AI Applications Lecture 7
Embedding Layer and Distances/Similarities

SUZUKI, Atsushi
Jing WANG

2025-09-19

Contents

1 Introduction

1.1 Review

In the previous lecture, we focused on the **token generator**, learning about **sampling**, which maps the **continuous output of a neural network** (such as a probability distribution for the next token) to a **discrete object in natural language** (a specific token). In this lecture, we reverse the perspective and make explicit the fact that **when using a neural network as the core of a token generator, it implicitly converts discrete objects (tokens) into continuous objects (real-valued vectors)**. The component responsible for this is the **embedding layer**.

1.2 Learning Outcomes

By the end of this lecture, students should be able to:

- Explain **what** an embedding layer does.
- **Calculate** the **distances and similarities** (Euclidean distance, standard inner product, cosine similarity) between token representations (vectors) obtained through embedding.

2 Mathematical Notations

Here is a review of the basic notation used in this lecture.

- **Definition:**
  - $(\mathrm{LHS}) := (\mathrm{RHS})$: Indicates that the left-hand side (LHS) is defined by the right-hand side (RHS). For example, $a := b$ means a is defined as b.
- **Set:**
  - Sets are often denoted by uppercase calligraphic letters. E.g., $\mathcal{A}$.
  - $x \in \mathcal{A}$: Indicates that the element x belongs to the set $\mathcal{A}$.
  - $\{\}$: The empty set.
  - $\{a, b, c\}$: The set consisting of elements a, b, c (roster notation).
  - $\{x \in \mathcal{A} | P(x)\}$: The set of elements in $\mathcal{A}$ for which the proposition $P(x)$ is true (set-builder notation).
  - $|\mathcal{A}|$: The number of elements in set $\mathcal{A}$ (in this lecture, generally used only for finite sets).

- $\mathbb{R}$: The set of all real numbers.
- $\mathbb{R}_{>0}$: The set of all positive real numbers.
- $\mathbb{R}_{\geq 0}$: The set of all non-negative real numbers.
- $\mathbb{Z}$: The set of all integers.
- $\mathbb{Z}_{>0}$: The set of all positive integers.
- $\mathbb{Z}_{\geq 0}$: The set of all non-negative integers.
- $[1, k]_{\mathbb{Z}}$: When k is a positive integer, $[1, k]_{\mathbb{Z}} \coloneqq \{1, 2, \ldots, k\}$, i.e., the set of integers from 1 to k. When $k = +\infty$, $[1, k]_{\mathbb{Z}} \coloneqq \mathbb{Z}_{>0}$, i.e., the set of all positive integers.

- **Function:**
  - $f : \mathcal{X} \to \mathcal{Y}$: Indicates that the function f is a map that takes an element from set $\mathcal{X}$ as input and outputs an element from set $\mathcal{Y}$.
  - $y = f(x)$: Indicates that the output is $y \in \mathcal{Y}$ when $x \in \mathcal{X}$ is input to function f.

- **Vector:**
  - In this course, a vector refers to a column of numbers.
  - Vectors are denoted by bold italic lowercase letters. E.g., $\boldsymbol{v}$.
  - $\boldsymbol{v} \in \mathbb{R}^n$: Indicates that the vector $\boldsymbol{v}$ is an n-dimensional real-valued vector.
  - The i-th element of vector $\boldsymbol{v}$ is denoted as v_i.

$$\boldsymbol{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}. \tag{1}$$

  - For two vectors $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^{d_{\mathrm{emb}}}$, the standard inner product

- **Sequence:**
  - Given a set $\mathcal{A}$, an $n \in \mathbb{Z}_{>0} \cup \{+\infty\}$, and a function $\boldsymbol{a} : [1, n]_{\mathbb{Z}} \to \mathcal{A}$, we call $\boldsymbol{a}$ a sequence of length n with elements from $\mathcal{A}$. When $n < +\infty$, it is a finite sequence, and when $n = \infty$, it is an infinite sequence.
  - Sequences are denoted by bold italic lowercase letters, similar to vectors. This is because a finite sequence can be seen as an extension of a real-valued vector. In fact, a finite sequence of real numbers can be regarded as a real-valued vector.
  - For a sequence $\boldsymbol{a}$ of length n with elements from $\mathcal{A}$, and for $i \in [1, n]_{\mathbb{Z}}$, the i-th component a_i is defined as $a_i \coloneqq \boldsymbol{a}(i)$.

  - When $n < +\infty$, a sequence $\boldsymbol{a}$ of length n with elements from $\mathcal{A}$ is determined by its elements $a_1, a_2, ..., a_n$, and is written as $\boldsymbol{a} = (a_1, a_2, ..., a_n)$. Similarly, if $\boldsymbol{a}$ is an infinite sequence, we write $\boldsymbol{a} = (a_1, a_2, ...)$.
  - The length of a sequence $\boldsymbol{a}$ is written as $|\boldsymbol{a}|$.

- **Matrix:**
  - Matrices are denoted by bold italic uppercase letters. E.g., $\boldsymbol{A}$.
  - $\boldsymbol{A} \in \mathbb{R}^{m,n}$: Indicates that the matrix $\boldsymbol{A}$ is a real-valued matrix with m rows and n columns.
  - The element in the i-th row and j-th column of matrix $\boldsymbol{A}$ is denoted as $a_{i,j}$.

$$\boldsymbol{A} = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{bmatrix}. \tag{2}$$

  - The transpose of a matrix $\boldsymbol{A}$ is denoted as $\boldsymbol{A}^\top$. If $\boldsymbol{A} \in \mathbb{R}^{m,n}$, then $\boldsymbol{A}^\top \in \mathbb{R}^{n,m}$, and

$$\boldsymbol{A}^\top = \begin{bmatrix} a_{1,1} & a_{2,1} & \cdots & a_{m,1} \\ a_{1,2} & a_{2,2} & \cdots & a_{m,2} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1,n} & a_{2,n} & \cdots & a_{m,n} \end{bmatrix}. \tag{3}$$

  - A vector is a matrix with one column, and its transpose can also be defined.

$$\boldsymbol{v}^\top = \begin{bmatrix} v_1 & v_2 & \cdots & v_n \end{bmatrix} \in \mathbb{R}^{1,n}. \tag{4}$$

- **Tensor:**
  - In this lecture, the term tensor simply refers to a multi-dimensional array. A vector can be considered a 1st-order tensor, and a matrix a 2nd-order tensor. Tensors of 3rd order or higher are denoted by an underlined bold italic uppercase letter, like $\underline{\boldsymbol{A}}$.
  - Students who have already learned about abstract tensors in mathematics or physics might feel uncomfortable calling a mere multi-dimensional array a tensor. However, if we consider that the basis is always fixed to the standard basis, we can identify the mathematical tensor with its component representation (which becomes a multi-dimensional array), thus maintaining (some) terminological consistency.

3 One-Hot Encoding

3.1 Motivation

The **appropriate distance relationships between tokens** in natural language are not known in advance. Therefore, to **treat all tokens symmetrically (equally)**, we use the simplest and most neutral representation, the **one-hot vector**.

Definition 3.1 (One-Hot Encoding)**.** Consider a finite set $\mathcal{V} = \{1, \ldots, d_{\mathrm{in}}\}$ with a vocabulary size of $d_{\mathrm{in}} \in \mathbb{Z}_{>0}$. For $i \in \mathcal{V}$, the one-hot vector $\mathrm{e}_i \in \{0, 1\}^{d_{\mathrm{in}}}$ is defined as:

$$(\mathrm{e}_i)_j := \begin{cases} 1 & (j = i) \\ 0 & (j \neq i) \end{cases} \quad (j \in [1, d_{\mathrm{in}}]_{\mathbb{Z}}). \tag{5}$$

Remark 3.1. For any $i \neq j$, we have $\|\mathrm{e}_i - \mathrm{e}_j\|_2 = \sqrt{2}$, which expresses neutrality (lack of prior assumptions) by ensuring that the **distance between all pairs of tokens is equal**.

4 Mapping to the Next Layer with a Fully-Connected Layer and the Emergence of Embedding

4.1 Symmetry and Fully-Connected Layer

When the **input is one-hot**, we have no choice but to treat **each input node equally**. On the input side, there are d_{in} scalar output nodes (each outputting the i-th component of e_i), and they are **symmetric**. Under this symmetry, the natural connection to the next set of nodes (with an intermediate representation dimension of d_{emb}) is a **fully-connected** layer. Partially introducing weight sharing risks treating **tokens asymmetrically**, so this is **generally not done** here. (Full sharing across all dimensions would eliminate any difference). Thus, when we assume a one-hot vector input and have a fully-connected layer where the weights are independent for each edge connecting the set of input nodes to another set of nodes, we call this part an **embedding layer**.

4.2 Matrix Representation and Equivalence to Linear Regression

As previously learned, a fully-connected layer is a **linear map** (equivalent to linear regression) and can be written using a matrix $\boldsymbol{W} \in \mathbb{R}^{d_{\mathrm{emb}}, d_{\mathrm{in}}}$ as:

$$\boldsymbol{z} := \boldsymbol{W}\boldsymbol{x} \quad (\boldsymbol{x} \in \mathbb{R}^{d_{\mathrm{in}}}, \ \boldsymbol{z} \in \mathbb{R}^{d_{\mathrm{emb}}}). \tag{6}$$

Here, $\boldsymbol{x}$ is a general input, which includes one-hot vectors.

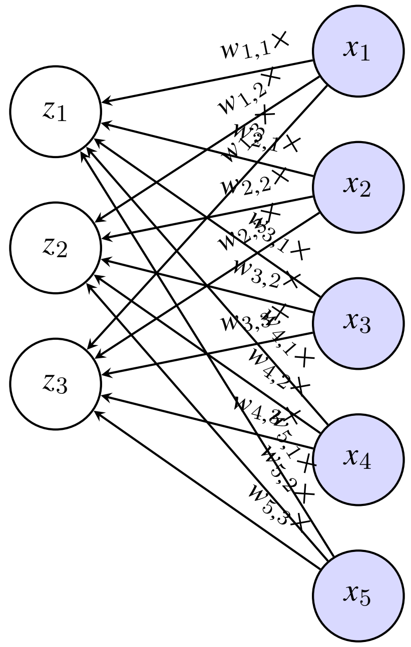

Figure 1: **Fully-connected** mapping from a one-hot input to the next layer (of dimension d_{emb}). The right side shows the input (d_{in} scalar nodes), and the left side shows the next-layer nodes (d_{emb} nodes). Each edge is assigned a weight $w_{i,j}$.

4.3 Active Edges for a One-Hot Input

In particular, when $\boldsymbol{x} = \mathrm{e}_i$, **only the weights in the i-th column are involved**:

$$\boldsymbol{W}\,\mathrm{e}_i = \boldsymbol{w}_i \quad \text{(where } \boldsymbol{w}_i \text{ is the } i\text{-th column of } \boldsymbol{W}\text{).} \tag{7}$$

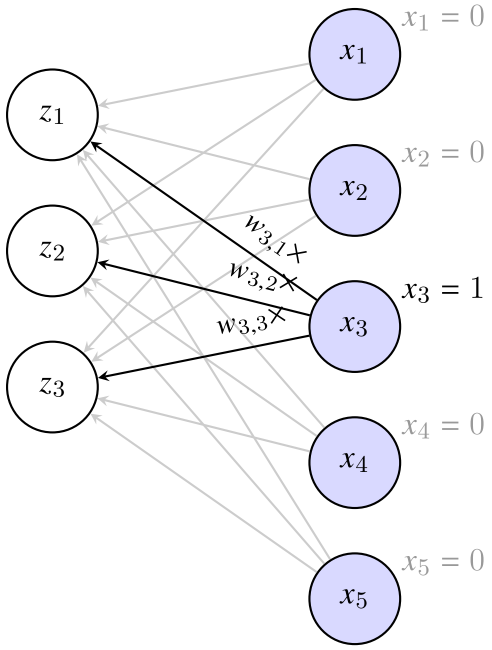

Figure 2: Edges that **activate** (black) when the input is $\boldsymbol{x} = \mathrm{e}_3$, and inactive edges (gray). In this case, $\boldsymbol{W}\mathrm{e}_3 = \boldsymbol{w}_3$ becomes the input to the next layer.

4.4 Equivalent Understanding of Embedding

Equation (7) means that "**feeding a one-hot vector e_i into the first fully-connected layer**" is **equivalent** to "**directly providing the column vector $\boldsymbol{w}_i$ to the group of output-side nodes (head nodes) of that layer**". Note that $\boldsymbol{w}_i$ is a **learnable parameter** that can be updated during training (while the correspondence between the entity and its numerical representation is fixed in the architecture formulation, the specific values of the column vector are determined by learning).

Definition 4.1 (Embedding)**.** The map defined by each column $\boldsymbol{w}_i$ of the matrix $\boldsymbol{W} \in \mathbb{R}^{d_{\mathrm{emb}}, d_{\mathrm{in}}}$,

$$\iota : \ i \in \{1, \ldots, d_{\mathrm{in}}\} \mapsto \boldsymbol{w}_i \in \mathbb{R}^{d_{\mathrm{emb}}} \tag{8}$$

is called an **embedding** or **embedding representation** [1–3].

Remark 4.1. It is customary to call $\boldsymbol{w}_i$ the **representation** of the i-th token or (though the terminology can be slightly confusing) the **embedding** of the i-th token. Mathematically, an "embedding" refers to an **injective map from a space with structure to another space that preserves the original structure in some sense**. In practice, this name is used with the expectation that the relationships between tokens in natural language will be **reflected** in the relationships within the vector space (if the training converges normally, the probability of columns being identical is extremely low, making the map effectively injective).

4.5 Vocabulary Extension (Adding Columns)

To add new token types later, we increase the **dimension of the one-hot vectors**. Graphically, this corresponds to **adding input nodes** and **adding a set of fully-connected edges** from them to the next layer. In matrix representation, we can simply **add new column vectors to $\boldsymbol{W}$** and make the number of columns match the new vocabulary size. Since each column corresponds one-to-one with a token, it is theoretically possible to **train only the new columns** (in practice, it is common to fine-tune the entire matrix simultaneously).

5 Distances and Similarities in the Embedding Space

If the embedding reflects semantic information to some extent, calculating the **relationships between vectors** can have practical applications [1,2]. Here, we will strictly define the most fundamental measures.

Definition 5.1 (Euclidean Distance)**.** For $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^{d_{\mathrm{emb}}}$, the **Euclidean distance** $d_{\mathrm{E}}(\boldsymbol{u}, \boldsymbol{v})$ between $\boldsymbol{u}$ and $\boldsymbol{v}$ is defined as:

$$d_{\mathrm{E}}(\boldsymbol{u}, \boldsymbol{v}) := \|\boldsymbol{u} - \boldsymbol{v}\|_2 = \sqrt{(\boldsymbol{u} - \boldsymbol{v})^\top (\boldsymbol{u} - \boldsymbol{v})}. \tag{9}$$

The **smaller** the Euclidean distance between two vectors, the more similar the concepts they represent can be considered. Although it is the most intuitive measure, it seems to be used less frequently in natural language processing compared to the other two discussed below.

Definition 5.2 (Standard Inner Product)**.** For $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^{d_{\mathrm{emb}}}$, the **Standard Inner Product** $\langle \boldsymbol{u}, \boldsymbol{v} \rangle$ of $\boldsymbol{u}$ and $\boldsymbol{v}$ is defined as:

$$\langle \boldsymbol{u}, \boldsymbol{v} \rangle := \boldsymbol{u}^\top \boldsymbol{v} = \sum_{k=1}^{d_{\mathrm{emb}}} u_k v_k. \tag{10}$$

The **larger** the standard inner product of two vectors, the more similar the concepts they represent can be considered. The most famous architecture that uses the standard inner product between vectors derived from token representation vectors is probably the Transformer [4].

Definition 5.3 (Cosine Similarity)**.** For $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^{d_{\mathrm{emb}}} \setminus \{\boldsymbol{0}\}$, the **cosine similarity** $\cos(\boldsymbol{u}, \boldsymbol{v})$ of $\boldsymbol{u}$ and $\boldsymbol{v}$ is defined as:

$$\cos(\boldsymbol{u}, \boldsymbol{v}) := \frac{\langle \boldsymbol{u}, \boldsymbol{v} \rangle}{\|\boldsymbol{u}\|_2 \|\boldsymbol{v}\|_2}. \tag{11}$$

As its name suggests, cosine similarity is the cosine of the angle between two vectors. The **larger** the cosine similarity of two vectors, the more similar the concepts they represent can be considered. It has many applications in natural language processing (e.g., [5]).

The following identity holds between these measures:

$$\begin{aligned} \|\boldsymbol{u} - \boldsymbol{v}\|_2^2 &= \|\boldsymbol{u}\|_2^2 + \|\boldsymbol{v}\|_2^2 - 2 \langle \boldsymbol{u}, \boldsymbol{v} \rangle \\ &= \|\boldsymbol{u}\|_2^2 + \|\boldsymbol{v}\|_2^2 - 2 \|\boldsymbol{u}\|_2 \|\boldsymbol{v}\|_2 \cos(\boldsymbol{u}, \boldsymbol{v}) \end{aligned} \tag{12}$$

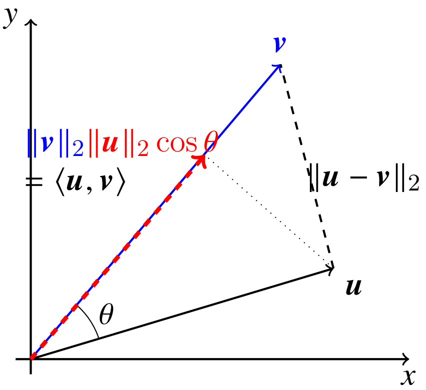
y
$\boldsymbol{v}$
$\|\boldsymbol{v}\|_2\|\boldsymbol{u}\|_2 \cos\theta$
$= \langle \boldsymbol{u}, \boldsymbol{v} \rangle$
$\|\boldsymbol{u} - \boldsymbol{v}\|_2$
$\boldsymbol{u}$
θ
x

Figure 3: Geometry of the **Euclidean distance** (line segment between the vector tips), **inner product** (length of the orthogonal projection), and **cosine similarity** (angle θ) between two vectors.

5.1 Examples and Exercises

Example 5.1 (Euclidean Distance)**.** For $\boldsymbol{u} = \begin{bmatrix} 2 \\ -1 \\ 3 \end{bmatrix}$ and $\boldsymbol{v} = \begin{bmatrix} 1 \\ 2 \\ -1 \end{bmatrix}$:

$$\boldsymbol{u} - \boldsymbol{v} = \begin{bmatrix} 2-1 \\ -1-2 \\ 3-(-1) \end{bmatrix} = \begin{bmatrix} 1 \\ -3 \\ 4 \end{bmatrix}, \tag{13}$$

$$\|\boldsymbol{u} - \boldsymbol{v}\|_2 = \sqrt{1^2 + (-3)^2 + 4^2} = \sqrt{1 + 9 + 16} = \sqrt{26}. \tag{14}$$

Exercise 5.1. Find the Euclidean distance $d_{\mathrm{E}}(\boldsymbol{a}, \boldsymbol{b})$ for $\boldsymbol{a} = \begin{bmatrix} -1 \\ 4 \\ 2 \end{bmatrix}$ and $\boldsymbol{b} = \begin{bmatrix} 3 \\ 0 \\ -2 \end{bmatrix}$.

Answer.

$$\boldsymbol{a} - \boldsymbol{b} = \begin{bmatrix} -1-3 \\ 4-0 \\ 2-(-2) \end{bmatrix} = \begin{bmatrix} -4 \\ 4 \\ 4 \end{bmatrix}, \tag{15}$$

$$\|\boldsymbol{a} - \boldsymbol{b}\|_2 = \sqrt{(-4)^2 + 4^2 + 4^2} = \sqrt{16 + 16 + 16} = \sqrt{48} = 4\sqrt{3}. \tag{16}$$

Example 5.2 (Inner Product)**.** For $\boldsymbol{u} = \begin{bmatrix} 2 \\ -1 \\ 3 \end{bmatrix}$ and $\boldsymbol{v} = \begin{bmatrix} 1 \\ 2 \\ -1 \end{bmatrix}$:

$$\langle \boldsymbol{u}, \boldsymbol{v} \rangle = 2 \cdot 1 + (-1) \cdot 2 + 3 \cdot (-1) = 2 - 2 - 3 = -3. \tag{17}$$

Exercise 5.2. Find the inner product $\langle \boldsymbol{a}, \boldsymbol{b} \rangle$ for $\boldsymbol{a} = \begin{bmatrix} -1 \\ 4 \\ 2 \end{bmatrix}$ and $\boldsymbol{b} = \begin{bmatrix} 3 \\ 0 \\ -2 \end{bmatrix}$.

Answer.

$$\langle \boldsymbol{a}, \boldsymbol{b} \rangle = (-1) \cdot 3 + 4 \cdot 0 + 2 \cdot (-2) = -3 + 0 - 4 = -7. \tag{18}$$

Example 5.3 (Cosine Similarity)**.** For $\boldsymbol{u} = \begin{bmatrix} 2 \\ -1 \\ 3 \end{bmatrix}$ and $\boldsymbol{v} = \begin{bmatrix} 1 \\ 2 \\ -1 \end{bmatrix}$:

$$\begin{aligned}
\langle \boldsymbol{u}, \boldsymbol{v} \rangle &= -3 \quad \text{(from Example 5.2)}, && (19) \\
\|\boldsymbol{u}\|_2 &= \sqrt{2^2 + (-1)^2 + 3^2} = \sqrt{4 + 1 + 9} = \sqrt{14}, && (20) \\
\|\boldsymbol{v}\|_2 &= \sqrt{1^2 + 2^2 + (-1)^2} = \sqrt{1 + 4 + 1} = \sqrt{6}, && (21) \\
\cos(\boldsymbol{u}, \boldsymbol{v}) &= \frac{-3}{\sqrt{14}\sqrt{6}} = \frac{-3}{\sqrt{84}} = \frac{-3}{2\sqrt{21}}. && (22)
\end{aligned}$$

Exercise 5.3. Find the cosine similarity for $\boldsymbol{a} = \begin{bmatrix} -1 \\ 4 \\ 2 \end{bmatrix}$ and $\boldsymbol{b} = \begin{bmatrix} 3 \\ 0 \\ -2 \end{bmatrix}$.

Answer.

$$\begin{aligned}
\langle \boldsymbol{a}, \boldsymbol{b} \rangle &= -7 \quad \text{(from Eq. (18))}, && (23) \\
\|\boldsymbol{a}\|_2 &= \sqrt{(-1)^2 + 4^2 + 2^2} = \sqrt{1 + 16 + 4} = \sqrt{21}, && (24) \\
\|\boldsymbol{b}\|_2 &= \sqrt{3^2 + 0^2 + (-2)^2} = \sqrt{9 + 0 + 4} = \sqrt{13}, && (25) \\
\cos(\boldsymbol{a}, \boldsymbol{b}) &= \frac{-7}{\sqrt{21}\sqrt{13}} = \frac{-7}{\sqrt{273}}. && (26)
\end{aligned}$$

6 Summary

In this lecture, we learned the following:

- Assuming a one-hot vector input, when there is a fully-connected layer where the weights are independent for each edge connecting the set of input nodes to another set of nodes, this part is called an **embedding layer**. An embedding layer is equivalent to a linear map by a matrix of weight parameters. When the input is e_i, the input to the next layer becomes the i-th column vector $\boldsymbol{w}_i$ of the weight matrix (Equation (7)).

- An embedding layer effectively converts each token i into a real-valued vector $\boldsymbol{w}_i$ composed of weight parameters. The target vector $\boldsymbol{w}_i$ is called the **representation (embedding)** of token i.

- To capture the relationships between embeddings, we strictly defined **Euclidean distance, inner product, and cosine similarity** (Equations (9)–(11)) and showed their **relationship** (Equation (12)) and geometry (Figure 3).

References

[1] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. Cambridge, MA: MIT Press, 2016.

[2] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” in Proceedings of Workshop at ICLR 2013, 2013.

[3] J. Pennington, R. Socher, and C. D. Manning, “Glove: Global vectors for word representation,” in Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543, 2014.

[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” in Proceedings of NeurIPS 2017, 2017.

[5] N. Reimers and I. Gurevych, “Sentence-bert: Sentence embeddings using siamese bert-networks,” arXiv preprint arXiv:1908.10084, 2019.