ESPOCH Congresses: The Ecuadorian Journal of S.T.E.A.M.
Volume 3, Issue no. 1, DOI 10.18502/espoch.v3i1.14466
Production and Hosting by Knowledge E

Conference Paper

Use of Machine Learning Models of the “Transformers” Type in the Construction of Services in a Gamified Web app

Utilización de modelos de aprendizaje automático del tipo “Transformers” en la construcción de servicios en una web app Gamificada

C Saavedra Escalante[1], D Alava Santana[1], F Moreira Moreira[2]*, and R Moreira Pico[2]

[1]Escuela Superior Politécnica Agropecuaria de Manabí “Manuel Félix López”. Calceta, Ecuador
[2]Grupo de Investigación SISCOM, ESPAM MFL. Calceta, Ecuador

IX CONGRESO INTERNACIONAL DE INVESTIGACIÓN DE LA RED ECUATORIANA DE UNIVERSIDADES Y ESCUELAS POLITÉCNICAS Y IX CONGRESO INTERNACIONAL DE CIENCIA TECNOLOGÍA EMPRENDIMIENTO E INNOVACIÓN SECTEI-ESPOCH 2022

Corresponding Author: F Moreira Moreira; email: fmoreira@espam.edu.ec

Published: 9 November 2023

Production and Hosting by Knowledge E

Abstract

The purpose of this document is to describe the use of a natural language processing model in the multiplatform system “Gamivity” by means of a sentence similarity algorithm to offer a personalized experience module based on the conceptual relationship between questions. For the selection process, certain criteria were chosen that will allow several pre-trained models under the “Transformers” architecture for evaluation, later. These criteria were the language with which the model was altered; Python was the programming language used for the implementation. Regarding the evaluation phase of the selected models, the “Sentence Transformers” library of the Python programming language was used. In addition, a work environment analogous to the module present in the “Gamivity” system was built, in which the development platform “Google Colab” was used to test these models. The criteria for choosing the candidate model were based on its effectiveness in relation to questions as well as the computational cost involved while performing the operations in the said model Based on the applied methodology, the model that yielded the best results was “paraphrase-multilingual-MiniLM-L12-v2,” modified with a large corpus of text in Spanish and 50 other languages, which showed a degree of precision. When it comes to conceptually relating the questions provided it was found to be optimal, having relatively low computational cost when performing these operations.

Keywords: *sentence transformers*, *sentence similarity, relate questions, personalized learning*.

Resumen

El presente documento, tiene como propósito el de describir la utilización de un modelo de procesamiento de lenguaje natural en el sistema multiplataforma “Gamivity”, por medio de un algoritmo de similitud de oraciones para ofrecer un módulo de experiencia personalizada a partir de la relación conceptual entre preguntas. Para el proceso de selección, se establecieron ciertos criterios que permitieron elegir varios modelos pre entrenados bajo la arquitectura “Transformers” para su posterior evaluación. Dichos criterios, fueron el idioma con el que fue entrenado el modelo, así como que el lenguaje de programación utilizado para la implementación fuese Python. En lo que concierne a la fase de evaluación de los modelos seleccionados, se hizo uso de la biblioteca “Sentence Transformers” del lenguaje de programación Python, además se construyó un entorno de trabajo análogo al módulo presente en el sistema “Gamivity”, en la plataforma de desarrollo “Google Colab” para poner a prueba dichos modelos, los criterios para la elección del modelo candidato, se resumen en la eficacia a la hora de relacionar preguntas, así como el coste computacional a la hora de realizar las operaciones involucradas en dicho proceso. A partir de la metodología aplicada, el modelo que mejor resultados generó fue “paraphrase-multilingual-MiniLM L12-v2”, entrenado con un gran corpus de texto en español, así como de otros 50 idiomas, el cual mostró un grado de precisión óptimo a la hora de relacionar conceptualmente las preguntas proporcionadas, así como su relativo bajo coste computacional a la hora de efectuar dichas operaciones.

Palabras Clave: *sentence transformers, sentence similarity, relacionar preguntas, aprendizaje personalizado.*

1. Introducción

Education transforms lives, and as established by UNESCO [1], it is a human right for all, throughout life, and access to instruction must be accompanied by quality. Education is a complex process in which the condition and nature of humanity [2] must be considered to establish its purpose and definition. Additionally, education plays a key role in providing people with the knowledge, skills, and competencies necessary to participate effectively in society [3]. Educating, then, is the process of facilitating learning through communication, teaching, or research.

Today, learning is considered a social activity that allows humans to acquire skills, abilities, or knowledge [4]. It must be personal, meaning that one cannot learn something well without first understanding its relevance and how it can be reflected in our personal situation. Additionally, it must be considered that each student acquires knowledge at different rates. For this reason, the student must actively involve themselves in the learning process while using strategies such as personalized learning, allowing the student to better assimilate their learning and increase their interest in the education they receive [5].

There are various ways to implement personalized learning, and one of them is to use applications or tools that support computer-assisted teaching and learning processes to build smarter systems. The term “intelligent” used in these systems is determined primarily by their continuous adaptability to the characteristics of the learning and knowledge of different users [6].

This is where neural networks in the subarea of natural language processing (NLP) play a crucial role in personalized learning, as most current research in this field revolves around a type of architecture known as “Transformers” [7], which have become the dominant paradigm for tasks such as natural language understanding and generation, surpassing even the efficiency of models based on convolutional and recurrent networks [8].

Transformer-type neural networks work mainly by receiving a sentence as input and converting it into two sequences. The first, a sequence of word vector embeddings, and the second, a sequence of positional encodings [9]. Word vector embeddings are a numerical representation of the text since it is essential to convert words to the embedded representation so that a neural network can process them. Positional encodings are a vector representation of the position of the word in the original sentence [10].

Considering the points described above, the main objective of this work is to implement a suitable natural language processing model that can be used in the multi-platform system "Gamivity". This will allow the incorporation of a personalized experience module that can be applied in the question relationship of the multi-platform system for the creation of personalized tests based on incorrectly answered questions. The purpose of this is to provide students who use this platform with a positive impact on their academic performance, contributing to an improvement in the quality of education, and generating a reference for best practices for other academic programs [11].

2. Materials and Methods.

Initially, several pre-trained models under the "Transformers" architecture were selected for further evaluation. These models were retrieved from the open-source artificial intelligence platform "Hugging Face" [12]. They are part of the Python framework "Sentence Transformers" [13], which is designed to create state-of-the-art embeddings for sentences, text, and images. For this selection, the following criteria were established that the implementation of the models had to meet, which can be summarized as follows: base model created under the "Transformers" architecture, trained on a corpus that includes at least the Spanish language, and they had to work under the Python library "sentence_transformers".

Consequently, to evaluate the selected models, a working environment was built on the development platform "Google Colab" [14]. This consists of an environment like what would be present in the core of the "Gamivity" system, under the same parameters and libraries that condition the functionalities associated with the personalized learning module. The main difference between this analog environment and the real one to be implemented is the way in which the data is supplied, since in the former, seed data is used, specifically a body of subject questions, which simulate those that would come through calls to the corresponding database.

3. Results and Discussion

3.1. Flowchart of the Gamivity system.

Next, we present in general terms how it is intended to implement the personalized learning module within the multi-platform system "Gamivity". To do this, as shown in (Figure **1**), starting from the questions manually added by the teacher, these would be supplied to the corresponding pre-trained NLP model in order to associate them with other existing questions in the general repository of questions. Furthermore, as shown in (Figure **2**), these associations are made so that the personalized learning controller can perform its basic operations, which can be summarized as building a personalized test segmented by classes made up of those questions related to the ones the student in question has failed.

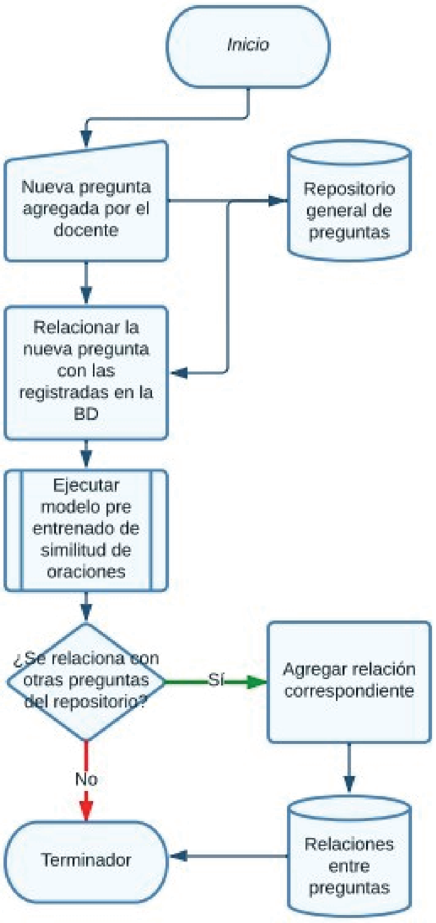

Inicio
Nueva pregunta agregada por el docente
Repositorio general de preguntas
Relacionar la nueva pregunta con las registradas en la BD
Ejecutar modelo pre entrenado de similitud de oraciones
¿Se relaciona con otras preguntas del repositorio?
Sí
Agregar relación correspondiente
No
Terminador
Relaciones entre preguntas

Figure 1

Flowchart of the "Gamivity" system – Aggregation of relationship between questions.

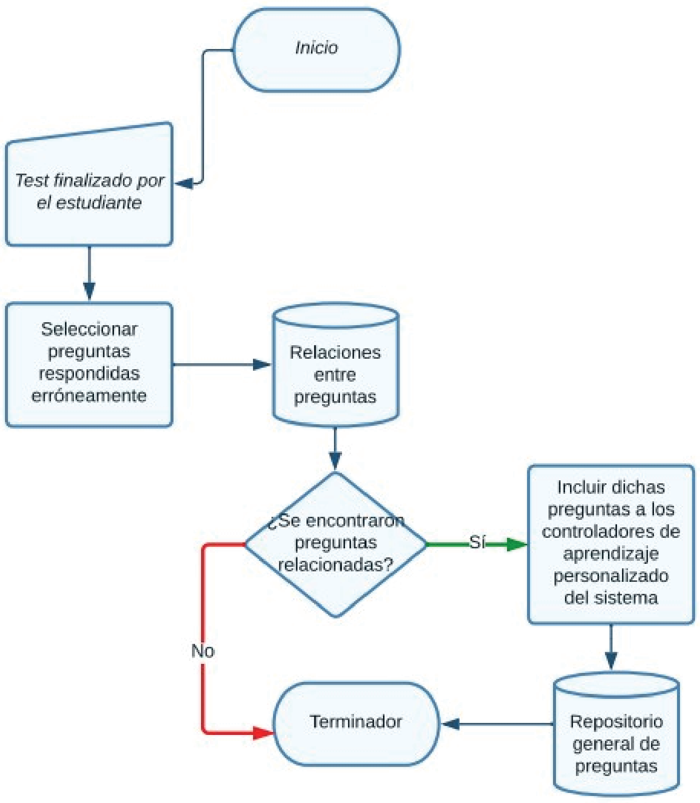

Inicio
Test finalizado por el estudiante
Seleccionar preguntas respondidas erróneamente
Relaciones entre preguntas
¿Se encontraron preguntas relacionadas?
Sí
No
Incluir dichas preguntas a los controladores de aprendizaje personalizado del sistema
Repositorio general de preguntas
Terminador

Figure 2

Flowchart of the "Gamivity" system – Personalized Learning Controller.

3.2. Selection of pre-trained models.

For the selection of models to evaluate, several aspects were considered, specifically that they were trained with corpora that include the Spanish language, as well as each of these models specializing in the task of calculating sentence similarity. Below are listed the different implementations of "Transformers" type base models selected. It should be noted that all of these were trained with a multi-language corpus [15].

1. paraphrase-multilingual-MiniLM-L12-v2
2. distiluse-base-multilingual-cased-v1
3. paraphrase-multilingual-mpnet-base-v2
4. paraphrase-MiniLM-L3-v2
5. clip-ViT-B-32-multilingual-v1

3.3. Testing environment of the selected models.

For the design of the testing environment similar to the one present in the personalized learning module of the "Gamivity" system, a body of questions was created regarding three subjects related to the computer science field with associated subtopics. Specifically, these subjects were programming in Python, digital electronic logic gates, and file systems in operating systems, as shown in (Table I). It should be noted that this body of questions contains fifteen records for each subtopic, which means a total of forty-five questions along with their correct answers.

Table 1

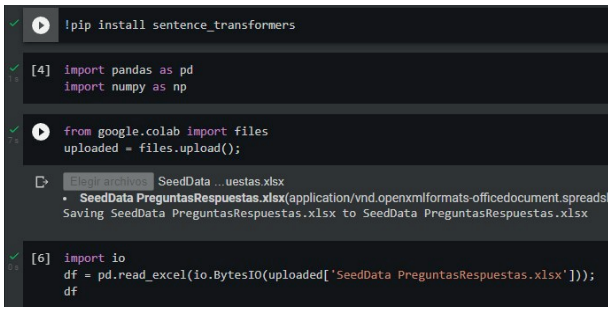
Sample of the question body used for the tests.

Python programming	**Digital Electronics (Logic gates, basic circuits)**	**Operating systems (Basic topics, File systems)**
What are generators in Python? - Functions that return an iterable set of elements	Which logic gate itself is a combinational circuit? - XOR	A file system is made up of: - Files (files) and Directories(folders)
How can you generate random numbers in Python? - import random random.random	What is the number of outputs of a full adder circuit? - Three	What does the partition table indicate? - The beginning, size and type of each partition
Is Python case sensitive? - Yes, he is sensitive.	What is the expression for the sum of a half adder if the inputs are A and B? -A XOR B	What do plain text files contain? - Only printable characters

In order to establish which pre-trained model performs best for the intended purpose, a simple criterion was chosen in which the correct association between questions carries the most weight, followed by the computational cost. The first part of this selection criterion consists of choosing three questions from the table, one question per theme, and associating it mostly with the other questions in the same column through cosine similarity calculation. That is, the model that generates the table that is most like the original based on this calculation will be the one elected according to this criterion. Lastly, in terms of the computational cost of the models, it is calculated based on the average execution time between the processes involved.

Regarding the coding of this environment, first, as shown in (Figure **3**), the "sentence_transformers" library is installed and the table with the questions that will be provided for the tests is loaded. Consequently, the respective models are loaded to map the questions to embeddings, as indicated in (Figure **4**). Once the model is loaded, the cosine similarity between the pivot question and the other questions is calculated, and this is the crucial part in which the execution time of the model is calculated, as shown in (Figure **5**). In this part, it is also defined how the related questions estimated by the model are retrieved, based on the criterion that the ten highest cosine similarity values calculated should correspond to the ten questions of the theme to which the pivot question belongs, provided that the cosine similarity threshold value is greater than 0.25.

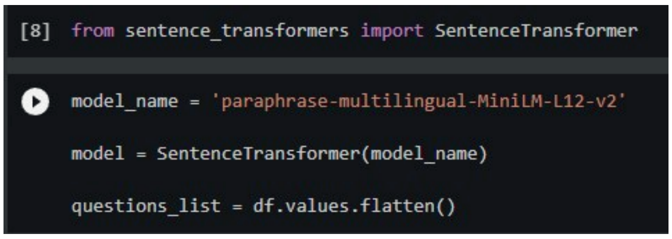
Finally, the corresponding score associated with the model's effectiveness in relating the questions is calculated. This value is defined by the expression: "final score = total associations + correct associations - incorrect associations." The value of the term "total associations" corresponds to a constant associated with the number of questions per theme, which is 15. The value of the variable "correct associations" refers to the number of questions that the model correctly related, that is, those that belong to the real theme, while the variable "incorrect associations" refers to those that do not. Of course, the value of the final score is maximum (30) when the value of "correct associations" is equal to 15.



```
!pip install sentence_transformers

[4] import pandas as pd
    import numpy as np

from google.colab import files
uploaded = files.upload();

Elegir archivos SeedData ...uestas.xlsx
• SeedData PreguntasRespuestas.xlsx(application/vnd.openxmlformats-officedocument.spreads
Saving SeedData PreguntasRespuestas.xlsx to SeedData PreguntasRespuestas.xlsx

[6] import io
    df = pd.read_excel(io.BytesIO(uploaded['SeedData PreguntasRespuestas.xlsx']));
    df
```

Figure 3

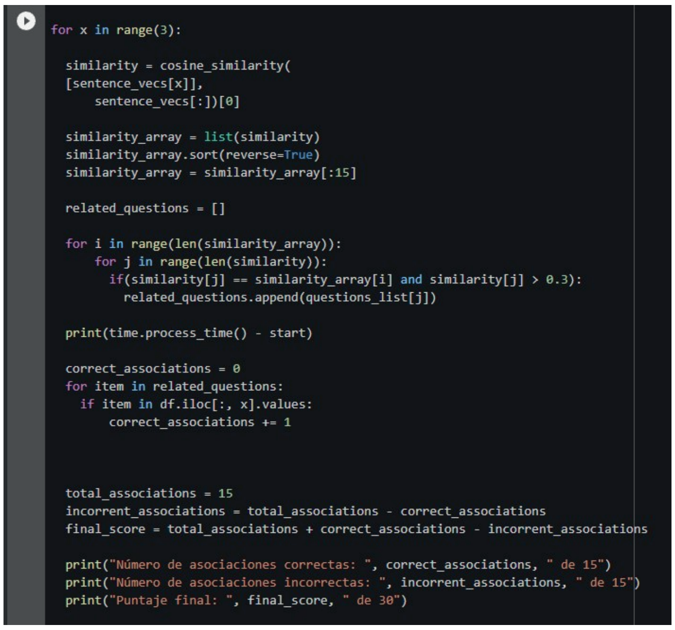
Installation/import of libraries and body of questions.



```
[8] from sentence_transformers import SentenceTransformer

model_name = 'paraphrase-multilingual-MiniLM-L12-v2'

model = SentenceTransformer(model_name)

questions_list = df.values.flatten()
```

Figure 4

Loading the models.



```
for x in range(3):

  similarity = cosine_similarity(
  [sentence_vecs[x]],
      sentence_vecs[:])[0]

  similarity_array = list(similarity)
  similarity_array.sort(reverse=True)
  similarity_array = similarity_array[:15]

  related_questions = []

  for i in range(len(similarity_array)):
      for j in range(len(similarity)):
        if(similarity[j] == similarity_array[i] and similarity[j] > 0.3):
          related_questions.append(questions_list[j])

  print(time.process_time() - start)

  correct_associations = 0
  for item in related_questions:
    if item in df.iloc[:, x].values:
        correct_associations += 1



  total_associations = 15
  incorrent_associations = total_associations - correct_associations
  final_score = total_associations + correct_associations - incorrent_associations

  print("Número de asociaciones correctas: ", correct_associations, " de 15")
  print("Número de asociaciones incorrectas: ", incorrent_associations, " de 15")
  print("Puntaje final: ", final_score, " de 30")
```

Figure 5

Calculation of the cosine similarity, execution time and the final score between models.

3.4. Final evaluation of the models

As shown in Table II, the selected models are listed along with their effectiveness score in relating questions for the three provided pivot questions, as well as the average execution time for the calculation. It should be noted that this average execution time was calculated based on six code executions for each model evaluation continuously. According to these calculations, the model that had the best effectiveness in relating questions was the first one listed, “paraphrase-multilingual-MiniLM-L12v2”. However, the one that showed the shortest average execution time was “paraphrase-MiniLML3-v2”.

4. Discussion

Based on the results presented, specifically in the diagrams of the “Gamivity” system, two main reasons are proposed as to why this work focused on these types of networks mentioned earlier and not on others such as the set of Word2Vec [16] or Glove [17] architectures. The first reason is that the word embeddings of the Word2Vec and Glove models are context-independent, since they generate only one vector/embedding for each word, combining all the different meanings of the word into a single vector [18]. This means that, for a given word, the associated contexts will be represented in a diffuse manner. Therefore, the main difference between these types of models is that both Word2Vec and Glove do not consider the order of words in their training, whereas

Table 2

Final evaluation.

Model	Score T1	Score T2	Score T3	Average Score (max. 30)	Average execution time
paraphrase-multilingual-MiniLM-L12-v2	20	22	24	22	1.219
distiluse-basemultilingual-cased-v1	22	22	16	20	1.7824
paraphrase-multilingual-mpnetbase-v2	20	18	22	20	3.2316
paraphrase-MiniLM-L3-v2	18	18	22	19.33	0.489
clip-ViT-B-32multilingual-v1	20	20	24	21.33	2.273

models based on “Transformers” do take into account this order and therefore represent a significant advancement over these networks proposed years ago [19].

On the other hand, the second reason for opting for this type of architecture lies in a technical aspect specific to this system’s implementation. Pre-trained “Transformer” models are quite efficient when it comes to performing tasks such as calculating the similarity between sentences [19], while Word2Vec or Glove models were initially designed to calculate similarity between words [16]. This means that if the system were to use one of these models, additional processes would have to be implemented to calculate the similarity between more complex sentences (in this case, questions); these processes could include the inclusion of a machine learning algorithm of the “keyword extraction” type to extract the keywords of a sentence and compare them with the keywords of another sentence [20]. In other words, “Transformer” networks are more convenient in this aspect because the calculations based on pre-trained models would be more direct and, in principle, less computationally expensive.

5. Conclusions

1. The implementation of an NLP model in the multiplatform system “Gamivity” is important for it to make the respective relationships between the questions belonging to the general question repository, thus being able to fulfill the objective of generating personalized tests based on questions that students have answered incorrectly.

2. Although there are countless pre-trained NLP models, the values obtained in the tests indicate that those that best adapted to the proposed needs were of the “Transformer” type, as they are efficient in understanding natural language, especially when estimating the similarity between sentences.
3. It was necessary to test the selected models in order to choose the model that generated the best results and had the lowest computational time. This was done to simulate from a high level what would be present in the personalized learning module in a real production environment of the final application.
4. Once the respective evaluation of the models was carried out, it was determined that the “paraphrase-multilingual-MiniLM-L12-v2” model obtained the best average efficiency when relating the questions. However, considering that this task in the application must be performed every time a teacher adds a new question, the execution time to complete it must be as minimal as possible. For this reason, the “paraphrase-MiniLM-L3-v2” model, which showed an average execution time up to 2.5 times shorter than the previously mentioned model, was established as the candidate model. Although this model was faster, it had a lower final score when associating the questions. It was established that the difference between them was not significant, especially considering that the body of questions used for these tests was not too large compared to the number of questions that would be present in a production environment of the application.

References

[1] UNESCO. “La educación transforma vidas”, UNESCO, 2021. https://es.unesco.org/themes/education/ (consultado jul. 04, 2022).

[2] León A. Qué es la educación. Scielo. 2017 Dec;11:595–604.

[3] Tu Índice para una Vida Mejor [Internet]. Oecdbetterlifeindex.org. 2015. Available from: https://www.oecdbetterlifeindex.org/es/topics/education-es/

[4] Roque Y, Sánchez A, López A, Fernández A, Moura D. Entorno de Aprendizaje Personalizado (PLE) para la asignatura de Investigación de Operaciones en Ingeniería Agrícola. Revista ciencias técnicas agropecuarias2016 Mar;25:55–59.

[5] UNESCO. “UNESCO Digital Library”, Aprendizaje personalizado, 2017. https://unesdoc.unesco.org/ark:/48223/pf0000250057_spa.

[6] Maraza B. “Hacia un aprendizaje personalizado en ambientes virtuales”, Campus virtuales: revista científica iberoamericana de tecnología educativa. 2016;20-29, 2016, [En línea]. Available: http://hdl.handle.net/11162/120601

[7] Muñoz M. Estudio de nuevos modelos de Deep Learning para el análisis y comprensión de grandes cantidades de datos. Zaragoza: Universidad Zaragoza; 2019.

[8] Vaswani A, et al., "Tensor2Tensor for Neural Machine Translation". arXiv, 2018. doi: 10.48550/ARXIV.1803.07416.

[9] Afán de Ribera I, Blanco G, Díaz V, Jiménez A. "Utilización de modelos de Transformers en la gestión de respuestas a preguntas en PLN. Aplicaciones a asistentes conversacionales". Colegio Universitario de Estudios Financieros. Madrid: CUNEF; 2021.

[10] Santander C. "Transformers: Redes Neuronales", Linkedin, 2021. https://www.linkedin.com/pulse/transformers-redes-neuronales-cristian-santander/

[11] Pérez Guerrero J, Ahedo Ruiz J. "La educación personalizada según García Hoz". Revista Complutense de Educación, vol. 31, núm. 2;153–161. abr. 2020, doi:10.5209/rced.61992.

[12] Hugging Face. Hugging Face – On a mission to solve NLP, one commit at a time. [Internet]. huggingface.co. Available from: https://huggingface.co/

[13] Reimers N, Iryna Gurevych. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. 2019 Aug 27.

[14] Google Colaboratory [Internet]. colab.research.google.com. [cited 2023 Jul 4]. Available from: https://colab.research.google.com/notebooks/intro.ipynbenvenida

[15] Reimers N, Iryna Gurevych. Making Monolingual Sentence Embeddings Multilingual using Knowledge Distillation. 2020 Apr 21.

[16] Mikolov T, Chen K, Corrado G, Dean J. "Efficient Estimation of Word Representations in Vector Space". arXiv, 2013. doi: 10.48550/ARXIV.1301.3781

[17] Pennington J, Socher RCDD. Manning, "Glove: Global vectors for word representation", en Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), 2014;1532–1543.

[18] Language Models and Contextualised Word Embeddings [Internet]. www.davidsbatista.net. Available from: https://www.davidsbatista.net/blog/2018/12/06/Word_Embeddings/

[19] Vaswani A, et al. Attention is all you need. Adv Neural Inf Process Syst. Volume 30. 2017.

[20] Firoozeh N, Nazarenko A, Alizon F, Daille B. Keyword extraction: Issues and methods. Natural Language Engineering. 2019 Nov 11;26(3):259–291.