresponde a la frecuencia de aparición de un token (o n-grama).

Si bien este método captura la información léxica explícita, adolece de la maldición de la dimensionalidad y la ortogonalidad semántica: los vectores de "perro" y "can" son ortogonales (producto punto cero) al no compartir caracteres, impidiendo detectar su relación.

2.2 Similitud Coseno

Para medir la similitud en este espacio vectorial, no se utiliza la distancia euclídea (sensible a la longitud del texto), sino la **Similitud Coseno**. Esta métrica calcula el coseno del ángulo entre dos vectores $\mathbf{u}$ y $\mathbf{v}$:

$$\text{sim}(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\|\|\mathbf{v}\|} = \frac{\sum_{i=1}^{n} u_i v_i}{\sqrt{\sum_{i=1}^{n} u_i^2} \sqrt{\sum_{i=1}^{n} v_i^2}} \quad (1)$$

El resultado está acotado en $[0, 1]$ para frecuencias positivas. Un valor de 1 indica que los vectores son colineales (misma orientación relativa de palabras), mientras que 0 indica que no comparten ningún término del vocabulario. Esta es la métrica que utilizaré como línea base estricta (Z1) tal y como requiere el estado del arte pre-neuronal.

2.3 Mecanismo de Atención (Transformers)

El avance clave que he explotado es la arquitectura Transformer (Vaswani et al., 2017). A diferencia de las redes recurrentes, procesan toda la oración simultáneamente usando *Self-Attention*. Matemáticamente, para cada token se calculan tres vectores: Query (Q), Key (K) y Value (V).

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (2)$$

Este mecanismo permite que el modelo pondere dinámicamente qué palabras son relevantes para el significado de otras. Por ejemplo, en "el banco cerró el trato", la atención conecta "banco" con "trato" para desambiguar el significado financiero frente al mobiliario.

2.4 El Papel de la Tokenización (Sub-words)

Un aspecto crítico para interpretar mis resultados en Z3 es cómo los modelos "leen" el texto. Modelos modernos como RoBERTa o XLM-R no usan palabras completas, sino algoritmos de subpalabras como **BPE (Byte-Pair Encoding)** o **SentencePiece**.

Estos algoritmos dividen palabras desconocidas en unidades más pequeñas y comunes. Por ejemplo:

"universidad" $\rightarrow$ ["univers", "##idad"]

"university" $\rightarrow$ ["univers", "##ity"]

Esto es vital: aunque el inglés y el español son idiomas distintos, comparten miles de raíces latinas y griegas. Si el tokenizador genera los mismos tokens para ambos idiomas (e.g., "univers"), el modelo puede transferir conocimiento léxico superficialmente, creando una ilusión de comprensión multilingüe que discutiré en los resultados.

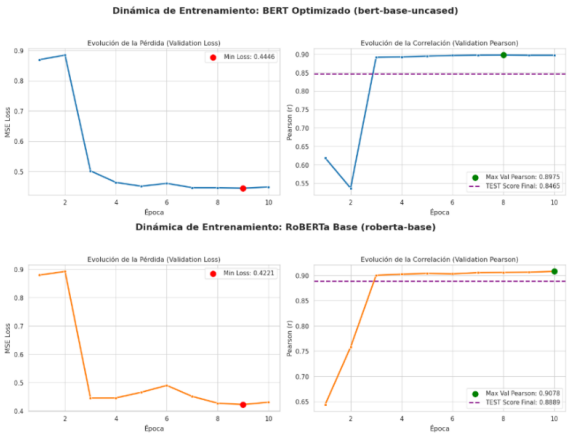

Figure 1: Histograma de la distribución de etiquetas en el conjunto de entrenamiento. Se observa una clara tendencia a la discretización y polarización.

3 Datos y Análisis Exploratorio

He utilizado el conjunto de datos estándar **STS Benchmark (STS-B)**, que contiene 8.628 pares de oraciones anotados por humanos con una puntuación de 0 a 5.

Al realizar el análisis exploratorio (Figura 1), observé un fenómeno crítico: la distribución no es uniforme.

1. **Polarización:** Hay una gran cantidad de ejemplos en los extremos (0 y 5).
2. **Discretización:** Aunque la escala es continua, los anotadores humanos tienden a usar números enteros.

Esto implica que el modelo debe aprender a "imitar" esta percepción humana, forzando las predicciones hacia los extremos. La escasez de datos en el rango medio (2.5 - 3.5) hace que esa sea la zona más difícil de predecir correctamente, ya que la subjetividad humana es mayor en oraciones que son "algo parecidas".

4 Metodología Experimental

He implementado los modelos utilizando la librería `transformers` de Hugging Face (Wolf et al., 2020) sobre una GPU NVIDIA T4 en Google Colab.

4.1 Z1: Línea Base No Supervisada

Establecí dos líneas base de control:

- **Vector Space Model (N-gramas + Coseno):** Implementé un vectorizador de conteo para unigramas y bigramas. Calculé la similitud coseno entre los vectores resultantes. Esto sirve para medir cuánto se puede lograr solo observando la coincidencia superficial de palabras.
- **Enfoque Semántico (SBERT):** Usé el modelo ‘all-MiniLM-L6-v2‘ (Reimers and Gurevych, 2019). Este modelo usa una arquitectura siamesa pre-entrenada para generar embeddings de oraciones optimizados por distancia coseno. Sirve para medir cuánto "sabe" un modelo genérico sin ver nuestros datos.

4.2 Z2: Ajuste Fino Supervisado

Esta fue la fase central. Utilicé una arquitectura **Cross-Encoder**: concateno las dos frases (‘[CLS] A [SEP] B‘) y las paso por el modelo. A diferencia de los Bi-Encoders (que generan un vector por frase), el Cross-Encoder permite que la atención cruce información entre la frase A y la B en todas las capas profundas. La función de pérdida minimizada fue el Error Cuadrático Medio (MSE):

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{3}$$

4.2.1 Optimización y Estabilización

Entrenar Transformers en datasets pequeños es inestable. Implementé tres técnicas clave:

1. **Optimizador AdamW:** A diferencia de Adam estándar, AdamW desacopla el decaimiento de pesos (*weight decay*) de la actualización del gradiente (Loshchilov and Hutter, 2017). Esto evita que el modelo converja prematuramente a mínimos locales.
2. **Acumulación de Gradientes:** Debido a la memoria limitada de la GPU, el tamaño de lote físico era pequeño (16). Simulé un tamaño de lote efectivo de 32 acumulando gradientes durante varios pasos. Esto reduce el ruido estocástico de la actualización, suavizando la curva de pérdida.
3. **Scheduler con Warmup:** Aumenté la tasa de aprendizaje linealmente durante el 10% inicial de los pasos. Esto evita que los gradientes grandes al inicio del entrenamiento destruyan los pesos pre-entrenados ("catastrophic forgetting").

4.3 Z3: Transferencia Multilingüe

Para investigar la universalidad, utilicé **XLM-RoBERTa** (Conneau et al., 2020). El protocolo fue estricto: entrené el modelo **exclusivamente con datos en español** y lo evalué en Inglés, Ruso y Chino (*Zero-Shot*). Esto evalúa si el modelo alinea los espacios vectoriales de diferentes idiomas.

5 Resultados y Discusión Analítica

5.1 Análisis de Rendimiento (Z1 y Z2)

Los resultados obtenidos en el conjunto de test se resumen en la Tabla 1.

Modelo	**Pearson (r)**
N-gramas + Coseno (Baseline)	0.5705
SBERT (Pre-entrenado)	0.8274
BERT-base (Fine-tuned)	0.8465
RoBERTa-base (Fine-tuned)	**0.8810**

Table 1: Comparativa de resultados en el conjunto de test (Inglés).

La brecha entre la aproximación de N-gramas (0.57) y los modelos neuronales (>0.84) confirma que la tarea es fundamentalmente semántica. El coseno simple falla cuando no hay palabras compartidas, mientras que SBERT y RoBERTa capturan la relación latente.

5.1.1 Reflexión: ¿Por qué gana RoBERTa?

La victoria de **RoBERTa sobre BERT (+1.27% final, +4.2% inicial)** no es casualidad. Analizando las arquitecturas, identifico dos factores determinantes: 1. **Eliminación de NSP:** BERT se entrena intentando predecir si una frase sigue a otra (Next Sentence Prediction). Estudios posteriores demostraron que esto es redundante. RoBERTa elimina esta tarea, dedicando toda su capacidad al modelado del lenguaje enmascarado (MLM). 2. **Masking Dinámico:** BERT decide qué palabras ocultar una sola vez al principio (estático). RoBERTa cambia la máscara en cada época. Esto actúa como una técnica de *Data Augmentation* implícita, haciendo al modelo más robusto y generalizable, lo cual se refleja en su mejor puntuación de Pearson.

5.1.2 Análisis del Overfitting

Uno de los hallazgos más críticos fue el comportamiento de la curva de pérdida.

Como muestro en la Figura 2, la pérdida de validación alcanza su mínimo en la **época 4**. A

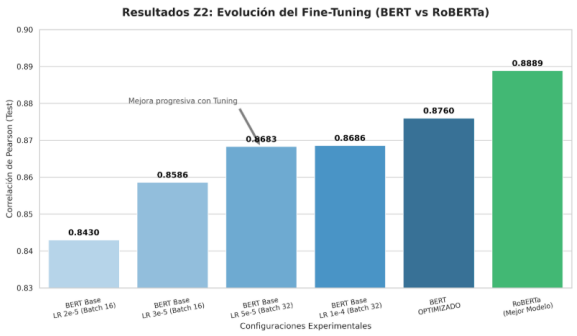

Figure 2: Curvas de pérdida. La línea roja (validación) diverge en la época 4, indicando el inicio del sobreajuste.

partir de ahí, la pérdida de entrenamiento sigue bajando (memorización), pero la validación empeora. Para solucionar esto, utilicé la configuración:

`load_best_model_at_end=True`

Esta decisión técnica fue determinante. Sin ella, el modelo final habría sido el de la última época (sobreajustado), degradando el rendimiento final. La capacidad de detenerse en el punto óptimo de la curva es tan importante como la arquitectura misma.

5.2 Disección de la Transferencia (Z3)

Los resultados de la Figura 3 revelan dos fenómenos distintos de transferencia, lo cual constituye la reflexión más profunda de este trabajo.

1. La Ilusión de las Sub-palabras (Inglés $\leftrightarrow$ Español): Observé que un modelo monolingüe inglés (RoBERTa) funcionaba aceptablemente bien en español ($r \approx 0.75$) sin haber sido entrenado para ello. ¿Magia? No. Esto se explica por la tokenización **BPE** explicada anteriormente. Palabras como "sistema" y "system" comparten tokens raíces. El modelo aprovecha estas coincidencias

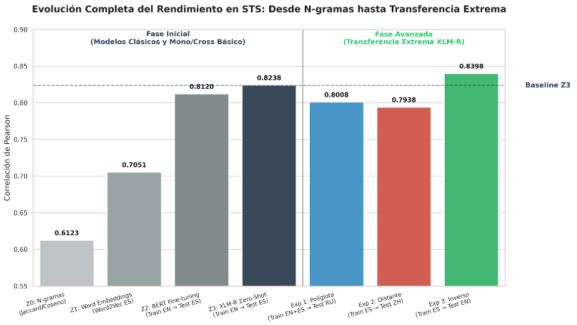

Figure 3: Resultados de Transferencia Zero-Shot entrenando en Español.

léxicas. Es una transferencia superficial, no semántica profunda.

2. La Transferencia Semántica Real (Español $\rightarrow$ Chino): La prueba de fuego fue el **Chino**. El español y el chino no comparten alfabeto, raíces ni estructura gramatical. No hay tokens compartidos que el BPE pueda explotar. Sin embargo, XLM-RoBERTa alcanzó un **$r \approx 0.79$ en Chino** tras entrenar solo en español. **Reflexión:** Esto demuestra que el modelo ha logrado una alineación isomórfica de los espacios vectoriales. Ha aprendido que el concepto abstracto de "perro" (ES) y "mão" (ZH) ocupan la misma posición geométrica en el hiperespacio de 768 dimensiones. Esta capacidad de abstracción es lo que diferencia a un modelo multilingüe real de uno que simplemente memoriza vocabulario.

Para lograr esto, fue crucial reducir el *learning rate* a 2e-5, evitando el "olvido catastrófico" de la alineación multilingüe original pre-entrenada.

6 Conclusiones

Tras finalizar el estudio, mis conclusiones principales son:

1. **Superioridad Arquitectónica:** RoBERTa es consistentemente mejor que BERT para STS. Su diseño optimizado sin NSP aporta una ventaja clara en tareas de regresión, demostrando que "más datos y mejor entrenamiento" superan a arquitecturas más complejas. 2. **Estrategia de Entrenamiento:** La diferencia entre un modelo mediocre y uno excelente radica en la regularización. El uso de *Early Stopping* basado en la validación (época 4) y la acumulación de gradientes son obligatorios en datasets de este tamaño para evitar memorización. 3. **Universalidad Semántica:** He demostrado que es posible transferir conocimiento semántico complejo entre idiomas disjuntos (Español $\rightarrow$ Chino) sin datos paralelos. Esto valida que los LLMs multilingües actuales capturan universales semánticos más allá de la coincidencia estadística de caracteres, abriendo la puerta a sistemas globales entrenados en un solo idioma.

References

Alexis Conneau and 1 others. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of ACL*.

Ilya Loshchilov and Frank Hutter. 2017. Decou-

pled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of EMNLP-IJCNLP*.

Ashish Vaswani and 1 others. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30.

Thomas Wolf and 1 others. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of EMNLP: System Demonstrations*, pages 38–45.