Análisis comparativo de arquitecturas de NLP para detectar similitudes entre escenarios en español

Gabriela Pérez[1,2], Catalina Mostaccio[1,], and Leandro Antonelli[1,3]

[1] LIFIA, Facultad de Informática, Universidad Nacional de La Plata
[2] UNAJ IIyA, Universidad Nacional Arturo Jauretche
[3] CAETI, Facultad de Tecnología Informática, Universidad Abierta Interamericana
{gperez, catty, lanto}@lifia.info.unlp.edu.ar

Resumen La ingeniería de requerimientos es una fase crítica en el desarrollo de software, ya que permite identificar y definir los requerimientos del sistema. Involucra clientes y desarrolladores, quienes deben comunicarse de manera efectiva a pesar de manejar vocabulario diferente. Uno de los artefactos utilizado para este propósito es el escenario, ya que permite especificar el conocimiento de un dominio utilizando lenguaje natural. La especificación de requerimientos implica un trabajo colaborativo, por lo tanto, es esencial detectar tempranamente escenarios similares, con el fin de evitar la duplicación de esfuerzos. Una técnica comúnmente utilizada para identificar similitudes entre oraciones es el uso de LLMs para generar representaciones vectoriales que capturan el significado semántico de las frases en un espacio de alta dimensionalidad. Sin embargo, tienden a generar falsos positivos cuando dos oraciones emplean términos similares con significados distintos, debido a la proximidad superficial de sus embeddings en el espacio vectorial. En este trabajo, se analizan arquitecturas de modelos de procesamiento de lenguaje natural basadas en modelos encoder-decoder para detectar similitudes entre escenarios escritos en español. Para abordar las limitaciones de los encoders tradicionales, se analizan otras estrategias que combinan eficientemente arquitecturas de codificación y decodificación. Esta investigación busca determinar si estas aproximaciones pueden aumentar la precisión y reducir la tasa de falsos positivos en escenarios con terminología diversa.

Keywords: Requerimientos, NLP, LLM, Similaridad Semántica

1. Introducción

La ingeniería de requerimientos constituye una etapa crítica en el ciclo de desarrollo de software ya que permite identificar y detallar, de manera precisa y temprana, los requerimientos del sistema en construcción. Los principales actores en este proceso son los desarrolladores y clientes, que suelen tener perspectivas diferentes. Los clientes, como expertos en el dominio, utilizan un lenguaje propio de su ámbito profesional mientras que los equipos de desarrollo utilizan un lenguaje más técnico, relacionado con la informática. A pesar de estas diferencias,

Proceedings of the 28th Workshop on Requirements Engineering (WER2025),
August 20-22, 2025, Rio de Janeiro-RJ, Brazil.
https://doi.org/10.29327/1588952.28-3

es esencial que puedan comunicarse de manera efectiva a través de artefactos que sean comprensibles por ambas partes. Uno de los artefactos ampliamente utilizado para este propósito son los escenarios [1], [2], [3] ya que permiten detallar el conocimiento de un dominio y pueden utilizarse para definir los requerimientos de un sistema y su dinámica simplemente utilizando lenguaje natural [4], evitando introducir un formalismos complejos. Esta característica los hace particularmente adecuados para ser producidos y comprendidos por los clientes. Los escenarios analizados en este trabajo corresponden específicamente a escenarios del dominio, aunque consideramos que el estudio realizado es igualmente aplicable a escenarios de fases posteriores en el ciclo de desarrollo.

El proceso de especificación de requerimientos implica la colaboración de un equipo donde cada integrante detalla ciertos aspectos del sistema, teniendo en cuenta otros artefactos ya creados. De no hacerlo adecuadamente puede dar lugar a la creación de escenarios superpuestos o duplicados. Esto puede deberse a que se utilizó terminología diferente para expresar una misma situación, o a la necesidad de crear un escenario adicional como una extensión del que se está desarrollando, lo que puede ocurrir desde distintas fuentes. Es importante contar con una herramienta que permita la detección temprana de escenarios similares, realizando un análisis semántico para que funcione aún si se utiliza una terminología diferente. Un requisito fundamental es su adaptación al contexto regional, lo que implica que debe funcionar eficazmente en idioma español y ser de código abierto permitiendo así su implementación en proyectos locales.

En la investigación propuesta por [5], se realiza una evaluación empírica sobre el desempeño de diversos modelos pre entrenados de procesamiento de lenguaje natural aplicados al español, específicamente orientados al análisis de similitud entre escenarios. Los resultados evidenciaron que, si bien los modelos mostraron un rendimiento aceptable, en determinados casos presentaban ciertas limitaciones. En particular, se observaron casos donde se detectaban similitudes inexistentes, lo que resultaba en falsos positivos. En este trabajo, se profundiza en el análisis de las problemáticas presentadas en [5] y se exploran técnicas específicas para optimizar sus resultados, con el objetivo de reducir la incidencia de falsos positivos y mejorar la precisión en la detección de similitudes. Esta investigación constituye una evolución que busca mejorar los métodos de detección de similitud semántica entre escenarios en el contexto hispanohablante.

El resto del trabajo se organiza de la siguiente manera. La sección 2 describe los trabajos relacionados. La sección 3 presenta brevemente los conceptos fundamentales que serán utilizados a lo largo del trabajo. En la sección 4 se proponen y evalúan cuatro estrategias diferentes para abordar la problemática de la similitud entre textos. En la sección 5 se aplican esas estrategias en un caso de uso específico dentro del dominio de la agricultura mostrando su desempeño en ese contexto. Por último, la sección 6 presenta las conclusiones y el trabajo futuro.

2. Trabajos relacionados

El análisis de similaridad en textos cortos ha evolucionado, pasando de enfoques clásicos basados en estadística a modelos de lenguaje más complejos. Técnicas como TF-IDF fueron reemplazadas por métodos que utilizan representaciones vectoriales de palabras, como word2vec [6] y GloVe [7], que capturan relaciones semánticas de manera más efectiva.

Una técnica comúnmente utilizada para evaluar la similitud entre textos consiste en generar esas representaciones vectoriales, conocidas como embeddings, que luego se comparan mediante la similitud del coseno. Esta medida calcula el ángulo entre los vectores, permitiendo evaluar el grado de relación entre los textos. Este enfoque permite capturar múltiples aspectos del significado y calcular el grado de relación entre ellos.

Los modelos de lenguaje de gran escala (LLMs), como BERT (Bidirectional Encoder Representations from Transformers) [8] , marcaron un avance significativo al emplear la arquitectura Transformer, que superó las limitaciones de las redes neuronales tradicionales, como las RNN y LSTM. A diferencia de los modelos previos que generaban representaciones vectoriales estáticas, BERT produce representaciones dinámicas que tienen en cuenta el contexto completo de una palabra o frase, lo que mejora la comprensión semántica. En su versión original, BERT genera embeddings a nivel de token que pueden ser promediados o combinados para representar oraciones completas. Sin embargo, esta aproximación inicial presentaba limitaciones para tareas directas de comparación semántica. Los Sentence Transformers, particularmente Sentence-BERT [9], representan un avance significativo en este campo. Mediante técnicas de redes siamesas y tripletas, estos transformadores pueden medir similitudes semánticas con gran precisión. Sus principales contribuciones incluyen la capacidad de generar embeddings de oraciones completas que capturan el contexto semántico íntegro de un texto, convirtiéndolos en herramientas poderosas para búsqueda semántica, agrupamiento de texto y análisis de similitud. No obstante, los embeddings generados por estos modelos han revelado limitaciones importantes en la representación vectorial de oraciones. Investigaciones como [10] muestran que los embeddings tienden a concentrarse en una región restringida del espacio, fenómeno conocido como “cono estrecho” o “anisotropía”. Esta característica implica que las representaciones no son isotrópicas (no están uniformemente distribuidas en todas las direcciones del espacio). Esto puede resultar en que palabras o frases que son semánticamente distintas tengan representaciones muy cercanas en el espacio vectorial y, como resultado, pueden tener una similitud de coseno alta.

Para superar este problema, en el trabajo de [11], los autores proponen un método de post-procesamiento denominado "whitening"que transforma los embeddings a un espacio ortogonal, abordando eficazmente el problema de la anisotropía. Esta técnica no solo mejora la isotropía de las representaciones semánticas, sino que también permite reducir la dimensión de los embeddings resultantes. Si bien los autores aplicaron esta técnica sobre BERT, en nuestra investigación se adaptó este enfoque para aplicarlo al modelo LaBSE específicamente con corpus en español, evaluando así su aplicabilidad a contextos hispanos.

De forma paralela, surge el paradigma de aprendizaje basado en prompts (prompt-learning) como un enfoque innovador para aprovechar el poder de los grandes modelos de lenguaje. Se centra en la formulación de instrucciones o consultas diseñadas cuidadosamente para que el modelo genere las respuestas deseadas sin necesidad de ajustar los parámetros del modelo. Investigaciones recientes, como [12], han desarrollado nuevos métodos para la tarea de Similaridad Textual Semántica (STS) utilizando prompts. Esta perspectiva emplea la técnica de Cadena de Pensamiento (Chain of Thought o CoT) en un contexto sin ejemplos previos (zero-shot). Al hacer que el modelo siga un proceso de razonamiento paso a paso, se logra una comparación más precisa y detallada del significado de los textos, mejorando su capacidad para identificar relaciones semánticas complejas. Mientras que dicho trabajo utiliza el modelo GPT en inglés, nuestra propuesta se basa en un modelo open source y aplicado sobre textos cortos en español.

3. Background

3.1. Ingeniería de Requerimientos y Escenarios

Los escenarios son herramientas eficaces que permiten explicar cómo funciona un sistema a través de la narración de historias. Este enfoque es efectivo porque permite incorporar detalles que son esenciales para una comprensión más clara y completa de su funcionamiento. Tanto desarrolladores como expertos del dominio pueden usarlos sin la necesidad de aprender formalismos complejos, lo que facilita la comunicación entre las partes interesadas. Además, pueden utilizarse en diferentes etapas del desarrollo de software, para mejorar la comprensión del comportamiento esperado del sistema. Leite [3] define un escenario con los siguientes atributos: (i) un título; (ii) un objetivo que debe ser alcanzado a través de la ejecución del escenario; (iii) un contexto que establece el punto de partida; iv) los recursos, que son objetos físicos o información que debe estar disponible; (v) los actores, que son agentes que realizan las acciones; y (vi) el conjunto de episodios. Cada episodio representa acciones que son realizadas por los actores utilizando los recursos disponibles.

Para diferenciar escenarios similares, se necesitan técnicas que ayuden a comparar su similitud, algunas de las cuales están basadas en LLMs y se presentan a continuación.

3.2. Grandes modelos de lenguaje

Los Transformers representan una arquitectura clave en el procesamiento del lenguaje natural. Desde su introducción en el trabajo 'Attention is All You Need' [13], han revolucionado el campo de la inteligencia artificial. Esta arquitectura está compuesta por dos partes principales: el encoder y el decoder. El encoder se encarga de procesar y transformar la información de entrada, convirtiendo secuencias de texto en representaciones vectoriales, conocidas como embeddings, que capturan el contexto de las palabras. Modelos como BERT y RoBERTa,

basados en el encoder, se destacan en tareas de comprensión de texto, como clasificación, reconocimiento de entidades y respuestas a preguntas.

Por otro lado, el decoder genera texto de manera secuencial, prediciendo cada nueva palabra basándose en las anteriores. Los modelos que usan únicamente decoders, como GPT, LLaMA, Claude y Mistral, son excelentes para generar texto fluido, completar prompts y tareas creativas. Finalmente, las arquitecturas completas de Transformer aprovechan las capacidades de ambos componentes para realizar tareas complejas de procesamiento de lenguaje.

Si bien los encoders y los embeddings han sido herramientas fundamentales en la evaluación de similitud semántica, recientemente ha emergido la ingeniería de prompts como un enfoque alternativo y complementario para abordar esta compleja tarea. En esta propuesta, la clave se encuentra en el diseño sistemático y la optimización de las instrucciones o consultas dirigidas a modelos decoders, con el objetivo de obtener respuestas más precisas y de mayor calidad.

Dado el gran número de modelos de lenguaje y estrategias de entrenamiento disponibles, es esencial contar con una metodología adecuada para compararlos objetivamente. Para ello, se utiliza comúnmente un conjunto de datos estándar, que sirven como entrada para cada uno de los modelos, permitiendo comparar sus resultados de manera consistente. Este enfoque facilita la aplicación de métricas de evaluación consistentes y justas, asegurando que tanto los modelos como las estrategias empleadas puedan ser comparados de manera precisa y objetiva.

3.3. Dataset utilizado

Para este trabajo, se ha seleccionado una versión multilingüe del STS Benchmark (Semantic Textual Similarity Benchmark) llamado stsb_multi_mt [14] que contiene pares de oraciones junto con una etiqueta que indica el puntaje de similitud. La tabla 1 muestra dos de estos pares. La puntuación varía de 0 a 5. Un puntaje de 5 indica que las oraciones son completamente equivalentes y expresan el mismo significado. Un puntaje de 4 significa que las oraciones son mayormente equivalentes, aunque difieren en detalles menores. Si la puntuación es 3, indica que las oraciones son aproximadamente equivalentes, pero falta información importante o difieren en algunos aspectos clave. Un puntaje de 2 indica que las oraciones no son equivalentes, pero comparten algunos detalles en común. Si la puntuación es 1 indica que las oraciones no son equivalentes, pero tratan sobre el mismo tema. Por último, si la puntuación es 0 indica que las oraciones son completamente diferentes, sin relación en su significado.

Este dataset contiene traducciones de las oraciones originales a varios idiomas, utilizando herramientas de traducción automática, lo que ocasionalmente resultó en errores menores. En este trabajo utilizamos exclusivamente el subconjunto de datos en español que está dividido en tres partes, 5749 muestras para entrenamiento , 1379 muestras para prueba y 1500 para validación.

En la figura 1 se presenta un histograma que ilustra la distribución de las puntuaciones asignadas a los pares de oraciones en el dataset. El eje horizontal representa los intervalos de puntuaciones, que van desde 0 hasta 1, divididos en segmentos como 0, 0.1, 0.1-0.2, y así sucesivamente. El alto de cada barra

Tabla 1. Algunas muestras del subconjunto español del dataset.

Textos	**Puntaje**
Un grupo de hombres juega al fútbol en la playa. Un grupo de chicos están jugando al fútbol en la playa.	3.6
Un hombre está sosteniendo una hoja. Un mono está luchando con un hombre.	0

indica la cantidad de pares de oraciones que recibieron una puntuación dentro del intervalo correspondiente. Se observa que la mayor parte de las puntuaciones se concentra en ciertos intervalos específicos, especialmente alrededor del valor 0 y en valores cercanos a enteros como 1, 2, 3, etc. Esto refleja la tendencia de los anotadores humanos a preferir ciertos valores enteros al evaluar la similitud entre oraciones. La presencia de ejemplos en varios intervalos proporciona un rango amplio y variado de datos que resulta adecuado para la evaluación de los modelos y estrategias aplicadas.

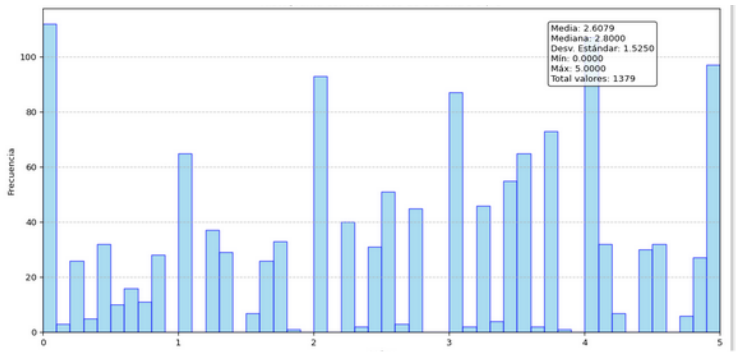

Figura 1. Distribución de las etiquetas de similitud semántica en el conjunto de datos de prueba.

Este subconjunto de datos es procesado por los modelos, y los resultados obtenidos se comparan con las puntuaciones originales del dataset. Para evaluar el desempeño de cada modelo, se utilizan métricas que calculan la correlación entre los puntajes generados por los modelos y las puntuaciones anotadas por humanos. Esto proporciona una medida objetiva para comparar su rendimiento.

3.4. Métricas de evaluación

Para evaluar el rendimiento de modelos se utilizan los coeficientes de correlación de Pearson y Spearman (ρ), que son métricas estadísticas que miden el grado de asociación entre dos variables, en este caso, entre las similitudes predichas por el modelo y las evaluaciones de referencia etiquetadas en el dataset.

El coeficiente de correlación de Pearson (r) mide la relación lineal entre las puntuaciones de similitud generadas por el modelo y las del dataset. Este coeficiente adopta valores entre -1 y 1, donde 1 indica una correlación positiva perfecta, 0 señala ausencia de correlación, y -1 representa una correlación negativa perfecta. Esta métrica resulta particularmente útil para determinar si las variaciones numéricas en las predicciones del modelo reflejan con precisión las variaciones en las evaluaciones humanas.

Por su parte, el coeficiente de correlación de Spearman (ρ) evalúa la relación monótona entre las predicciones del modelo y las referencias humanas, centrándose en el orden o ranking de las similitudes más que en sus valores absolutos. Esta métrica resulta especialmente valiosa para determinar si el modelo clasifica correctamente los pares de textos según su grado de similitud, incluso cuando la relación no es estrictamente lineal.

4. Estrategias analizadas

4.1. Utilizando un modelo codificador

Para llevar a cabo este experimento, se seleccionó LaBSE (Language-agnostic BERT Sentence Embedding) [16] un modelo pre-entrenado para la tarea de similitud semántica, disponible en la plataforma Hugging Face [15], que es compatible con varios idiomas, incluido el español. Este modelo es una variante de la arquitectura BERT diseñada específicamente para la codificación de oraciones completas y el cálculo de la similitud semántica entre ellas. Aunque LaBSE no fue entrenado exclusivamente para el español, su capacidad multilingüe le permite trabajar con más de 50 idiomas, generando embeddings de 768 dimensiones.

Tabla 2. Ejemplos donde se muestra el limite de los embeddings .

Textos	**Coseno**	**Etiqueta**
Un autobús escolar amarillo estacionado en un campo. Un caballo marrón en un campo verde.	0.7041261	0.0
No necesitas ningún visado. No necesitas salsa en absoluto.	0.70859253	0.0

Estrategia 1: En este enfoque se utilizó el modelo encoder para calcular los embeddings de cada par de oraciones y se utilizó similitud coseno entre ellos como medida de similitud semántica. Los resultados obtenidos, se presentan en la Tabla 3, estrategia 1, muestran los coeficientes de correlación de Spearman (ρ) y Pearson, indicando una correlación entre las puntuaciones del dataset y las evaluaciones humanas. En ambas métricas los valores obtenidos son superiores a 0.72, que sugiere un rendimiento general aceptable del modelo. Sin embargo, un análisis más profundo revela discrepancias significativas en ciertos casos de prueba donde la similitud coseno entre embeddings falla en capturar matices semánticos que resultan evidentes para los evaluadores humanos. La Tabla 2

presenta dos oraciones del conjunto de datos, junto con sus etiquetas correspondientes y los valores asignados por la similitud coseno de los embeddings. Estos casos particulares evidencian que las métricas basadas en la similitud coseno entre embeddings pueden diferir significativamente de los juicios humanos sobre la similitud semántica.

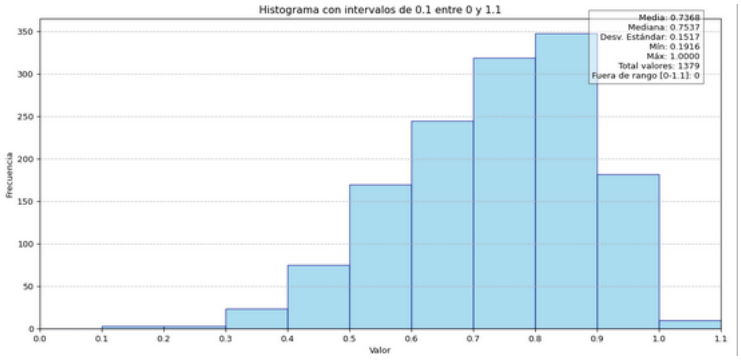

Figura 2. Histograma de la similitud de coseno entre los embeddings obtenidos con LaBSE.

El comportamiento del resto de las predicciones se puede observar mediante un histograma, que muestra la cantidad de ejemplos agrupados en diferentes rangos de predicción. En la figura 2, se puede observar que la mayoría de las predicciones superan el umbral de 0.5, lo cual indica que los embeddings tienden a identificar una similitud considerable entre muchos pares de textos, incluso cuando sus significados subyacentes pueden ser diferentes. Este fenómeno puede atribuirse a la anisotropía, que provoca una distribución no uniforme de los embeddings en el espacio vectorial, concentrándose en regiones específicas. Esto distorsiona las mediciones basadas en el coseno, ya que la distancia angular entre los vectores no refleja las relaciones semánticas, lo que afecta negativamente los cálculos de similitud.

Tabla 3. Coeficientes de Spearman y Pearson para las estrategias evaluadas.

	Modelos	**Tipo**	**Spearman**	**Pearsons**
Estrategia 1	LaBSE	Encoder	0.7291	0.7297
Estrategia 2	Labse - whitenning	Encoder	0.7633	0.7675
Estrategia 3	Gemma 3	Decoder	0.7830	0.7800
Estrategia 4	Gemma 3 + Labse	Decoder + encoder	0.7454	0.7569

Estrategia 2: Para abordar las limitaciones identificadas en la estrategia anterior, se implementó la técnica de whitening como post-procesamiento de los embeddings generados por LaBSE. Esta transformación tiene como objetivo

transformar los datos para que tengan una media de cero y una covarianza de identidad haciendo que los datos sean más .ésféricos", eliminando la correlación entre las dimensiones y asegurando que las características tengan varianzas iguales. Para lograrlo, primero se calcula la media y la matriz de covarianza de los datos del conjunto de entrenamiento. Esto se hace con el objetivo de centrarlos en torno a cero y disminuir las correlaciones entre las características. Luego, a partir de la matriz de covarianza se calcula una matriz de transformación, W, que se usa para transformar los datos de entrenamiento y posteriormente los de prueba. La matriz W garantiza que los datos de prueba se alineen de la misma manera que los datos de entrenamiento. Esta transformación puede mejorar el rendimiento de modelos, haciendo que el modelo sea más eficiente en la captura de relaciones relevantes.

Para evaluar la efectividad de esta estrategia, se calcularon los coeficientes de correlación de Spearman (ρ) y Pearson (r) presentados en la 3, Estrategia 2. Los valores obtenidos, superiores al uso del encoder solo, fueron mayores a 0.76 en ambas métricas. Estas métricas superan las obtenidas con el enfoque sin post-procesamiento, se observa una mejora significativa, lo que indica que el whitening tiene un impacto positivo en el rendimiento del modelo. Es importante resaltar que el proceso de whitening requiere contar con datos de entrenamiento, ya que la transformación depende de las estadísticas (media y covarianza) calculadas sobre este conjunto. Sin datos de entrenamiento disponibles, no sería posible calcular los valores necesarios para aplicar correctamente esta estrategia.

4.2. Utilizando un modelo decodificador

Estrategia 3: Como tercera estrategia, se implementó una aproximación basada en ingeniería de prompts. Para la generación de texto, se utilizó Gemma 3, que es un modelo que lanzó Google como alternativa de código abierto. Este modelo es multimodal, procesando tanto texto como imágenes, y cuenta con una ventana de contexto de 128K, con soporte para más de 140 idiomas y su diseño compacto permite su implementación en dispositivos con recursos limitados.

En este caso se proporciona al modelo un prompt estructurado similar a las instrucciones originales que siguieron los anotadores humanos, enriquecido con ejemplos balanceados extraídos del conjunto de entrenamiento. Estos ejemplos cuidadosamente seleccionados permiten al modelo calibrar mejor sus evaluaciones y producir puntuaciones más alineadas con los criterios originales. El modelo recibe los datos de entrada y simplemente produce una puntuación directa. Se calcularon los coeficientes de correlación de Spearman (ρ) y Pearson (r) entre las puntuaciones generadas por el modelo y las evaluaciones humanas, cuyos resultados se presentan en la Tabla 3, Estrategia 3. Los valores obtenidos fueron superiores a 0.79 en ambas métricas, superando tanto al enfoque del encoder simple como al método con whitening.

Si bien esta estrategia muestra mejores resultados, presenta una limitación. El uso de un modelo decodificador como Gemma 3 para cada título de escenario implica un coste computacional significativo, lo que se traduce en tiempos de respuesta considerablemente más largos en comparación con las estrategias

anteriores. Este factor debe considerarse al evaluar la viabilidad de implementación en aplicaciones que requieran respuestas en tiempo real o procesamiento de grandes volúmenes de datos.

4.3. Utilizando estrategias híbridas

En esta sección se describen estrategias que incluyen tanto a un modelo codificador como a uno decodificador.

Estrategia 4: En esta estrategia se evalúa el impacto de la paráfrasis en la similitud semántica. Debido a que los textos son generalmente muy breves, enriquecerlos mediante paráfrasis podría mejorar la capacidad de los embeddings para capturar mejor la semántica de los textos. En esta estrategia se utiliza el modelo decoder en dos oportunidades por cada ejemplo: primero para obtener un parafraseo del texto original y luego para extraer la acción principal del texto. Aunque este enfoque implica un mayor costo computacional, puede ser realizado en segundo plano sin afectar el rendimiento del sistema.

El siguiente prompt es el utilizado para poder obtener la paráfrasis del texto original utilizando Gemma 3.

```
prompt = f'''Tu tarea es parafrasear un texto de forma clara,
simple y concisa. No se debe extender el significado.
Solo responder con la propuesta de reescritura, sin ninguna otra
información. La respuesta debe ser en español.
La frase es : {texto}
```

Posteriormente se utiliza el modelo encoder para calcular los embeddings y obtener cuatro valores de similitud de coseno:

(i) entre los embeddings de los textos originales.

(ii) entre los embeddings de las respectivas paráfrasis generadas.

(iii) entre los embeddings de textos concatenados con sus paráfrasis y

(iv) entre los embeddings de las acciones principales extraídas.

Todos los embeddings pueden ser calculados y almacenados en segundo plano, lo que permite optimizar el tiempo de procesamiento.

Para determinar la contribución óptima de cada coseno, se llevó a cabo un proceso iterativo de ajuste sobre el dataset de testing. Es decir, por cada muestra del dataset, se calcularon los cuatro cosenos y se obtuvo un coseno ponderado. Tras recorrer todo el dataset, se calcularon los coeficientes de correlación de Spearman y Pearson, presentados en la Tabla 3 Estrategia 4.

Los pesos establecidos fueron (i) * 0.16 + (ii) * 0.20 +(iii) * 0.50 + (iv) * 0.14. Esta aproximación busca proporcionar un contexto semántico más rico que el que ofrecería la oración original por sí sola. Los resultados muestran coeficientes de correlación superiores a 0.74, lo que representa una mejora significativa respecto a los enfoques tradicionales basados únicamente en embeddings directos.

5. Aplicando las estrategias a escenarios de agricultura

Como se mencionó previamente, la definición de escenarios suele ser una tarea colaborativa que involucra a un equipo de trabajo. Por ello, es fundamental

simplificar este proceso garantizando que, cada vez que se cree un nuevo escenario, pueda compararse con los ya existentes para identificar de forma inmediata si ese escenario ya ha sido abordado. Se elige el dominio de la agricultura porque

Tabla 4. Escenarios seleccionados para el análisis.

id	**Título**	**id**	**Título**
1	Eliminar las malezas	8	Cosechar los tomates de forma manual
2	Quitar las malas hierbas	9	Realizar el podado de las plantas
3	Controlar las plagas	10	Controlar las plagas e insectos
4	Despuntar las inflorescencias	11	Regar las plántulas de tomate
5	Regar las plantas de tomate	12	Cosechar los tomates en racimos
6	Controlar las enfermedades bacterianas	13	Controlar las enfermedades virales
7	Prevención de enfermedades fungosas	14	Realizar la poda de forma manual

posee la particularidad que prácticas que persiguen el mismo objetivo se pueden realizar con diferentes técnicas o herramientas. Esto lo convierte en un ejemplo valioso para mostrar el análisis y la interpretación de los resultados en la búsqueda de escenarios similares. Dado que el título suele ser el primer elemento que se redacta en un escenario, nos enfocamos en compararlo con los títulos de escenarios previamente definidos. De este modo, el autor podrá verificar la existencia de otros similares y decidir si es necesaria su creación.

Teniendo en cuenta el trabajo de [5], partimos del mismo conjunto de 14 escenarios previamente definidos, que forman parte de un conjunto de 150, elaborados por profesionales de la industria informática. Los títulos de estos escenarios se detallan en la Tabla 4.

Tabla 5. Resultados de la encuesta realizada a expertos.

	Título nuevo escenario	**Resultados esperados**
Título 1	Realizar fumigación para controlar plagas	id 3, id 6, id 7, id 10, id 13
Título 2	Recortar ramas de la planta	id 4, id 9, id 14
Título 3	Distribuir agua en los cultivos	id 5, id 11
Título 4	Erradicar vegetación indeseada	id 1, id 2
Título 5	Recolectar los tomates maduros	id 8, id 12

El objetivo es simular la creación de nuevos escenarios para evaluar cada estrategia. Para ello, se proponen los siguientes títulos a incorporar: “Realizar fumigación para controlar plagas”, “Recortar ramas de la planta”, “Distribuir agua en los cultivos”, “Erradicar vegetación indeseada”, “Recolectar los tomates maduros”. Estos títulos fueron elegidos para asegurar una diversidad sintáctica y permitir una mejor evaluación.

Para validar los resultados obtenidos, se tomó la opinión de expertos presentada en [5], quienes seleccionaron entre los 14 escenarios presentados anteriormente, aquellos que consideraban como los resultados esperados. Es importante

destacar que en este tipo de análisis no existe una única respuesta correcta, ya que la interpretación varía según diferentes criterios. La tabla 5 presenta los resultados de la consulta. Para facilitar su análisis, las respuestas fueron reorganizadas de acuerdo con el identificador del escenario. Se puede observar que, en el título 1, los resultados esperados incluyen los escenarios con id 6 y 7. Aunque estos escenarios no comparten palabras, sí comparten el propósito subyacente de la acción deseada (prevenir y controlar enfermedades). En el título 2, los escenarios relevantes (id 4, 9 y 14) no contienen ninguna palabra en común con el nuevo título, pero son similares a este. En los títulos 3 y 4, vemos que ninguna de las dos respuestas esperadas tiene palabras en común con el título nuevo. En "Distribuir agua en los cultivos", se espera que los escenarios similares estén relacionados con "regar", a pesar de que se utilizan diferentes términos. En cuanto al título 5, se observa que la palabra "tomate" está presente, aunque otros escenarios también la contienen, pero no comparten la semántica de la frase.

Tabla 6. Resultados obtenidos con las estrategias definidas.

	Título 1 (5 rtas)	**Título 2** (3 rtas)	**Título 3** (2 rtas)	**Título 4** (2 rtas)	**Título 5** (2 rtas)
Estrateg. 1	**id 3 (0.70)**	**id 9 (0.69)**	id 9 (0.65)	**id 2 (0.58)**	id 11 (0.76)
	id 10 (0.69)	id 2 (0.64)	id 4 (0.50)	id 9 (0.57)	id 5 (0.73)
	id 13 (0.62)	id 1 (0.53)	**id 5 (0.48)**	id 5 (0.39)	**id 12 (0.71)**
	id 6 (0.61)	id 5 (0.53)	id 6 (0.46)	id 3 (0.38)	**id 8 (0.66)**
	id 7 (0.59)	id 12 (0.50)	id 10 (0.45)	id 14 (0.37)	id 2 (0.47)
Estrateg. 2	**id 10 (0.69)**	**id 9 (0.64)**	id 9 (0.52)	id 9 (0.54)	id 11 (0.62)
	id 3 (0.66)	id 2 (0.56)	**id 5 (0.33)**	**id 2 (0.52)**	**id 12 (0.59)**
	id 7 (0.55)	id 1 (0.46)	id 2 (0.32)	id 5 (0.39)	id 5 (0.55)
	id 13 (0.54)	id 5 (0.42)	id 3 (0.30)	id 3 (0.38)	**id 8 (0.52)**
	id 6 (0.52)	id 8 (0.37)	id 10 (0.30)	id 14 (0.35)	id 2 (0.34)
Estrateg. 3	**id 3 (4.6)**	**id 9 (3.8)**	**id 5 (3.8)**	**id 1 (3.8)**	**id 8 (3.8)**
	id 7 (3.8)	**id 14 (3.8)**	**id 11 (3.8)**	**id 2 (3.8)**	**id 12 (3.8)**
	id 10 (3.8)	**id 4 (3.2)**	id 2 (2.8)	id 4 (3.2)	id 5 (3.2)
	id 6 (3.2)	id 1 (2.8)	id 8 (2.8)	id 3 (2.8)	id 11 (3.2)
	id 13 (3.2)	id 2 (2.8)	id 12 (2.8)	id 9 (2.8)	id 4 (3.1)
Estrateg. 4	**id 3 (0.66)**	**id 9 (0.69)**	id 9 (0.59)	**id 2 (0.84)**	**id 12 (0.70)**
	id 10 (0.64)	id 2 (0.61)	**id 5 (0.51)**	**id 1 (0.63)**	**id 8 (0.68)**
	id 7 (0.60)	**id 14 (0.55)**	id 13 (0.49)	id 9 (0.62)	id 11 (0.63)
	id 13 (0.57)	id 1(0.53)	id 3 (0.47)	id 3 (0.55)	id 5 (0.62)
	id 6 (0.56)	id 11 (0.49)	**id 11 (0.47)**	id 10 (0.52)	id 2 (0.47)

Los resultados se presentan en la tabla 6. Para cada nuevo título, se enumeran los cinco escenarios más similares, junto con los valores de similitud correspondientes. Las filas de la tabla representan los resultados de una estrategia, mientras que las columnas, los distintos resultados de cada estrategia para un mismo título. Se destacan en negrita los resultados que coinciden con nuestras expectativas.

Si bien los resultados del método híbrido propuesto muestran una mejora respecto al uso exclusivo de embeddings, identificamos un potencial significativo para optimizar aún más el rendimiento. El análisis detallado de estas limitaciones ha abierto caminos prometedores para el refinamiento. Particularmente, estamos explorando con resultados alentadores dos enfoques posteriores: (1) un análisis más profundo respecto a los valores obtenidos de los diferentes cosenos, con la posibilidad de extraer otras características (2) la implementación de una fase posterior donde los candidatos principales sean procesados por un decoder para determinar, en una única consulta, cuáles muestran mayor similitud conceptual.

6. Conclusiones y trabajos futuros

Como se mencionó anteriormente, los embeddings, prometen capturar relaciones semánticas entre elementos textuales, pero presentan limitaciones significativas como la anisotropía. El uso de técnicas como el whitening puede mitigar parcialmente este problema pero su efectividad depende críticamente de la disponibilidad de un conjunto de datos representativo del dominio para calcular la transformación adecuada, lo cual representa una limitación que hace inviable su aplicación.

Por otro lado, los modelos decoder tienen una capacidad significativamente superior para interpretar contextos lingüísticos sofisticados. Sin embargo, escribir el prompt adecuado para que el modelo responda correctamente puede ser dificultoso. Incluso cuando se logra, el modelo puede, en ocasiones, dar respuestas incorrectas o no cumplir con la tarea esperada. Además, su implementación está restringida por la longitud de ventana de contexto del modelo utilizado y el alto costo computacional al realizar las consultas, lo que dificulta su aplicación en sistemas de procesamiento en tiempo real.

En este contexto, la estrategia híbrida parece ser la mejor opción. La combinación de la eficiencia de los embeddings con los modelos decoder permite alcanzar un equilibrio entre precisión semántica y eficiencia computacional debido a que el procesamiento más costoso realizado por los modelos decoder puede llevarse a cabo en segundo plano, facilitando su uso en aplicaciones en tiempo real. De manera similar, el cálculo de los embeddings también puede realizarse de forma asincrónica. Si bien existen otros modelos decoder que probablemente ofrezcan un rendimiento superior, nuestra elección se enfoca en utilizar uno de código abierto que sea compatible con el idioma español. Este enfoque híbrido se presenta como una solución que mejora el desempeño de los modelos basados en embeddings ya que aprovecha al máximo las fortalezas de cada tecnología.

Finalmente, como parte de los trabajos futuros, se propone integrar la estrategia híbrida y evaluarla en un conjunto más voluminoso de datos, de forma que nos permita analizar su desempeño tanto en términos de tiempo de respuesta en tiempo real, como en la capacidad de medir adecuadamente la similitud. Además, se planea realizar un análisis más detallado sobre qué parte de la oración tiene mayor peso en el cálculo de la similitud, ya que consideramos que este es un aspecto clave para mejorar la efectividad de la determinación de similitud

entre dos oraciones cortas. También se explorará la extracción de más características de la oración, para mejorar aún más la capacidad de captura de relaciones semánticas relevantes.

Referencias

1. Alexander, I., Maiden, N.: Scenarios, stories, and use cases: the modern basis for system development. Computing Control Engineering Journal 15(5), 24–29 (2004).
2. Carrol, J. M.: Five reasons for scenario-based design. Proceedings of the 32nd Annual Hawaii International Conference on Systems Sciences, (1999).
3. Leite, J. C. S. d. P., Rossi, G., Balaguer, F., Maiorana, V., Kaplan, G., Hadad, G. Oliveros, A.: Enhancing requirements baseline with scenarios, Requirements Engineering Journal, vol. 2, no. 4, pp. 184-198 (1997).
4. Antonelli, L., Delle Ville, J., Dioguardi, F., Fernandez, A., Tanevitch, L., Torres, D.: An Iterative and Collaborative Approach to Specify Scenarios using Natural Language. Workshop on Requirements Engineering (WER) 2022. pp. , DOI 10.29327/1298262.25-2. (2022).
5. Pérez, G., Mostaccio, C., Antonelli, L.: Evaluación de modelos de procesamiento de lenguaje natural para medir similaridad entre escenarios escritos en español. Brasil. Porto Alegre. 2024. Workshop in Requirements Engineering 2024 (WER 2024).
6. Mikolov,T., Chen, K., Corrado, G., Dean, J.: Efficient Estimation of Word Representations in Vector Space. Proceedings of Workshop at ICLR. (2013).
7. Pennington J., Socher R., y Manning C.: GloVe: Global Vectors for Word Representation. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). https://nlp.stanford.edu/projects/glove/ (2014).
8. Devlin, J., Chang, M., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. NAACL-HLT, pages 4171–4186 -2019.
9. Reimers, N., Gurevyc, I. : Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). (2019)
10. Kawin Ethayarajh. How contextual are contextualized word representations? comparing the geometry of BERT, ELMo, and GPT-2 embeddings. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 55–65, Hong Kong, China. Association for Computational Linguistics. (2019).
11. Jianlin Su, Jiarun Cao, Weijie Liu, Y. Ou.: Whitening Sentence Representations for Better Semantics and Faster Retrieval. 10.48550/arXiv.2103.15316. (2021).
12. Hussain, Musarrat, Rehman, Ubaid Ur, Nguyen, Tri and Lee, Sungyoung. (2024). CoT-STS: A Zero Shot Chain-of-Thought Prompting for Semantic Textual Similarity. 135-139. 10.1145/3639592.3639611.
13. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A., Kaiser, L., Polosukhin, I. : Attention is all you need - 31st Conference on Neural Information Processing Systems, pages 5998–6008 (NIPS 2017), Long Beach, CA, USA. (2017).
14. STSbenchmark dataset https://huggingface.co/datasets/ Philip-May/stsb_multi_mt
15. Hugging Face, https://huggingface.co/, Accedido Marzo 2025.
16. LaBSE - Language-agnostic BERT Sentence Embedding. (https://aclanthology.org/2022.acl-long.62/) (Feng et al., ACL 2022)