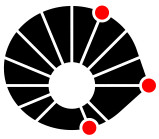

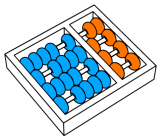

A bibliographic survey of Neural Language Models with applications in topic modeling and clinical studies

F. Leal *A. Santanchè* *C. B. Medeiros*

Technical Report - IC-24-01 - Relatório Técnico
August - 2024 - Agosto

UNIVERSIDADE ESTADUAL DE CAMPINAS
INSTITUTO DE COMPUTAÇÃO

A bibliographic survey of Neural Language Models with applications in topic modeling and clinical studies

Fagner Leal* André Santanchè† Claudia Bauzer Medeiros‡

Abstract

This text presents a literature review of Neural Language Models, which are deep neural networks to encode a given language. The scope of this review covers two main topics: (i) Transformers-based Neural Networks, established as state-of-the-art in addressing Natural Language Processing (NLP) problems and a suitable approach to train Language Models; and (ii) Neural Language Models that compress the statistical semantics of textual data into word vectors. These word vectors computationally represent the basic units of the language at hand.

In fact, obtaining a computational representation for textual constructs is a long-standing problem that has challenged diverse NLP approaches. We analyzed the usage of language models for Topic Modeling and for Semantic Annotation of Virtual Patients. The establishment of transformers-based language models opens up vast possibilities and perspectives on interdisciplinary topics. This text concludes with a critical analysis addressing issues regarding applications based on language models.

1 Introduction

Natural Language Processing (NLP) is a field of Computer Science whose goal is to convert human language into a representation that is interpretable by computers. It is an interdisciplinary research area that incorporates concepts from various other fields, such as Statistics and Linguistics.

Manning and Schütze [1] classify NLP methods into statistical and non-statistical approaches. Statistical approaches rely on patterns that commonly occur in a language, while non-statistical approaches focus on mapping and computationally implementing the rules that structure the language. The distinction between statistical and non-statistical approaches has roots grounded in the philosophical debate surrounding the perspectives of Rationalism and Empiricism [1].

In the epistemological realm [2], Rationalism claims the ideas of deductive reasoning are possible because they are innate, prior to all experience. In turn, Empiricism states that none of our ideas are innate, and the mind would be a blank tablet when we are born. Subsequently, Kant considered both the concept of active mind (from rationalism) and the role of sensations (from empiricism) as essentials in knowledge acquisition. In turn, Bertrand Russell "explicitly rejected the existence of innate ideas" [2]. The debate remains open and has led to the development of several philosophical schools.

In the field of Linguistics, the rationalist perspective is characterized by the belief in the existence of an innate language fixed in the human brain through genetic inheritance. Advocated by

*Inst. de Computação, UNICAMP, 13083-852 Campinas, SP. fagnerleal@ufpa.br. Apoio CAPES
†Inst. de Computação, UNICAMP, 13083-852 Campinas, SP. santanche@ic.unicamp.br
‡Inst. de Computação, UNICAMP, 13083-852 Campinas, SP. cmbm@ic.unicamp.br

Noam Chomsky [1], rationalism has been crucial to the development of the theory of Formal Languages, which serves as the foundation for current programming languages. Formal languages constitute a special class of language that lacks ambiguity and, therefore, can be interpreted/compiled by computers. The ability to interpret a language in a non-ambiguous manner is essential for a computer to execute commands instructed by humans through a programming code [1].

In contrast to programming languages, natural languages are inherently ambiguous, since a word or phrase can have more than one meaning [3]. In natural language cases, the empiricist perspective assumes that, instead of pre-constructed linguistic structures, the human mind possesses generic operations of association, generalization, and pattern recognition. These cognitive abilities, combined with a rich sensory system, enable humans to learn detailed language structures. This hypothesis forms the basis of Machine Learning methods that use statistical models to recognize patterns and complex structures in a dataset. This statistical approach is grounded in the Information Theory developed by Claude Shannon [1].

Manning and Schütze [1] point out that "the difference between the approaches is not absolute but one of degree", as rationalism believes "the key parts of language are innate – hardwired in the brain at birth as part of the human genetic inheritance" while empiricism believes in an innate capacity to develop language through generalizations such that "a baby's brain begins with general operations for association, pattern recognition, and generalization, and that these can be applied to a rich sensory input available to the child to learn the detailed structure of natural language".

This philosophical debate remains an open question; however, its practical utility is valuable as it theoretically underpins various areas of computer science.

More recently, statistical approaches have advanced the state-of-the-art in various NLP tasks. This progress can be attributed to, among other factors: (1) advances in computational capacity; (2) recent deep neural network models capable of retaining significantly more information than previously proposed neural models; and (3) the development of more efficient techniques for handling the vast amount of information available on the Web.

The rest of this text is organized as follows: Sections 2 and 3 give some background on foundations of Neural Networks, Deep Neural Networks, Transformers, as well as the history and development of the so-called Neural Language Models. Sections 4 and 5 review related work of two case studies that involve neural networks and transformers: Topic Modeling and Semantic Annotations of virtual patients. Section 6 briefly discusses more recent work developed in research in Language Models. This is followed by a section that critically analyzes language models addressing some interdisciplinary aspects, finishing with concluding remarks.

2 Neural Networks

Several recent advances in the field of Natural Language Processing (NLP) are attributed to the mellowing of Deep Neural Network models, which are more sophisticated types of Artificial Neural Networks. This section describes some relevant issues in Neural Network architectures, followed in the subsequent section by Language Models in the context of such networks

2.1 Artificial Neural Networks

An artificial neural network — a computational abstraction inspired by the biological nervous system — is an interconnected network of artificial neurons organized in layers. Typically, neural networks perform Supervised Learning, where the network receives successive sets of pre-labeled training samples and must infer the corresponding output for each input sample. For example, a neural network can be trained to recognize cancerous tumors in computed tomography images

based on labeled images previously presented to the model. After this training phase, the neural network is capable of making inferences about new images that were not observed by the model during its network training [4].

Example - Sentiment Analysis through Neural Networks Sentiment Analysis through neural networks is an NLP task whose objective is to classify sentences based on their sentiment polarity $C = Positive, Negative, Neutral$. In the training phase, iteratively the network is fed by pairs of $\{sentence, label\}$ contained in the training set. In each training step, let s be the sentence to be classified, y the corresponding label, and h the output of the classification, representing the class inferred for s by the algorithm. The sentence s is represented by a feature vector $x = (x_1, x_2, ..., x_n)$. Let $\theta = (\theta_1, \theta_2, ..., \theta_m)$ be a vector of parameters (weights) of each neuron. Classification works as follows. Vector x is propagated through the network's layers, adjusting the parameters θ of each neuron based on their contribution to constructing the output y. Figure 1 illustrates a neural network classifying the sentence "I liked this movie". The neural network produces an output vector $o = (o_1, o_2, o_3)$ containing the algorithm's hypotheses regarding the probabilities of the sentence belonging to each of the possible classes in C, where the highest one is chosen as the algorithm's hypothesis $h = positive$.

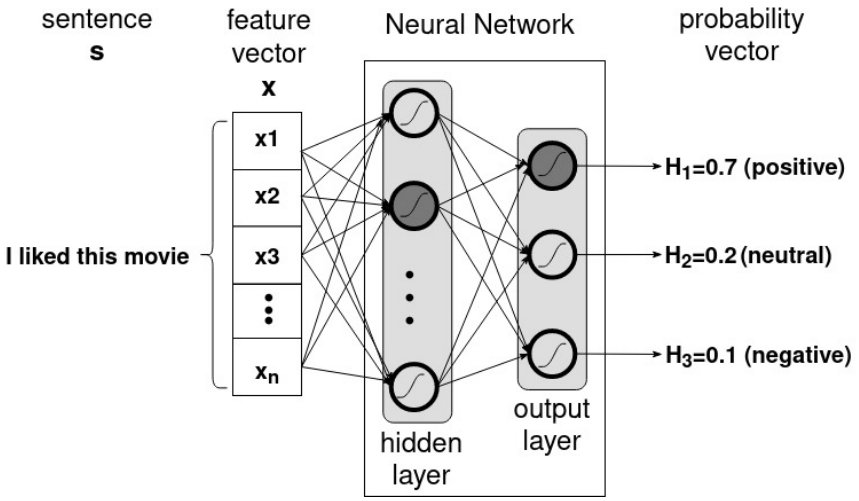

Figure 1: Neural network performing sentiment analysis. Adapted from [4].

Each neuron in the network has a transfer function (or activation function) $f(\Sigma)$ that operates on the weight parameter θ_i of the neuron and the input feature x_i, as illustrated in Figure 2. Several transfer functions can be employed, including the sigmoid function:

$$f(\Sigma) = \frac{1}{1 + e^{-\sum_{i=1}^{n} \theta_i x_i}} \tag{1}$$

During the training phase, each sentence is fed and processed in a training step. A training step of a neural network involves two main mechanisms [4]:

- Forward Propagation: Propagates the sample sentence s through the neural network until it reaches the final layer. The final layer produces the hypothesis h containing the probability of s being classified as *positive*, *neutral* or *negative*. The difference between the hypothesis h inferred by the algorithm and the true class y annotated in the training set indicates the

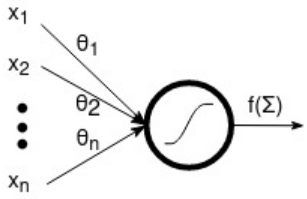

Figure 2: The functioning of the artificial neural unit.

contribution (or responsibility) of each parameter θ_i to the error measured when classifying s.

- Backpropagation: Adjusts each parameter θ_i based on its contribution to the error calculated between the hypothesis h and the true class y annotated in the training set. The larger the contribution of the neuron, the greater the adjustment in its parameters should be. The magnitude of the adjustment in the parameters θ_i can be controlled by the hyperparameter[1] η, which typically has a value close to 0.1.

The algorithm completes one training epoch when it processes all the samples in the training set. The number of epochs is also a hyperparameter. At the end of the training, the network parameters have been calibrated to solve the task for which it was trained [4].

An architecture with at least one hidden layer of neurons (as depicted in Figure 1) is known as a Multi-Layer Perceptron [4]. Other models implement different architectures, transfer functions, propagation mechanisms, etc. Deep neural networks models serve more robust architectures, including Recurrent Neural Networks, Convolutional Neural Networks, and the more recently introduced Transformers.

2.1.1 Recurrent Neural Networks

A recurrent neural network is suitable for solving problems with a sequential aspect [6], as observed in various NLP problems (e.g., sentiment analysis, Named Entity Recognition (NER), etc.) Recurrent neural networks leverage the inherent sequential aspect in textual constructs. Broadly speaking, the sequential aspect implies that each term w_i in a given sentence s depends on the preceding term w_{i-1}. For example, for a neural network handling the sentence "She is excellent at her role as a", the probability of the next word being "doctor" is immensely higher than being "and" [5], as illustrated by Equation 2.

$$P(\texttt{doctor}|\texttt{She is excellent at her role as a}) \gg P(\texttt{and}|\texttt{She is excellent at her role as a}) \quad (2)$$

The Long-Short Term Memory (LSTM) model [11] is a sophisticated type of recurrent neural network that achieves considerable success in addressing NLP problems due to its mechanism for deciding which information to retain and which to discard [6]. Thus, the LSTM is capable of capturing contextual information from terms that are distant from the term being currently processed. However, the LSTM experiences performance degradation in scenarios involving very long sentences. This issue is known as Long-Term Dependencies, which arises when a word depends on words that are far apart in the sentence.

[1]Hyperparameters are variables that control the overall behavior of the network. Do not confuse with the term "parameter" that refers to the weights assigned to each neuron in the network.

The LSTM has unidirectional contextual memory, restricting its usage to acquiring information solely from preceding terms (in the case of the left-to-right version of LSTM) or solely from subsequent terms (in the right-to-left LSTM) [7]. Additionally, as a specific type of recurrent neural network, the LSTM faces several challenges during the training phase, such as gradient explosion and gradient vanishing [8]. Furthermore, the sequential nature of the LSTM precludes the parallelization of the training process [9].

Despite these limitations, LSTM is often successfully employed in the Sequence Translation (or sequence-to-sequence) task, which aims to transform a given sequence s of arbitrary length into a corresponding sequence t of pre-defined fixed size n [10]. This fixed-size representation can be used to perform other NLP tasks (e.g., Language Modeling, Machine Translation, Speech Recognition, Question-Answering) through Transfer Learning techniques.

A significant advancement in this sequence-to-sequence task (and consequently in the NLP research field) was accomplished by the Transformer, a novel deep neural network model that overcomes the issue of long-term dependencies and provides a means to obtain bidirectional contextual memory. The next section focuses on the Transformer.

2.2 Transformers

The Transformer model, introduced in the paper "Attention is All You Need" [9], addresses the sequence-to-sequence problem more efficiently than the LSTM. The Transformer is a deep neural network model that captures the **context** through the Attention Mechanism. With this mechanism, the Transformer has advanced the state-of-the-art in NLP tasks that can be modeled as an instance of the sequence-to-sequence problem.

The Attention Mechanism The Transformer model overcomes the issue of long-term dependencies through the attention mechanism, which can capture global dependencies in the input sentence regardless of the distance between words [9]. Thus, this model can encode the context of the input sentence into the vector representation produced as output. The attention mechanism provides the Transformer with the ability to focus on words that are relevant to achieving the goal of the task being performed. There are various attention mechanisms [12, 13]. The Transformer specifically employs Self-Attention (or Intra-Attention), which involves a weighted sum of vectors resulting from successive linear transformations of the matrices Q, K, and V (query, key, and value, respectively) [14].

Matrices K and V comprehend the Neural Memory [15, 16][2] in which each row corresponds to a word (although, in other settings, it could be a character, a sentence, etc) within the training set, and the columns hold the distribution of weights [13] learned by the network. This distribution accounts for the context learned during the training phase. According to [15], "each key vector k_i captures a particular pattern (or set of patterns) in the input sequence, and that its corresponding value vector v_i represents the distribution of terms that follows said pattern".

The core of an attention mechanism is the computation over the matrices K, Q and V to infer an appropriate context representation. In each training step, the current word being processed (the query q) is matched against the K and V matrices to search for a key-value pair corresponding to the given query q. In this training process, the matrices undergo the series of operations illustrated in Figure 3, which is a visual representation of the Attention Function Equation 3 to map a query

[2] in fact, initially [15] showed the Feed-Forward sublayer can be seen as a neural memory, and [16] claimed a Feed-Forward sublayer can be seen as an attention layer, therefore we extend the concept of neural memory to refer also to the attention layer

and a set of key-value pairs into an output [9]. Here, we use the notation c to refer to this output vector.

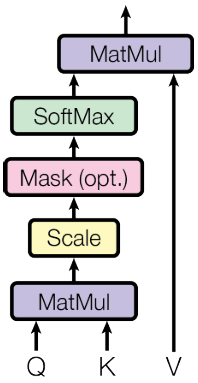

Figure 3: Operations performed by the layers of Self-Attention on the matrices Q, K, and V. Source: [9]

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_K}})V \tag{3}$$

The attention mechanism operates over the K, Q, and V matrices in order to highlight the prominent patterns observed in the training dataset. The mechanism is about matching the context of a query q (the context of the given sentence) against the most similar context accumulated on the neural memory of $\{K, V\}$ whereupon the network gets information to accomplish the demanded task (e.g., the translation task in the original Transformer model [9]). As a byproduct of translation, the attention mechanism generates word alignments [12].

As an example, Figure 4 shows an alignment matrix derived from translating an input sentence (in the column) from English to German (in the rows). This matrix helps to visualize how the attention gets the correspondences between all words in a sentence: it highlights the "attention" each word in the target sentence pays to the words in the source sentence.

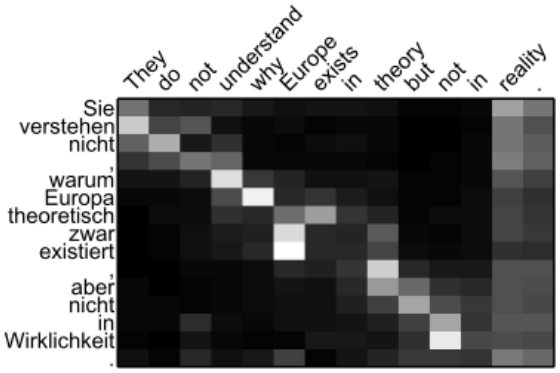

Figure 4: Word alignments derived from a translation task from English to German. Source: [12]

The original Transformer architecture (Figure 5) consists of two stacks of N layers of Encoders and Decoders. Subsequent researcy proposed different architectures [17], for example encoder-only such as BERT [7], and decoder-only such as GPT [18].

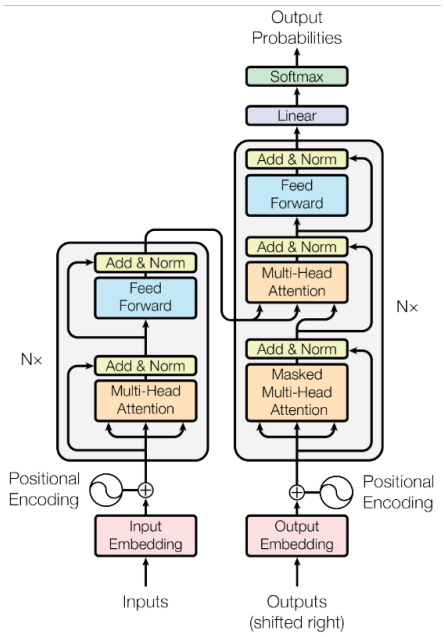

Figure 5: Transformer architecture. Source: [9]

Encoders An Encoder correlates each word $p_i \in s$ with all other words in the sentence s. Its result is a fixed-size vector representation c that encapsulates information regarding the sentence context.

The input sequence flows through the N layers of encoders. Each encoder receives Q, K, and V from the previous one. Within each encoder, there is a Multi-Head Attention unit (implementing Equation 3) and a Feed-Forward Network. The representation c generated by the last encoder is then sent to the decoder.

Decoders A Decoder uses the vector c received from the Encoder layers to generate the translated sequence t. The layers of the decoder are configured similarly to the encoder concerning the pair of multi-head attention and feed-forward layers. In addition to these, the decoder has a third layer called Masked Multi-Head Attention, responsible for ensuring that the decoder obtains information only from the preceding terms in the sequence t to preserve the auto-regressive property [9].

Multi-Head Attention In the so-called Multi-Head Attention, the Transformer projects the vectors Q, K, and V into multiple multi-heads with different learned linear projections. The resulting vectors are concatenated and projected again, resulting in the final vector representation, as depicted in Figure 6. Thus, the Transformer can pay attention to different representation subspaces at different positions [9], in addition to leveraging parallel computation.

Transfer Learning Through Transfer Learning techniques, a Transformer can be effectively applied to a range of NLP tasks, despite its initial design for sequence-to-sequence tasks. This versatility stems from the Transformer's capability to encode statistical patterns common to various NLP tasks into the matrices K, Q and V .

Fine-Tuning – a form of transfer learning – involves initial training on a source task τ_i followed by an adjustment of the learned parameters from τ_i to be applicable to solving a target task

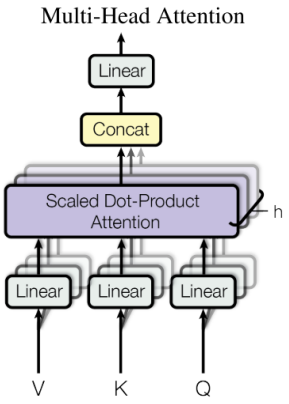

Figure 6: Schema of Multi-Head Attention. Source: [9]

$\tau_j \neq \tau_i$ [20]. Through fine-tuning, the parameters of the pre-trained model are easily adaptable to other NLP tasks. There are several approaches that implement fine-tuning, such as Google BERT [7] and OpenAI GPT [18].

The reuse of knowledge acquired in performing a generic task was first studied in computer vision research and, with the advent of the Transformer, has been extensively investigated in NLP [20]. One advancement in NLP was the realization that matrices K and V learned by the Transformer serve as a universal representation for textual constructs, capturing prominent syntactic and semantic aspects in the given textual construction. These aspects reveal idiosyncrasies embedded in the model during its training.

The neural memory of K and V embeds a Neural Language Model, which comprehends a new generation of Word Representations [19] (also known as Word Vectors, or Word Embeddings) for use in transfer learning across various work tasks. To our knowledge, the Transformer is currently the most robust and suitable model for pre-training these Language Models, which will be further explored in the next section.

3 Neural Language Models

This section discusses Language Models in the context of Neural Networks.

Given that each NLP task exhibits a distinct set of relevant features influencing its behavior and outcomes [4], it is crucial to carefully select an appropriate set of features to represent sentences given as input to the algorithm that has been selected. This stage of feature definition, known as Feature Engineering, is one of the major challenges in developing machine learning algorithms [4] and must be carefully conducted, as it has a significant impact on the algorithm's results. These features are typically stored in Feature Vectors.

In some cases, features are curated by experts, while in other scenarios features are collected through an automated process. Additionally, the criteria for feature selection are task-specific. For instance, in the case of Named Entity Recognition (NER), the features indicating whether a term should be considered a named entity encompass: term with the first letter capitalized, term preceded by a definite article, grammatical classes (e.g., noun, proper noun), prefixes and suffixes (e.g., diseases commonly ending with "it"), foreign words, term position within the sentence, term frequency in the training corpus, presence of other entities in the sentence, etc. [21].

Feature engineering often deals with the presence of redundant features, and also with features

that are important in a given context but may not be as relevant in other contexts, among other obstacles. As a result, feature engineering is typically a costly and time-consuming process.

There is a recent trend on using Word Representations to alleviate the burden of effort spent in the feature engineering phase, as they provide efficient text representations that enhance the performance of classification algorithms applied to NLP tasks. Indeed, this is a rapidly growing research area that has been experiencing intense development due to recent advances in Deep Learning methods and the increased capacity for parallel processing.

3.1 Word Vector Representation

Also informally known as Word Embedding, word vector representation is, in summary, a numeric vector used to represent a unit of text (which can be a word, a document, a paragraph, etc.) given as input to NLP algorithms. This computational representation of textual data serves as an alternative to hand-designed feature vectors generated during the initial feature engineering phase in classical NLP pipelines.

One of the pioneer models of word representation is the One-Hot Vector, which represents each word as a vector of size $|W|$ (i.e., one vector dimension for each word in the vocabulary W) containing values of 1 in the dimension corresponding to the given word, while all other dimensions receive the value 0. Table 1 provides an example of a one-hot vector encoding for four words from a hypothetical vocabulary of size $|V| = 6$.

Table 1: One-hot representation.

	heart	drug	disease	therapy	kidney	chest
heart	1	0	0	0	0	0
drug	0	1	0	0	0	0
...			...			
disease	0	0	1	0	0	0
therapy	0	0	0	1	0	0

Unfortunately, the one-hot representation is inefficient due to its high dimensionality (one dimension for each word in the vocabulary) and its nature as a sparse matrix model, as each vector is composed of 0 values in most of its dimensions [19]. Distributional Representations are alternatives to mitigate these drawbacks.

Distributional Representations are generally based on matrices whose values are relative to the distribution of words in a specific context of that word. A context can be the entire document, a section of a document, other words nearby or around the word, among others. Typically, the context is defined in terms of window size and direction [19]. For instance, a context could be characterized by a window of the last 3 terms preceding each word, or a window of terms both to the left and right of each word.

In addition to the context, it is necessary to define the metrics to be used. One of the most common metrics is the co-occurrence of a pair of words, which can be recorded in a co-occurrence matrix of dimensions $|W| * |W|$. Another possibility is to use the widely adopted Term Frequency – Inverse Document Frequency (TF-IDF), which measures how discriminative a word is in a given collection of documents, i.e., how frequent a word is in a given document while being rare in other documents in the collection.

These distributional representations are vectors that, based on the distributional hypothesis [106, 107], contain contextual information generated through word counting, so that words occurring

in the same contexts tend to have similar meanings [108]. Such representations are suitable for tasks like classification and text retrieval; however, the challenge remains open to configure them appropriately for use in sequence labeling tasks (such as NER) [19].

An alternative approach involves generating representations through unsupervised training. These representations, commonly known as Distributed Representations, are typically induced using neural language models through training on a Language Modeling task. Unlike distributional representations that count the frequency of words in a given context, distributed representations are small – i.e., usually with a size between 50 and 1000 dimensions – and dense [19] – i.e., most dimensions contain values other than 0.

3.2 The Language Modeling Task

Language Modeling is an NLP task whose goal is to estimate the probability distribution of words in a given sentence. Traditionally, this estimation is achieved by predicting the next word based on the preceding words in the sentence [22] using the chain rule of probability [23]:

$$P(w_1, w_2, ..., w_{t-1}, w_t) = P(w_1)P(w_2|w_1)P(w_3|w_1, w_2)\dots P(w_t|w_1, w_2, \dots, w_{t-1}) \quad (4)$$

This equation predicts the probability that a word w will be used at position t following the previous words $w_1, w_2, ..., w_{t-1}$ in a given sentence. It is reasonable to expect that the term "beach" has a higher likelihood than the word "jail" of being the next word used in the sentence "I like to be in this…". Stated differently, a language model assesses the probability of a specified sentence existing within the modeled language [24].

Neural language modeling produces a set of weights (i.e., parameters) that are incrementally adjusted to minimize the loss during network training. The adjusted weights are used to induce word embeddings, whose similarity to other embeddings in the vocabulary indicates that these words occur in similar contexts in the given training set [25].

Neural language models implement a form of Unsupervised Learning and therefore do not require pre-labeled input data. The scarcity of manually annotated resources is a problem for Supervised Learning-based approaches, as is the case in many NLP algorithms, since generating pre-labeled data can be a costly and time-consuming task [18].

By using unsupervised learning, language models leverage the vast amount of unlabeled text available on the web, which is an immeasurable source of linguistic knowledge to be embedded in such models. Unsupervised training allows the model to automatically learn the latent features associated with syntactic and semantic properties.

There are many approaches to training language models, such as [26, 22, 27, 28, 29]. In a pioneering work, Bengio et al. [26] proposed a distributed representation model that effectively overcame the problem known as the Curse of Dimensionality. This problem was a barrier to training language models using neural networks; in earlier models, each word in the vocabulary was treated as a random variable which resulted in a network with a massive number of parameters, making training computationally infeasible due to the high computational cost involved.

Collobert and Weston [22] introduced a convolutional neural network to jointly train various NLP tasks through semi-supervised learning. The approach combines unsupervised learning, specifically language modeling, with supervised learning of other tasks in the pipeline, such as NER, part-of-speech tagging, etc. Therefore, this work demonstrated how to utilize embeddings learned in an unsupervised manner in the training of supervised tasks, rather than manually designed features.

Mikolov et al. [27] introduced Word2Vec, a language model with a simple and efficient neural network that has few hidden layers precisely to minimize the computational complexity caused by the non-linear hidden layer of deep neural network models. Word2Vec is provided in two similar versions: Continuous Bag-of-Words (CBOW), which predicts a target word based on the context words (4 words to the left and 4 words to the right) without considering the order of these words; and Continuous Skip-Gram, which predicts a target word based on another word within a specified range. Word2Vec captures different kinds of word similarity, going beyond basic syntax regularities. By employing simple algebraic operations on the word vectors, it is possible to observe that there is a similarity between the words "big" and "bigger" in the same way as between the words "small" and "smaller". Notably, an intriguing outcome arises from the operation $vector(King) - vector(Man) + vector(Woman) \approx vector(Queen)$.

Pennington et al. [28] introduced the GloVe model (Global Vectors for Word Representation), which generates global vectors in the sense that they contain statistical information regarding the entire training corpus. It employs a hybrid approach that combines co-occurrence matrices with distributed representations.

Bojanowski et al. [29] describe FastText as an extension of the Continuous Skip-Gram model that incorporates the morphological structure of words. It represents each word based on its internal sub-terms (e.g., the vector for the word "where" is generated by summing the vectors of the n-grams <wh, whe, her, ere, re>). Thus, FastText addresses the Out-of-Vocabulary (OoV) problem by enabling the representation of words not present in the training set.

The models presented up to this point are considered static embeddings [30, 31, 32], since they assign a unique representation to each word in the vocabulary, thereby limiting their ability to handle polysemy (when a word has different meanings depending on the context in which it appears).

3.3 Context-aware Models

Recent work has considered the so-called context-sensitive word representations [30]. These contextualized language models have dynamic representation spaces, so that a specific term can have different representations depending on the specifics of the text in which it is found.

For instance, the Embeddings from Language Models (ELMo) model [33] derives context from the internal states of a bidirectional LSTM network that traverses both the right and left contexts of the current term. ELMo concatenates the internal states of the two layers to produce context-sensitive embeddings from both directions.

The Generative Pre-trained Transformer (GPT) [18] applies the Transformer architecture to a Language Modeling task to pre-train universal embeddings adaptable to various NLP tasks—such as Natural Language Inference, Question-Answering, Semantic Similarity, and Text Classification—through Fine-Tuning. Both ELMo and GPT use unidirectional language models to learn to represent context. The bidirectional context created by ELMo is a concatenation of two unidirectional contexts learned by different networks.

The BERT (Bidirectional Encoder Representations from Transformers) language model [7] employs the Transformer architecture to train on the Masked Language Modeling task – a variation of the traditional Language Modeling seen in Equation 2.4. In this task, the model receives a training sentence with one of its terms hidden by a mask (e.g., "I like this [X] and ventilated room") and it must uncover the term hidden behind the mask. By accomplishing the task objective, the network generates a Bidirectional Context that incorporates information statistically relating each term within the sentence to all neighboring terms present in the sentence. The bidirectional context captures various facets and features — e.g., long-term dependencies, hierarchical

relationships, sentiments — that are relevant to task completion [20]. The awareness of the bidirectional context is a key aspect that enabled BERT to achieve the state of the art in 11 NLP tasks [7].

3.4 Neural Models for Sentences

The arrival of word representations has inspired other approaches to generate vector representations for larger text segments, such as phrases, sentences, paragraphs, and even entire documents. Inspired by CBOW [27], the Sent2Vec model [34] generates Sentence Vectors by averaging the vectors of the constituent n-grams in the input sentence. In fact, sentence vectors generated by averaging the vectors of all words in the sentence are quite robust models [105].

Mikolov et al. [35] propose a method for encoding idiomatic expressions, i.e., terms or phrases that have a meaning derived from the composition of their components, which is different from the meanings of the individual terms. For example, the expression "Boston Globe" represents the name of a newspaper, and its meaning is distinct from the simple combination of the individual terms "Boston" and "Globe". Additionally, the article describes some interesting properties of the Skip-Gram model, such as the Additive Property, which yields semantically coherent results. For example, in the vector space produced by Skip-Gram, the result of `vector(Russia) + vector(river)` is close to `vector(Volga River)`, while `vector(Germany) + vector(capital)` is close to `vector(Berlin)`. Such observations suggest that it is possible to obtain a non-obvious understanding of language by using Vector Arithmetic on word vectors [35].

Through a generalization of Skip-Gram, the SkipThought model [36] encodes a sentence by predicting the surrounding sentences. SkipThought implements an encoder-decoder model: the encoder maps words to a sentence vector, which is then utilized by the decoder to predict the surrounding sentences.

The InferSent model [37] employs a BiLSTM siamese network with a final layer of max-pooling. InterSent works as follows: the model is trained in a supervised fashion using the Stanford Natural Language Inference (SNLI) dataset [38], surpassing the results of unsupervised methods such as Skip-Thought. The SNLI dataset comprises 570,000 sentence pairs annotated with labels `contradiction`, `entailment`, and `neutral`. InferSent results suggest that Natural Language Inference (NLI) is a highly suitable task for sentence embeddings training.

The Sentence-BERT (SBERT) model [39] uses siamese and triplet networks (i.e., different networks with tied weights) to generate sentence embeddings. The training step of Sentence-BERT takes as input a pair of sentences and a similarity value between them. Initially, Sentence-BERT applies a pooling operation on the BERT embeddings to obtain a fixed-size representation (usually 768) for each sentence. As shown in Figure 7, in the end, a single fixed-size representation is generated based on the similarity between the two representations.

4 Neural Networks and Topic Modeling

Topic Modeling [58, 59, 60, 61, 62] is an unsupervised Natural Language Processing challenge whose problem is discovering topics that represent an overview of the textual collection under analysis. A topic model explicitly represents the latent semantic structure [63] – or gist [59] – of a textual collection.

The Topics Model concept refers to a discrete probability distribution describing the connections between words, topics, and documents [64]. Topics are word combinations that demonstrate idiosyncrasies in the linguistic distribution of the corpus under analysis [65]. Topic models are explicit representations that probabilistically associate documents with topics and topics with words [58].

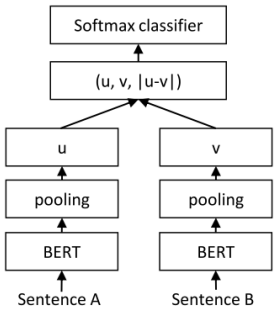

Figure 7: The architecture of the siamese network of SBERT. Source: [39].

Many methodologies address the Topic Modeling problem, such as Latent Semantic Analysis (LSA) based on linear algebra and its probabilistic version pLSA. Such methods apply dimensionality reduction to the documents represented in a Bag-of-Words format. Bag-of-words representations are adequate since, by hypothesis, word order is not a determining factor in these methods [58].

The Latent Dirichlet Allocation (LDA) [58] is a probabilistic model for discrete data collections, such as textual data. The LDA uses two complementary distributions: a topic over words distribution that describes the relationship between topics and words; and a second distribution that allocates topics to documents [64]. For LDA, a document is a random mixture of latent topics, which in turn are probability distributions over vocabulary words. The LDA is a dimensionality reduction technique [58] besides is one of the most robust and efficient methods for topic modeling.

The inferential machinery of LDA is capable of solving problems modeled in a multi-level structure, for example, Collaborative Filtering, in which the dataset comprises a collection of users, which in turn has a list of preferred objects. In this case, users and objects are analogous to documents and words in the document, respectively [58]. Therefore, LDA applies to problems beyond the textual domain.

Recent works [63, 66, 67, 68, 69, 61, 70, 71] investigate Neural Topic Modeling (NTM), a current research trend that combines topic modeling with Neural Language Models. Indeed, this is a rapidly growing area of research. [67, 69] discourse on the similarity between the topics produced via LDA and NTM. NTM-based works claim to produce more interpretable topics than prior methods, yielding improvements in the state-of-the-art concerning topic coherence measure [61].

Top2vec [63] infers topic vectors by applying vector algebra over the neural vectors of words and documents embedded in the same vector space. Each topic corresponds to a centroid of a cluster of documents and takes the closer word vectors as its most representative words. The approach infers the optimal number of topics through the HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise) algorithm.

The TopicBERT [70] topic model recognizes topics combining Transformers, community detection techniques in graphs, and named entity recognition.

[62] presented the BERTopic model based on clusters of documents grouping vectors inferred through the Sentence-BERT framework [39]. Subsequently, the approach assigns the clusters to a c-TF-IDF matrix that indicates the most representative words of each topic. c-TF-IDF is a variation of the classic TF-IDF algorithm, taking clusters as the unit of analysis instead of documents.

Although many research efforts use hierarchical clustering algorithms – such as HDBSCAN – their focus is not on the hierarchy, as they apply dimensionality reduction that interferes with hierarchical analysis. We provide an analysis of the inferred hierarchy of topics and its relationship with the arrival of Language Models based on the Transformer architecture and its attention

mechanism.

Related works have studied hierarchical structures of topics [72, 73, 74]. In particular, [72] showed important aspects of the hierarchy of topics and proposed the recursive Chinese restaurant process (rCRP) method to generate hierarchical topic structures with unbounded depth and width. Their study analyzed metrics to characterize Hierarchical Structures to organize topics, such as Hierarchical Affinity and Topic Specialization.

5 Neural Networks to Produce Semantically Annotated Virtual Patients

This section explores the use of Neural Networks to help clinical studies. More specifically, it highlights the use of automatic semantic annotations to enhance knowledge about Virtual Patients.

5.1 Semantic Annotation of Virtual Patients

In this section we give a background about Semantic Web and Named Entity Recognition (NER) approaches to perform semantic annotation on free-texts.

These topics are the foundations of our method to Semantic Annotation described on Chapter 3 of this thesis. Our paper "Harena Semantics: a framework to support Semantic Annotation in Citizen Science systems" published at the 15^{th} International Conference on Health Informatics 2022 (HealthInf) is based on that chapter. The paper is intended to enable easy semantic annotation over natural language sequences contained within Virtual Patients data.

5.1.1 Semantic Annotations

Semantic annotation [40] is the process to connect textual sentences to the resources of the Semantic Web, towards linking the natural language texts to networks of ontologies. The Semantic Web is a network designed to be manipulated both by humans and computer agents. As envisioned by Tim Berners-Lee et al., [41], it is a extension of the current Word Wide Web (not a separate one), which structures the information in a format that better enables computers to process their content in a meaningful way.

As far as we know, there is not a standardized, established definition of the approaches used for semantic annotation. Some works define them as Concept Normalization [42, 43], Entity Linking [44], Entity Typing [45], and so on.

Usually, the semantic annotation process involves some kind of Named Entity Recognition procedure. Named Entity Recognition (NER) is a Natural Language Processing (NLP) task to identify and classify entity types, such as People, Organization and Location. In the biomedical domain, research works focus on Gene, Protein, Disease, Chemical, Anatomy, etc.

There are many approaches to implement NER tasks. Recent works using statistical approaches have leveraged the NER state-of-the-art by using Deep Neural Networks to learn Language Models [22, 7]. These Language Models encode syntactic and semantic information in Word Vectors in such a way that those vectors with similar meanings have similar representations. One can build NER methods by feeding the word vectors as feature vectors to a downstream algorithm that decides if it should or not tag a given term as a named entity.

There are works [46, 47, 48] specializing BERT vectors to the biomedical and clinical domains. [47] introduce BioBERT, a BERT-based model specialized in biomedical language. BioBERT is pre-trained on large-scale biomedical corpora composed of PubMed abstracts and PMC full-text articles. BioBERT outperformed the state-of-the-art models in a bunch of experiments over NER

tasks. BioBERT trained several models to recognize different named entities (e.g., one model to handle diseases, another one to deal with proteins, and so on). We based our NER implementation on the BioBERT model, extending it to recognize multiple entities (anatomy, chemicals, and disease) in a single model.

Other initiatives also address NER by extending language models, refining word vectors, or fine-tuning the model to specialized datasets [44, 49, 42]. BERN [50] is a neural biomedical multi-type NER tool based on the BioBERT model. The model uses a separate NER model for each entity type, and then combines the results using decision rules to handle overlapping entities (i.e., when the models tag a term with different tags, for instance, two models tagging the term "androgen" as gene and chemical). Differently, our NER model does not need decision rules, since the language model itself contains the statistical context to decide which entity is the most likely to occur. It is due to our model is trained on a dataset containing all the entities of interest, avoiding training one model to each entity. The model is based only on a Statistical approach (Empiricist) since does not rely on pre-defined rules.

5.1.2 Virtual Patients authoring

In the context of Medical Education, a clinical case comprises a medical narrative of situations occurring in real clinical environments. Lecturers use clinical cases as pedagogical resources to teach clinical practices to medical students. According to Šuster [110] "a case report is a detailed description of a clinical case that focuses on rare diseases, unusual presentation of common conditions and novel treatment methods".

There is a wide spectrum of strategies to simulate patients for students' training [51, 52, 53, 54, 55]. The adopted strategy depends on the available resources, the goal expected from the training, the level of structure in the data and the desired expressiveness of the clinical narrative of simulation.

Virtual Patients (VP) [51] are designed to present scenarios and narratives of a Clinical Case, guided by computers. They represent the Clinical Case in a graph of states affording structured guidance.

The adoption of Virtual Patients enables interesting research. For example, Hege et al. present a tool to foster the acquisition of clinical reasoning skills through Virtual Patients and Concept Maps [56].

The OpenLabyrinth [53] is a system for authoring virtual patients. The OLabX project [57] – an extension of OpenLabyrinth – uses mEducator schema to discover, retrieve, share, and reuse medical educational resources.

Our approach to Virtual Patients authoring differs from related work, as it departs from a markdown-derived language, apt for human writing, reading, and annotation, combining it with automatically generated superimposed annotations produced via the semantic annotation process. In the paper we analyzed semantic annotation within the Harena system specifically designed to manage a collection of Virtual Patients.

6 More recent NLP research

We have witnessed a heated research focus around the release of ChatGPT. The ChatGPT system uses the models of the GPT family – GPT-3 [18], GPT-4 [76], etc – to perform text generation in a dialog style [75]. GPT is a decoder-only language model fine-tuned through Reinforcement Learning from Human Feedback [17].

Recent research focuses on increasing the network size by developing Large Language Models containing billions of parameters [17]. GPT-3 contains 175 billions of parameters, Llama 65 B [77], Chinchilla 70B, PaLM 540B, BLOOM 176B [78]. Studies [79, 77] have found not necessarily the larger model results to best performance at inference time, but there is a trade-off between the model and dataset sizes such that the best model would be a smaller model trained longer (i.e., on more samples).

A recent trend is to incorporate language models into larger systems. For example, DALL-E[3] [80] is a system to generate images from text prompts. Sora[4] is a system capable of generating high-fidelity videos from input text based on diffusion models [81]. Gemini [82] is a multimodal model trained on different modalities of data such as image, audio, and video. However, some systems and applications lack academic references to describe the techniques employed and details of integrating the theoretical models in the systems. Usually, the implementation details are referred to in web pages. Many of the recent research on language models are described in technical reports uploaded to repositories which do not account for peer-review processes. This difficulties the scrutiny of the real advancements in this research area and the establishment of a reliable ground of scientific validation.

There are works focusing on demonstrating the linguistic capabilities of language models. The results of the study by Tenney et al. [83], suggest that the initial layers of BERT networks concentrate on basic syntactic information, while the higher layers focus on high-level semantic information. Ettinger [84] applies tests based on psycholinguistic studies to assess the language models' ability to capture linguistic features. The results suggest that probability distributions are sensitive to linguistic distinctions, such as semantic roles, pragmatic reasoning, common sense, etc. These aspects would be evidence of idiosyncrasies embedded in the model during its training.

Diverse surveys aim to review the methods and techniques employed on the last released large language models. Other surveys [85, 86, 87] address the evaluation of language models. The work of [86] categorizes a bunch of methods for the evaluation of language models in terms of faithful explainability.

There are works analyzing the language capabilities of language models and comparing their procedures with the functioning of the human brain. For example, Sejnowski [88] hypothesized that the intelligence of language models is a mirror that reflects the intelligence of the person using such a system. The paper of [89] claims that, although GPT models lack mechanisms of consciousness from a cognitive science perspective, they have already passed the Turing test and therefore can successfully imitate human language capabilities.

However, there is neither consensus nor definition about which type of analysis should be employed to evaluate the language models. This state of affairs is perhaps not surprising, since neural networks are examples of Complex Systems [90] and therefore, are essentially holistic and interdisciplinary. Thus, neural networks for language models would be machines "as complex as the systems they model and therefore they will be equally difficult to analyse" [90]. This situation resembles the difficulty of validating models based on a relativist, holistic philosophy of science [2]. By such approach the "The criterion of practical use has taken the place of formal rigor [...] validation becomes a semiformal, conversational process" than "a matter of formal accuracy". Therefore, the emergence of neural language models demand the research and development of new validation methods to assess their capacities, considering their holistic and interdisciplinary nature.

[3] https://openai.com/dall-e-3

[4] https://openai.com/sora#research

7 Critical Analysis

Up to this point, we concentrated on the Transformer and its capability to handle global context in input sequences. This advantage enables the Transformer to successfully train Language Models, leveraging huge amounts of data. However, there are open issues: (i) the difficulty to interpret [91] the inner workings of neural networks, and consequently, of transformers; (ii) the distributed nature [90] of Language Models hinders control of the patterns represented (for example, societal biases [92]).

The Attention Mechanism and the Transformer model have elevated the state-of-the-art in various NLP tasks that have long been challenges in this research area. Such improvements suggest that a new level of language understanding can be achieved by using attention-based models to capture the patterns that structure textual sentences. The invention of attention-based language models can be stated as a revolution in the NLP research field, symbolizing a significant technological leap forward in this field. However, there are different perspectives regarding the actual advancements achieved in a scientific context beyond computer science.

In a position paper, Bender and Koller [93] argue that language models are not, a priori, capable of understanding the **real meaning** of processed texts, as they are trained only on textual forms (i.e., the linguistic signal). This is based on the definition of meaning as the relationship between a linguistic form and an intention of communication. This would imply that there is a portion of meaning attributed to extra-textual information not present in the training set. The authors draw attention to the imprudent use of certain terms (such as understanding, comprehension, etc.) as academic terminology when reporting research results in the field.

On the other hand, Sahlgren and Carlsson [94] argue that if meaning produces effects on form, then a language model should at least be able to observe and learn these effects.

This debate addresses issues that have historically been studied in various research areas. Therefore, it is necessary to analyze the results obtained by this so-called NLP revolution with caution, as it raises expectations and interests from different actors in society—companies, states, political groups, and even the public at large.

For example, AlphaFold [95] is a Google project aimed at predicting the 3D structure of proteins—an essential challenge within Biology [96]—using neural language models. These proteins could be applied in projects for new therapies for infectious diseases, less allergenic foods, and also for potential malicious applications—such as the development of toxic proteins as biological weapons [97]. There are also examples of work in Law [98, 99] or in Geosciences and Petroleum Engineering [100]. The work by McGuffie and Newhouse [109], warns about potential language models trained to generate content based on radical ideologies (white supremacy, anti-Semitism, etc.) with the aim of disseminating extremist thoughts.

Given that neural language models are trained to recognize prominent patterns in the training set, it is expected that they capture — and consequently reproduce — racist, classist, sexist, misogynistic, homophobic, xenophobic biases, and other patterns historically perpetuated in society. For example, the results from Silva et al. [92] indicate that Transformers exhibit a statistically significant tendency to infer female and Afro-American subjects in contexts of emotive words, thus highlighting an embedded racial bias in these opaque-box models.

The existence of bias in neural models raises concerns, in particular in sensitive situations such as the development of methods for automatic student evaluation [101], the classification of patients with Opioid Use Disorder using longitudinal health data [102], or the exploration of the connection between cannabis use and depression disorder through Twitter post content [103], and so on. Research applying language models to study such complex and interdisciplinary topics should take into account the knowledge and perspectives of other fields to avoid oversimplifications and

the establishment of spurious correlations.

In this sense, it is crucial to address ethical issues in Artificial Intelligence. This includes implementing best practices for developing open-source Neural Language Models. This is essential to ensure individual freedom in an era where we store a multitude of personal information on the Internet. It also helps prevent the complexity [104] of this new technology from being used to mask biases and interests. Open-source code allows, to some extent, auditability of inferred classifications and coded rules, enabling the verification of results.

The introduction of machine learning algorithms everywhere, and NLP in particular, can impact the way we work, relate, learn, and develop. Therefore, there is a need for education at all levels aimed at teaching people how to use, understand, develop, and consume these tools in a healthy manner. Considering the breadth of the impacts that neural language models can have on human life, a pedagogical project is needed to guide towards a sustainable and ethical use of neural language models that also serve to address real and widely discussed societal problems, rather than solely serving the economic and market interests of the few who hold and dominate this technology.

8 Conclusion

The research efforts over the past decades on neural networks have led to the establishment of the Transformer model. A key aspect of the Transformer is its awareness of the global context within the training collection by "paying attention" to all the terms surrounding the current term being processed (not only to the n-grams to left or right as in the previously proposed approaches).

In this report, we reviewed the Transformer as a suitable approach to train Language Models that efficiently compress the global context of text collections by encoding the statistical patterns prominent in the language. Diverse long-standing NLP problems were suddenly solved by approaches applying transformers-based language models. This demonstrates and corroborates the impressive performance of Neural Language Models. Here we reviewed two of these tasks: Named Entity Recognition and Topic Modeling.

The arrival of neural language models shed many paths of discovery and improvements in processes, studies, and scientific discovery. Despite their undeniable success, the complexity and innovation of such technological tools raises concerns about: (i) evaluation of such models; and (ii) their ethical applicability – as voted by the November 2021 Unesco Assembly [111]

References

[1] C. Manning and H. Schütze, *Foundations of statistical natural language processing,* MIT press (1999).

[2] Y. Barlas and S. Carpenter, *Philosophical roots of model validation: two paradigms,* Wiley Online Librar (1990).

[3] B. Goertzel, *Chaotic logic: Language, thought, and reality from the perspective of complex systems science,* Springer Science & Business Media (2013).

[4] M. Kubat, *An introduction to machine learning,* Springer (2017).

[5] C. Thrampoulidis, *Implicit Bias of Next-Token Prediction,* arXiv preprint (2024).

[6] S. Ruder, *Neural transfer learning for natural language processing,* NUI Galway (2019).

[7] J. Devlin, M. Chang, K. Lee, and K. Toutanova, *Bert: Pre-training of deep bidirectional transformers for language understanding,* arXiv preprint (2018).

[8] R. Pascanu, T. Mikolov and Y. Bengio, *On the difficulty of training recurrent neural networks,* International conference on machine learning (2013).

[9] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, L. Kaiser and I. Polosukhin, *Attention is all you need,* arXiv preprint (2017).

[10] I. Sutskever, O. Vinyals and Q. Le, *Sequence to sequence learning with neural networks,* Advances in neural information processing systems (2014).

[11] S. Hochreiter and J. Schmidhuber, *Long short-term memory,* MIT Press (1997).

[12] M. Luong, H. Pham and C. Manning, *Effective approaches to attention-based neural machine translation,* arXiv preprint (2015).

[13] A. Galassi, M. Lippi and P. Torroni, *Attention in natural language processing,* IEEE Transactions on Neural Networks and Learning Systems (2020).

[14] G. Kobayashi, T. Kuribayashi, S. Yokoi and K. Inui, *Attention is Not Only a Weight: Analyzing Transformers with Vector Norms,* Conference on Empirical Methods in Natural Language Processing (EMNLP) (2020).

[15] M. Geva, R. Schuster, J. Berant and O. Levy, *Transformer feed-forward layers are key-value memories,* arXiv preprint (2020).

[16] S. Sukhbaatar, E. Grave, G. Lample, H. Jegou and A. Joulin, *Augmenting self-attention with persistent memory,* arXiv preprint (2019).

[17] S. Tian, Q. Jin, L. Yeganova, P. Lai, Q. Zhu, X. Chen, Y. Yang, Q. Chen, W. Kim and D. Comeau, *Opportunities and challenges for ChatGPT and large language models in biomedicine and health,* Briefings in Bioinformatics (2024).

[18] A. Radford, K. Narasimhan, T. Salimans and I. Sutskever, *Improving language understanding with unsupervised learning,* Technical report, OpenAI (2018).

[19] J. Turian, L. Ratinov and Y. Bengio, *Word representations: a simple and general method for semi-supervised learning,* Annual Meeting of the Association for Computational Linguistics (2010).

[20] J. Howard and S. Ruder, *Universal language model fine-tuning for text classification,* arXiv preprint (2018).

[21] D. Nadeau and S. Sekine, *A survey of named entity recognition and classification,* Lingvisticae Investigationes (2007).

[22] R. Collobert and J. Weston, *A unified architecture for natural language processing: Deep neural networks with multitask learning,* International Conference on Machine Learning (2008).

[23] Y. Bengio, *Neural net language models,* `http://www.scholarpedia.org/article/Neural_net_language_models` (2008).

[24] C. Huyen, *Evaluation Metrics for Language Modeling,* hhttps://thegradient.pub/understanding-evaluation-metrics-for-language-models/ (2019).

[25] D. Jurafsky and J. Martin, *Speech and Language Processing. An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition,* MIT Press (2000).

[26] Y. Bengio, R. Ducharme, P. Vincent and C. Janvin, *A neural probabilistic language model,* The Journal of Machine Learning Research (2003).

[27] T. Mikolov, K. Chen, G. Corrado and J. Dean, *Efficient estimation of word representations in vector space,* arXiv preprint (2013).

[28] J. Pennington, R. Socher and C. Manning, *Glove: Global vectors for word representation,* Conference on Empirical Methods in Natural Language Processing (EMNLP) (2014).

[29] P. Bojanowski, E. Grave, A. Joulin and T. Mikolov, *Enriching word vectors with subword information,* Transactions of the Association for Computational Linguistics (2017).

[30] K. Ethayarajh, *How contextual are contextualized word representations? comparing the geometry of BERT, ELMo, and GPT-2 embeddings,* Transactions of the Association for Computational Linguistics (2019).

[31] K. Kalyan, S. Sangeetha, *Secnlp: A survey of embeddings in clinical natural language processing,* Journal of Biomedical Informatics (2020).

[32] N. Tawfik and M. Spruit, *Evaluating sentence representations for biomedical text: Methods and experimental results,* Journal of Biomedical Informatics (2020).

[33] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee and L. Zettlemoyer, *Deep contextualized word representations,* arXiv preprint (2018).

[34] M. Pagliardini, P. Gupta and M. Jaggi, *Unsupervised learning of sentence embeddings using compositional n-gram features,* arXiv preprint (2017).

[35] T. Mikolov, I. Sutskever, K. Chen, G. Corrado and J. Dean, *Distributed representations of words and phrases and their compositionality,* Advances in Neural Information Processing Systems (2013).

[36] R. Kiros, Y. Zhu, R. Salakhutdinov, R. Zemel, R. Urtasun, A. Torralba and S. Fidler, *Skip-thought vectors,* Advances in Neural Information Processing Systems (2015).

[37] R. Kiros, Y. Zhu, R. Salakhutdinov, R. Zemel, R. Urtasun, A. Torralba and S. Fidler, *Supervised learning of universal sentence representations from natural language inference data,* arXiv preprint (2017).

[38] S. Bowman, G. Angeli, C. Potts and C. Manning, *A large annotated corpus for learning natural language inference,* arXiv preprint (2015).

[39] N. Reimers and I. Gurevych, *Sentence-bert: Sentence embeddings using siamese bert-networks,* arXiv preprint (2019).

[40] C. Jonquet, N. Shah, C. Youn, M. Musen, C. Callendar and M. Storey, *NCBO annotator: semantic annotation of biomedical data,* International Semantic Web Conference (2009).

[41] T. Berners-Lee, J. Hendler and O. Lassila, *The Semantic Web,* Scientific American (2001).

[42] Z. Miftahutdinov, A. Kadurin, R. Kudrin and E. Tutubalina, *Drug and disease interpretation learning with biomedical entity representation transformer,* arXiv preprint (2021).

[43] R. Doğan, R. Leaman and Z. Lu, *NCBI disease corpus: a resource for disease name recognition and concept normalization,* Journal of Biomedical Informatics (2014).

[44] M. Basaldella, F. Liu, E. Shareghi and N. Collier, *COMETA: A corpus for medical entity linking in the social media,* arXiv preprint (2020).

[45] E. Choi, O. Levy, Y. Choi and L. Zettlemoyer, *Ultra-fine entity typing,* arXiv preprint (2018).

[46] E. Alsentzer, J. Murphy, W. Boag, W. Weng, D. Jin, T. Naumann and M. McDermott, *Publicly available clinical BERT embeddings,* arXiv preprint (2019).

[47] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. So and J. Kang, *BioBERT: a pre-trained biomedical language representation model for biomedical text mining,* Bioinformatics (2020).

[48] L. Akhtyamova, P. Martínez, K. Verspoor and J. Cardiff, *Testing contextualized word embeddings to improve NER in Spanish clinical case narratives,* IEEE Access (2020).

[49] Y. Lyu and J. Zhong, *DSMER: A Deep Semantic Matching Based Framework for Named Entity Recognition,* European Conference on Information Retrieval (2021).

[50] D. Kim, J. Lee, C. So, H. Jeon, M. Jeong, Y. Choi, W. Yoon, M. Sung and J. Kang, *A neural named entity recognition and multi-type normalization tool for biomedical text mining,* IEEE Access (2019).

[51] D. Cook and M. Triola, *Virtual patients: a critical literature review and proposed next steps,* Medical Education (2009).

[52] K. Freeman, S. Thompson, E. Allely, A. Sobel, S. Stansfield and W. Pugh, *A virtual reality patient simulation system for teaching emergency response skills to US Navy medical providers,* Prehospital and Disaster medicine (2001).

[53] T. Călinici and V. Muntean, *Open labyrinth–a web application for medical education using virtual patients,* Applied Medical Informatics (2010).

[54] A. Takeuchi, T. Kobayashi, M. Hirose, T. Masuda, T. Sato and I. Ikeda, *Arterial pulsation on a human patient simulator improved students' pulse assessment,* Journal of Biomedical Science and Engineering (2012).

[55] T. Chaplin, B. Thoma, A. Petrosoniak, K. Caners, T. McColl, C. Forristal, C. Dakin, J. Deshaies, E. Raymond-Dufresne and M. Fotheringham, *Simulation-based research in emergency medicine in Canada: priorities and perspectives,* Journal of the Canadian Association of Emergency Physicians (2020).

[56] I. Hege, A. Kononowicz and M. Adler, *A clinical reasoning tool for virtual patients: design-based research study,* JMIR Medical Education (2017).

[57] E. Dafli, P. Antoniou, L. Ioannidis, N. Dombros, D. Topps and P. Bamidis, *Virtual patients on the semantic Web: a proof-of-application study,* Journal of Medical Internet Research (2015).

[58] D. Blei, A. Ng and M. Jordan, *Latent Dirichlet Allocation,* The Journal of Machine Learning Research (2003).

[59] T. Griffiths, M. Steyvers and J. Tenenbaum, *Topics in semantic representation,* Psychological Review (2007).

[60] D. Mimno, H. Wallach, E. Talley, M. Leenders and A. McCallum, *Optimizing semantic coherence in topic models,* Conference on Empirical Methods in Natural Language Processing (2011).

[61] A. Hoyle, P. Goel, A. Hian-Cheong, D. Peskov, J. Boyd-Graber and P. Resnik, *Is Automated Topic Model Evaluation Broken? The Incoherence of Coherence,* Advances in Neural Information Processing Systems (2021).

[62] M. Grootendorst, *BERTopic: Neural topic modeling with a class-based TF-IDF procedure,* arXiv preprint (2022).

[63] D. Angelov, *Top2vec: Distributed representations of topics,* arXiv preprint (2020).

[64] J. Boyd-Graber, Y. Hu and D. Mimno, *Applications of topic models,* Foundations and Trends® in Information Retrieval (2017).

[65] G. Bouma, *Normalized (pointwise) mutual information in collocation extraction,* Biennial GSCL Conference (2009).

[66] A. Dieng, F. Ruiz and D. Blei, *Topic modeling in embedding spaces,* Transactions of the Association for Computational Linguistics (2020).

[67] L. Thompson and D. Mimno, *Topic modeling with contextualized word representation clusters,* arXiv preprint (2020).

[68] N. Peinelt, D. Nguyen, and M. Liakata, *tBERT: Topic models and BERT joining forces for semantic similarity detection,* Annual Meeting of the Association for Computational Linguistics (2020).

[69] H. Zhao, D. Phung, V. Huynh, Y. Jin, L. Du and W. Buntine, *tBERT: Topic models and BERT joining forces for semantic similarity detection,* arXiv preprint (2021).

[70] M. Asgari-Chenaghlu, M. Feizi-Derakhshi, M. Balafar and C. Motamed, *TopicBERT: A cognitive approach for topic detection from multimodal post stream using BERT and memory–graph,* Chaos, Solitons & Fractals (2021).

[71] A. Kulkarni, A. Hengle, P. Kulkarni and M. Marathe, *Cluster Analysis of Online Mental Health Discourse using Topic-Infused Deep Contextualized Representations,* International Workshop on Health Text Mining and Information Analysis (2021).

[72] J. Kim, D. Kim, S. Kim and A. Oh, *Modeling topic hierarchies with the recursive chinese restaurant process,* International Conference on Information and Knowledge Management (2012).

[73] M. Isonuma, J. Mori, D. Bollegala and I. Sakata, *Tree-structured neural topic model,* Annual Meeting of the Association for Computational Linguistics (2020).

[74] H. Yan, L. Gui and Y. He, *Hierarchical interpretation of neural text classification,* Computational Linguistics (2022).

[75] Y. Liu, T. Han, S. Ma, J. Zhang, Y. Yang, J. Tian, H. He, A. Li, M. He and Z. Liu, *Summary of chatgpt-related research and perspective towards the future of large language models,* Meta-Radiology (2023).

[76] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, *Gpt-4 technical report,* arXiv preprint (2023).

[77] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro and F. Azhar, *Llama: Open and efficient foundation language models,* arXiv preprint (2023).

[78] T. Le Scao, A. Fan, C. Akiki, E. Pavlick, S. Ilić, D. Hesslow, R. Castagné, A. Luccioni, F. Yvon and M. Gallé, *Bloom: A 176b-parameter open-access multilingual language model,* arXiv preprint (2022).

[79] J. Hoffmann, S. Borgeaud, A. Mensch, E. Buchatskaya, T. Cai, E. Rutherford, D. Casas, L. Hendricks, J. Welbl and A. Clark, *Training compute-optimal large language models,* arXiv preprint (2022).

[80] J. Betker, G. Goh, L. Jing, T. Brooks, J. Wang, L. Li, L. Ouyang, J. Zhuang, J. Lee and Y. Guo, *Improving image generation with better captions,* Computer Science (2023).

[81] W. Peebles and S. Xie, *Scalable diffusion models with transformers,* International Conference on Computer Vision (2023).

[82] R. Anil, S. Borgeaud, Y. Wu, J. Alayrac, J. Yu, R. Soricut, J. Schalkwyk, A. Dai and A. Hauth, *Gemini: a family of highly capable multimodal models,* arXiv preprint (2023).

[83] I. Tenney, P. Xia, B. Chen, A. Wang, A. Poliak, R. McCoy, N. Kim, B. Van Durme, S. Bowman and D. Das, *What do you learn from context? probing for sentence structure in contextualized word representations,* arXiv preprint (2019).

[84] A. Ettinger, *What BERT is not: Lessons from a new suite of psycholinguistic diagnostics for language models,* Transactions of the Association for Computational Linguistics (2020).

[85] F. Xu, Q. Lin, J. Han, T. Zhao, J. Liu and E. Cambria, *Are large language models really good logical reasoners? a comprehensive evaluation from deductive, inductive and abductive views,* arXiv preprint (2023).

[86] Q. Lyu, M. Apidianaki and C. Callison-Burch, *Towards faithful model explanation in nlp: A survey,* Computational Linguistics (2024).

[87] Y. Chang, X. Wang, J. Wang, Y. Wu, L. Yang, K. Zhu, H. Chen, X. Yi, C. Wang and Y. Wang, *A survey on evaluation of large language models,* Transactions on Intelligent Systems and Technology (2023).

[88] T. Sejnowski, *Large language models and the reverse turing test,* Neural Computation (2023).

[89] G. Dodig-Crnkovic, *How GPT Realizes Leibniz's Dream and Passes the Turing Test without Being Conscious,* Computer Sciences & Mathematics Forum (2023).

[90] S. Scott, *Approaching complexity,* Faraday Discussions (2023).

[91] Z. Lipton, *The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery,* Queue (2018).

[92] A. Silva, P. Tambwekar and M. Gombolay, *Towards a comprehensive understanding and accurate evaluation of societal biases in pre-trained transformers,* Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (2021).

[93] E. Bender and A. Koller, *Climbing towards NLU: On meaning, form, and understanding in the age of data,* Annual Meeting of the Association for Computational Linguistics (2020).

[94] M. Sahlgren and F. Carlsson, *The singleton fallacy: Why current critiques of language models miss the point,* Frontiers in Artificial Intelligence (2021).

[95] M. AlQuraishi, *AlphaFold at CASP13,* Bioinformatics (2019).

[96] E. Callaway, *It will change everything': DeepMind's AI makes gigantic leap in solving protein structures,* Nature (2020).

[97] J. Vig, A. Madani, L. Varshney, C. Xiong, R. Socher and N. Rajani, *Bertology meets biology: Interpreting attention in protein language models,* arXiv preprint (2020).

[98] I. Chalkidis, M. Fergadiotis, P. Malakasiotis, N. Aletras and I. Androutsopoulos, *LEGAL-BERT: The muppets straight out of law school,* arXiv preprint (2020).

[99] E. Elwany, D. Moore and G. Oberoi, *Bert goes to law school: Quantifying the competitive advantage of access to large legal corpora in contract understanding,* arXiv preprint (2019).

[100] J. Massot, *How Named Entity Recognition and Document Comprehension Unlock Geosciences and Engineering Semantic Search without Big Data,* EAGE Digitalization Conference and Exhibition (2020).

[101] E. Mayfield and A. Black, *Should you fine-tune BERT for automated essay scoring?,* Workshop on Innovative Use of NLP for Building Educational Applications (2020).

[102] S. Fouladvand, J. Talbert, L. Dwoskin, H. Bush, A. Meadows, L. Peterson, R. Kavuluru and L. Chen, *Predicting Opioid Use Disorder from Longitudinal Healthcare Data using Multi-stream Transformer,* arXiv preprint (2021).

[103] S. Yadav, U. Lokala, R. Daniulaityte, K. Thirunarayan, F. Lamy and A. Sheth, *When they say weed causes depression, but it's your fav antidepressant": Knowledge-aware Attention Framework for Relationship Extraction,* PloS one (2021).

[104] J. Assange, *Cypherpunks: liberdade e o futuro da internet,* Boitempo Editorial (2015).

[105] T. Kenter, A. Borisov and M. De Rijke, *Siamese cbow: Optimizing word embeddings for sentence representations,* arXiv preprint (2016).

[106] J. Firth, *A synopsis of linguistic theory,* Studies in Linguistic Analysis (1957).

[107] Z. Harris, *Distributional Structure,* Word (1954).

[108] M. Baroni, G. Dinu and G. Kruszewski, *Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors,* Annual Meeting of the Association for Computational Linguistics (2014).

[109] K. McGuffie and A. Newhouse, *The radicalization risks of GPT-3 and advanced neural language models,* arXiv preprint (2020).

[110] S. Šuster and W. Daelemans, *Clicr: A dataset of clinical case reports for machine reading comprehension,* arXiv preprint (2018).

[111] Unesco Assembly *Ethics of Artificial Intelligence - a Unesco Recommendation.* November (2021). https://www.unesco.org/en/artificial-intelligence/recommendation-ethics