Evaluation of natural language processing models to measure similarity between scenarios written in Spanish

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

[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, gmaltempo}@lifia.info.unlp.edu.ar

Abstract. *Requirements engineering is a critical phase in software development; it seeks to understand and document system requirements from early stages. Typically, requirements specification involves close collaboration be- tween customers and development teams. Customers contribute their expertise in the domain language, while developers use more technical, computational terms. Despite these differences, achieving mutual understanding is crucial.*

One of the most widely used artifacts for this purpose is scenarios. In environments where multiple actors write scenarios, duplication is common. Thus, there is a need for mechanisms to detect similar scenarios and prevent redundancy. In this paper we empirically evaluate several pre-trained Natural Language Processing models to analyze the semantic similarity between scenarios in Spanish, identifying words or phrases with equivalent meanings. It is important to note that the analysis is performed in this language to contribute to the region.

Finally, we present a tool that facilitates the creation of new scenarios by identifying potential similarities with existing ones. The tool supports multiple models, allowing users to select the most appropriate one to detect similar scenarios accurately during the definition process.

1. Introduction

Requirements engineering is a crucial and foundational phase in software systems development. It aims at thorough, early-on understanding and documentation of the requirements for the system under development. When this is not properly executed, issues may arise in later phases of development, the resolution of which is more complex. These is- sues may include missing or incorrect functionalities, inconsistencies within the system, and misunderstandings between developers and clients, among others. Clients and development teams usually operate in different environments and use different terminologies.

Clients are domain experts who provide in-depth knowledge of the

Cadernos do IME - Série Informática
e-ISSN: 2317-2193 (online)
DOI: 10.12957/cadinf.2024.87935

problem and whose language is related to that domain. Development teams, on the other hand, use the language of computing. Despite these differences, both parties need to communicate effectively and understand each other using mutually intelligible natural language artifacts. One of the artifacts widely used for this purpose is scenarios [Alexander and Maiden 2004], [Carrol 1999], as they allow for the specification of do- main knowledge. Scenarios can be used to define both the requirements of a system and its dynamics in natural language [Antonelli et al. 2022] free of complex formalisms, thus making them suitable for production and understanding by the client. It is important to mention that the scenarios analyzed in this work are domain scenarios. However, we understand that the study as conducted also applies to scenarios in later phases of the development cycle.

In the requirements specification process, the task is not commonly a single person's responsibility, but rather involves the collaboration of a team, potentially composed of several members. Each team member must detail certain aspects of the system while considering any other artifacts that have already been created, as failing to do so properly may lead to the creation of redundant scenarios. Redundancy may be due to the use of different terminology to express the same situation, or the need to create an additional scenario as an extension of that being developed, which could arise from diverse sources. In this context, it would be extremely useful to have a tool that enables the early detection of similar scenarios by performing a semantic analysis, allowing it to function even if different terminology is used.

The main objective of this work is to conduct an empirical evaluation on the performance of pre-trained natural language processing models in the context of Spanish, to analyze the similarity between scenarios. Working with phrases in this language poses an additional challenge due to the linguistic complexity and the wide range of existing ex- pressions and terms. This decision was made with a view to develop and provide a useful tool for the region. This choice is particularly relevant given that most pre-trained models are developed for English.

Tests and comparisons were conducted among the following approaches: TF-IDF, FastText, and the most popular SBERT models based on the number of downloads recorded at the time of writing. Finally, we present a tool developed to assist users in the process of defining scenarios, utilizing the previously analyzed models.

The rest of this paper is organized as follows: Section 2 discusses related work. Section 3 briefly presents the concepts (background) to be used throughout the paper. Section 4 describes the strategy adopted to experiment with the selected models and presents the results obtained. Section 5 includes a detailed discussion on the challenges encountered during the work. Section 6 introduces the tool developed to facilitate scenario creation. Finally, Section 7 presents the conclusions and future work.

2. Related Work

Estimating the similarity between texts is one of the challenging research problems that remain open in the field of Natural Language Processing. The ability to measure the similarity between sentences is fundamental for a wide range of applications, such as information retrieval, document clustering, plagiarism detection, and question answering, among others. [Sunilkumar and Shaji 2019] provide a study of semantic text similarity, classifying different types of approach, including corpus-based, knowledge-based, and string-based methods. For example, the proposal in [Delle Ville et al. 2023] uses a string- based approach, which involves employing a method based on Jaccard similarity to analyze and cluster scenarios. It is important to note that this is a syntactic proposal, therefore synonyms are treated as distinct words and their semantic relationship is not taken into account in these cases.

In another vein, several works utilize a corpus-based approach [Turian et al. 2010] highlight the importance of pre-trained word embeddings as a fundamental resource in modern natural language processing systems, as they provide significant improvements over embeddings learned from scratch. Along the same line, [Zebari and Ahmed 2023] evaluate the effectiveness of semantic similarity methods for comparing academic texts and essays. This study is focused on the efficient processing of lengthy documents, where time management was a crucial factor to consider. The language used in this study was English. Meanwhile, in the study by [Patricoski et al. 2022], an evaluation of pre-trained BERT models was conducted to compare semantic similarity among unstructured texts from clinical essays. In collaboration with researchers from Johns Hopkins University, seven BERT models pre-trained specifically for medical applications were compared. All the texts analyzed were written in English. In our case, the scenarios we face involve short texts and a limited number of examples in Spanish.

3. Background

3.1. Scenarios

Scenarios are useful tools for explaining how a system works through storytelling. The effectiveness of this approach lies in the possibility of incorporating details that are essential for a clearer and more complete understanding of the system's functionality. Both developers and domain experts can use scenarios without the need to learn complex formalisms, thus facilitating communication between stakeholders. Scenarios can be used at different stages of software development to improve understanding of the system's expected behavior.

According to [Leite et al. 1997], a scenario consists of the following key attributes: (i) a title; (ii) a goal that must be achieved by executing the scenario; (iii) a context that establishes the starting point; (iv) resources, which are the physical objects or information that must be available; (v) actors, who are agents that perform the actions; and (vi) a set of episodes. Each episode represents actions to be taken by the actors using the available resources.

Table 1. Example of a scenario within the domain of Agriculture.

Attributes	Description
Title	Planting tomato seeds
Goal	Placing tomato seeds in the seedbed
Context	Prepared seedbed
Resources	Tomato seeds, seedbed
Actors	Gardener, agricultural engineer
Episodes	The agricultural engineer chooses the tomato seeds. The gardener places the tomato seeds in the seedbed. The gardener sprays the seedbed with water.

Table 1 presents a specific scenario within the domain of Agriculture, namely that of Tomato Cultivation. This domain has been chosen because agriculture uniquely allows for the same objectives to be achieved through various techniques and tools. This characteristic makes it an intriguing example for demonstrating the analysis and interpretation of results in the search for similar scenarios. To accomplish this, techniques for assessing text similarity are required, some of which are outlined below.

3.2. Similarity Measuring Techniques

In natural language processing (NLP), it is often necessary to compare different words or phrases with each other, or to identify patterns within a text. In many cases, it is of interest not only to find exact matches between two texts but also to measure their proximity or similarity between them when no perfect match exists. A commonly employed technique for assessing similarity between texts is to create a vector representation of words or phrases in a high dimensional space, known as embeddings. Each dimension of this vector captures one aspect of the meaning of the word or phrase. These vectors are then compared using some measure of similarity to determine whether or not they are similar. The following sections present different techniques for vectorizing text.

TF-IDF - Term Frequency-Inverse Document Frequency

TF-IDF is a statistical technique commonly used in NLP to evaluate the relative importance of a word in a set of documents or corpus. The core idea behind this technique is to identify the words that occur most frequently in the text and, at the same time, to take into account their overall rarity. The inverse document frequency component reduces the weight of terms that occur frequently across all documents, thereby giving higher weight to less frequent words.

The process of calculating the similarity between two sentences begins with a pre-processing, which may include removing punctuation marks, converting text to lowercase, removing stop words, and applying lemmatization or stemming to reduce words to their base forms. Next, TF-IDF values are calculated for each term in the sentence set. This includes calculating the term frequency (TF), which measures how often a term occurs in a sentence, and the inverse document frequency (IDF), which measures the rarity of the term within the entire corpus. In addition to TDF-IDF there are other techniques that rely on training neural network models on large text datasets, which allow them to learn vector representations of words.

Word Embeddings

Word2Vec, GloVe, and fastText are widely recognized embedding methods used in NLP. Word2Vec is a technique developed by Google in 2013 [Mikolov et al. 2013] that enables learning vector representations of words efficiently from large text corpora. It its able to capture both semantic and syntactic relationships between words. However, it generates word embeddings independently, which can generate problems with respect to polysemous words, that is, those that have different meanings in various contexts.

In contrast, GloVe (Global Vectors for Word Representation), developed at Stan- ford University in 2014 [Pennington et al. 2014], represents a significant improvement over Word2Vec. GloVe builds a global vector representation of words by considering both the co-occurrence of words and their global co-occurrence relationships in the text corpus. This allows GloVe to capture not only the local semantics of words but also the wider semantic relationships between words in the corpus.

Finally, fastText, developed by Facebook AI Research in 2016 [Bojanowski et al. 2017], has the ability to generate word representations by considering subwords or n-grams. This enables fastText to capture information at both the word and subword levels, making it especially useful for languages with rich morphology or compound words. Such capability allows fastText to effectively manage rare or out-of-vocabulary words, proving beneficial across a wide range of NLP tasks.

As the field of NLP evolved, more advanced and specialized approaches have emerged. These include the pre-trained language models described below.

Large Language Models

Large language models (LLMs) are artificial intelligence systems that were trained in an unsupervised or semi-supervised fashion on large volumes of text to perform NLP-related tasks. They are based on an architecture known as Transformers, which has gained great relevance in this area. Since its introduction in the paper “Attention is All You Need” [Vaswani et al. 2017], this architecture has replaced recurrent neural networks (RNNs) and Long Short-Term Memory (LSTM) networks due to its superior performance. Well- known models in NLP, such as BERT [Devlin et al. 2018], GPT [Radford et al. 2023], and T5 [Raffel et al. 2019], are based on the transformer architecture. These pre-trained models are ready-to-use without requiring additional tuning. However, if there is a need to adapt a model for a specific task or to improve its performance in a particular domain, it is possible to fine-tune the model using a dataset suitable for that task.

It is important to note that BERT is designed to process pairs of sentences and is not optimized for generating embeddings from single sentences, as mentioned in [Reimers and Gurevych 2019]. In order to overcome this limitation, a variant of Bert called Sentence-BERT has been developed, which uses Siamese networks and triplets. This adaptation significantly extends the scope of BERT, enabling its application to new tasks that were previously unfeasible with the standard version. In the following sections, we present the experiments carried out to evaluate these techniques, using versions that include Spanish.

4. Model Similarity Evaluations

As mentioned above, scenario definition is usually a collaborative task involving a team. The aim is to simplify this process by ensuring that each time a new scenario is created, it can be compared against the existing ones. Thus it will be possible to immediately identify if a scenario being created has already been developed by another team member. To accomplish this task, we first create an embedding for each scenario using the models described in the previous sections. We then compare the embedding corresponding to the new title against each of the existing scenario embeddings. To determine which embeddings are closest to the embedding of the new title we use the cosine between the vectors. The closer the cosine value is to 1, the greater the semantic similarity between the elements represented by the vectors.

As the title is usually the first thing that is written in a scenario, our focus will be on comparing this title with the titles of the other scenarios. However, on certain occasions, we may also want to evaluate the similarity between the titles and objectives, or even between the titles, objectives, and contexts of different scenarios. This approach allows the author of a new scenario to determine whether it is necessary to write a new scenario after checking for the existence of similar situations. In this first phase of experimentation, we will focus exclusively on comparing the titles of the scenarios.

In this context, an important question arises regarding the establishment of a threshold value to determine the similarity between two scenarios. This value, known as the cosine similarity threshold, defines the boundary that determines whether two titles are considered semantically similar. The actual threshold can vary depending on the technique employed. In this study, instead of setting a specific bound for this threshold, the scenarios are ranked from highest to lowest similarity and a configurable quantity is set to return as many similar scenarios as requested.

We begin by presenting the already defined scenarios against which we will com- pare the title of the new scenario. Due to space limitations, 14 have been selected from a set of 150 scenarios created by IT industry professionals; the titles are detailed in Table 2.

Table 2. Scenarios selected for analysis

id	Title of a defined scenario
1	Eliminar las malezas [Remove Weeds]
2	Quitar las malas hierbas [Remove Weeds]
3	Controlar las plagas [Control Pests]
4	Despuntar las inflorescencias [Pinch Flower Clusters]

5	Regar las plantas de tomate [Water Tomato Plants]
6	Controlar las enfermedades bacterianas [Control Bacterial Diseases]
7	Prevención de enfermedades fungosas [Prevent Fungal Diseases]
8	Cosechar los tomates de forma manual [Harvest Tomatoes Manually]
9	Realizar el podado de las plantas [Prune the Plants]
10	Controlar las plagas e insectos [Control Pests and Insects]
11	Regar las plántulas de tomate [Water Tomato Seedlings]
12	Cosechar los tomates en racimos [Harvest Tomatoes in Clusters]
13	Controlar las enfermedades virales[Control Viral Diseases]
14	Realizar la poda de forma manual [Perform Manual Pruning]

The tests performed consist of simulating the creation of a new scenario and evaluating how the model responds regarding which of the existing scenarios are the most similar to the new one. The proposed titles for these new scenarios are as follows: “Realizar fumigación para controlar plagas” [Perform fumigation to control pests], “Recortar ramas de la planta” [Trim branches of the plant], “Distribuir agua en los cultivos” [Distribute water in the crops], “Erradicar vegetación indeseada” [Eradicate unwanted vegetation] and “Recolectar los tomates maduros” [Harvest ripe tomatoes]. These titles were chosen to ensure enough syntactic diversity for a better evaluation of the models. Additionally, in order to validate the results obtained, a survey was carried out among a group of experts, who were asked to select, among the 14 scenarios presented above, those that could be considered as expected results. It is important to note that in this type of analysis, there is no absolute truth or single correct result, as interpretations can vary according to different criteria and perspectives. Therefore, the experts’ opinions were used as a reference to evaluate and validate the results obtained.

Although the responses were presented in a different order, they were reorganized according to the scenario identifier to facilitate their analysis. Table 3 shows the results of the survey. For Title 1, the expected outcomes include the scenarios with IDs 6 and 7, which do not have identical words but do share the underlying purpose of the desired action (disease prevention and control). For Title 2, the scenarios with IDs 4 and 14 do not have identical words compared to the new title but they still show similarities. For Titles 3 and 4, none of the expected responses have words in common with the new title. For example, in the query “Distribuir agua en los cultivos” [Distribute water in the crops], it is expected that similar scenarios will relate to “regar” [watering], despite the use of different terminology. In Title 5, the word “tomate” [tomato] is present but, although other scenarios also include it, they do not have the same semantic context.

Table 3. Results of experts’ survey

	Title of new scenario	**Expected results**
Title 1	Realizar fumigación para controlar plagas [Perform fumigation to control pests]	id 3, id 6, id 7, id 10, id 13
Title 2	Recortar ramas de la planta [Trim branches of the plant]	id 4, id 9, id 14

Title 3	Distribuir agua en los cultivos [Distribute water in the crops]	id 5, id 11
Title 4	Erradicar vegetación indeseada [Eradicate unwanted vegetation]	id 1, id 2
Title 5	Recolectar los tomates maduros [Harvest ripe tomatoes]	id 8, id 12

4.1. Evaluating TF-IDF Performance

For the tests performed, the TfidfVectorizer library from sklearn was used to transform the scenario titles into a TF-IDF matrix. The scenarios were preprocessed by removing stop words and lemmatizing the terms. The TF-IDF matrix was then generated and cosine similarity was calculated to determine the similarity between the original scenarios and the query scenarios. The first row under the headings of Table 4 shows the scenarios and the similarity values corresponding to each query. The columns show the results for each title. For example, in the case of Title 1, five answers were expected but only four were obtained. Only those scenarios with a similarity value greater than zero were included. The results that match the expected ones are highlighted in bold. The obtained similarity value is shown in brackets.

In Title 1, we can see that the similarity value is 1. This phenomenon occurs because the term "fumigación" [fumigation] is not present in the original corpus, i.e. it was not found in the 14 original scenarios. As it is a new word for the model, the algorithm ignores it in the similarity calculations, resulting in a similarity score of 1.

For Title 2, only one correct answer was found out of the three expected, beside the scenario with ID 5, which is incorrect. In the case of Titles 3 and 4, no similar scenarios were found, probably because there were no words in common with the scenarios presented. This result shows that the analysis focuses mainly on the syntactic structure rather than on semantic meaning. Similarity is determined by the choice of words rather than by the conceptual relationships between them. This underlines the importance of considering both lexical content and semantic context when comparing texts. With regard to Title 5, the expected answers are found, but they do not occupy the first positions.

Table 4. Results from TF-IDF and fastText

	Title 1 **(5 ans.)**	**Title 2** **(3 ans.)**	**Title 3** **(2 ans.)**	**Title 4** **(2 ans.)**	**Title 5** **(2 ans.)**
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)

4.2. Evaluating fastText Performance

The second model evaluated was fastText, chosen for its ability to capture morphology and semantic relations in languages with rich morphological structures, such as Spanish. We used one pre-trained model from those available for 157 languages in [fastText 2024], namely that corresponding to Spanish. These models generate embeddings of 300 dimensions, but in order to adapt them to the available hardware resources, we had to reduce their dimension to 100 positions.

The results are shown at the bottom of Table 4. Again, the results that matched our expectations are highlighted in bold. In the cases where TF-IDF was not effective, particularly the predictions for Titles 3 and 4, we observe that fastText provides relevant results. In all cases, fastText identified at least one of the expected results. However, it sometimes gives unexpected answers, for example for Title 3, or does not cover all the expected results.

4.3. Evaluating Sentence BERT Models performance

In order to carry out these tests, pre-trained models for the semantic similarity task were selected, which are available in the Hugging Face platform [Hugging Face 2024], and can be used with Spanish sentences. All the selected models are variants of SBERT (Sentence- BERT), specifically designed to encode complete sentences and compute semantic similarity between them. The four most popular models of the last month were chosen, the most downloaded model having 2.85 million downloads and the least downloaded one with more than 460 thousand downloads.

They were also chosen to be diverse in terms of the size of the embeddings they generate. Table 5 shows the list of these models, together with the number of downloads and the size of the embeddings they generate. It can be seen that Model 1 generates an embedding of 384 dimensions while Model 3 generates an embedding of 768 dimensions. None of the models has been trained exclusively for the Spanish language, but are compatible with several languages (at least 50).

Table 5. Selected models.

	Model name	Downloads	Embedding
Model 1	paraphrase-multilingual-MiniLM-L12-v2	2.49M	384
Model 2	distiluse-base-multilingual-cased-v2	877k	512
Model 3	paraphrase-multilingual-mpnet-base-v2	460k	768
Model 4	multilingual-e5-small	2.85M	384

For each new title, the five most similar scenarios and the corresponding similarity values are listed. The results for each model are shown in Table 6. In this table, rows show the results for the individual model, while columns show the different results for the same title across the different models. The results that are in line with our expectations are highlighted in bold. It is clear from the table that it is not possible to define a single similarity threshold that is effective for all models.

Table 6. Results obtained using the five selected SBERT models.

	Title 1	**Title 2**	**Title 3**	**Title 4**	**Title 5**
	(5 ans.)	**(3 ans.)**	**(2 ans.)**	**(2 ans.)**	**(2 ans.)**
Model 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)
Model 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)
Model 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)
Model 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 11 (0.89)	id 1 (0.88)

However, it might be possible to de- fine different thresholds for each model. For Title 1 "Realizar fumigación para controlar plagas", one of the models shows a perfect match, while the others successfully identify 4 out of the 5 expected identifiers, but add one unexpected result. For Title 2 “Recortar ramas de una planta”, it can be seen that all the models answers include the scenario with ID 9, which corresponds to "Realizar el podado de las plantas". Most of the models’ answers also include the scenario with ID 4, “Despuntar las inflorescencias”, although they do not share any words syntactically. Only Model 4 includes the scenario with ID 14, “Realizar la poda de forma manual”. For Title 3 "Distribuir agua en los cultivos", all models identify as highly similar the scenario with ID 5, "Regar las plantas de tomate", and some also identify the scenario with ID 11, "Regar las plántulas de tomate". This contrasts with the results obtained with the TF-IDF model. For Title 4 “Erradicar vegetación indeseada” it can be seen that Model 1 does not identify any of the expected responses, whereas the other models do, although they differ in the positions in which they find them. As in the case of Title 3, it can be seen that Title 4 does not share any words with the response scenarios. Finally, for Title 5 “Recolectar los tomates maduros”, it can be seen that the models include the expected responses, but also incorporate other responses that contain the term “tomate”, such as the scenarios with IDs 5 and 11, although they are not related to the action of harvesting.

The results obtained from the tests showed consistency across all models, despite variations in their architectures. This observation indicates a certain degree of stability and overall reliability in the models, suggesting their potential utility in a range of practical applications.

5. Discussion

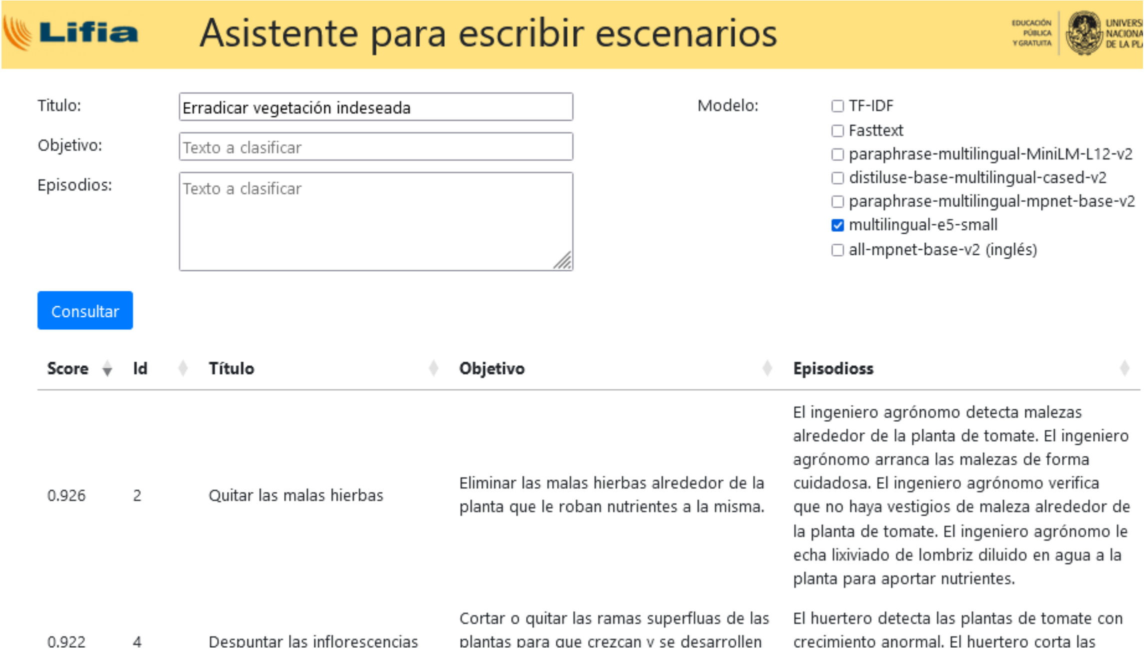
Our study focused on empirically evaluating natural language processing models to determine the similarity between scenarios written in Spanish. The choice of this language faced us with significant challenges due to the lack of models trained in Spanish. For example, we could not find a version of Word2Vec to work with Spanish sentences. In the case of fastText, we did find a version to work in Spanish, but were limited by the availability of resources needed to use it. FastText provides a template for the Spanish language, but we were limited when using it in Google Colab. The standard version gen- erates embeddings of 300 values, but due the limited hardware resources available in this environment, we had to reduce that dimension to 100 values to be able to use that model.

There are pre-trained LLMs that include the Spanish language, but the availability and variety of these models is considerably smaller compared to models for English or other more widely spoken languages. This smaller number of available models can represent a significant challenge for those working on language processing applications in this language. Furthermore, the situation is similar if one tries to perform fine-tuning of the models, as the limited availability of datasets in this language makes this task very difficult. All this shows the need for further development and availability of resources in Spanish in the field of natural language processing.

6. Tool Developed for Assisting in the Definition of New Scenarios

This section introduces the tool designed to support the process of defining new scenarios. Figure 1 shows a screenshot of the tool. When defining a new scenario, the user starts by entering its title. The tool then displays the existing scenarios sorted by similarity, i.e. first those that are most similar to the title entered. In this way, the user can check whether a scenario similar to the one being defined already exists. On the right side of the interface there are checkboxes for selecting the model to be used to obtain the embeddings of each scenario. In this version, we have included all the models evaluated in this work.

When the “Consultar” [consult] button is clicked, the existing scenarios are listed at the bottom of the tool, sorted by their similarity to the title entered, using the selected model. Figure 1 shows the behavior of the tool when a scenario with the title “Erradicar vegetación indeseada” [Eradicate unwanted vegetation] is created using Model 4. It can be seen that the most similar scenario has the ID 2, along with the similarity value and other scenario fields.



Lifia Asistente para escribir escenarios

Titulo: Erradicar vegetación indeseada

Objetivo: Texto a clasificar

Episodios: Texto a clasificar

Modelo:
- ☐ TF-IDF
- ☐ Fasttext
- ☐ paraphrase-multilingual-MiniLM-L12-v2
- ☐ distiluse-base-multilingual-cased-v2
- ☐ paraphrase-multilingual-mpnet-base-v2
- ☑ multilingual-e5-small
- ☐ all-mpnet-base-v2 (inglés)

Score ▼	Id	Título	Objetivo	Episodioss
0.926	2	Quitar las malas hierbas	Eliminar las malas hierbas alrededor de la planta que le roban nutrientes a la misma.	El ingeniero agrónomo detecta malezas alrededor de la planta de tomate. El ingeniero agrónomo arranca las malezas de forma cuidadosa. El ingeniero agrónomo verifica que no haya vestigios de maleza alrededor de la planta de tomate. El ingeniero agrónomo le echa lixiviado de lombriz diluido en agua a la planta para aportar nutrientes.
0.922	4	Despuntar las inflorescencias	Cortar o quitar las ramas superfluas de las plantas para que crezcan y se desarrollen	El huertero detecta las plantas de tomate con crecimiento anormal. El huertero corta las

Figure 1. Tool developed for assisting in the definition of new scenarios.

As none of the models have completely correct answers, the ability to query multiple models helps to more accurately determine if there are scenarios similar to the one being defined.

7. Conclusions and future work

Our study focused on the empirical evaluation of natural language processing models to measure the similarity between scenarios written in Spanish. Specifically, we focused on the application of three sentence similarity techniques: TF-IDF, fastText and SBERT. We observe that a major drawback of TF-IDF compared to more advanced models is its limitation for capturing the semantic and contextual complexity of natural language. It does not take into account sentence structure nor the meaning of words in a particular context, which makes it difficult to work with synonyms. It also requires all values to be recalculated when new data, i.e. new titles, are introduced. On the other hand, TF-IDF is sensitive to noise such as typing errors, irrelevant words or words outside the original corpus.

In fastText, we found that it performed better than TF-IDF and demonstrated a greater ability to capture the semantics of the texts. Despite the fact that we had to limit the model by reducing the size of the generated embedding in order to be able to use it with the available resources, the results were satisfactory and showed a significant improvement.

Finally, tests carried out on SBERT networks showed promising results, although some limitations were identified. These networks offer a deeper semantic representation of texts, but do not find the expected answers. One of the challenges encountered was the variability of the similarity thresholds established in each network, which makes it difficult to define a single threshold for all the networks evaluated.

To conclude, since no model perfectly matches the expectations, the implemented tool allows the user to consult several models in order to determine more precisely whether the scenario to be defined has already been defined by a similar one. This flexibility in model selection makes it possible to adapt to different situations. This significantly improves the efficiency and accuracy of the process. As future work, we plan to refine the existing pre-trained linguistic models using a specific dataset to assess semantic similarity more accurately. However, prior to this fine-tuning, it is essential to build a dataset that is representative and suitable for our particular task.

References

Alexander, I. and Maiden, N. (2004). Scenarios, stories, and use cases: the modern basis for system development. *Computing Control Engineering Journal*, 15(5):24–29.

Antonelli, L., Delle Ville, J., Dioguardi, F., Fernandez, A., Tanevitch, L., and Torres,

D. (2022). An iterative and collaborative approach to specify scenarios using natural language. In *Proceedings of the Workshop on Requirements Engineering (WER).*

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2017). Enriching word vectors with subword information. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP).*

Carrol, J. M. (1999). Five reasons for scenario-based design. In *Proceedings of the 32nd Annual Hawaii International Conference on Systems Sciences*.

Delle Ville, J., Torres, D., Fernández, A., and Antonelli, L. (2023). An approach to cluster scenarios according to their similarity using natural language processing. In *IX Jornadas Iberoamericanas de Interacción Humano – Computadora (JIHCI 2023)*, Universidad de La Matanza, Argentina.

Devlin, J., Chang, M., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidi- rectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

fastText (2024). fasttext homepage. Accessed February

2024. Hugging Face (2024). Hugging face. Accessed

February 2024.

Leite, J. C. S. d. P., Rossi, G., Balaguer, F., Maiorana, V., Kaplan, G., Hadad, G., and Oliveros, A. (1997). Enhancing requirements baseline with scenarios. *Requirements Engineering Journal*, 2(4):184–198.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. In *Proceedings of the Workshop at the International Conference on Learning Representations (ICLR)*.

Patricoski, J., Kreimeyer, K., Balan, A., Hardart, K., Tao, J., and Hopkins, J. (2022).

An evaluation of pretrained bert models for comparing semantic similarity across unstruc- tured clinical trial texts. *Stud Health Technol Inform*, 289:18–21.

Pennington, J., Socher, R., and Manning, C. (2014). Glove: Global vectors for word rep- resentation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. (2023). Improving language understanding by generative pre-training. gpt-4 technical report. Technical report, Ope- nAI.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. (2019). Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Reimers, N. and Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natu- ral Language Processing and the 9th International Joint Conference on Natural Lan- guage Processing (EMNLP-IJCNLP)*.

Sunilkumar, P. and Shaji, P. A. (2019). A survey on semantic similarity. In *Proceedings of the International Conference on Advances in Computing, Communication and Control (ICAC3)*.

Turian, J., Ratinov, L., and Bengio, Y. (2010). Word representations: A simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL ’10)*, pages 384–394.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. In *Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017)*, pages 5998–6008.

Zebari, R. and Ahmed, N. (2023). Evaluating the efficacy of semantic similarity methods for comparison of academic thesis and dissertation texts. *Science Journal of University of Zakho*, 11.