Evaluación de modelos de procesamiento de lenguaje natural para medir similaridad entre escenarios escritos 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; busca comprender y documentar los requisitos del sistema desde etapas tempranas. Con frecuencia, la especificación de requerimientos es realizada de forma conjunta entre clientes y el equipo de desarrollo. Los clientes aportan conocimientos profundos en el lenguaje del dominio, mientras que los equipos de desarrollo utilizan términos más informáticos. A pesar de esto, la comprensión mutua es esencial. Uno de los artefactos más utilizados para este propósito son los escenarios. En entornos donde múltiples actores escriben escenarios, es común la duplicación. Es necesario contar con algún mecanismo que posibilite la detección de escenarios similares para evitar tal duplicación. En este trabajo se evalúan empíricamente varios modelos pre-entrenados de Procesamiento de Lenguaje Natural para analizar la similitud semántica entre escenarios en español, identificando palabras o expresiones con significados similares. Es importante destacar que el análisis se realiza en este lenguaje para brindar una contribución a la región. Finalmente, se presenta una herramienta que simplifica la creación de nuevos escenarios, mostrando la eventual existencia de similares. Permite operar con varios modelos y ofrece la posibilidad de seleccionar entre ellos para determinar de forma más precisa si existen escenarios similares al que se está definiendo.

Keywords: Ingeniería de Requerimientos · Escenarios · BERT · Similaridad Semántica · NLP · Sentence BERT.

1. Introducción

La ingeniería de requerimientos es una etapa crítica y fundamental en el desarrollo de sistemas de software. Su objetivo es comprender y documentar en etapas tempranas y de forma precisa los requerimientos del sistema que se está construyendo. Cuando esto no se realiza correctamente, pueden surgir problemas durante las etapas posteriores del desarrollo, cuya resolución resultaría más compleja. Estos problemas pueden incluir funcionalidades faltantes o incorrectas, inconsistencias en el sistema, malos entendidos entre los desarrolladores y

Proceedings of the 27th Workshop on Requirements Engineering (WER24), August 07-09, 2024, Buenos Aires, Argentina.
DOI: 10.29327/1407529.27-4

los clientes, entre otros. En la mayoría de los casos, los clientes y los equipos de desarrollo operan en entornos distintos y utilizan terminologías diferentes.

Los clientes son expertos en el dominio, aportan un conocimiento profundo del problema y su lenguaje está relacionado a ese ámbito, mientras que los equipos de desarrollo utilizan un lenguaje más relacionado a la informática. A pesar de esas diferencias, es esencial que puedan comunicarse de manera efectiva y comprenderse mutuamente a través de artefactos en lenguaje natural que sean comprensibles por ambas partes. Uno de los artefactos ampliamente utilizado para este propósito son los escenarios [1], [4] ya que permiten la especificación del conocimiento de un dominio. Pueden utilizarse tanto para definir los requerimientos de un sistema como su dinámica utilizando lenguaje natural [2], sin introducir un formalismo complejo, por lo que son adecuados para ser producidos y comprendidos por el cliente. Es importante mencionar que los escenarios analizados en este trabajo son escenarios del dominio. Sin embargo entendemos que el estudio realizado aplica también para escenarios de fases posteriores en el ciclo de desarrollo.

En el proceso de especificación de requerimientos es habitual que la tarea no sea responsabilidad de una única persona, sino que implique la colaboración de un equipo de trabajo, compuesto posiblemente por varios miembros. Cada uno de ellos debe detallar ciertos aspectos del sistema, teniendo en cuenta otros artefactos ya creados, ya que de no hacerlo adecuadamente puede dar lugar a la creación de escenarios redundantes. 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. En este contexto, resultaría de gran utilidad 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.

El objetivo principal de este trabajo es realizar una evaluación empírica del desempeño de modelos pre-entrenados para el procesamiento del lenguaje natural en el contexto del español, con el fin de analizar la similitud entre escenarios. Esto implica enfrentar un desafío adicional al trabajar con frases en este idioma, debido a la complejidad lingüística y la amplia gama de expresiones y términos existentes. Se tomó esta decisión con el propósito de desarrollar y ofrecer una herramienta útil en la región. Esta elección resulta especialmente relevante dado que la mayoría de los modelos pre-entrenados están desarrollados para el inglés.

Se llevaron a cabo pruebas y comparaciones entre los siguientes enfoques: TF-IDF, FastText y los modelos SBERT más populares según la cantidad de descargas. Finalmente, se presenta una herramienta diseñada para asistir en el proceso de definición de escenarios, usando los modelos previamente analizados.

El resto del trabajo está organizado de la siguiente manera: en la sección 2 se discuten trabajos relacionados. La sección 3 presenta brevemente los conceptos (*background*) que serán utilizados a lo largo del trabajo. En la sección 4 se describe la estrategia utilizada para experimentar con los modelos seleccionados, y se muestran los resultados obtenidos. La sección 5 incluye una discusión detallada

sobre los problemas que enfrentamos en la realización del trabajo. La sección 6 presenta la herramienta desarrollada que facilita la creación de escenarios. Por último, la sección 7 presenta las conclusiones y el trabajo futuro.

2. Trabajos relacionados

Estimar la similitud entre textos es uno de los problemas de investigación desafiantes y abiertos en el campo del Procesamiento de Lenguaje Natural. La capacidad de medir la similitud entre oraciones es fundamental para una amplia gama de aplicaciones como búsqueda de información, agrupación de documentos, detección de plagio, respuesta a preguntas, entre otras. [16] ofrece un estudio de la similitud semántica de texto, clasificando diferentes tipos de enfoques, como los basados en corpus, los basados en el conocimiento y los basados en cadenas. Por ejemplo, la propuesta de [5] utiliza un enfoque basado en cadena, que implica utilizar una aproximación basada en el método de similitud de Jaccard para analizar y agrupar escenarios. Es importante destacar que esta propuesta es sintáctica, lo que implica que los sinónimos se consideran como palabras distintas y no se tiene en cuenta su relación semántica en estos casos.

Por otro lado, hay varios trabajos que utilizan el enfoque basado en corpus. [17] destaca la importancia de los *embeddings* pre-entrenados de palabras como un recurso fundamental en los sistemas modernos de procesamiento de lenguaje natural, ya que ofrecen mejoras significativas con respecto a los *embeddings* aprendidos desde cero. Continuando con esta línea [19] realizó una evaluación de la eficacia de los métodos de similitud semántica para la comparación de textos académicos y ensayos académicos. Este estudio se enfocó en el procesamiento eficiente de documentos extensos, donde la gestión del tiempo fue un factor crucial a tener en consideración. El lenguaje utilizado en este estudio fue el inglés. Por su parte, en el estudio de [11], se llevó a cabo una evaluación de modelos BERT pre-entrenados para comparar la similitud semántica entre textos no estructurados de ensayos clínicos. En colaboración con investigadores de la Universidad *Johns Hopkins*, se compararon siete modelos BERT pre-entrenados específicamente para aplicaciones médicas. Todos los textos analizados estaban escritos en inglés. En nuestro caso, nos enfrentamos a escenarios que implican textos cortos y un número limitado de ejemplos en español.

3. Background

3.1. Escenarios

Los escenarios son herramientas útiles que permiten explicar cómo funciona un sistema a través de la narración de historias. Este enfoque es efectivo porque es una forma de incorporar detalles que son esenciales para una comprensión más clara y completa de su funcionamiento. Tanto los desarrolladores como los expertos pueden usarlos sin la necesidad de aprender formalismos complejos. Esto facilita la comunicación entre las partes interesadas y se pueden utilizar

en diferentes etapas del desarrollo de software, para mejorar la comprensión del comportamiento esperado del sistema.

Leite [9] 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.

Tabla 1. Ejemplo de un escenario en el dominio de la Agricultura.

Atributos	**Descripción**
Título	Sembrar semillas de tomate
Objetivo	Colocar las semillas de tomate en el almácigo
Contexto	Almácigo preparado
Recursos	Semillas de tomate, almácigo
Actores	Huertero, ingeniero agrónomo
Episodios	El ingeniero agrónomo elige las semillas de tomate.
	El huertero coloca las semillas de tomate en el almácigo.
	El huertero pulveriza con agua el almácigo

En la tabla 1 se describe un escenario específico del dominio de la Agricultura, en particular, el Cultivo de Tomates. Se elige este dominio porque la agricultura posee la particularidad que prácticas que persiguen el mismo objetivo se pueden realizar con diferentes técnicas o herramientas. Esto lo convierte en un ejemplo interesante para mostrar el análisis y la interpretación de los resultados en la búsqueda de escenarios similares. Para poder realizar esto, se necesitan técnicas que permitan evaluar la similaridad entre textos, algunas de las cuales se presentan a continuación.

3.2. Análisis de Similaridad

En el procesamiento del lenguaje natural (NLP) es frecuente tener la necesidad de comparar diferentes palabras o frases entre sí o de buscar patrones dentro de un texto. En muchos casos es de interés encontrar, no solamente las coincidencias exactas entre dos textos, sino también el tener una medida de aproximación o similitud entre éstas cuando la coincidencia no es perfecta. Una técnica comúnmente empleada para evaluar la similitud entre textos consiste en crear una representación vectorial de palabras o frases en un espacio de alta dimensionalidad, conocido como *embedding*, donde cada dimensión del vector puede capturar un aspecto del significado de la palabra o frase. Luego, estos vectores se comparan utilizando alguna medida de similitud, para determinar si son o no similares.

En las siguientes secciones se presentan formas de vectorización de texto.

TF-IDF - Frecuencia de término – frecuencia inversa de documento
TF-IDF (del inglés *Term Frequency-Inverse Document Frequency*) es una técnica estadística utilizada en NLP que permite evaluar la importancia relativa de una palabra respecto a un conjunto de documentos o corpus. La idea detrás de esta técnica es identificar las palabras que aparecen con más frecuencia en el texto. La frecuencia inversa del documento permite disminuir el peso de los términos que aparecen con alta frecuencia en el total de los documentos y otorga mayor valor a las palabras menos comunes.

Para calcular la similaridad entre dos oraciones, primero se realiza un preprocesamiento, que puede incluir la eliminación de los signos de puntuación, la conversión a minúsculas, la eliminación de palabras comunes (*stop words*), y la lematización o *stemming* que permite reducir las palabras a su forma base. Luego, se calculan los valores TF-IDF para cada término en el conjunto de oraciones. Esto implica calcular la frecuencia de término (TF) y la frecuencia inversa de documento (IDF) para cada término en cada oración. La frecuencia de término mide cuántas veces aparece un término en una oración, mientras que la frecuencia inversa mide la rareza del término en el corpus.

Existen otras técnicas que se basan en el entrenamiento de modelos de redes neuronales en grandes conjuntos de datos de texto lo que les permite aprender representaciones vectoriales de palabras.

Word Embeddings
Word2Vec, GloVe y FastText son métodos de *embedding* ampliamente reconocidos y utilizados en el NLP. Word2Vec es una técnica desarrollada por Google en 2013 [10] que permite aprender representaciones vectoriales de palabras de manera eficiente a partir de grandes corpus de texto. Permite capturar relaciones semánticas y sintácticas entre palabras. Sin embargo, genera *embeddings* de palabras de manera independiente y puede generar problemas respecto a palabras polisémicas que tienen diferentes significados en diferentes contextos. En contraste, GloVe (*Global Vectors for Word Representation*) desarrollado en la Universidad de Stanford en 2014 [12], representa una mejora significativa con respecto a Word2Vec. GloVe se centra en la construcción de una representación vectorial global de palabras al considerar tanto la coocurrencia de palabras como su relación de coocurrencia global en el corpus de texto. Esto permite que GloVe capture no solo la semántica local de las palabras, sino también las relaciones semánticas más amplias entre palabras en el corpus. Por otro lado, fastText, desarrollado por Facebook AI Research en 2016 [3], se distingue por su capacidad para generar representaciones de palabras considerando subpalabras o n-gramas. Puede capturar información tanto a nivel de palabra como a nivel de subpalabra, lo que lo hace especialmente útil para idiomas con una rica morfología o palabras compuestas. Esta capacidad única permite que fastText maneje de manera efectiva palabras poco comunes o fuera del vocabulario, y ha demostrado ser útil en una amplia gama de tareas de procesamiento de lenguaje natural.

Continuando con la evolución del NLP, surgieron enfoques más avanzados y especializados, como los modelos de lenguaje pre-entrenados que se describen a continuación.

Grandes modelos de lenguaje
Los grandes modelos de lenguaje (LLM por sus siglas en inglés) son sistemas de inteligencia artificial que se entrenan de forma no supervisada con grandes volúmenes de texto para realizar tareas relacionadas con NLP. Se basan en una arquitectura llamada *Transformers*, que ha ganado gran relevancia en esta área. Desde su presentación en el artículo *Attention is all you need* [18], ha reemplazado a las redes neuronales recurrentes (RNN) y a las redes LSTM (Long Short-Term Memory), ya que ofrece resultados superiores. Modelos destacados en NLP, como BERT [6], GPT [13] y T5 [14], se basan en la arquitectura *transformer*. Estos modelos, pre-entrenados están listos para ser utilizados sin necesidad de realizar un ajuste adicional. Sin embargo, si se desea adaptar el modelo a una tarea específica o mejorar su rendimiento en un dominio particular, es posible realizar ajustes o *fine-tuning* del modelo utilizando un conjunto de datos adecuado para esa tarea.

Es importante considerar que BERT está diseñado para procesar pares de oraciones y no está optimizado para generar un *embedding* a partir de una sola oración, como se menciona en [15]. Para abordar esta limitación surge Sentence-BERT, una variante de BERT que aprovecha redes siamesas y tripletas para este propósito. Esta adaptación amplía significativamente el alcance de BERT, permitiendo su aplicación en nuevas tareas que previamente no eran abordables con la versión estándar. A continuación, presentaremos los experimentos realizados para evaluar las técnicas utilizando versiones que incluyen el español.

4. Experimentos realizados

Como se mencionó previamente, la definición de escenarios suele ser una tarea colaborativa que involucra a un equipo de trabajo. Por lo tanto, el objetivo es simplificar este proceso asegurando que cada vez que se cree un nuevo escenario, pueda ser comparado con los escenarios ya existentes. Esto permitirá identificar de inmediato si se está intentando abordar un escenario que ya ha sido desarrollado. Para llevar a cabo esta tarea, primero necesitamos construir un vector o *embeddig* para cada escenario utilizando los modelos vistos en las secciones previas. Posteriormente, compararemos el vector correspondiente al nuevo título con cada uno de los vectores de los otros títulos. Para determinar qué vector está más cercano empleamos el cálculo del coseno entre vectores. Cuanto más cercano sea el valor del coseno a 1, mayor será la similitud semántica entre los elementos que representan los vectores.

Dado que generalmente el título es lo primero que escribimos en un escenario, nos centraremos en comparar este título con el de otros escenarios. Sin embargo, en ciertas ocasiones, también podemos querer evaluar la similitud entre los títulos y los objetivos, o incluso entre los títulos, los objetivos y los contextos de los

diferentes escenarios. Así, el autor del escenario podrá determinar si es necesario redactar uno nuevo al verificar la existencia de situaciones similares. En esta etapa inicial de experimentación, nos enfocaremos exclusivamente en comparar los títulos de los escenarios.

En este contexto, surge el interrogante sobre la posibilidad de establecer un valor para determinar que dos escenarios son similares. Este valor, conocido como umbral de similitud del coseno, establece el límite que determina si dos títulos se consideran semánticamente similares o no y puede variar según la técnica utilizada. En este estudio, en lugar de fijar un límite específico para este umbral, los escenarios se ordenan de mayor a menor por similitud y se establece una cantidad configurable, para devolver tantos escenarios similares como sea solicitado.

Comenzamos presentando los escenarios que consideramos previamente definidos. Por restricciones de espacio, se seleccionaron 14 de un conjunto de 150, que fueron elaborados por profesionales de la industria informática, cuyos títulos se detallan en la tabla 2.

Tabla 2. 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

Las pruebas que se realizan consisten en simular la creación de escenarios nuevos y evaluar cómo el modelo responde en términos de similitud. Los títulos propuestos para los nuevos escenarios son los siguientes: “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 definidos para asegurar una diversidad sintáctica que permita una mejor evaluación de los modelos. Además, para validar los resultados obtenidos, se llevó a cabo una encuesta entre un grupo de expertos, solicitando que seleccionen dentro de los 14 escenarios presentados anteriormente, cuáles podrían considerarse los resultados esperados. Es importante destacar que en este tipo de análisis no existe una verdad absoluta o un único resultado correcto, ya que la interpretación puede variar según diferentes criterios y enfoques. Por lo tanto, la opinión de los expertos se utilizó como una referencia para evaluar y validar los resultados obtenidos. Aunque las respuestas fueron dadas en diferente orden, pero fueron reorganizadas de acuerdo con el identificador del escenario para facilitar su análisis. En la tabla 3 se presenta el resultado de la encuesta. Se puede observar que, en el título 1, los resultados esperados incluyen los escenarios

Tabla 3. 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

con id 6 y 7. Aunque estos escenarios no comparten palabras, sí comparten el propósito subyacente de la acción deseada (prevenir y controlar enfermedades). Respecto al título 2, se evidencia que los escenarios con id 4 y 14 no contienen ninguna palabra idéntica al nuevo título; sin embargo, son similares a él. En los títulos 3 y 4, vemos que ninguna de las dos respuestas esperadas tienen palabras en común con el título nuevo. Por ejemplo, en la consulta "Distribuir agua en los cultivos", se espera que los escenarios similares estén relacionados con "regar", a pesar 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.

4.1. Pruebas realizadas con TF-IDF

Para realizar las pruebas, se empleó la biblioteca TfidfVectorizer de sklearn para transformar los títulos de los escenarios en una matriz TF-IDF. Los escenarios fueron preprocesados mediante la eliminación de las palabras comunes (*stopwords*) y la lematización de los términos. Posteriormente, se generó la matriz TF-IDF y se calculó el coseno para determinar la similitud entre los escenarios originales y los de consulta. En la parte superior de la tabla 4 se muestran los escenarios y los valores de similitud correspondientes para cada consulta. Las columnas representan los resultados por título. Por ejemplo, para el título 1 se esperaban 5 respuestas, pero se obtuvieron 4. Se incluyen únicamente aquellos escenarios cuyo valor de similitud es mayor que cero. Resaltamos en negritas los resultados que coinciden con los esperados. Entre los paréntesis se encuentra el valor de similitud obtenido.

En el título 1, se puede observar que el valor de similitud es 1. Esto se debe a que la palabra "fumigación" no está presente en el corpus original, es decir, no se encontraba entre los 14 escenarios originales. Dado que es una palabra nueva para el modelo, el algoritmo la ignora durante los cálculos de similitud. En cuanto al segundo título solo se encontró una respuesta correcta de las tres esperadas y da como resultado el escenario con id 5 que es incorrecto. Respecto a los títulos 3 y 4, no se hallaron escenarios similares, posiblemente debido a que no comparten las mismas palabras con los escenarios presentados. Este resultado resalta que el análisis se centra principalmente en la estructura sintáctica, en lugar de considerar su significado semántico. La similitud se determina más por la selección de palabras que por la relación conceptual entre ellas. Esto remarca

Tabla 4. Resultados obtenidos con TF-IDF y fastText.

	Título 1 (5 rtas)	**Título 2** (3 rtas)	**Título 3** (2 rtas)	**Título 4** (2 rtas)	**Título 5** (2 rtas)
TDF-IDF	**id 3 (1.0)** **id 10 (0.74)** **id 6 (0.30)** **id 13 (0.30)**	**id 9 (0.65)** id 5 (0.61)	-	-	id 5 (0.49) id 11 (0.46) **id 12 (0.46)** **id 8 (0.42)**
FastText	**id 3 (0.78)** **id 10 (0.73)** **id 13 (0.68)** **id 6 (0.67)** id 1 (0.65)	**id 14 (0.78)** **id 9 (0.74)** id 5 (0.70) id 11 (0.67) id 8 (0.64)	id 12 (0.85) id 8 (0.83) **id 5 (0.73)** **id 11 (0.72)** id 9 (0.71)	id 10 (0.61) **id 1 (0.60)** id 3 (0.52) id 6 (0.52) id 7 (0.51)	**id 12 (0.86)** **id 8 (0.79)** id 10 (0.55) id 11 (0.55) id 5 (0.54)

la importancia de tener en cuenta tanto el contenido léxico como el contexto semántico al realizar la comparación de textos. Respecto al título 5, aunque se encuentran las respuestas esperadas, no ocupan las primeras posiciones.

4.2. Pruebas realizadas con fastText

El segundo modelo que evaluamos fue fastText, seleccionado por su capacidad para capturar la morfología y las relaciones semánticas en idiomas con una morfología rica, como el español. Utilizamos un modelo pre-entrenado seleccionado entre los disponibles para 157 idiomas en [7], específicamente el correspondiente al español. Normalmente, estos modelos generan vectores con una dimensión de 300 posiciones, pero para adaptarlos a los recursos de hardware disponibles, tuvimos que reducir su dimensión a 100 posiciones.

En la parte inferior de la tabla 4 pueden observarse los resultados obtenidos. Una vez más, resaltamos en negritas los resultados que coinciden con nuestras expectativas. En aquellos casos donde TF-IDF no fue efectivo, como las predicciones sobre el título 3 y 4, observamos que fastText sí proporciona resultados relevantes. En todos los escenarios, fastText identifica al menos uno de los resultados esperados. Sin embargo, en ocasiones, ofrece respuestas inesperadas, como por ejemplo para el título 3, o no abarca todos los resultados esperados.

4.3. Pruebas realizadas con modelos Sentence BERT

Para realizar estas pruebas se seleccionaron modelos pre-entrenados en la tarea de similaridad semántica, disponibles en la plataforma Hugging Face [8] y que pueden ser utilizados en español. Todos los modelos seleccionados son variantes de SBERT (Sentence-BERT), diseñados específicamente para la codificación de oraciones completas y el cálculo de la similitud semántica entre ellas. Se eligieron los cuatro modelos más populares del último mes, siendo el más descargado con 2.85 millones de descargas y el menos descargado con más de 460 mil descargas. También fueron seleccionados para tener variedad en cuanto al tamaño del *embedding* que generan. En la tabla 5 se presenta la lista de estos modelos, junto

con la cantidad de descargas y la dimensión de los *embeddings*. Puede verse que el modelo 1 genera un *embedding* de 384 dimensiones mientras que el modelo 3 uno de 768 dimensiones. Ninguno de los modelos fue entrenado exclusivamente para español, sino que son compatibles con varios idiomas (al menos 50).

Tabla 5. Redes seleccionadas.

	Nombre del modelo	**Descargas**	**Embedding**
Modelo 1	paraphrase-multilingual-MiniLM-L12-v2	2.49M	384
Modelo 2	distiluse-base-multilingual-cased-v2	877k	512
Modelo 3	paraphrase-multilingual-mpnet-base-v2	460k	768
Modelo 4	multilingual-e5-small	2.85M	384

Para cada nuevo título se enumeran los cinco escenarios más similares, junto con los valores de similitud correspondientes. Los resultados para cada modelo se presentan en una tabla 6. En dicha tabla, las filas representan los resultados de un modelo, mientras que las columnas muestran los distintos resultados de los diferentes modelos para un mismo título. Se destacan en negritas los resultados que coinciden con nuestras expectativas. Al observar la tabla, se hace evidente que no es posible establecer un umbral de similitud único que sea efectivo para todos los modelos. No obstante, podría ser factible definir umbrales diferentes para cada modelo. En la primera consulta, para el título 1 "Realizar fumigación para controlar plagas" uno de los modelos muestra una coincidencia perfecta, mientras que en los otros, la coincidencia es de 4 sobre los 5 valores esperados, añadiendo un valor que no era esperado. En el título 2, "recortar ramas de una planta", se observa que todos los modelos incluyen como respuesta el escenario con id 9, que corresponde a "realizar el podado de las plantas". La mayoría también incluye al escenario con id 4, "despuntar las inflorescencias", a pesar que sintácticamente no comparten ninguna palabra. Únicamente el modelo 4 incorpora el escenario con id 14, "realizar la poda de forma manual". Para el título 3, "distribuir agua en los cultivos", todos los modelos identifican como altamente similares al escenario con id 5, "regar las plantas de tomate", y algunos también lo hacen con el escenario id 11, "regar las plántulas de tomate". Esto contrasta con lo obtenido en el modelo TF-IDF. Para el título 4, "Erradicar vegetación indeseada", se observa que el modelo 1 no identifica ninguna de las respuestas esperadas, mientras que los otros modelos sí lo hacen, aunque varían en las posiciones en las que las encuentran. Al igual que en el título 3, se observa que el título no comparte palabras con los escenarios de las respuestas. Finalmente, en el título 5, "Recolectar los tomates maduros", se observa que los modelos incluyen las respuestas esperadas, pero también incorporan respuestas que contienen la palabra "tomate", como los escenarios con id 5 y 11, aunque no están relacionadas con la acción de recolectar.

Podemos observar que los resultados obtenidos en las pruebas realizadas fueron consistentes en todos los modelos, a pesar de las diferencias en sus arquitec-

Tabla 6. Resultados obtenidos con los cuatro modelos SBERT seleccionados.

	Título 1 (5 rtas)	**Título 2** (3 rtas)	**Título 3** (2 rtas)	**Título 4** (2 rtas)	**Título 5** (2 rtas)
Modelo 1	**id 3 (0.92)**	**id 9 (0.79)**	id 9 (0.58)	id 4 (0.68)	id 11 (0.91)
	id 10 (0.83)	id 11 (0.64)	id 4 (0.49)	id 9 (0.67)	id **12 (0.91)**
	id 13 (0.69)	id 5 (0.63)	**id 5 (0.47)**	id 5 (0.53)	id 5 (0.88)
	id 6 (0.67)	id 12 (0.62)	**id 11 (0.47)**	id 11 (0.52)	**id 8 (0.86)**
	id 7 (0.64)	**id 4 (0.59)**	id 12 (0.40)	id 10 (0.46)	id 9 (0.58)
Modelo 2	**id 3 (0.73)**	**id 9 (0.88)**	id 9 (0.59)	id 9 (0.83)	id 11 (0.93)
	id 10 (0.58)	**id 4 (0.65)**	id 4 (0.45)	id 4 (0.68)	**id 12 (0.90)**
	id 6 (0.50)	id 5 (0.61)	id 2 (0.43)	**id 2 (0.62)**	id 5 (0.87)
	id 13 (0.49)	id 2 (0.59)	**id 5 (0.42)**	id 5 (0.60)	**id 8 (0.82)**
	id 2 (0.44)	id 11 (0.46)	**id 11 (0.37)**	**id 1 (0.51)**	id 2 (0.42)
Modelo 3	**id 3 (0.88)**	**id 9 (0.81)**	id 9 (0.54)	id 4 (0.77)	id 11 (0.91)
	id 10 (0.83)	id 2 (0.69)	**id 5 (0.48)**	**id 2 (0.76)**	id 5 (0.89)
	id 7 (0.71)	**id 4 (0.68)**	id 2 (0.47)	id 9 (0.67)	**id 8 (0.84)**
	id 6 (0.66)	id 11 (0.58)	id 4 (0.44)	**id 1 (0.57)**	**id 12 (0.82)**
	id 1 (0.60)	id 5 (0.57)	**id 11 (0.41)**	id 5 (0.56)	id 9 (0.59)
Modelo 4	**id 3 (0.70)**	id 5 (0.93)	id 9 (0.88)	**id 2 (0.92)**	**id 8 (0.93)**
	id 10 (0.69)	id 11 (0.92)	id 3 (0.87)	id 4 (0.92)	id 11 (0.92)
	id 11 (0.62)	**id 9 (0.92)**	**id 5 (0.86)**	**id 1 (0.91)**	id 5 (0.92)
	id 6 (0.61)	**id 14 (0.90)**	**id 11 (0.86)**	id 3 (0.90)	**id 12 (0.92)**
	id 7 (0.59)	id 12 (0.89)	id 4 (0.86)	id id 11 (0.89)	id 1 (0.88)

turas. Esto sugiere cierta estabilidad y fiabilidad general en los modelos, lo que hace posible su utilización en diversas aplicaciones prácticas.

5. Discusión

Nuestro trabajo se centró en evaluar de forma empírica modelos de procesamiento de lenguaje natural para medir la similitud entre escenarios escritos en español. La elección de este idioma nos llevó a enfrentar desafíos significativos debido a la falta de modelos entrenados en español. Por ejemplo, no pudimos encontrar una versión de Word2Vec en español. En el caso de FastText, a pesar de haber encontrado una versión en español, nos vimos limitados por la disponibilidad de recursos. Aunque fastText proporciona un modelo para el idioma español, encontramos ciertas limitaciones al utilizarlo en Google Colab. Normalmente, los *embeddings* de fastText tienen una dimensión de 300 posiciones, pero debido a las restricciones de los recursos de hardware disponibles en este entorno, nos vimos obligados a reducir su dimensión a 100 posiciones para lograr una ejecución adecuada.

Si bien para el caso de los LLMs hay modelos pre-entrenados que incluyen el español, la disponibilidad y variedad de estos modelos es considerablemente menor en comparación con los modelos para el inglés u otros idiomas de mayor difusión. Este problema puede representar un desafío significativo para quienes

trabajan en aplicaciones de procesamiento del lenguaje en este idioma. Además, la situación es similar si se intenta realizar *fine-tuning* de los modelos, ya que la escasez de conjuntos de datos en este idioma dificulta mucho esta tarea. Todo esto resalta la necesidad de un mayor desarrollo y disponibilidad de recursos en español en el campo del procesamiento del lenguaje natural.

6. Herramienta desarrollada

En esta sección se introduce una herramienta diseñada para agilizar el proceso de definición de escenarios. En la Figura 1 se muestra una captura de pantalla de la herramienta. Al iniciar una nueva definición, los usuarios pueden comenzar escribiendo el título correspondiente y, en ese momento, verificar cuáles son los escenarios más similares en relación con ese título. Para realizar la búsqueda de escenarios similares, en la parte derecha de la interfaz se encuentran unos casilleros que permiten seleccionar el modelo que se utilizará. En esta versión, hemos incluido todos los modelos evaluados en el trabajo.

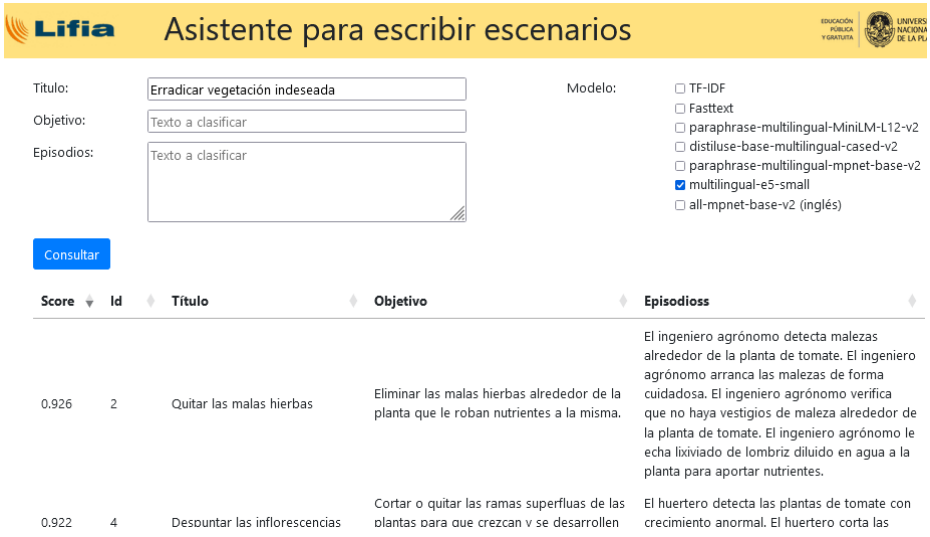

Figura 1. Herramienta para asistir en la creación de escenarios.

Posteriormente, al hacer clic en el botón "Consultar", en la parte inferior de la interfaz se muestran los escenarios ordenados por similaridad con respecto al título ingresado utilizando el modelo seleccionado. En la Figura 1 se puede observar el comportamiento al crear un escenario con el título "Erradicar vegetación indeseada" utilizando el modelo 4. Puede verse que el escenario más similar tiene el id 2, junto con el valor de similitud y otros detalles del escenario.

Dado que ninguno de los modelos arroja respuestas completamente correctas, la capacidad de consultar varios modelos ayuda a determinar de manera más precisa si existen escenarios similares al que se está definiendo.

7. Conclusiones y trabajos futuros

Nuestro estudio se enfocó en la evaluación empírica de modelos de procesamiento del lenguaje natural para medir la similitud entre escenarios escritos en español. Específicamente, nos centramos en la aplicación de tres técnicas de búsqueda de similitud de oraciones: TF-IDF, FastText y SBERT. Pudimos observar que una desventaja significativa de TF-IDF en comparación con modelos más avanzados es su limitación para capturar la complejidad semántica y contextual del lenguaje natural. Este método no considera la estructura de la oración ni el significado de las palabras en un contexto específico, lo que dificulta trabajar con sinónimos. Además, requiere recalcular todos los valores cuando se introducen nuevos datos, es decir, nuevos titulos. Por otro lado es sensible al ruido, como errores tipográficos, palabras irrelevantes o fuera del corpus original.

En el caso de FastText, encontramos que funciona mejor que TF-IDF y demuestra una mayor capacidad para capturar la semántica de los textos. A pesar de haber tenido que reducir la dimensión del *embedding* generado para poder utilizarlo con los recursos disponibles, los resultados fueron satisfactorios y demostraron una mejora significativa.

Por otro lado, las pruebas realizadas en las redes SBERT mostraron resultados prometedores, aunque se identificaron algunas limitaciones. Aunque estas redes ofrecen una representación semántica más profunda de los textos, no lograron encontrar las respuestas esperadas. Uno de los desafíos encontrados fue la variabilidad en los umbrales de similitud establecidos en cada red, lo que dificulta la definición de un umbral único para todas las redes evaluadas.

Finalmente, dado que no existe un modelo que se ajuste perfectamente a las expectativas, la herramienta implementada permite consultar varios modelos para determinar de manera más precisa si el escenario que se intenta definir ya tiene uno similar previamente establecido. Esta flexibilidad en la selección de modelos permite adaptarse a diferentes situaciones, lo que mejora significativamente la eficacia y precisión del proceso. Como trabajos futuros planeamos realizar un ajuste fino de los modelos de lenguaje pre-entrenados existentes utilizando un conjunto de datos específico para evaluar la similitud semántica de manera más precisa. Sin embargo, antes de llevar a cabo este ajuste fino, es esencial construir un conjunto de datos que sea representativo y adecuado para nuestra tarea particular.

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. 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).
3. Bojanowski, P, Grave E, Joulin A y Mikolov: Enriching Word Vectors with Subword Information. NLP Conference on Empirical Methods in Natural Language - 2017
4. Carrol, J. M.: Five reasons for scenario-based design. Proceedings of the 32nd Annual Hawaii International Conference on Systems Sciences, (1999).
5. Delle Ville, J., Torres, D., Fernández, A., Antonelli, L. : An Approach to Cluster Scenarios According to their Similarity using Natural Language Processing, IX Jornadas Iberoamericanas de Interacción Humano – Computadora (JIHCI 2023), 13 al 15 de Septiembre, Universidad de La Matanza, Argentina. (2023).
6. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. 2018. http://arxiv.org/abs/1810.04805 (2018)
7. fastText Homepage, https://fasttext.cc/, Accedido febrero 2024.
8. Hugging Face, https://huggingface.co/, Accedido febrero 2024.
9. 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).
10. Mikolov,T., Chen, K., Corrado, G., Dean, J.: Efficient Estimation of Word Representations in Vector Space. Proceedings of Workshop at ICLR. (2013).
11. Patricoski J, Kreimeyer K, Balan A, Hardart K, Tao J; Hopkins, J. : An Evaluation of Pretrained BERT Models for Comparing Semantic Similarity Across Unstructured Clinical Trial Texts. Stud Health Technol Inform. 2022 Jan 14;289:18-21. doi: 10.3233/SHTI210848. (2022)
12. 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).
13. Radford, A., Narasimhan K., Salimans T., Sutskever I.: Improving Language Understanding by Generative Pre-Training. GPT-4 Technical Report. https://cdn.openai.com/papers/gpt-4.pdf. (2023)
14. Raffel C, Shazeer N, Roberts A., Lee K., Narang S., Matena M., Zhou Y., Li W., Liu P.: Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. https://arxiv.org/abs/1910.10683. (2019)
15. 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)
16. Sunilkumar, P., Athira Shaji, P. : A Survey on Semantic Similarity. International Conference on Advances in Computing, Communication and Control (ICAC3). 20-21 December 2019. Mumbai, India. DOI:10.1109/ICAC347590.2019.9036843 (2019).
17. Turian, J., Ratinov, L, Bengio, Y.: Word representations: A simple and general method for semi-pervised learning. In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, ACL '10, pages 384–394. (2010).
18. 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).
19. Zebari, R. y Ahmed, N.:. Evaluating of Efficacy Semantic Similarity Methods for Comparison of Academic Thesis and Dissertation Texts. Science Journal of University of Zakho 10.25271/sjuoz.2023.11.3.1120. Agosto 2023.