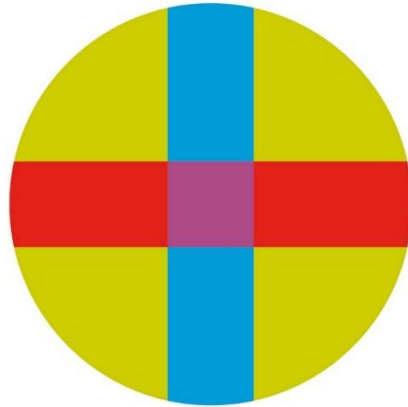


UNIVERSITY CEU - SAN PABLO

POLYTECHNIC SCHOOL

BIOMEDICAL ENGINEERING DEGREE



BACHELOR THESIS

**LARGE LANGUAGE MODEL OF BIOMEDICAL
CONCEPTS EXTRACTED FROM SCIENTIFIC
LITERATURE**

Author: Cristina Jiménez de Abadal
Supervisors: Carlos Óscar Sorzano, Javier Tejedor

July 2024



Datos del alumno

NOMBRE:

Datos del Trabajo

TÍTULO DEL PROYECTO:

Tribunal calificador

PRESIDENTE:

FDO.:

SECRETARIO:

FDO.:

VOCAL:

FDO.:

Reunido este tribunal el ____/____/____, acuerda otorgar al Trabajo Fin de Grado presentado por Don _____ la calificación de _____.

ACKNOWLEDGMENTS

Agradezco a Carlos Óscar, por su paciencia y su inmensa ayuda a lo largo del proyecto.

A el CEU y a su profesorado, en especial a Javier, por su ayuda a lo largo de mis estudios.

A mi familia por su apoyo y ayuda constante.

Y a mis compañeros por su amistad y cariño.

ABSTRACT

This thesis introduces a tool aimed at aiding medical professionals in symptom analysis and diagnosis among other things by employing natural language processing techniques. The program initiates by parsing scientific literature to extract relevant information, which is subsequently embedded for the model's learning process through triplet loss. Utilizing advanced NLP (Natural Language Processing) techniques, particularly a Siamese model, textual descriptions of patient symptoms provided by physicians are processed, enabling the generation of informative links to scientific resources. The program's ability to seamlessly integrate bilingual resources (both English and Spanish) is an important factor, as it enhances its effectiveness across various linguistic environments. Visualizations in the form of scatterplots further enhance understanding by depicting the relationships between input symptoms and relevant cases. This program represents a significant advancement in leveraging artificial intelligence to augment clinical decision-making, offering a promising avenue for improving patient outcomes and advancing medical practice.

RESUMEN

Este trabajo presenta una herramienta para apoyar el análisis y diagnóstico de síntomas médicos. Utiliza técnicas avanzadas de procesamiento de lenguaje natural, como un modelo Siamese, para analizar literatura científica y extraer información relevante. Esta información se incorpora al modelo para su aprendizaje. Los profesionales médicos ingresan descripciones de síntomas, y el programa genera enlaces a recursos científicos. Este presenta capacidades multilingües, ampliando su utilidad en diversos contextos lingüísticos. Las visualizaciones, como gráficos de dispersión, clarifican las relaciones entre síntomas y casos. En resumen, el programa es un avance significativo en la inteligencia artificial aplicada a la toma de decisiones clínicas, prometiendo mejorar los resultados para los pacientes y avanzar en la práctica médica a nivel global.

INDEX

1 INTRODUCTION	1
1.1 MOTIVATION	1
1.2 OBJECTIVES	2
1.3 THESIS OUTLINE	3
2 ARTIFICIAL NEURAL NETWORKS AND TRIPLET LOSS	4
2.1 UNSUPERVISED LEARNING	4
2.2 NEURAL NETWORKS	4
2.2.1 Siamese Neural Network	6
2.3 TRIPLET LOSS	8
3 MATERIALS AND METHODS	11
3.1 PARSING OF HTMLS AND PDFS	13
3.1.1 Parsing of HTMLs	13
3.1.2 Parsing of PDFs	16
3.2 EMBEDDING OF THE PHRASES	16
3.3 SEMANTIC EMBEDDING	17
3.4 SEMANTIC SEARCH	19
4 RESULTS	21
4.1 QUANTITATIVE RESULTS: SIMPLE INPUT QUERY	23
4.2 QUANTITATIVE RESULTS: ADVANCED INPUT QUERY	25
5 DISCUSSION	27
5.1 INPUT PROCESSING AND ANALYSIS	27
5.2 GENERATION OF RELEVANT LINKS	27
5.3 INTEGRATION OF MULTILINGUAL RESOURCES	27
5.4 VISUALIZATION THROUGH SCATTERPLOTS	28
5.5 PRECISION RESULT EVALUATION	28
5.6 LIMITATIONS	29
6 CONCLUSIONS	30
7 REFERENCES	32

FIGURE INDEX

FIGURE 1 NEURON ANATOMY	5
FIGURE 2 ARTIFICIAL NEURON	5
FIGURE 3 NEURAL NETWORK.....	6
FIGURE 4 ARCHITECTURE OF A SIAMESE NEURAL NETWORK WITH TRIPLET LOSS	8
FIGURE 5 REPRESENTATION OF TRIPLET LOSS OBJECTIVE	9
FIGURE 6 STEPS DIAGRAM	12
FIGURE 7 HTML STRUCTURE.....	14
FIGURE 8 WEBSITE EXAMPLE	14
FIGURE 9 FINAL TEXT FILE CREATED FROM PARSING.....	15
FIGURE 10 VISUAL REPRESENTATION OF AN EMBEDDING PROCESS	17
FIGURE 11 LOSS FUNCTION REPRESENTATION THROUGH EACH BATCH	19
FIGURE 12 USER INTERFACE INTERACTION	21
FIGURE 13 PROGRAM URL RESPONSE.....	21
FIGURE 14 WEBSITE EXAMPLE FROM ONE OF THE URL RESULTS FOR “ABDOMINAL PAIN”	22
FIGURE 15 SCATTERPLOT PRODUCED FOR “ABDOMINAL PAIN”	22
FIGURE 16 SCATTERPLOT PRODUCED BY “SWOLLEN ANKLES”	23

TABLE INDEX

TABLE 1 DIFFERENT USER TEXTS AND TRUE POSITIVES ASSOCIATED 24

1 INTRODUCTION

1.1 Motivation

In the world of healthcare, making well-informed decisions holds the uppermost importance, whether it is diagnosing a patient, selecting treatment options, or producing other critical decisions that significantly impact patient outcomes. However, the field of medicine is vast and continuously evolving, with new discoveries and insights emerging over time. Despite the immense wealth of knowledge available, it is humanly impossible for healthcare professionals to retain and recall every detail, especially in areas outside their specialized domains. In addition, there are times where a disease needs to be quickly identified to start treatment or it can impact negatively on the patients' health [1].

Recognizing this limitation, there is a growing recognition of the need for tools and platforms that can support physicians in decision-making processes. These tools have the potential to minimize or even eliminate medical errors, reduce costs, and enhance both patient and physician satisfaction. By providing a platform capable of integrating a vast amount of information extracted from scientific literature and providing evidence-based recommendations, medical professionals can take it into account to make better-informed decisions, thus, providing a better quality of care to the patients [1].

For instance, imagine a scenario where a physician encounters a patient presenting with a series of symptoms. Instead of relying solely on memory or manual literature searches, the physician can employ a decision-support platform. By adding the patient's symptoms into the platform, the system can quickly retrieve, and present relevant findings and treatment recommendations derived from a comprehensive analysis of scientific literature.

The development and use of decision-support platforms represents a promising advancement in healthcare, offering a valuable resource for physicians to deal with the complexity of medical decision-making effectively. By leveraging technology to augment clinical expertise, these platforms have the potential to enhance patient outcomes, optimize resource utilization, and ultimately, advance the quality of healthcare delivery.

1.2 Objectives

This project aims to develop a tool with the objective of improving medical practice by providing assistance to healthcare professionals in symptom analysis and diagnosis. Essentially, the project aims to create a comprehensive software tool capable of processing textual descriptions such as symptoms and returning a series of URLs to websites related to the input with accuracy and efficiency.

To achieve this ambitious goal, the project will leverage advanced Natural Language Processing (NLP) techniques, including the integration of a Siamese model. This will include:

- Development a tool capable of extracting relevant information from scientific literature and loading it into text files.
- Embedding of the information into computer readable values through NLP techniques.
- Training of the neural network with the embedded data and re-embed it through the trained model.
- Obtain a tool that correctly interprets an input text and gives a proper answer.

Through continuous learning and refinement, the tool aims to evolve into a trusted ally for healthcare professionals, offering support in understanding the complexities of medical diagnosis and treatment.

In addition, it is essential to emphasize that the program does not aim to replace the expertise of healthcare professionals, but stands as a tool designed to complement and enhance the decision-making process, ultimately leading to better patient outcomes, reduced medical error and improved healthcare delivery.

Overall, this project represents an advance of medical informatics, promising to reshape the dynamics of healthcare delivery and help doctors make better and more informed decisions about different aspects of medicine as well as provide information to those who search.

1.3 Thesis outline

This document is organized as follows:

- Section 1: Introduction. It describes the motivation of the development of the tools and presents the objectives for the development of the project.
- Section 2: Neural Networks and Triplet Loss. It provides an overview of what Neural Networks are and introduces Siamese Neural Networks, the one that will be implemented in this project. It will also introduce triplet loss, the loss function implemented in this network as the learning method of the model.
- Section 3. Materials and Methods. It provides a detailed description of the process that has been done to develop the tool. This includes the parsing of the data, embedding, training of the model, and finally a tool that can return a series of URLs related to an input value.
- Section 4. Results. It provides an overview of the tool and how it works: What the user will see and what type of response the program will deliver.
- Section 5. Discussion. Various topics will be mentioned: The results of the model, including a brief analysis of the precision of the model. Some limitations that have appeared during the project will be mentioned.
- Section 6. Conclusions. This will provide a final summary of the projects, including the different objectives, a brief summary of the limitations that have appeared and interesting future work that could be done.

2 ARTIFICIAL NEURAL NETWORKS AND TRIPLET LOSS

For the project we had to consider different aspects of machine learning, such as whether to implement supervised or unsupervised learning and what type of neural network to use.

Machine learning is a method utilized for computers to learn from data and be able to improve continuously. Through the process of machine learning two main learning approaches can be implemented, supervised learning and unsupervised learning [1].

2.1 Unsupervised Learning

The two main learning approaches in machine learning are supervised learning and unsupervised learning. The main difference between these two is that in supervised learning an existing pattern of data is given to the model to then use with new data while in unsupervised learning the model must find this pattern in the data without any previous indication [2].

In unsupervised learning, it is unclear what type of dataset is being worked with and so it is not possible to provide a training dataset or label the data. The main objective of this type of learning is for the model to discover the hidden patterns that are within the data without the intervention of an exterior party [1].

2.2 Neural Networks

Neural Networks are inspired by how the brain works. The brain is capable of completing a wide range of complex tasks and has a structure that allows them to process information while also transmitting orders to the rest of the body at the same time [3].

The brain and nervous system are composed of cells called neurons, which are the ones in charge of receiving, processing and transmitting information. They are composed of three main parts, the cell body, also known as the soma, the dendrites and the axon. Each of these parts completes an important role in the transmission of information [3].

The cell body, or soma, is located in the nucleus, and regulates essential cellular functions. Branching out from the soma are dendrites, short projections specialized in receiving input signals from neighboring neurons. These signals, once received, are processed, and integrated within the neuron. And lastly, the axons, typically longer than dendrites, serve as conduits for transmitting signals, known as action potentials, in response to the received inputs. The connection from one neuron to another is known as synaptic terminals or synapses [3]. This can be seen in Figure 1.

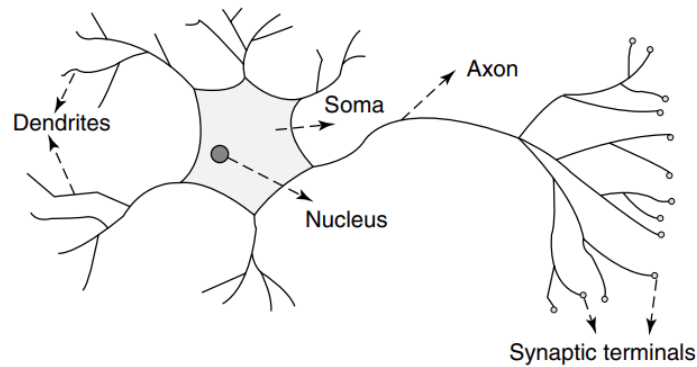


Figure 1 Neuron Anatomy [3]

In the context of neural networks, an artificial neuron is also known as a node. The connections to from one node to another (which can also be called as a synapses) are represented through connection weights that are determined by the association of input signals and the features of the nodes are represented by a transfer function [3]. So, the artificial neural network learns by adjusting the input weights through the transfer function [3]. Figure 2 shows an artificial neuron that receives these weights and, through the transfer function, develops a corresponding output.

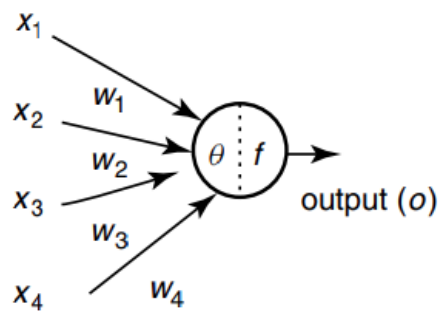


Figure 2 Artificial neuron [3]

The architecture of a neural network is divided into three layers, the input layer where the network first receives the data from the exterior, the hidden layer, and the output layer where the final results are provided. The number of hidden layers is determined by the user, and as the number of hidden layers in a neural network increases the more complex the network will be created [1].

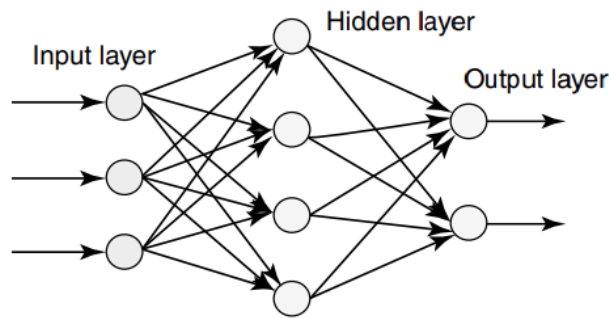


Figure 3 Neural Network [3]

As mentioned earlier, this project will use a unsupervised learning approach. This means that an unlabeled input vector will be given to each input node. The network with this information will then adjust its weights taking into account the training algorithm and the hidden patterns withi the data and finally create its own function capable of determining the output from the input vectors [3].

2.2.1 Siamese Neural Network

For the development of the model, a Siamese Neural network was employed. The Siamese Neural Networks (SNNs) were proposed in 1993 as a solution to a problem that was arising in methods where similarity was measured. This network is ideal when dealing with problems that involve measuring similarities between two concepts [4].

A Siamese Network refers to a specific architecture where two identical neural networks (often referred to as "twins" or "branches") share the same parameters and weights and are trained simultaneously on two different inputs. In the final step of the process, the outputs obtained are compared employing a distance metric system, in this

case Euclidean distance. Similar outputs will have a distance closer to 0 while those considered different will have a distance value closer to 1.

In this model, the input parameters that are introduced into the network can either be pairs or triplets. The network that accepts two input values can also be referred to as a “Twin Neural Network” where the paired elements are compared and the network learns if these are similar or dissimilar through a loss function, commonly the contrastive loss function [4].

The network that accepts triplets can also be called “Triplet Networks” which will be the one employed for this project. Instead of having a pair we will have three input elements. The first element will be compared to the other two, one being a similar element from the same origin and the other a dissimilar element from a different origin. Through this method the model learns to differentiate and classify these different elements. The loss function in these type of network is called triplet loss [4].

In Figure 4 we can see a diagram of how the network works. The Siamese network accepts three inputs (anchor, positive, negative) and during the learning method, it adjusts its weights employing triplet loss as its loss function.

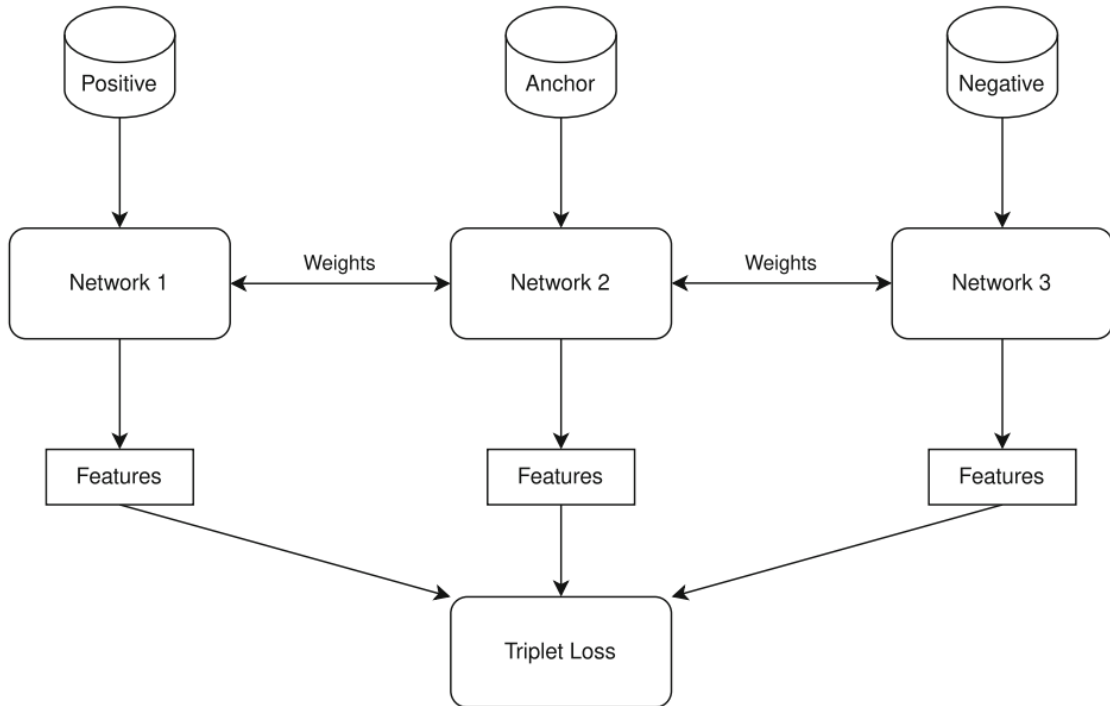


Figure 4 Architecture of a Siamese Neural Network with triplet loss [4]

2.3 TRIPLET LOSS

The learning method chosen for the neural network model was triplet loss. Triplet loss is a loss function commonly used in neural network models for learning embeddings, particularly in tasks such as image recognition, facial recognition, and information retrieval. It is especially relevant in scenarios where the goal is to learn embeddings that capture semantic similarity or dissimilarity between data points, such as images, texts, or other high-dimensional data, such as what is being done in the development of this project [5].

In the context of our neural network model dealing with vectors representing queries based on scientific literature, triplet loss can be a powerful learning method. It uses groups of three items called triplets. These triplets consist of an item (anchor, data point for which we want to learn a meaningful embedding representation), a similar item (positive), and a dissimilar item (negative). Each item is an embedding, the anchor and

positive item come from the same “origin” (ex. Article) while the negative comes from a different one [4, 5].

The goal of the loss function is to minimize the distance between the anchor and positive items while maximizing the distance between the anchor and negative items. It is a way for the model to understand the concept of similarities and dissimilarities. The mathematical representation is

$$\sum_i^N [\| f(x_i^a) - f(x_i^p) \|_2^2 - \| f(x_i^a) - f(x_i^n) \|_2^2 + \alpha]$$

Equation 1 Triplet Loss [5]

Where “N” is the number of batches of triplets, “a” refers to the anchor item, “p” refers to the positive item, “n” refers to the negative item, $f(x)$ accepts an input x , and “ α ” refers to the bias.

The first half of the equation tries to minimize the distance between the anchor item and the positive item, which denotes the Euclidean distance between these two. And the second half tries to maximize the distance between the anchor and negative item [5].

The loss function is minimized when the distance between the anchor and the positive sample is smaller than the distance between the anchor and the negative sample by at least a specified margin [5]. This can be seen in Figure 5, where the model tries to create more distance between the negative value and the anchor while trying to decrease the distance between the anchor and the positive value.

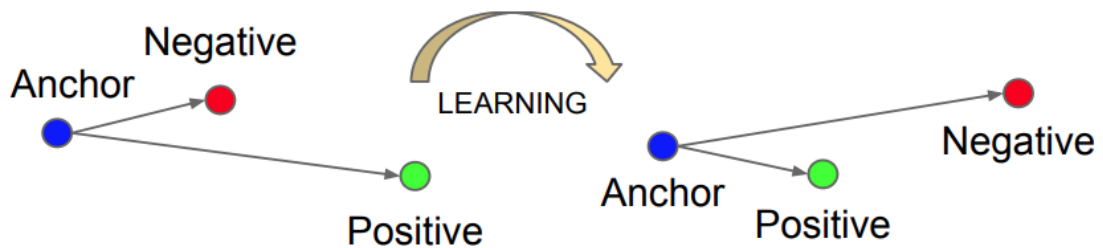


Figure 5 Representation of triplet loss objective [5]

So, by utilizing triplet loss as the learning method for the Siamese neural network model, it aims to learn embeddings that effectively represent queries based on scientific literature in a semantically meaningful way. This can lead to improved performance in tasks such as similarity search, document retrieval, and information organization.

3 MATERIALS AND METHODS

The development of the tool required various steps. Firstly, the data needed to be prepared, which includes extracting it from scientific literature and transforming it into computer readable format. Then, create a neural Network model which then went through a learning process for it to be capable of performing a semantic embedding. And finally provide a program, capable of receiving an input text and conducting a semantic search employing distance search. The code used for this process is uploaded into the GitHub repository as *CrisJimAbadal / LLMBiomedicalConcepts* [6].

Here are the steps in more detail:

- First, the extraction of the information through parsing of HTMLs and PDFs and loading it into text files. By leveraging web scraping tools in Python, relevant scientific literature is extracted and stored by phrases in their corresponding text files.
- Then, the embedding of the information which is crucial to transform it into computer readable data for the model to understand by employing Large Language Model (LLMs) tools. This embedding is multilingual as it will be processing phrases in both English and Spanish, but without the biomedical contextualization.
- Following, the training of the model. It will then be able to semantically embed the vectors, this time the biomedical meaning will be taken into account. The model will consist of a Siamese Neural Network that will learn through a Triplet Loss function and will then be able
- And finally, the creation of a program capable of conducting a semantic search and providing the correct URLs. For this, a FAISS index will be used, all the embeddings will be loaded into it and will then serve as a library capable of finding similar embeddings from an input query using distance metrics, specifically Euclidean distance.

The following diagram provides a visual representation of the steps that will be completed. These are the four main steps, the first two (1, 2) consist of the data preparation, the third (3) consist of the model learning and semantic embedding, and finally the last step (4) conducts a semantic search.

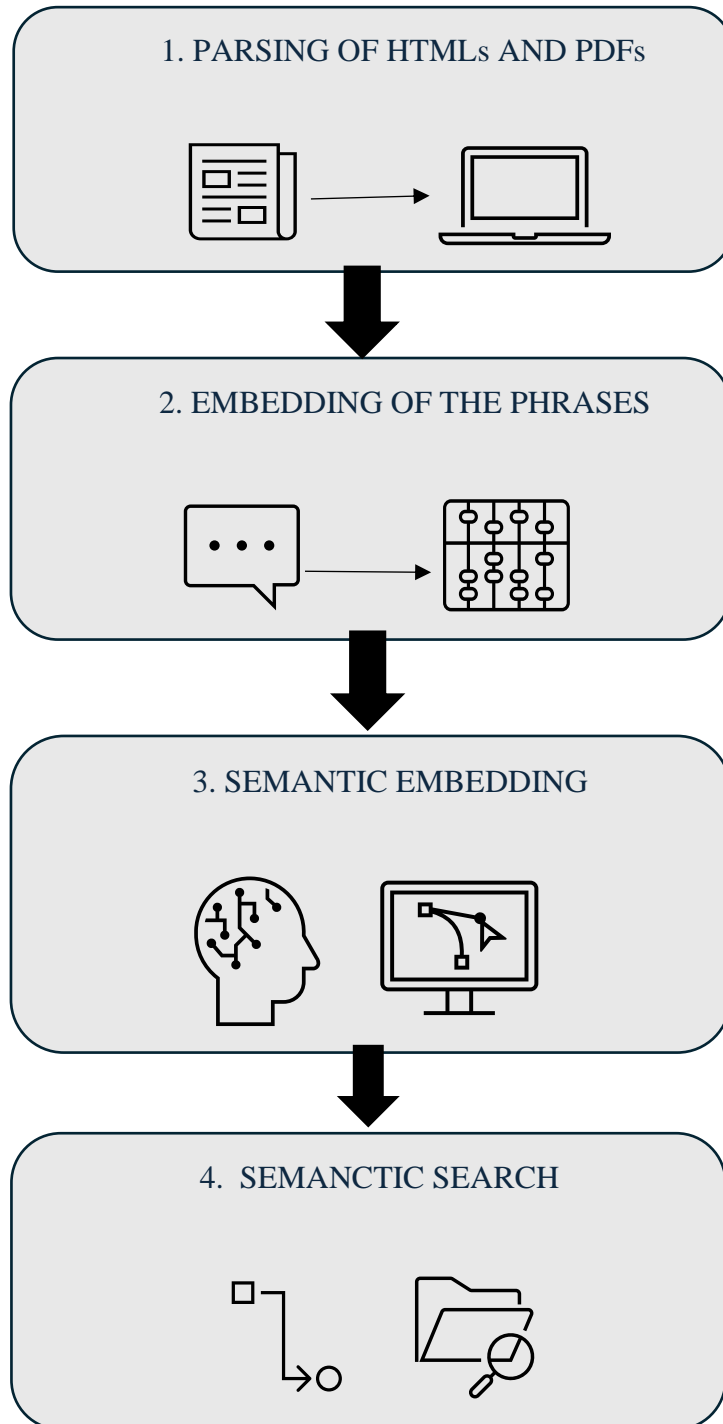


Figure 6 Steps Diagram

3.1 Parsing of HTMLs and PDFs

The first step of the preparation of the data was to extract meaningful information from scientific literature, which will be the parsing section. In this project, we dealt with scientific literature contained in both HTML and PDF formats. So, two different programs were developed for the data extraction, one for the HTMLs and another for the PDFs.

3.1.1 Parsing of HTMLs

In this first step, we conducted a parsing code for the HTMLs. For the extraction of the HTMLs we employed *BeautifulSoup* a web scraping tool, which allows the extraction of data from HTML and XML documents in Python [7]. This specific tool was chosen because of its availability in python including other qualities such as increased performance, portability and accuracy rate compared to other web scraping tools such as Scrapy and selenium [7].

With *BeautifulSoup* we are able to parse HTML code and extract the data from it. We are also able to create a structured object called a “parse tree object” from the HTML code which allows us to parse and process [8].

There were two necessary libraries to be imported for this part of the process. The first library, *os*, a helpful library for dealing with routes and file paths, contains a series of functions that allows us to read and write through the file system. The second library imported was *bs4*, which is especially important as it contains the *BeautifulSoup* tool for parsing the HTMLs.

The next step is to create the parse tree object leveraging the *BeautifulSoup* parser function and extract the necessary information from the html file.

In Figure 6 we can see an example of how HTMLs are structured and how in the formation is stored, and Figure 7 shows how it is seen on the website.

```

</span><span class="hide-offscreen">Expanda sección</span></button></div><div class="sm-live-area hide-offscreen" aria-live="polite">
</div></div><div class="section-body" id="section-1"><p>Muchas personas tienen intolerancia a los alimentos. Este término generalmente se
</span><span class="hide-offscreen">Expanda sección</span></button></div><div class="sm-live-area hide-offscreen" aria-live="polite">
</div></div><div class="section-body" id="section-2"><p>Los síntomas generalmente comienzan en un lapso dos horas después de comer. A veces
sal, goteo nasal
test" href="https://medlineplus.gov/spanish/ency/article/003120.htm">Calambres estomacales</a>, diarrea, náuseas o vómito </li></ul>
</span><span class="hide-offscreen">Expanda sección</span></button></div><div class="sm-live-area hide-offscreen" aria-live="polite">
</div></div><div class="section-body" id="section-3"><p>Algunas veces, se emplean pruebas cutáneas o exámenes de sangre para confirmar qu
</span><span class="hide-offscreen">Expanda sección</span></button></div><div class="sm-live-area hide-offscreen" aria-live="polite">
</div></div><div class="section-body" id="section-4"><p>Si sospecha que usted o su hijo tienen alergia a un alimento, consulte a un espe
</span><span class="hide-offscreen">Expanda sección</span></button></div><div class="sm-live-area hide-offscreen" aria-live="polite">
</div></div><div class="section-body" id="section-5"><p>Los siguientes grupos pueden proveer información sobre las alergias alimentarias:
</span><span class="hide-offscreen">Expanda sección</span></button></div><div class="sm-live-area hide-offscreen" aria-live="polite">

```

Figure 7 HTML structure

Un sitio oficial del Gobierno de Estados Unidos [Así es como usted puede verificarlo](#)

NIH Biblioteca Nacional de Medicina

MedlinePlus
Información de salud para usted

Menú Búsqueda Engl

Página Principal → Enciclopedia médica → Alergia alimentaria

Alergia alimentaria

Es un tipo de respuesta inmunitaria desencadenada por el consumo de huevos, maní, leche, mariscos u otro tipo específico de alimento.

Causas

Muchas personas tienen intolerancia a los alimentos. Este término generalmente se refiere a acidez, cólicos, dolor de estómago o diarrea que pueden ocurrir después de comer alimentos como:

- Productos del maíz
- Leche de vaca y productos lácteos (generalmente debido a la intolerancia a la lactosa)
- Trigo y otros granos que contienen gluten (intolerancia al gluten o celiaquía)

Temas de salud relacionados

Alergia a los alimentos

Imágenes

MiPlato

Anafilaxia

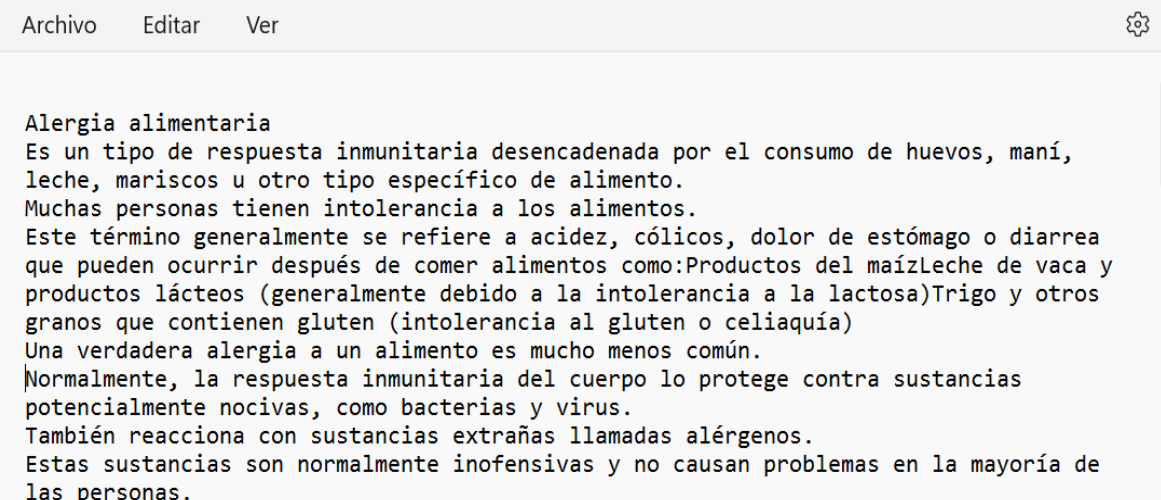
Figure 8 Website example

As we can see from the information provided by Figure 7, HTMLs are distributed through different sections which have their own tags allowing the information to be identified. By comparing it to the website, we can see that, in this example, all the important text is contained in the “p” tags.

With *Beautifulsoup* we are able to create what is called a “parse tree object”. With this object, we are able to do various operations including navigating, searching and modifying the HTML document [8]. To extract the desired text, the html tag in which it is contained has to be previously indicated to know where to find it.

The indicated tag varies depending on the distribution of the HTML file. In this case, the information was contained in the “p” tags, but this can change with different documents. Another method that can be employed to extract the text that can be universally employed is “text = soup.body.get_text()”. With this line of code, all text is extracted. It is beneficial as it works for all HTMLs, but all the text is extracted including the irrelevant data and filtering has to be conducted to eliminate any unwanted text.

The final step would be to store the information into a text file separating each line into phrases, as we can see in Figure 9.



```

Archivo  Editar  Ver  ⚙️

Alergia alimentaria
Es un tipo de respuesta inmunitaria desencadenada por el consumo de huevos, maní,
leche, mariscos u otro tipo específico de alimento.
Muchas personas tienen intolerancia a los alimentos.
Este término generalmente se refiere a acidez, cólicos, dolor de estómago o diarrea
que pueden ocurrir después de comer alimentos como: Productos del maíz, Leche de vaca y
productos lácteos (generalmente debido a la intolerancia a la lactosa), Trigo y otros
granos que contienen gluten (intolerancia al gluten o celiaquía)
Una verdadera alergia a un alimento es mucho menos común.
Normalmente, la respuesta inmunitaria del cuerpo lo protege contra sustancias
potencialmente nocivas, como bacterias y virus.
También reacciona con sustancias extrañas llamadas alérgenos.
Estas sustancias son normalmente inofensivas y no causan problemas en la mayoría de
las personas.

```

Figure 9 Final text file created from parsing

3.1.2 Parsing of PDFs

We also had to create a code that parsed through the files that were in PDF format. For this part, *BeautifulSoup* was not implemented as there was no need. The text was extracted directly from the PDF document, instead.

For the PDF parsing the *fitz* library was imported. This has a module capable of extracting information from PDF documents [9]. With this function, we were able to open the pdf document and extract its contents.

To make sure any unwanted phrases were not saved into the text file, a filtering was conducted where a personalized list of stop words was created containing words such as “web” or “email” that have no medical relevancy but often appeared in the documents. For each phrase extracted, if it contained any of the stop words, the whole phrase was disregarded and was not added into the text file.

3.2 Embedding of the phrases

The second phase of the preparation of data was to embed the phrases extracted earlier into a computer readable format. Embedding is the transformation of non-numerical values, such as phrases or images, into numerical ones that can be interpreted by the computer.

First, we had to import the necessary libraries. For this process *LangChain* and *HuggingFace* were leveraged for the embedding of phrases.

LangChain is an open-source framework available in python that helps in the development of applications that use LLMs [10]. Through *LangChain* we were able to import *HuggingFaceEmbeddings*, an embedding model of *HuggingFace*. It allows to turn the textual phrases into numerical values and has several properties such as capturing the semantic meaning of the embeddings or the contextual information [11].

In figure 10 we can see a possible way to create sentence representations by using this framework. We can take advantage of the embedding of each word and calculate the embedding of the whole sentence based on those. As we can see in the figure, the whole sentence is embedded by words and then a high-dimensional vector is created.

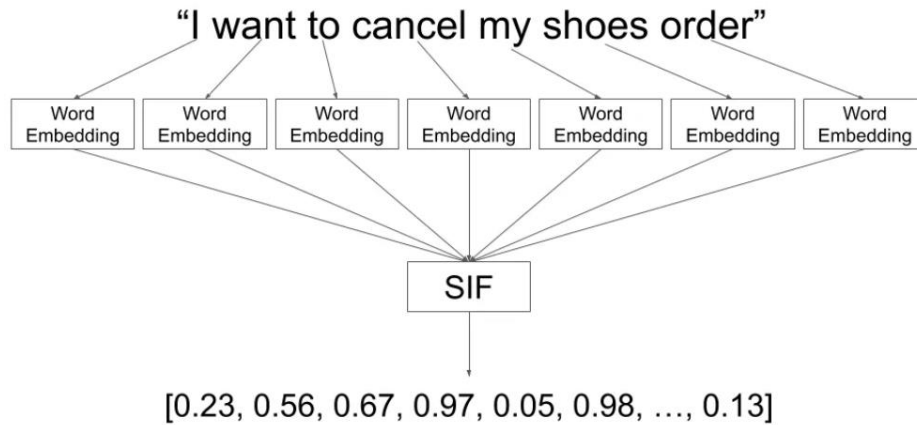


Figure 10 Visual representation of an embedding process [14]

3.3 Semantic Embedding

To do the semantic embedding we first have to create the Siamese Model and train it with the embeddings created before. From what it has learned, it will then be able to perform its own embedding employing semantic understanding which will allow for the program to then do the semantic search.

For the creation of our Siamese network, we utilized *Pytorch*, a Python library popularly used for deep learning workflows. It allows us to define layers, load the data, run optimizers and do the training process [11]. For this, *torch* library was imported.

The Siamese network was defined as a class with two methods, one indicating the layers that are chained together in a sequential matter, and another defining the forward pass of the network. This method takes in three inputs which would be the anchor, positive and negative embeddings.

For the Siamese Networks' first method, four layers were defined, two outer layers (input and output) and two hidden layers. Four layers were defined as it provides

enough capacity for the model to learn useful patterns while reducing the risk of overfitting. Through torch, the “nn.Linear” function was applied to create the layers.

Another class was also defined indicating the triplet loss function that the Siamese network will train from. In this class, the loss function is defined, and an average loss value is returned for each batch (anchor, positive and negative embeddings).

For the training of the model, a few values had to be defined such as the number of epochs, the learning rate and the margin value.

- The number of epochs indicates the iterations the model has to do within one same batch. The value was set to 10 as we are dealing with a large data set which needs enough iterations to show good performance but not too much that would cause overfitting of the model.
- The learning rate of the model was set to 0.001, a common value to use in deep learning models. This value controls how slow or fast a model is going to learn, as it defines the changing rate of the weights of the neural network.
- Finally, the margin value indicates the minimum distance value required between the anchor-positive pair and anchor-negative pair during the triplet loss computation. This allows the model to learn about the distinction between similar and dissimilar values. The margin value was set to 1, which is a commonly used value as it is not too small or too big. If the value is too small the model will not learn the difference between the similar and dissimilar items, and if it is too big, the model will find it difficult to satisfy the condition.

So, the model processed all the embeddings with 10 epochs for each batch, a learning rate a 0.001 and a margin value of 1.

Figure 11 shows how the model iterates each time through a new epoch, and how the loss is reduced. As we can see there may be times where the loss slightly increases but it is all part of the learning process. For each batch, the loss function can vary.

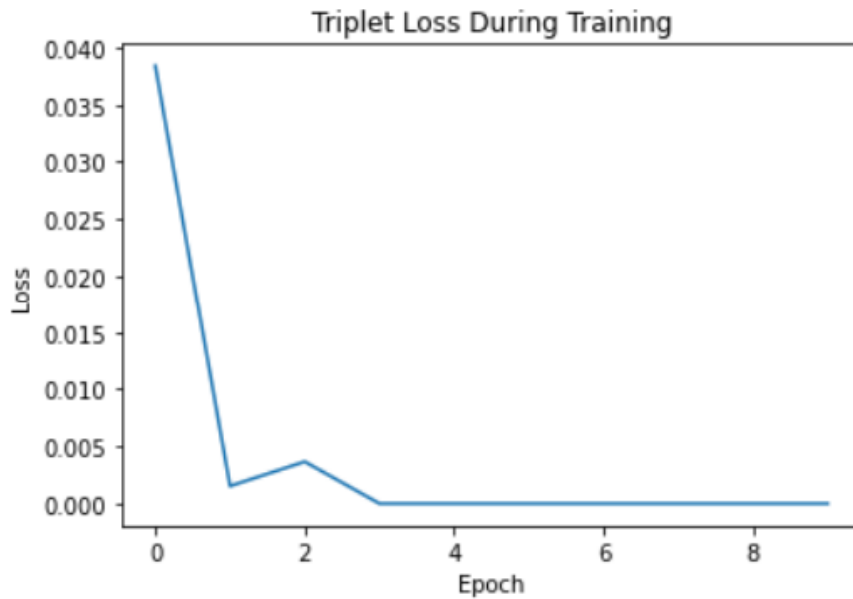


Figure 11 Loss function representation through each batch

The next step would be to do the semantic embedding. For this we had to load the previously trained model into python and pass all the vectors through it creating the semantic embedding. Finally, they are stored in the corresponding NumPy binary files and passed to the semantic search.

3.4 Semantic Search

This is the final step of the process, where a *FAISS* module is created and serves as a vector database to search for the closest K embeddings after providing an input query.

Vector databases are databases capable of storing large collections of embedded vectors. They do not only offer vector storage but have other capabilities such as vector search that is conducted through similarity search between the items [12].

FAISS is a library developed by Facebook that provides Approximate Nearest Neighbor (ANN) algorithms and serves as a vector search library. The basic structure of

FAISS is the index. The index serves as a storage unit for database vectors. During search operations, a query vector is passed onto FAISS, and the index returns the closest vectors based on Euclidean distance [12].

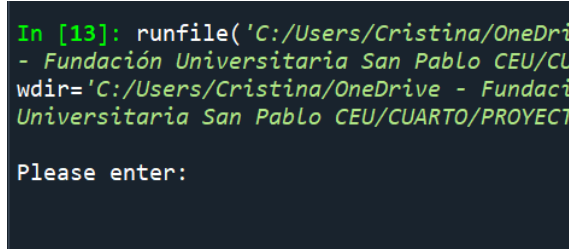
The first step of the semantic search is to create a FAISS index and load all the semantically embedded vectors from the previous step into this index. For this, the *FAISS* library was imported into the environment. The index was created through “faiss.IndexFlatL2” function. This creates a “Flat” index specifically for similarity search, this is, an array for vectors, specifying “L2” parameter, which means that the nearest neighbor search will be done through Euclidean distances.

While the embeddings are being loaded into the FAISS index, another list will be created. This list consists of a series of IDs that are associated to each embedding. It is created as the embeddings are being individually loaded into the index. The ID represents the folder where the embedding is stored. This folder also contains other features such as the URL where the embedding originated from, which will be necessary for the next step of the semantic search. This way, the embeddings are controlled and the URLs for each embedding can be identified.

After having loaded all the embeddings into the index, the user interaction code will be created. First, we will need to employ an embedding function for the input query. From the embedded query, the *FAISS* index will search for the *K* nearest neighbors, where *K* is a value indicated by the user, using the previously mentioned Euclidean distance.

4 RESULTS

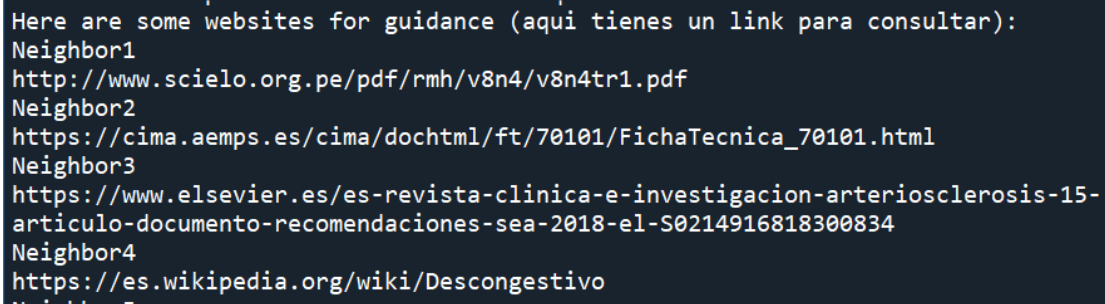
The tool receives an input from a doctor (or anyone who is using the program), which includes, for example, textual descriptions of a patient's symptoms. Figure 12 shows what the user would see and where they would introduce the text:



```
In [13]: runfile('C:/Users/Cristina/OneDrive - Fundación Universitaria San Pablo CEU/CU
wdir='C:/Users/Cristina/OneDrive - Fundación Universitaria San Pablo CEU/CUARTO/PROYECT
Please enter:
```

Figure 12 User interface interaction

The program generates a series of links based on the input text. These links point to resources, articles, or other information considered relevant by the program that could provide additional help in understanding or diagnosing the symptoms. The program, based on the Euclidean distance to determine the closest embeddings to the input text, provides the content shown in Figure 13.



```
Here are some websites for guidance (aquí tienes un link para consultar):
Neighbor1
http://www.scielo.org.pe/pdf/rmh/v8n4/v8n4tr1.pdf
Neighbor2
https://cima.aemps.es/cima/dochtml/ft/70101/FichaTecnica_70101.html
Neighbor3
https://www.elsevier.es/es-revista-clinica-e-investigacion-arteriosclerosis-15-articulo-documento-recomendaciones-sea-2018-el-S0214916818300834
Neighbor4
https://es.wikipedia.org/wiki/Descongestivo
Neighbor5
```

Figure 13 Program URL response

The URL serves as an address that points to a website that could be of use to the physician. Figure 14 shows the website associated to the neighbor 4 URL. This would be an article in Spanish that talks about the indigestion of corrosive substances and indicates that pain in the abdomen is a possible symptom and that if this pain gets worse, it is necessary to investigate the patient's condition through radiology techniques.

Los síntomas de compromiso esofágico son básicamente disfagia y odinofagia, mientras que dolor epigástrico, arcada o vómito (restos tisulares o sangre), indican compromiso gástrico (4). Un compromiso de todas las capas del esófago puede originar extensión de la injuria al mediastino y el paciente rápidamente llega a estar agudamente enfermo, produciéndose dolor severo y persistente, subesternal o en la espalda, vómito, síntomas respiratorios, fiebre, taquicardia y shock (6). Una perforación gástrica o del esófago distal produce una peritonitis (10,22). La perforación esofágica o gástrica , puede ocurrir en cualquier momento dentro de las primeras dos semanas, de aquí que si existe algún cambio en la condición clínica del paciente (tal como el empeoramiento de un dolor abdominal o aparición de dolor torácico), debe investigarse rápidamente con técnicas radiológicas (4,13,27). También se ha descrito como consecuencia de la injuria cáustica el desarrollo de fístula esófago-traqueal (13) y hematemesis masiva debido al desarrollo de una fístula aortica (4).

Figure 14 Website example from one of the URL results for “abdominal pain”

Additionally, the program produces two scatterplots. These plots visualize the K nearest neighbors generated by both the Siamese model and a generic embedding technique, in this case the 15th nearest neighbors. This value of K was chosen to get a deeper visual representation of the FAISS index search. Each point on the scatterplot represents a neighbor, and the position of the point reflects the similarity between the user’s text. This can be seen in figure 15.

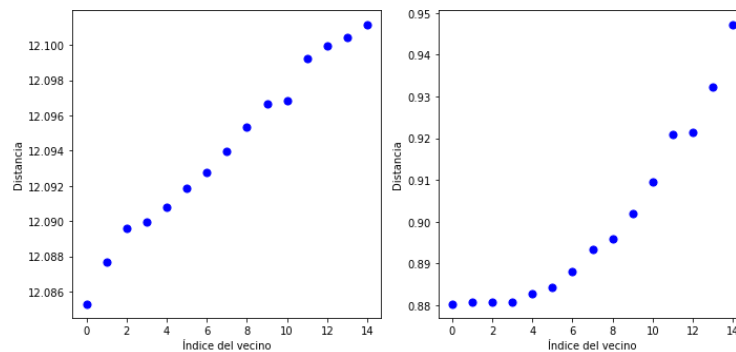


Figure 15 Scatterplot produced for “abdominal pain”

Another example of a scatterplot but with the user introducing the text in English “swollen ankles” is shown in Figure 15.

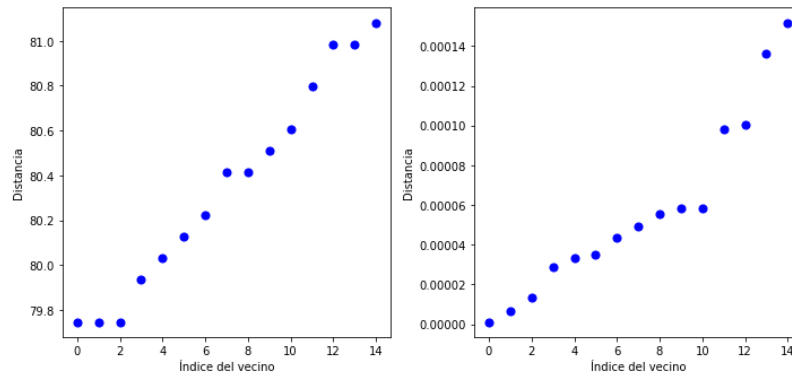


Figure 16 Scatterplot produced by “Swollen ankles”

4.1 Quantitative results: simple input query

The efficiency of the model was determined by gathering a diverse dataset comprising different text inputs from the user along with associated links and calculating the precision. In addition, we will ensure that the dataset covers a wide range of topics and scenarios relevant to the application domain.

We will annotate the dataset by manually verifying the association between the provided text and the corresponding links. This annotation process serves as ground truth for evaluating the model's performance.

For this we will generate predictions for the inputs in the dataset and set the K as 5, so it will generate the 5 nearest neighbors to have a value that allows for further analysis. For each of these results, we will compare the generated link with the text input and determine if they are truly associated correctly.

For it to be truly associated, the website linked to the URL provided must contain some type of relationship to the text inside the meaningful information section. In this case, the neighbor will be considered as “true positive”, the other case, it will be considered as “false positive”. The precision metric was used for system evaluation. This

measures the proportion of instances that the model correctly identifies as positive out of all instances it identifies as positive [13]. It gives an indication of how reliable the model's positive predictions are and is given by Equation (2)

$$\text{Precision} = \text{TP} / (\text{TP} + \text{FP}) \quad (2)$$

For experiments, we generated 10 different text inputs and sorted the results as “true positive” or “true negative” by manually verifying the association between the provided text and the corresponding links. Table 1 shows different phrases that the search tool has processed and the number of true positives that the program returned.

Table 1 Different User texts and true positives associated

User text	TP	User text	TP
1. “Picor en la garganta”	3	6. “Piel irritada”	4
2. “Mareos espontáneos”	4	7. “Caída de pelo”	2
3. “Decreased appetite”	2	8. “Dolor de cuello”	4
4. “Cansancio”	4	9. “Dry hands”	4
5. “Inflammation of the esophagus”	3	10 “Parálisis de la pierna derecha”	3

Now we will calculate the positive predictions from the values extracted from the search. We have processed 10 different user texts and determined the number of URLs provided were “true positive” or “false positive”, the results were:

TP: 33

FP: 17

Precision = $33/(33 + 17) = 0.66$

So, the precision of the model in this case would be 0.66 or 66%. This means that 66% of the instances that the model predicts as positive are indeed true positives.

4.2 Quantitative results: Advanced input query

This tool should be able to outperform other platforms such as google. This means, it should be able to accept a large query of various sentences and give an appropriate response. For this test, we entered the next query:

“Esta mañana me he levantado con un fuerte dolor abdominal. Decidí tomarme un analgésico y descansar un poco más, pero el dolor no cedía. Preocupado, llamé a mi médico para pedir consejo. Me sugirió que monitorizara mis síntomas y que si el dolor persistía o empeoraba, acudiera a urgencias. Después de unas horas, noté que el dolor se concentraba en el lado derecho del abdomen y comenzaba a sentir náuseas.”

From this input query, the tool provided the next URLs representing the 5 closest neighbors:

Neighbor 1: <https://medlineplus.gov/spanish/ency/article/000817.htm>

Neighbor 2: <https://medlineplus.gov/spanish/ency/article/000817.htm>

Neighbor 3: <https://medlineplus.gov/spanish/ency/article/000817.htm>

Neighbor 4: <https://medlineplus.gov/spanish/ency/article/000817.htm>

Neighbor 5: <https://medlineplus.gov/spanish/ency/article/000817.htm>

Even though these five neighbors are from the same website, when we access the URL, we can see that it does in fact have useful information regarding the input query.

The website provides a wide range of information about food allergies including possible symptoms that can arise from them. Amongst the numerous general symptoms, some of them encompass nausea and abdominal pain, which match the query. In addition, it also gives an indication of which steps must be followed in case of a need for medical assistance, which is similar to the input text. However, the website does not make any mention about concentration of the pain to the right side of the abdomen.

5 DISCUSSION

The tool presented serves as a powerful tool in helping medical professionals by using advanced natural language processing techniques and embedding models to analyze textual descriptions of patient symptoms. This discussion focuses on the key components of the program and their implications for medical practice. This includes the text input process and its analysis, the generation of relevant links upon the textual input, the program capabilities of processing text in both Spanish and English, the visualization through scatterplots of the K nearest neighbors that were identified and an analysis of the efficiency of the model by analyzing the contents that the URLs provided contain.

5.1 Input Processing and Analysis

One of the fundamental functionalities of the program involves processing textual descriptions of patient symptoms provided by doctors. This input serves as the foundation for subsequent analysis and investigation. By leveraging NLP techniques, the program can interpret and extract relevant information from the textual input, enabling it to generate meaningful insights.

5.2 Generation of Relevant Links

Upon processing the input symptoms, the program generates a series of links that are potentially related to the provided information. These links guide users to a range of resources, articles, or relevant information sources that can assist them in understanding or diagnosing the symptoms they are investigating. This functionality enhances the program's utility by providing medical professionals with access to a diverse range of resources and knowledge, thereby facilitating informed decision-making.

5.3 Integration of Multilingual Resources

An interesting aspect of the program is its ability to integrate multilingual resources seamlessly. For instance, the program does not only direct users to scientific literature in English, but also Spanish, as it is seen in the example discussing abdominal pain and associated symptoms. This feature expands the scope of accessible information and enhances the program's applicability across diverse linguistic contexts.

5.4 Visualization through Scatterplots

In addition to textual recommendations, the program produces visualizations in the form of scatterplots to aid in understanding the relationships between the input symptoms and relevant cases. These scatterplots visualize the K nearest neighbors identified by both the Siamese model and a generic embedding technique. By visually representing the similarity between the user's text and other cases, these scatterplots offer valuable insights into the clustering and distribution of related information, further supporting the diagnostic process.

5.5 Precision result evaluation

The evaluation process involved testing the model on 10 different text inputs, manually verifying the associations between the provided text and URLs, and then calculating the number of true positives and false positives generated by the model. The precision metric helps assess the reliability of the model's predictions in this context, with a higher precision indicating a higher proportion of correct predictions relative to incorrect ones.

This evaluation was also conducted with different values of K, and concluded that as the value of K decreased, precision increased and while the value of K increased, the precision of the model decreased. In addition, more complex input phrases, such as those that contain more than one symptom had less precise results. Reasons of this could be that the model does not integrate sufficient information to have enough websites that contain this information, or the model may have needed another iteration during the training process.

Another conclusion is that the text inputs written in Spanish generally seemed to have a higher number of true positives than those texts written in English. This could be due to the fact that most of the websites from which the information was extracted had contents written in Spanish, so there were fewer websites in English for the model to learn from. But we have to bear in mind that this evaluation was very superficial and just gives us a bit of guidance of the precision of the model.

5.6 Limitations

Limitations that have appeared during the development of this project include problems with web scrapping, the embedding process, and the prolonged time it took for the preparation of data and running of the code.

Firstly, the collection of information during the parsing of HTMLs had some difficulties. This was due to the fact that many groups of websites contained a different HTML structure and therefore the important information was contained in different sections depending on each HTML. It is possible that some irrelevant information was extracted, and the model processed that information considering it as medical data, which could have affected the results of the program.

And secondly, there were challenges in the preparation of data. First the embedding of the phrases extracted from the scientific literature took an immense amount of time to finish executing and secondly, the model also took a very long amount of time to process the embeddings and learn from it. This is possibly due to the large content of data (approximately 31000 different files to process), resource constraints such as limited memory and external dependencies such as the performance of the computer used for the development of the project.

6 CONCLUSIONS

This tool exemplifies the convergence of natural language processing techniques and medical informatics, offering insights for enhancing clinical decision-making. By seamlessly integrating with the workflow of healthcare professionals, the program is a promising tool for the process of symptom analysis and diagnosis, providing practitioners with rapid, evidence-based insights.

Through its input processing abilities, the program transforms textual descriptions of patient symptoms into the generation of relevant links to scientific literature and resources, coupled with the utilization of embeddings from a Siamese model, and augments the diagnostic process by providing clinicians with access to a broad range of relevant information.

Essentially, the program represents a way of using artificial intelligence to improve clinical decision-making, offering a series of computational results and medical expertise. As the landscape of healthcare continues to evolve, such innovative solutions hold tremendous promise in driving improvements in patient outcomes, ultimately contributing to the advancement of medical practice and the betterment of human health.

Limitations that have appeared during the development of the project include problems with web scrapping, the embedding process, and the prolonged time it took for the preparation of data and running of the code. Factors that influenced these limitations include the diversity of the different HTML structures, the large contents of data that were worked with and the computers own limited memory and capacity among other different factors.

Finally, some future work that could be interesting to consider is as follows: to give the model more scientific literature that could improve its accuracy and provide more comprehensive results; exploring alternative learning methods, such as contrastive loss, and different neural networks, such as convolutional neural networks, to determine if there are any performance improvements; use of devices with larger capacity, to speed-up the time execution, as well as more memory resources; integration of patient data and learn from it to provide more personal results for each patient; and integrate some level of security measures such as excluding patients' name and other medically irrelevant

information. In conclusion, while the program has shown promising results in leveraging NLP and embedding models for medical text analysis, there is potential for further development.

7 REFERENCES

- [1] Jayatilake, S. M. D. A. C., & Ganegoda, G. U. (2021). Involvement of Machine Learning Tools in Healthcare Decision Making. *Journal Of Healthcare Engineering*, 2021, 1-20.
- [2] Muhathir, M., Sibarani, T. T. S., & Al-Khowarizmi, A. (2020). Analysis K-Nearest Neighbors (KNN) in Identifying Tuberculosis Disease (Tb) By Utilizing Hog Feature Extraction. *Al'adzkiya International Of Computer Science And Information Technology (AIoCSIT) Journall*, (pp. 33-38).
- [3] Abraham, A. (2005). Artificial neural networks. *Handbook of measuring system design*.
- [4] Serrano, N., & Bellogín, A. (2023). Siamese neural networks in recommendation. *Neural Computing and Applications*, 35(19), 13941-13953.
- [5] Schroff, F., Kalenichenko, D., & Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 815-823).
- [6] CrisJimAbadal. (s. f.). GitHub - CrisJimAbadal/LLM^{BiomedicalConcepts}: LARGE LANGUAGE MODEL OF BIOMEDICAL CONCEPTS EXTRACTED FROM SCIENTIFIC LITERATURE. GitHub. <https://github.com/CrisJimAbadal/LLM^{BiomedicalConcepts}.git>
- [7] Dikilitaş, Y., Çakal, Ç., Okumuş, A. C., Yalçın, H. N., Yıldırım, E., Ulusoy, Ö. F., Macit, B., Kırkaya, A. E., Yalçın, Ö., Erdoğan, E., & Sayar, A. (2024). Performance Analysis for Web Scraping Tools: Case Studies on BeautifulSoup, Scrapy, Htmlunit and Jsoup. In *Lecture notes in networks and systems* (pp. 471-480).
- [8] Yevsieiev V. A Program for Analyzing the Structure of a Web site Development Using the Parsing Method Based on the Python / V. Yevsieiev, S. Maksymova, Ahmad Alkhalailah // *Journal of Universal Science Research*, 2024, 2(4), 172-183.
- [9] Buakhao, R. (2024). Extracting Known Side Effects from Summaries of Product Characteristics (SmPCs) Provided in PDF Format by the European Medicines Agency (EMA) using BERT and Python.
- [10] Topsakal, O., & Akinci, T. C. (2023). Creating Large Language Model Applications Utilizing LangChain: A Primer on Developing LLM Apps Fast. *International Conference On Applied Engineering And Natural Sciences*, 1(1), 1050-1056.
- [11] Bandi, A., & Kagitha, H. (2024). A Case Study on the Generative AI Project Life Cycle Using Large Language Models. *Proceedings of 39th International Confer*, 98 (pp. 189-199).
- [12] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., ... & Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.
- [13] Douze, M., Guzhva, A., Deng, C., Johnson, J., Szilvasy, G., Mazaré, P., Lomeli, M., Hosseini, L., & Jégou, H. (2024). The Faiss library. *arXiv (Cornell University)*. <https://doi.org/10.48550/arxiv.2401.08281>
- [14] Ferreira, D. (2022, 1 febrero). What Are Sentence Embeddings and why Are They Useful? *Medium*.