

Schritt 2: Embedding — Zahlen bekommen Bedeutung



Was bisher geschah: Der Tokenizer hat unseren Satz in Token-IDs verwandelt:

„Die Katze sitzt auf der Matte“ → [0, 1, 2, 3, 4, 5]

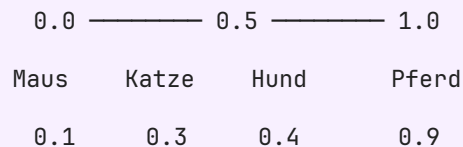
Das Problem: Die Zahl 0 sagt dem Transformer nichts über die Bedeutung von „Die“. Ist „Katze“ (1) näher an „Matte“ (5) als an „sitzt“ (2)? Die IDs sind willkürlich!

Was ist ein Embedding?

Analogie: Koordinaten auf einer Landkarte

Stell dir vor, jedes Wort hat einen **Ort** in einem mehrdimensionalen Raum. Wörter mit ähnlicher Bedeutung liegen nahe beieinander.

In 1D (einer Zahl) könntest du Tiere auf einer Linie anordnen:



Aber mit nur 1 Dimension kann man nicht ausdrücken, dass „Katze“ und „Hund“ beides Haustiere sind, während „Katze“ und „Tiger“ beides Katzen sind. **Dafür braucht man mehr Dimensionen!**

GPT-2 nutzt **768 Dimensionen**. GPT-4 vermutlich ~12.000.

Wir nutzen **4 Dimensionen** — genug zum Verstehen, wenig genug zum Rechnen.

So funktioniert es

Ein Embedding ist einfach eine **Nachschlagetabelle**:

Token-ID → Tabelle nachschlagen → Vektor (Liste von Zahlen)

Woher kommen die Zahlen?

Am Anfang sind sie **zufällig**! Während des Trainings lernt der Transformer, welche Zahlen gut sind. Ähnliche Wörter bekommen dabei automatisch ähnliche Vektoren.

Für unser Papier-Modell nehmen wir feste Beispiel-Werte.

Die Embedding-Tabelle

Das ist das Herzstück dieses Schritts. **Drucke diese Seite aus und lege sie neben deine Token-Kärtchen.**

Für jede Token-ID schlägst du hier die 4 Zahlen nach.

Unsere 4 Dimensionen stellen (vereinfacht) dar:

Dimension	d_1	d_2	d_3	d_4
Bedeutung (vereinfacht)	Artikel?	Lebewesen?	Aktion?	Objekt?

Token	ID	d_1 Artikel?	d_2 Lebewesen?	d_3 Aktion?	d_4 Objekt?
Die	0	0.9	0.1	0.0	0.1
Katze	1	0.0	0.9	0.1	0.2
sitzt	2	0.0	0.1	0.9	0.0
auf	3	0.5	0.0	0.3	0.4
der	4	0.9	0.1	0.0	0.1
Matte	5	0.0	0.0	0.0	0.9

Schau genau hin:

- „Die“ (ID:0) und „der“ (ID:4) haben fast den gleichen Vektor → sie bedeuten Ähnliches!
- „Katze“ (ID:1) hat eine hohe Zahl bei „Lebewesen?“ → das passt!
- „Matte“ (ID:5) hat eine hohe Zahl bei „Objekt?“ → auch das passt!
- „sitzt“ (ID:2) hat eine hohe Zahl bei „Aktion?“ → Verb erkannt!

In Wirklichkeit haben die Dimensionen keine klaren Bedeutungen wie „Lebewesen?“ — der Transformer lernt selbst, was jede Dimension darstellt. Wir benutzen diese Labels nur, damit du sehen kannst *warum* ähnliche Wörter ähnliche Zahlen haben.

Übung: Nachschlagen — Von Token-IDs zu Vektoren

Mach mit!

1. Nimm deine Token-Kärtchen vom Tokenizer (Blatt 2 — Wörter)
2. Dreh sie auf die Rückseite (ID-Seite)
3. Schlage jede ID in der Tabelle nach
4. Schreibe den Vektor auf ein neues Kärtchen

Schritt für Schritt:

Die ID: 0	→ Zeile 0 nachschlagen →	[0.9, 0.1, 0.0, 0.1]
Katze ID: 1	→ Zeile 1 nachschlagen →	[0.0, 0.9, 0.1, 0.2]
sitzt ID: 2	→ Zeile 2 nachschlagen →	[0.0, 0.1, 0.9, 0.0]
auf ID: 3	→ Zeile 3 nachschlagen →	[____, ____, ____, __]
der ID: 4	→ Zeile 4 nachschlagen →	[____, ____, ____, __]
Matte ID: 5	→ Zeile 5 nachschlagen →	[____, ____, ____, __]

Tipp: Die letzten 3 darfst du selbst ausfüllen! Schau in der Embedding-Tabelle auf der vorherigen Seite nach.

Das Ergebnis: Die Embedding-Matrix

Wenn du alle Vektoren untereinander schreibst, bekommst du eine **Matrix** — eine Tabelle aus Zahlen. Das ist der eigentliche Input für den Transformer!

Lege deine Vektor-Kärtchen untereinander, in der Reihenfolge des Satzes:

	ID	d_1	d_2	d_3	d_4
Die	0	0.9	0.1	0.0	0.1
Katze	1	0.0	0.9	0.1	0.2
sitzt	2	0.0	0.1	0.9	0.0
auf	3	0.5	0.0	0.3	0.4
der	4	0.9	0.1	0.0	0.1
Matte	5	0.0	0.0	0.0	0.9

↑ Das ist eine **6 × 4 Matrix** (6 Tokens × 4 Dimensionen)

Was du jetzt hast:

Aus dem Satz „Die Katze sitzt auf der Matte“ ist eine Tabelle aus Zahlen geworden. **Ab jetzt arbeitet der Transformer nur noch mit diesen Zahlen!** Der Text ist vergessen — alles ist Mathematik.

Übung: Ähnlichkeit berechnen

Wie misst man, ob zwei Wörter ähnlich sind? Man berechnet das **Skalarprodukt** (Dot Product) ihrer Vektoren:


Multipliziere die Zahlen an gleicher Position und addiere alles.

Beispiel: Wie ähnlich sind „Die“ und „der“?

	d₁	d₂	d₃	d₄	
Die	0.9	0.1	0.0	0.1	
der	0.9	0.1	0.0	0.1	
Multiplizieren	0.9×0.9	0.1×0.1	0.0×0.0	0.1×0.1	
Ergebnis	0.81	0.01	0.00	0.01	Summe = 0.83

0.83 — sehr hoch! → „Die“ und „der“ sind ähnlich. ✓

Jetzt du: Wie ähnlich sind „Katze“ und „Matte“?

 Hier selbst ausfüllen

	d₁	d₂	d₃	d₄	
Katze	0.0	0.9	0.1	0.2	
Matte	0.0	0.0	0.0	0.9	
Multiplizieren	0.0×0.0	0.9×0.0	0.1×0.0	0.2×0.9	
Ergebnis					Summe = ____

Ist das Ergebnis hoch oder niedrig? Was sagt das über die Ähnlichkeit?

Bonus-Übung: Alle Ähnlichkeiten auf einen Blick

Berechne das Skalarprodukt für jedes Wort-Paar und trag es in die Tabelle ein. Diese Tabelle ist wichtig — sie kommt im nächsten Schritt (Self-Attention) wieder!

 Hier selbst ausfüllen

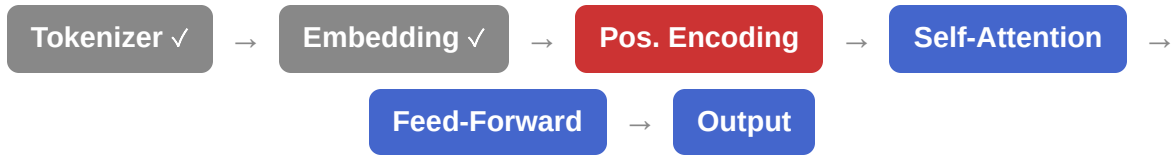
	Die	Katze	sitzt	auf	der	Matte
Die	0.83				0.83	
Katze		0.86				
sitzt			0.82			
auf				0.50		
der	0.83				0.83	
Matte						0.81

Hinweis: Die Diagonale (Wort mit sich selbst) ist immer am höchsten. Ein paar Werte sind schon eingetragen — füll den Rest aus!

Tipp: Die Tabelle ist symmetrisch! „Die·Katze“ = „Katze·Die“

Zusammenfassung: Was hast du gebaut?

Schicht	Input	Output	Papier-Modell
Tokenizer	Text	Token-IDs [0,1,2,3,4,5]	Kärtchen mit Wörtern/IDs
Embedding	Token-IDs	6×4 Matrix aus Zahlen	Nachschlagetabelle + Vektor-Kärtchen



Nächster Schritt: Positional Encoding

Die Embedding-Matrix hat ein Problem: Sie weiß nicht, in welcher **Reihenfolge** die Wörter stehen! „Die Katze sitzt auf der Matte“ und „Matte der auf sitzt Katze Die“ haben die gleiche Matrix — nur die Zeilen sind vertauscht.

Das Positional Encoding fügt **Positions-Informationen** hinzu: Jeder Vektor bekommt ein Signal, das sagt „Ich bin das 1. Wort“, „Ich bin das 2. Wort“ usw.