metrics

MDPI

Review

Embeddings for Efficient Literature Screening: A Primer for Life Science Investigators

Carlo Galli [1], Claudio Cusano [2], Stefano Guizzardi [1], Nikolaos Donos [3] and Elena Calciolari [3,4,*]

[1] Histology and Embryology Laboratory, Department of Medicine and Surgery, University of Parma, Via Volturno 39, 43126 Parma, Italy; carlo.galli@unipr.it (C.G.); stefano.guizzardi@unipr.it (S.G.)
[2] Department of Electrical, Computer and Biomedical Engineering, University of Pavia, Via Ferrata 1, 27100 Pavia, Italy; claudio.cusano@unipv.it
[3] Centre for Oral Clinical Research, Institute of Dentistry, Faculty of Medicine and Dentistry, Queen Mary University of London, London E1 2AD, UK; n.donos@qmul.ac.uk
[4] Department of Medicine and Surgery, Dental School, University of Parma, 43126 Parma, Italy
* Correspondence: elena.calciolari@unipr.it

Abstract: As the number of publications is quickly growing in any area of science, the need to efficiently find relevant information amidst a large number of similarly themed articles becomes very important. Semantic searching through text documents has the potential to overcome the limits of keyword-based searches, especially since the introduction of attention-based transformers, which can capture contextual nuances of meaning in single words, sentences, or whole documents. The deployment of these computational tools has been made simpler and accessible to investigators in every field of research thanks to a growing number of dedicated libraries, but knowledge of how meaning representation strategies work is crucial to making the most out of these instruments. The present work aims at introducing the technical evolution of the meaning representation systems, from vectors to embeddings and transformers tailored to life science investigators with no previous knowledge of natural language processing.

Keywords: embedding; life science; academic publications

Citation: Galli, C.; Cusano, C.; Guizzardi, S.; Donos, N.; Calciolari, E. Embeddings for Efficient Literature Screening: A Primer for Life Science Investigators. *Metrics* **2024**, *1*, 1. https://doi.org/10.3390/metrics1010001

Academic Editor: Ying Huang

Received: 6 August 2024
Revised: 12 September 2024
Accepted: 26 September 2024
Published: 30 September 2024

1. Introduction

The world of academic publishing is rapidly accelerating. More research is being conducted by an increasing number of researchers worldwide, and more journals are available; publication turnaround is faster [1]. As a result, the number of scientific reports that appear every year online is growing at an exponential rate [2]. For any researcher, understanding the current literature is essential for producing new science and generating new knowledge. Consequently, the ability to sift through vast and increasingly complex data repositories has become a critical skill. Additionally, with the volume of publications far surpassing the capacity of individual scholars to keep pace, reviews—whether narrative or systematic—are becoming increasingly important for summarizing state-of-the-art knowledge in various fields. Reviews are gaining popularity, especially in the life sciences, and more researchers are undertaking such projects. To conduct reviews of the literature, however, online databases must be queried [3,4], usually with keyword-based approaches, i.e., searching for entries that include one or more desired keywords in their text, usually in their title or abstract [5]. This approach can be very effective, provided that investigators use appropriate search terms and query syntax and that articles use the desired wording [6]. Keyword-based searches, however, are only partially efficient because search results are commonly inflated by large numbers of off-topic publications. We have recently demonstrated that even well-conducted literature searches in the biomedical field can include unrelated articles—up to 30% in our experience—likely due to text ambiguity or the use of polysemic expressions, which makes literature screening cumbersome and

time-consuming [7]. Thus, normally investigators have to spend much time and effort to clean their search results, often through manual screening [8]. Over the years, many investigators have worked intensively to improve search strategies and enhance the quality of search results [9–11].

2. A Semantic Space Odyssey

Fortunately, natural language processing (NLP) and machine learning are progressing at a very fast pace, and new tools and techniques are being developed that could greatly enhance efficiency and effectiveness in this process. Several of these systems are now available, offering powerful solutions for streamlining the search process [12]. Many of these algorithms are capable of understanding the central theme or "aboutness" of a document, which is a crucial capability for artificial intelligence systems aiming to discriminate between relevant and irrelevant literature. This type of data analysis is known as topic modeling, and numerous studies have been published on the application of topic modeling techniques to search, screen, and analyze scientific literature, including in the life sciences, due to the enhanced potential of newer algorithms [13–15]. The core of these algorithms lies in their ability to create numerical representations of text meaning using vectors, or special versions thereof called embeddings [16].

The concept that a word, a sentence, or a whole document can be represented as a vector in a semantic space based on its co-occurrence with other words within a given lexical context has origins in the work of Leonard Bloomfield and Zellig Harris that is the basis of Distributional Semantics (DS) [17,18]. In contrast to other semantics theories, which mainly focus on the cognitive structures that associate an object in the world (a referent) to a word, according to DS, the meaning of a word is defined by its linguistic context. While this idea is not unique to DS, the theory emphasizes that a word's meaning can be represented through its context. According to a famous quote by J.R. Firth, "You shall know a word by the company it keeps!".

An example of this could be the somewhat quirky sentence:

> "She took a *xylinth*, and applying a very light pressure, she was able to remove most of the calculus."

The readers may not be aware of what a *xylinth* is (it is admittedly a made-up word), and they may not have heard this word at all, but based on the context, most readers would likely infer that this unusual word refers to some kind of tool, possibly a surgical instrument, used in dental practice to remove calculus—maybe from alien teeth.

Distributional semantics undoubtedly possesses an advantage that can be leveraged in NLP, i.e., that all the elements that are necessary to describe the meaning of a word are contained within the sentence itself. A meaning representation rooted in Distributional Semantics does not need to refer to any other knowledge repository, lexicon, or ontology outside the very sentence that is being represented, and this makes computing leaner and faster.

There are several approaches for constructing semantic vectors from a document or corpus of documents in the field of NLP. One approach to sentence vectors is utilizing co-occurrence matrices that measure the frequency of words appearing together in the text [19]. These vectors are typically long, sparse (i.e., containing many zeros), and generally inefficient.

Consider the following two simple sentences to better understand the process:

(a) Red blood cells transport oxygen.
(b) Inflamed tissues are red.

From these two sentences, we can create a vocabulary, W, that includes all the words listed by their lemma (i.e., their base or dictionary form):

$$W = [\text{Red, blood, cell, to transport, oxygen, inflamed, tissue, to be}]$$

We can now convert each sentence into a vector, potentially ignoring common grammatical words like articles or prepositions, which are referred to as stopwords in NLP [20], by assigning the value 1 for each word of *W* when it is present in the sentence and 0, when it is absent (Table 1).

Table 1. Creation of a sentence-term matrix through the use of a dictionary.

W	Red	Blood	Cell	To Transport	Oxygen	Inflamed	Tissue	To Be
Sent. (A)	1	1	1	1	1	0	0	0
Sent. (B)	1	0	0	0	0	1	1	1

The first sentence now possesses a vectorial representation [1,1,1,1,1,0,0,0], while the second can be represented by the vector [1,0,0,0,0,1,1,1].

While these vectors can be useful for certain simple NLP tasks, they come with significant limitations. These vectors are sparse, containing many zeros, because each sentence includes only a small number of words from the W vocabulary. The length of W is determined by the size of the vocabulary, which can be quite large—ranging from tens to hundreds of thousands of words, depending on the language and topic. Another major limitation is that sentence representations are independent of word order, meaning that sentences like:

(a) Red blood cells transport oxygen
(a′) Oxygen transports red blood cells

have identical representations, even though their meaning is different. This technique is known as the "Bag of Words" (BoW) model, as it treats all words in a document as if they were placed together in a single "bag," disregarding their order [21]. Additionally, this method complicates the accurate representation of sentence semantics and is further limited by the absence of intonation and prosody, elements present in spoken language but absent in written text.

Common words, which often contribute little to the sentence's meaning, may be overemphasized in the vector representation within a document-term matrix. To obviate that, weighing procedures have been developed, such as the Term Frequency-Inverse Document Frequency (Tf/Idf) algorithm, which increases the weight of rarer words [22] while reducing the bearing of more frequent words (to highlight the peculiarities of individual sentences).

A radical alternative to sparse vectors is the use of *dense* vectors, commonly known as embeddings in this context. Dense vectors are numerical sequences where most, if not all, elements are non-zero. Embeddings represent words or phrases in a continuous vector space, and several algorithms have been developed to compute them [23]. One example is the Word2Vec algorithm, introduced by Mikolov in 2013, which uses shallow neural networks with a single hidden layer to create word embeddings. This approach has been widely influential [24]. Word2vec relies on the fundamental premise of the distributional hypothesis, assuming that words appearing in similar contexts share similar meanings. The first step is to set a context window, i.e., how many words around a given word we should consider; it could be ± 1 word or more, although there is consensus that the context should not exceed ± 3–4 words [25]. Consider the following sentence, which is likely uncontroversial:

(a) Neurology is a difficult but interesting topic

We would typically eliminate "is", "a", and "but" because they are stopwords and do not add much to the semantic content of the embedding, at least in BoW algorithms. This leaves us with four semantically rich lexical elements:

(b) Neurology difficult interesting topic

For each word in the sentence, we identify its context. Assuming a ± 1 context window, the context for "neurology" is "difficult". If "difficult" is the target word, its context will be "neurology", "interesting", and so on. We can thus create tuples of (target, context) words:

(neurology, difficult),
(difficult, neurology), (difficult, interesting)
(interesting, difficult), (interesting, topic)
(topic, interesting)

Now that we have tuples of target–context words, Word2Vec offers two primary methods for generating word embeddings: the Continuous Bag of Words (CBOW) and skip-gram models. In the CBOW model, the objective is to predict a target word based on its context words within a specified window. Conversely, the skip-gram model flips this process by aiming to predict context words given a target word. Word2vec uses a neural network architecture where the input layer corresponds to the context (or target) words, and the output layer contains its counterpart in the tuple. Training adjusts the weights between these layers to improve the model's ability to predict one word based on the other. The word embeddings are extracted from the hidden layer's weights, representing words as dense, continuous vectors in a high-dimensional space.

In practice, the algorithm assigns an arbitrary vector to each word using the 'one-hot encoding' approach, for example:

Neurology	=	[1,0,0,0]
Difficult	=	[0,1,0,0]
Interesting	=	[0,0,1,0]
Topic	=	[0,0,0,1]

Word2vec would then apply weights (w_1 in Figure 1, for skip-gram architecture) to the input vector that would then get adjusted to maximize the similarity of the output of the neural network to the vector of the target word. So, for instance, Figure 1 shows how the target word "neurology" (one-hot encoded representation [1,0,0,0]) would be processed with its context word "difficult" (one-hot encoded representation [0,1,0,0]). The one-hot encoded representation of "neurology" is multiplied by a matrix of values (known as weights) in the hidden layer to produce a dense vector, also known as word embedding. This embedding vector is passed to the output layer, where it is used to predict the probability of the context word "difficult". The output is a function that converts the raw scores into probabilities for each possible word in the vocabulary. Since the correct context word in this case is "difficult" [0,1,0,0], the algorithm calculates the error by comparing the predicted probability distribution with the actual one-hot encoded representation of "difficult".

The error is then propagated back through the network using backpropagation. The weights between the input and hidden layers (word embeddings) and between the hidden and output layers are adjusted based on this backpropagated error. Over many iterations with different target–context word pairs, the network would thus learn to adjust the weights in a way that words that appear in similar contexts end up having similar vector representations.

Once trained, Word2Vec generates word embeddings by discarding the original input vectors and taking only the weights associated with each word in the input layer of the neural network (Figure 2). In simpler terms, the trained model captures the meaning of words by retaining only the learned weights (i.e., the embeddings), which are a representation of each word's meaning based on its context in the text. The nature of the learning task ensures that words appearing in similar contexts are assigned similar weights and, therefore, similar embeddings. For example, the word "cat" will have an embedding closer to "dog" than to "airplane", since "cat" and "dog" often occur in similar contexts.

Moreover, semantic structures tend to be preserved by the embedding, so it is typical to observe relations like

$$\text{King} - \text{Man} + \text{Woman} \approx \text{Queen}$$

using those weights to produce embeddings that encode semantic similarities and relationships among words. Word2Vec's efficiency and ability to capture semantic meaning have made it a cornerstone of natural language processing, enabling a wide range of downstream applications such as text classification, sentiment analysis, and recommendation systems [26–29].

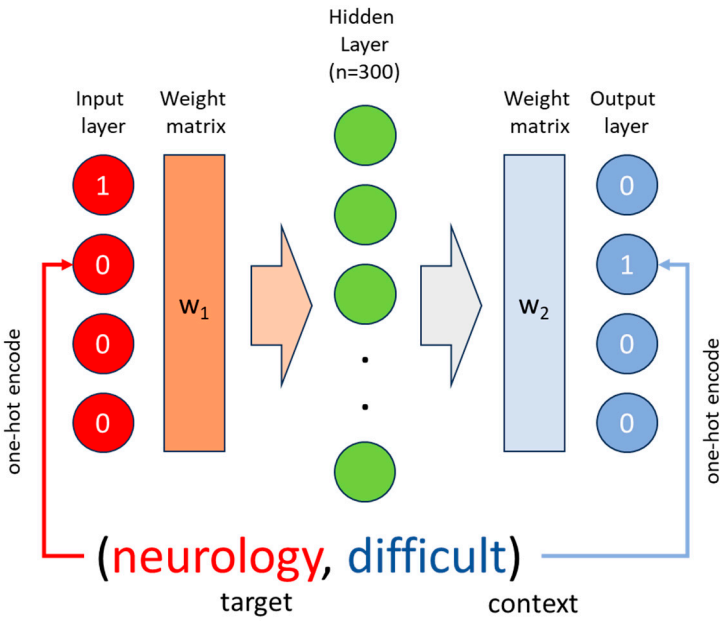

Figure 1. Diagram representing the architecture of the Word2Vec shallow neural network.

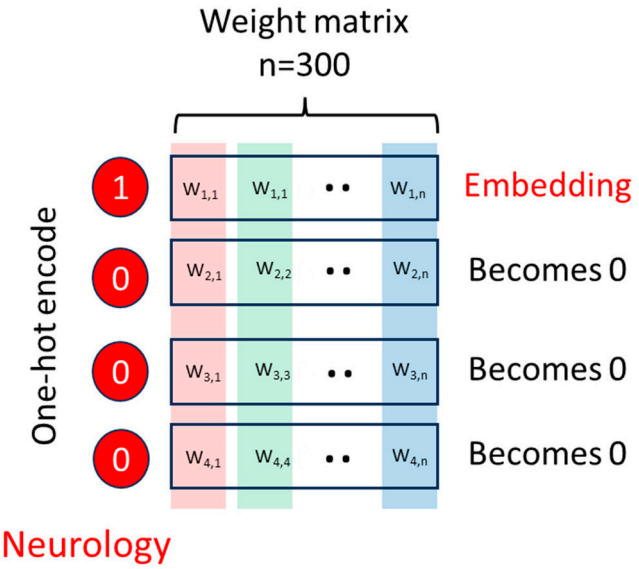

Figure 2. Word2Vec creates word embeddings by discarding the input one-hot encode vector and retaining the weights.

Recent evidence suggests that embedding-based models actually align with human cognitive representations. Hebart et al. [30] demonstrated that human similarity judgments for objects can be captured through a low-dimensional, interpretable embedding space.

This alignment may help explain why embeddings perform so well in various tasks, as they reflect fundamental patterns in how humans perceive and categorize information.

Once word embeddings are generated, entire sentences or documents can be represented by averaging or combining the embeddings of individual words.

Some researchers have explored the geometric limitations of word embeddings within Euclidean spaces, which often lead to overestimated similarities between distant words and underestimated similarities between nearby ones. Wang et al. [31] described the Global-Locality Preserving Projection (GLPP) method, which re-embeds word vectors by considering both local and global manifold structures, thus improving the accuracy of semantic information captured in word vectors. This refinement could significantly enhance the utility of transformers and other advanced models, which sometimes struggle with integrating local and global context effectively [32].

3. The Good, the Bad, the Ugly

Word2vec and other similar word embeddings are based on analyzing a small context window. This severely limits their capability to capture the meaning of full sentences, paragraphs, or documents. This limitation is inherited by the Multi-Layer Perceptron (MLP) architecture of the neural network [33]. The connections between neurons in an MLP are, in fact, fixed. Any input will be subjected to exactly the same sequence of operations. Therefore, the input size must be uniform and defined before training.

A more complex architecture, specially designed to deal with variable-length input sequences, is that of Recurrent Neural Networks (RNNs) [34]. In an RNN, the input is a sequence of vectors (e.g., obtained by applying word2vec to the words in a sentence). Each vector is processed by the RNN together with another vector representing a "state" of the computation. The result is an output vector and a new state. The state works as a short-term memory, aggregating information about all the previous input vectors processed so far. In some sense, in RNNs, the context window is formed by all vectors preceding the one being processed.

However, the capacity of the state vector is limited, and the memory of basic RNNs is typically very short. To improve that, many complex models have been proposed, such as Long–Short-Term Memory (LSTM, [35]) and Gated Recurrent Units (GRU, [36]).

RNNs became the de facto standard for most applications in natural language processing. They are especially suitable for the implementation of autoregressive models, that is, models that anticipate the next word in a sentence, given all the preceding ones [37].

4. Transformers, More than Meets the Eye

In recent years, Transformers have become a significant advancement in the field of literature screening, greatly improving research efficiency. Their introduction can be traced back to the seminal paper "Attention is All You Need" by Vaswani et al. in 2017 and has since paved the way for state-of-the-art NLP models such as the Bidirectional Encoder Representations from Transformers (BERT) or the Generative Pre-trained Transformer (GPT), among others [38].

Transformers offer an alternative architecture that addresses some limitations of traditional word embedding learning methods by incorporating contextual information for downstream tasks. Transformers are specifically designed to handle data sequences, making them ideal for NLP tasks requiring a deep understanding of word context. This is achieved through the inclusion of self-attention, a mechanism that allows the model to assign varying levels of importance to different segments during processing.

Unlike traditional neural networks, where connections between the neurons are fixed at design time, networks with attention mechanisms may learn to route information adaptively, greatly improving their capabilities. Given a sequence of vectors x_1, $x\_(2)$, . . ., x_T (perhaps vectors in a word embedding), self-attention uses three matrices of learnable

weights to linearly project them into three $T \times D$ matrices of keys (K), values (V), and queries (Q). Then, the following expression is computed:

$$Y = softmax(\frac{KQ^T}{\sqrt{D}})V$$

where the product KQ^T computes a matching score for each combination of a key and a query vector. The softmax operator converts the set of scores for each query into coefficients in the [0, 1] range, and the last product computes convex combinations of the value vectors by using those coefficients. The $\sqrt{D}$ term is just needed to scale the scores appropriately. The result is the $T \times D$ matrix Y, whose rows y_1, $y\_(2)$, . . ., y_T represent a transformed version of the input sequence. Note that all input vectors contribute to all output vectors, but with a weight determined by the matching between keys and queries. Note also that self-attention does not inherently consider the position in the sequences. To address this limitation, positional encodings are combined with word embeddings to provide a numerical representation of the relative positions of each token within the sequence.

Several models are based on transformer architecture, but their structural differences may reflect specific uses. For instance, although having similar basic architectural concepts, BERT and GPT have unique architectures that are suited for different tasks. BERT is designed as a bidirectional model, which means it considers the context from both directions (left and right) when encoding text. For tasks where comprehending the complete context is essential, such as sentence classification and question answering, BERT is especially well-suited [39]. On the other hand, GPT is an autoregressive model, generating text one word at a time while predicting the next word based on prior context. This makes GPT particularly effective for tasks like text generation, where fluency and coherence are very important [40].

A recent report by Goldstein et al. [41] has provided further insight into the capabilities of transformer-based models, demonstrating that the contextual embeddings generated by these models closely align with the patterns of neural activity observed in human language processing regions, such as the inferior frontal gyrus. This suggests that transformer models like GPT not only mirror computational principles of language processing but also encode semantic information in a way that resembles the human brain's natural language processing mechanisms.

So, briefly, transformers are neural networks that include multiple self-attention layers and many other standard layers [42]. They progressively transform the input by aggregating information across all positions in the sequence. In the case of language models, transformers are trained to predict the next token or to fill artificial gaps in sentences taken from a large corpus of text. After training, the model can be used as is or fine-tuned for a specific downstream task such as sentiment analysis, sentence similarity assessment, natural language inference, question–answer systems, and reading comprehension evaluations. We recently demonstrated that sentence transformers excel at capturing the meaning of specific sentences, such as biomedical article titles, and effectively distinguish subtle differences between articles [43]. Currently, sentence transformers have achieved a level of performance that enables them to be used as a foundation for conducting literature searches in the biomedical field and across the broader scientific landscape [44,45], with transformer-based literature reviews becoming increasingly common [13–15,46–50].

5. Far Away, So Close

A literature search requires the operator to be able to discriminate among papers [51]. This typically involves the investigator having specific research questions that need to be answered. The process of selecting or discarding the literature involves comparing the retrieved articles to the research query [52]. Simply put, if the topics are closely related, the article is retained; otherwise, it is discarded. For an artificial intelligence (AI) system to function effectively, the algorithm must have a way to measure the relatedness of a document's meaning to other documents [53]. Fortunately, once embeddings are created,

their similarity can be measured mathematically, allowing for an assessment of how similar the meaning of words or sentences is. One such method is cosine similarity, a widely used concept in NLP that measures the similarity between two non-zero vectors in a multi-dimensional space. Specifically, it quantifies the cosine of the angle between these vectors when they are represented as points in this space. In simple terms, cosine similarity measures how closely aligned two vectors are, regardless of their size. This property makes it particularly valuable for comparing documents or text data, where the vectors can represent the frequency of words or other features. To compute cosine similarity, the cosine of the angle between two vectors is calculated as the dot product of the vectors divided by the product of their magnitudes [54].

The similarity score ranges from -1 to 1. A score of 1 indicates perfect similarity (when the vectors point in the same direction), 0 indicates no similarity (when they are orthogonal), and -1 indicates perfect dissimilarity (when the vectors point in opposite directions). Cosine similarity offers advantages over other metrics, such as Jaccard similarity [55], because it takes into account not only the presence or absence of elements but also their relative frequencies. Moreover, cosine similarity is computationally efficient and can handle high-dimensional data effectively.

The similarity between embeddings can also be represented graphically, though there is a notable obstacle to that: we cannot plot graphs in more than three dimensions due to the physical constraints of the reality we live in, and vectors can have hundreds and thousands of dimensions. Fortunately, there are mathematical methods to reduce the number of dimensions of a vector, preserving as much information as possible. One common approach to dimensionality reduction is Principal Component Analysis (PCA), introduced in the early 20th century [56] and which identifies orthogonal axes (principal components) along which the data vary the most. By projecting the data onto a subset of these principal components, PCA reduces dimensionality while minimizing information loss (Figure 3) [57].

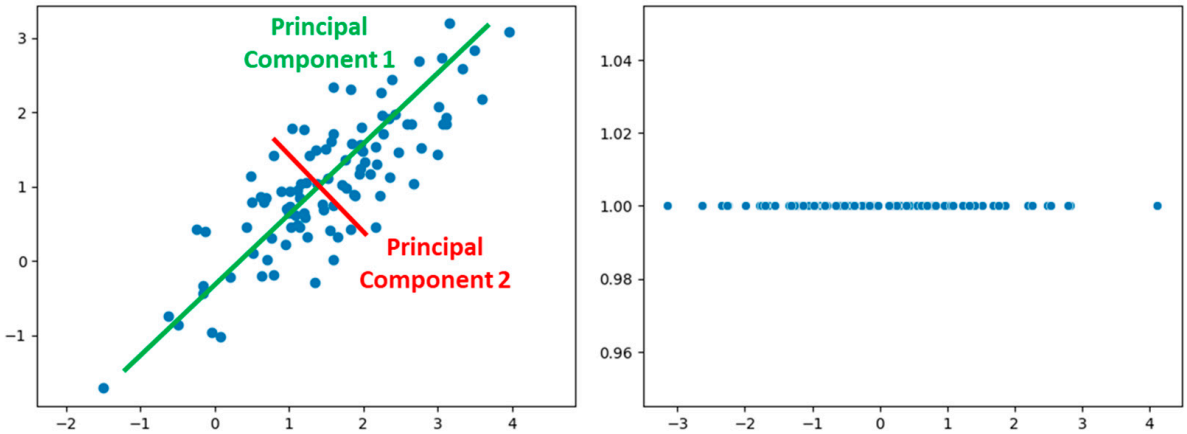

Figure 3. Principal Component Analysis is one algorithm for dimensionality reduction, which works by picking the components (i.e., the dimensions) along which the data show the greatest degree of variation.

Another widely used technique is t-Distributed Stochastic Neighbor Embedding (t-SNE), which focuses on preserving the pairwise similarity structure between data points, making it particularly useful for visualization and clustering tasks [58]. Linear Discriminant Analysis (LDA) is another method that seeks to maximize class separability in supervised learning settings, making it valuable for classification problems [59]. Non-linear dimensionality reduction techniques, such as Isomap and Locally Linear Embedding (LLE), address scenarios where data relationships are more complex and cannot be captured by linear transformations [60]. Isomap constructs a low-dimensional representation by estimating the geodesic distances between data points on a manifold, preserving the intrinsic geometry

of the data. LLE, on the other hand, seeks to retain local relationships between data points, making it suitable for preserving the fine-grained structure of the data. More recently, UMAP (Uniform Manifold Approximation and Projection) has been introduced to preserve the topological structure of high-dimensional data. UMAP creates a high-dimensional graph where each data point is connected to its nearest neighbors and then maps this structure into a lower-dimensional graph [61].

As an example, we can take a set of 10 titles from the literature:

a. Porous titanium granules in the treatment of peri-implant osseous defects: a 7-year follow-up study, reconstruction of peri-implant osseous defects: a multicenter randomized trial [62],
b. Porous titanium granules in the surgical treatment of peri-implant osseous defects: a randomized clinical trial [63],
c. D-plex500: a local biodegradable prolonged-release doxycycline-formulated bone graft for the treatment of peri-implantitis. A randomized controlled clinical study [64],
d. Surgical treatment of peri-implantitis with or without a deproteinized bovine bone mineral and a native bilayer collagen membrane: a randomized clinical trial [65],
e. Effectiveness of the enamel matrix derivative on the clinical and microbiological outcomes following surgical regenerative treatment of peri-implantitis. A randomized controlled trial [66],
f. Surgical treatment of peri-implantitis using enamel matrix derivative, an rct: 3- and 5-year follow-up [67],
g. Surgical treatment of peri-implantitis lesions with or without the use of a bone substitute—a randomized clinical trial [68],
h. Peri-implantitis—reconstructive surgical therapy [69].

And turn them into embeddings using the freely available pre-trained all-MiniLM-L6-v2 transformer model. This model has a transformer-based architecture that can turn any sentence into an embedding.

As an example, a portion of the 384-dimensional embedding for the first title in the list is shown below:

−2.29407530e−02, 1.49187818e−03, 9.16108266e−02, 1.75204929e−02, −8.36422145e−02, −6.10548146e−02, 8.30101445e−02, 3.96910682e−02, 1.58667186e−04, −2.62408387e−02, −7.69069120e−02, 4.60811984e−03, 8.64421800e−02, 7.87990764e−02, −4.33325134e−02, 2.49587372e−02, 2.24952400e−02, −2.90464610e−02, 3.59166898e−02, 4.27976809e−02, 7.94209242e−02, −5.87367006e−02, −6.49892315e−02, −8.70294198e−02, −5.51731326e−02, 4.95349243e−03, −3.01233679e−02, −3.23325321e−02, −1.54273247e−03, 5.24741262e−02, −7.11492598e−02, 5.16711324e−02, −4.42666225e−02, −6.38814121e−02, 6.46011531e−02, −4.63259555e−02, −9.23364013e−02, −3.56980823e−02, −9.30937752e−02, 1.27522862e−02, 5.05894162e−02, 5.07237464e−02, −9.00633708e−02, 6.91129547e−03, 4.79323231e−02, −6.69493945e−03, 1.27279535e−01, −6.33438602e−02, 2.78936550e−02, −3.34392674e−02, −6.21677283e−03, −4.32619415e−02, 5.89787960e−02, −9.10086110e−02, −2.79910862e−02, −5.80033176e−02, −5.82423434e−02, −6.41746866e−03, 4.17577056e−03, −1.90278993e−03, 6.72984421e−02, −4.39932309e−02, . . . 1.52898552e−02, 9.40597132e−02, −3.60338315e−02

The only way we can plot such an embedding is by reducing its dimensions to 3 or 2, so that we can use those as cartesian coordinates in a 3D or 2D graph, respectively. By applying PCA, the very same embedding can be reduced to:

0.5363835, 0.1012947

Each title is therefore now represented by only two values, which are less effective in describing all the semantic nuances of the title but can be used as cartesian coordinates in an x, y space, i.e., these reduced embeddings can now be plotted as dots within a semantic space in a 2D scatterplot (Figure 4A), where the distance between dots is a visual representation of the semantic distance between titles, or, for an even more detailed representation of the relationship between titles, in a 3D plot (Figure 4B).

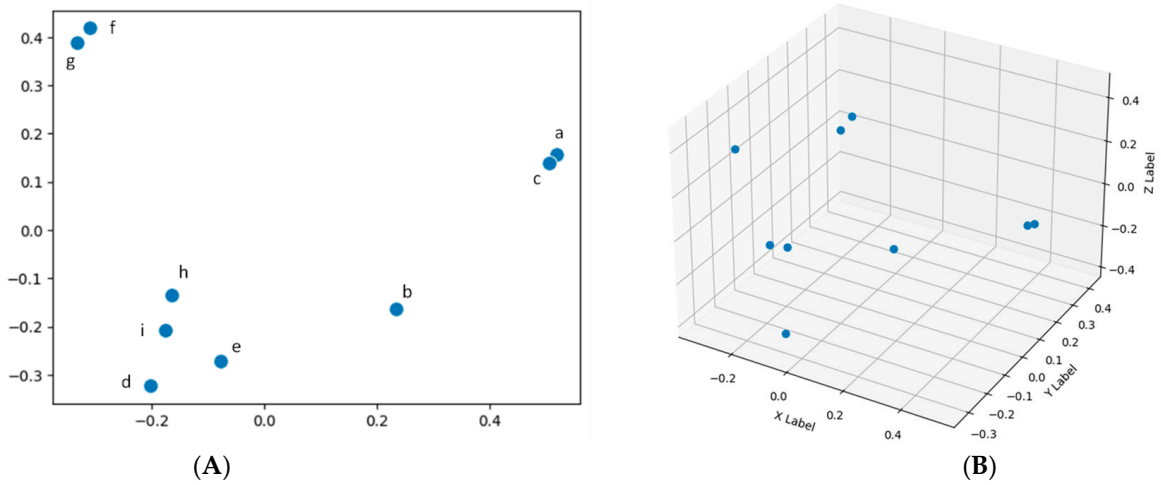

Figure 4. Embeddings can be reduced to two or three dimensions and used as Cartesian coordinates within a semantic space in (**A**) 2D or (**B**) 3D scatterplots. The closer the points in the scatterplot, the closer the semantics of the words.

However, dimensionality reduction comes with challenges and trade-offs [70]. Aggressive reduction may cause information loss, and selecting the appropriate method and number of dimensions requires careful consideration and experimentation [71]. Moreover, it may not always be suitable for all datasets; its effectiveness depends on the data's inherent structure and the specific task at hand. Nevertheless, when applied with discernment, dimensionality reduction techniques can be invaluable tools for simplifying complex data, improving model performance, and gaining insights from high-dimensional datasets in a wide range of applications [72].

6. Everything Everywhere All at Once

The development of Large Language Models (LLMs) like the GPT series developed by OpenAI [73] has significantly transformed NLP and its applications [74]. These models are "large" both in terms of the volume of data they are trained on and the complexity of their underlying neural networks [75]. LLMs stand out for their ability to understand, generate, and interpret human language, enabling them to perform various language-based tasks [76]. As these models evolve, becoming larger and more sophisticated, their potential in the literature review and information retrieval expands exponentially [77,78]. The ability to sift through the vast expanses of the internet and extract precise data and information is becoming very real, bolstered by the continuous release of open models and advancements in the underlying technology. Meta's introduction of the Llama series, or Google's Gemini or the Pathways Language Model (PaLM), highlights the competition among private and public stakeholders in developing LLMs that are both powerful and capable of nuanced understanding [79].

Combining LLMs with Retrieval-Augmented Generation (RAG) techniques enhances their ability to generate contextually relevant content by retrieving information from various online sources [80]. This approach could radically transform how we retrieve, use, and interact with information online. RAG works by combining a retrieval component with a generative model, enabling the system to fetch real-time information from external sources without relying solely on pre-trained knowledge.

Despite these advancements and the tools that are becoming available [81], the integration of AI in literature search and evidence collection still necessitates human interaction. This synergy between human oversight and AI's computational power ensures that the outputs are not only relevant but also accurate. The goal, however, remains to achieve a level of sophistication where AI can autonomously conduct literature reviews with

minimal to no human intervention, a milestone that seems increasingly attainable as the technology progresses.

The practical applications of such models are already being realized and implemented at an unprecedented pace. Looking ahead, the trajectory of LLMs suggests a continued enhancement in their ability to process and understand language at a level of depth and subtlety previously unattainable. The integration of these models into information retrieval and research processes promises to revolutionize how we navigate and utilize the wealth of knowledge available online, moving us closer to a world where AI can autonomously manage complex tasks of data and information retrieval with high sensitivity and sensibility.

7. Conclusions

This brief introduction has reviewed the role of embeddings in current NLP techniques, particularly in the analysis of scientific literature. By representing words, sentences, and entire documents as dense vectors in a semantic space, embeddings enable A.I. models to capture complex contextual relationships that traditional methods struggle to handle. Transformer-based models, such as BERT and GPT, use these embeddings to understand nuanced meanings and reduce ambiguity, and they potentially offer very effective tools for identifying relevant literature, opening up opportunities for life science and biomedical researchers that need new ways to face ever-increasing literature corpora.

Embeddings are thus a powerful tool for representing and processing the meaning of texts and provide a foundation for more accurate, efficient, and scalable literature screening. As these techniques are progressively refined and their applications across various domains are explored, embeddings will remain central to the ongoing evolution of natural language processing and information retrieval. For those interested, we have provided a simple Python code as Supplemental Material. This file can be uploaded in Jupyter Notebooks or Google Colab, and readers will have an opportunity to encode sentences as embeddings using transformer models and explore sentence similarity.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/metrics1010001/s1.

Author Contributions: Conceptualization, C.G.; methodology, C.C.; investigation, N.D.; data curation, C.C.; writing—original draft preparation, C.G. and E.C.; writing—review and editing, S.G.; supervision, N.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: Data are available upon request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Hanson, M.A.; Barreiro, P.G.; Crosetto, P.; Brockington, D. The Strain on Scientific Publishing. *Quant. Sci. Stud.* **2024**, 1–29. [CrossRef]
2. Landhuis, E. Scientific Literature: Information Overload. *Nature* **2016**, *535*, 457–458. [CrossRef] [PubMed]
3. Dickersin, K.; Scherer, R.; Lefebvre, C. Systematic Reviews: Identifying Relevant Studies for Systematic Reviews. *BMJ* **1994**, *309*, 1286–1291. [CrossRef]
4. Bramer, W.M.; Rethlefsen, M.L.; Kleijnen, J.; Franco, O.H. Optimal Database Combinations for Literature Searches in Systematic Reviews: A Prospective Exploratory Study. *Syst. Rev.* **2017**, *6*, 245. [CrossRef]
5. Lu, Z. PubMed and beyond: A Survey of Web Tools for Searching Biomedical Literature. *Database* **2011**, *2011*, baq036. [CrossRef] [PubMed]
6. Jin, Q.; Leaman, R.; Lu, Z. PubMed and beyond: Biomedical Literature Search in the Age of Artificial Intelligence. *EBioMedicine* **2024**, *100*, 104988. [CrossRef]
7. Galli, C.; Cusano, C.; Meleti, M.; Donos, N. Topic Modeling for Faster Literature Screening Using Transformer-Based Embeddings. 2024. Available online: https://www.preprints.org/manuscript/202407.2198/v1 (accessed on 10 September 2024).
8. Grivell, L. Mining the Bibliome: Searching for a Needle in a Haystack? *EMBO Rep.* **2002**, *3*, 200–203. [CrossRef]

9. Wilczynski, N.L.; Haynes, R.B.; Team, H. Developing Optimal Search Strategies for Detecting Clinically Sound Prognostic Studies in MEDLINE: An Analytic Survey. *BMC Med.* **2004**, *2*, 23. [CrossRef]
10. Zhang, L.; Ajiferuke, I.; Sampson, M. Optimizing Search Strategies to Identify Randomized Controlled Trials in MEDLINE. *BMC Med. Res. Methodol.* **2006**, *6*, 23. [CrossRef]
11. Heintz, M.; Hval, G.; Tornes, R.A.; Byelyey, N.; Hafstad, E.; Næss, G.E.; Bakkeli, M. Optimizing the Literature Search: Coverage of Included References in Systematic Reviews in Medline and Embase. *J. Med. Libr. Assoc.* **2023**, *111*, 599–605. [CrossRef]
12. Khalil, H.; Ameen, D.; Zarnegar, A. Tools to Support the Automation of Systematic Reviews: A Scoping Review. *J. Clin. Epidemiol.* **2022**, *144*, 22–42. [CrossRef] [PubMed]
13. Samsir, S.; Saragih, R.S.; Subagio, S.; Aditiya, R.; Watrianthos, R. BERTopic Modeling of Natural Language Processing Abstracts: Thematic Structure and Trajectory. *J. Media Inform. Budidarma* **2023**, *7*, 1514. [CrossRef]
14. Karabacak, M.; Margetis, K. Natural Language Processing Reveals Research Trends and Topics in The Spine Journal over Two Decades: A Topic Modeling Study. *Spine J.* **2024**, *24*, 397–405. [CrossRef] [PubMed]
15. Raman, R.; Pattnaik, D.; Hughes, L.; Nedungadi, P. Unveiling the Dynamics of AI Applications: A Review of Reviews Using Scientometrics and BERTopic Modeling. *J. Innov. Knowl.* **2024**, *9*, 100517. [CrossRef]
16. Jurafsky, D.; Martin, J.H. Vector Semantics and Embeddings. In *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models*, 3rd ed.; Online manuscript released 20 August; 2024. Available online: https://web.stanford.edu/~jurafsky/slp3 (accessed on 25 September 2024).
17. Turney, P.D.; Pantel, P. From Frequency to Meaning: Vector Space Models of Semantics. *J. Artif. Intell. Res.* **2010**, *37*, 141–188. [CrossRef]
18. Harris, Z.S. Distributional Structure. *Word* **1954**, *10*, 146–162. [CrossRef]
19. Erk, K. Vector Space Models of Word Meaning and Phrase Meaning: A Survey. *Lang. Linguist. Compass* **2012**, *6*, 635–653. [CrossRef]
20. Saif, H.; Fernandez, M.; He, Y.; Alani, H. On Stopwords, Filtering and Data Sparsity for Sentiment Analysis of Twitter. In Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC 2014), Reykjavik, Iceland, 26–31 May 2014.
21. Zhang, Y.; Jin, R.; Zhou, Z.-H. Understanding Bag-of-Words Model: A Statistical Framework. *Int. J. Mach. Learn. Cybern.* **2010**, *1*, 43–52. [CrossRef]
22. Jing, L.-P.; Huang, H.-K.; Shi, H.-B. Improved Feature Selection Approach TFIDF in Text Mining. In Proceedings of the International Conference on Machine Learning and Cybernetics, Beijing, China, 4–5 November 2002; IEEE: New York, NY, USA, 2002; Volume 2, pp. 944–946.
23. Wang, S.; Zhou, W.; Jiang, C. A Survey of Word Embeddings Based on Deep Learning. *Computing* **2020**, *102*, 717–740. [CrossRef]
24. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient Estimation of Word Representations in Vector Space. *arXiv* **2013**, arXiv:1301.3781.
25. Di Gennaro, G.; Buonanno, A.; Palmieri, F.A.N. Considerations about Learning Word2Vec. *J. Supercomput.* **2021**, *77*, 12320–12335. [CrossRef]
26. Al-Saqqa, S.; Awajan, A. The Use of Word2vec Model in Sentiment Analysis: A Survey. In Proceedings of the 2019 international Conference on Artificial Intelligence, Robotics and Control, Cairo, Egypt, 14–16 December 2019; pp. 39–43.
27. Haider, M.M.; Hossin, M.A.; Mahi, H.R.; Arif, H. Automatic Text Summarization Using Gensim Word2vec and K-Means Clustering Algorithm. In Proceedings of the 2020 IEEE Region 10 Symposium (TENSYMP), Dhaka, Bangladesh, 5–7 June 2020; IEEE: New York, NY, USA, 2020; pp. 283–286.
28. Ibrohim, M.O.; Setiadi, M.A.; Budi, I. Identification of Hate Speech and Abusive Language on Indonesian Twitter Using the Word2vec, Part of Speech and Emoji Features. In Proceedings of the 1st International Conference on Advanced Information Science and System, Singapore, 15–17 November 2019; pp. 1–5.
29. Jatnika, D.; Bijaksana, M.A.; Suryani, A.A. Word2vec Model Analysis for Semantic Similarities in English Words. *Procedia Comput. Sci.* **2019**, *157*, 160–167. [CrossRef]
30. Hebart, M.N.; Zheng, C.Y.; Pereira, F.; Baker, C.I. Revealing the Multidimensional Mental Representations of Natural Objects Underlying Human Similarity Judgements. *Nat. Hum. Behav.* **2020**, *4*, 1173–1185. [CrossRef]
31. Wang, B.; Sun, Y.; Chu, Y.; Yang, Z.; Lin, H. Global-Locality Preserving Projection for Word Embedding. *Int. J. Mach. Learn. Cybern.* **2022**, *13*, 2943–2956. [CrossRef]
32. Liu, Q.; Kusner, M.J.; Blunsom, P. A Survey on Contextual Embeddings. *arXiv* **2020**, arXiv:2003.07278.
33. Kruse, R.; Mostaghim, S.; Borgelt, C.; Braune, C.; Steinbrecher, M. Multi-Layer Perceptrons. In *Computational Intelligence: A Methodological Introduction*; Springer: Cham, Switzerland, 2022; pp. 53–124.
34. Salehinejad, H.; Sankar, S.; Barfett, J.; Colak, E.; Valaee, S. Recent Advances in Recurrent Neural Networks. *arXiv* **2017**, arXiv:1801.01078.
35. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
36. Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning Phrase Representations Using RNN Encoder-Decoder for Statistical Machine Translation. *arXiv* **2014**, arXiv:1406.1078.
37. Yin, W.; Kann, K.; Yu, M.; Schütze, H. Comparative Study of CNN and RNN for Natural Language Processing. *arXiv* **2017**, arXiv:1702.01923.

38. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention Is All You Need. *Adv. Neural Inf. Process Syst.* **2017**, *30*, 6000–6010.
39. Chernyavskiy, A.; Ilvovsky, D.; Nakov, P. Transformers:"The End of History" for Natural Language Processing? In *Machine Learning and Knowledge Discovery in Databases. Research Track: European Conference, ECML PKDD 2021, Bilbao, Spain, 13–17 September 2021*; Proceedings, Part III 21; Springer: Cham, Switzerland, 2021; pp. 677–693.
40. Patwardhan, N.; Marrone, S.; Sansone, C. Transformers in the Real World: A Survey on NLP Applications. *Information* **2023**, *14*, 242. [CrossRef]
41. Goldstein, A.; Grinstein-Dabush, A.; Schain, M.; Wang, H.; Hong, Z.; Aubrey, B.; Schain, M.; Nastase, S.A.; Zada, Z.; Ham, E. Alignment of Brain Embeddings and Artificial Contextual Embeddings in Natural Language Points to Common Geometric Patterns. *Nat. Commun.* **2024**, *15*, 2768. [CrossRef] [PubMed]
42. Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; Funtowicz, M.; et al. Transformers: State-of-the-Art Natural Language Processing. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Online, 16–20 November 2020; Association for Computational Linguistics: Stroudsburg, PA, USA, 2020; pp. 38–45.
43. Galli, C.; Donos, N.; Calciolari, E. Performance of 4 Pre-Trained Sentence Transformer Models in the Semantic Query of a Systematic Review Dataset on Peri-Implantitis. *Information* **2024**, *15*, 68. [CrossRef]
44. Guizzardi, S.; Colangelo, M.T.; Mirandola, P.; Galli, C. Modeling New Trends in Bone Regeneration, Using the BERTopic Approach. *Regen. Med.* **2023**, *18*, 719–734. [CrossRef] [PubMed]
45. Wang, Z.; Chen, J.; Chen, J.; Chen, H. Identifying Interdisciplinary Topics and Their Evolution Based on BERTopic. *Scientometrics* **2023**, 1–26. [CrossRef]
46. Qian, J.; Kang, Y.; He, Y.; Hu, H. Topic Modeling Analysis of Chinese Medicine Literature on Gastroesophageal Reflux Disease: Insights into Potential Treatment. *Chin. J. Integr. Med.* **2024**, 1–9. [CrossRef]
47. Jeon, E.; Yoon, N.; Sohn, S.Y. Exploring New Digital Therapeutics Technologies for Psychiatric Disorders Using BERTopic and PatentSBERTa. *Technol. Forecast. Soc. Chang.* **2023**, *186*, 122130. [CrossRef]
48. Lindelöf, G.; Aledavood, T.; Keller, B. Dynamics of the Negative Discourse toward COVID-19 Vaccines: Topic Modeling Study and an Annotated Data Set of Twitter Posts. *J. Med. Internet Res.* **2023**, *25*, e41319. [CrossRef]
49. Li, H.; Lu, X.; Wu, Y.; Luo, J. Research on a Data Mining Algorithm Based on BERTopic for Medication Rules in Traditional Chinese Medicine Prescriptions. *Med. Adv.* **2023**, *1*, 353–360. [CrossRef]
50. Karabacak, M.; Jagtiani, P.; Carrasquilla, A.; Jain, A.; Germano, I.M.; Margetis, K. Simplifying Synthesis of the Expanding Glioblastoma Literature: A Topic Modeling Approach. *J. Neuro-Oncol.* **2024**, *169*, 601–611. [CrossRef]
51. Bramer, W.M.; De Jonge, G.B.; Rethlefsen, M.L.; Mast, F.; Kleijnen, J. A systematic approach to searching: An efficient and complete method to develop literature searches. *J. Med. Libr. Assoc. JMLA.* **2018**, *106*, 531–541. [CrossRef] [PubMed]
52. Patrick, L.J.; Munro, S. The Literature Review: Demystifying the Literature Search. *Diabetes Educ.* **2004**, *30*, 30–38. [CrossRef] [PubMed]
53. Farouk, M. Measuring Text Similarity Based on Structure and Word Embedding. *Cogn. Syst. Res.* **2020**, *63*, 1–10. [CrossRef]
54. Li, B.; Han, L. Distance Weighted Cosine Similarity Measure for Text Classification. In *Intelligent Data Engineering and Automated Learning–IDEAL 2013, Proceedings of the 14th International Conference, IDEAL 2013, Hefei, China, 20–23 October 2013*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 611–618.
55. Ivchenko, G.I.; Honov, S.A. On the Jaccard Similarity Test. *J. Math. Sci.* **1998**, *88*, 789–794. [CrossRef]
56. Pearson, K. LIII. On Lines and Planes of Closest Fit to Systems of Points in Space. *Lond. Edinb. Dublin Philos. Mag. J. Sci.* **1901**, *2*, 559–572. [CrossRef]
57. Labrín, C.; Urdinez, F. Principal Component Analysis. In *R for Political Data Science*; Chapman and Hall/CRC: Boca Raton, FL, USA, 2020; pp. 375–393.
58. Van der Maaten, L.; Hinton, G. Visualizing Data Using T-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2579–2605.
59. Xanthopoulos, P.; Pardalos, P.M.; Trafalis, T.B.; Xanthopoulos, P.; Pardalos, P.M.; Trafalis, T.B. Linear Discriminant Analysis. In *Robust Data Mining*; Springer: New York, NY, USA, 2013; pp. 27–33.
60. Friedrich, T. *Nonlinear Dimensionality Reduction with Locally Linear Embedding and Isomap*; University of Sheffield: Sheffield, UK, 2002.
61. McInnes, L.; Healy, J.; Melville, J. Umap: Uniform Manifold Approximation and Projection for Dimension Reduction. *arXiv* **2018**, arXiv:1802.03426.
62. Andersen, H.; Aass, A.M.; Wohlfahrt, J.C. Porous Titanium Granules in the Treatment of Peri-Implant Osseous Defects—A 7-Year Follow-up Study. *Int. J. Implant. Dent.* **2017**, *3*, 50. [CrossRef]
63. Wohlfahrt, J.C.; Lyngstadaas, S.P.; Rønold, H.J.; Saxegaard, E.; Ellingsen, J.E.; Karlsson, S.; Aass, A.M. Porous Titanium Granules in the Surgical Treatment of Peri-Implant Osseous Defects: A Randomized Clinical Trial. *Int. J. Oral Maxillofac. Implant.* **2012**, *27*, 401–410.
64. Emanuel, N.; Machtei, E.E.; Reichart, M.; Shapira, L. D-PLEX. *Quintessence Int.* **2020**, *51*, 546–553.
65. Renvert, S.; Giovannoli, J.; Roos-Jansåker, A.; Rinke, S. Surgical Treatment of Peri-implantitis with or without a Deproteinized Bovine Bone Mineral and a Native Bilayer Collagen Membrane: A Randomized Clinical Trial. *J. Clin. Periodontol.* **2021**, *48*, 1312–1321. [CrossRef] [PubMed]

66. Isehed, C.; Holmlund, A.; Renvert, S.; Svenson, B.; Johansson, I.; Lundberg, P. Effectiveness of Enamel Matrix Derivative on the Clinical and Microbiological Outcomes Following Surgical Regenerative Treatment of Peri-implantitis. A randomized controlled trial. *J. Clin. Periodontol.* **2016**, *43*, 863–873. [PubMed]
67. Isehed, C.; Svenson, B.; Lundberg, P.; Holmlund, A. Surgical Treatment of Peri-implantitis Using Enamel Matrix Derivative, an RCT: 3-and 5-year Follow-up. *J. Clin. Periodontol.* **2018**, *45*, 744–753. [CrossRef]
68. Renvert, S.; Roos-Jansåker, A.; Persson, G.R. Surgical Treatment of Peri-implantitis Lesions with or without the Use of a Bone Substitute—A Randomized Clinical Trial. *J. Clin. Periodontol.* **2018**, *45*, 1266–1274. [CrossRef] [PubMed]
69. Nct Peri-Implantitis—Reconstructive Surgical Therapy. 2017. Available online: https://clinicaltrials.gov/show/NCT03077061 (accessed on 10 April 2022).
70. Ahmad, N.; Nassif, A.B. Dimensionality Reduction: Challenges and Solutions. In *ITM Web of Conferences*; EDP Sciences: Les Ulis, France, 2022; Volume 43, p. 01017.
71. Sumithra, V.; Surendran, S. A Review of Various Linear and Non Linear Dimensionality Reduction Techniques. *Int. J. Comput. Sci. Inf. Technol.* **2015**, *6*, 2354–2360.
72. Zebari, R.; Abdulazeez, A.; Zeebaree, D.; Zebari, D.; Saeed, J. A Comprehensive Review of Dimensionality Reduction Techniques for Feature Selection and Feature Extraction. *J. Appl. Sci. Technol. Trends* **2020**, *1*, 56–70. [CrossRef]
73. Liu, X.; Zheng, Y.; Du, Z.; Ding, M.; Qian, Y.; Yang, Z.; Tang, J. *GPT Understands, Too*; AI Open: San Francisco, CA, USA, 2023.
74. Thirunavukarasu, A.J.; Ting, D.S.J.; Elangovan, K.; Gutierrez, L.; Tan, T.F.; Ting, D.S.W. Large Language Models in Medicine. *Nat. Med.* **2023**, *29*, 1930–1940. [CrossRef]
75. Kaddour, J.; Harris, J.; Mozes, M.; Bradley, H.; Raileanu, R.; McHardy, R. Challenges and Applications of Large Language Models. *arXiv* **2023**, arXiv:2307.10169.
76. Wei, J.; Tay, Y.; Bommasani, R.; Raffel, C.; Zoph, B.; Borgeaud, S.; Yogatama, D.; Bosma, M.; Zhou, D.; Metzler, D. Emergent Abilities of Large Language Models. *arXiv* **2022**, arXiv:2206.07682.
77. Hersh, W.R. Search Still Matters: Information Retrieval in the Era of Generative AI. *arXiv* **2023**, arXiv:2311.18550. [CrossRef]
78. Zhu, Y.; Yuan, H.; Wang, S.; Liu, J.; Liu, W.; Deng, C.; Dou, Z.; Wen, J.-R. Large Language Models for Information Retrieval: A Survey. *arXiv* **2023**, arXiv:2308.07107.
79. Hadi, M.U.; Qureshi, R.; Shah, A.; Irfan, M.; Zafar, A.; Shaikh, M.B.; Akhtar, N.; Wu, J.; Mirjalili, S. A Survey on Large Language Models: Applications, Challenges, Limitations, and Practical Usage. *Authorea Preprints* **2023**. Available online: https://www.techrxiv.org/doi/full/10.36227/techrxiv.23589741.v1 (accessed on 10 September 2024).
80. Lozano, A.; Fleming, S.L.; Chiang, C.-C.; Shah, N. Clinfo. Ai: An Open-Source Retrieval-Augmented Large Language Model System for Answering Medical Questions Using Scientific Literature. In Proceedings of the Pacific Symposium on Biocomputing 2024, Waimea, HI, USA, 3–7 January 2024; World Scientific: Singapore, 2023; pp. 8–23.
81. Agarwal, S.; Laradji, I.H.; Charlin, L.; Pal, C. LitLLM: A Toolkit for Scientific Literature Review. *arXiv* **2024**, arXiv:2402.01788.