M05d - Multi-Head-Attention

Stand: 03.2025

1 Fünf mögliche Köpfe und ihre Spezialisierungen:

1.1 Kopf 1 – Grammatikalische Struktur

- Erkennt die Beziehungen zwischen Subjekt, Prädikat und Objekt.
- Z. B. "Katze" → "jagt" (Subjekt-Prädikat), "jagt" → "Maus" (Prädikat-Objekt).

Beispiel Attention-Matrix:

```
Die Katze jagt eine kleine Maus

Die 

Katze 

jagt 

eine 

kleine 

Maus 

Maus
```

👉 Fokus auf Verb-Beziehungen ("Katze jagt", "jagt Maus").

1.2 Kopf 2 – Semantische Beziehungen (Synonyme, Kategorien)

- Erkennt, dass "Katze" und "Maus" beides Tiere sind.
- Verstärkte Verbindungen zwischen ähnlichen Wörtern.

Beispiel Attention-Matrix:

```
Die Katze jagt eine kleine Maus

Die 

Katze 

jagt 

eine 

Herrich Maus
```

👉 "Katze" und "Maus" haben eine semantische Verbindung.

1.3 Kopf 3 – Bestimmung von Modifikatoren (Adjektive & Artikel)

- Achtet auf die Beziehungen zwischen Hauptwort und seinen Modifikatoren.
- Z. B. "eine" und "kleine" beziehen sich auf "Maus".

Beispiel Attention-Matrix:

```
Die Katze jagt eine kleine Maus

Die 

Katze 

jagt 

eine 

kleine 

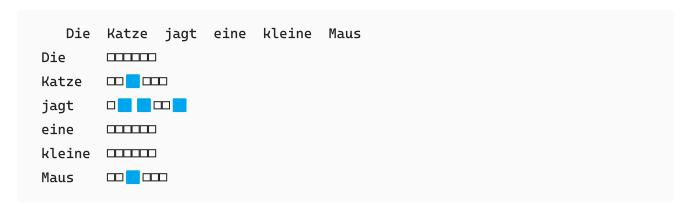
Maus
```

👉 "eine" und "kleine" sind wichtig für "Maus".

1.4 Kopf 4 – Fokus auf das Verb (wichtige Handlung)

- Achtet auf das Verb und dessen direkte Verknüpfung mit dem Subjekt und Objekt.
- "Katze jagt Maus" bekommt hohe Gewichtung.

Beispiel Attention-Matrix:

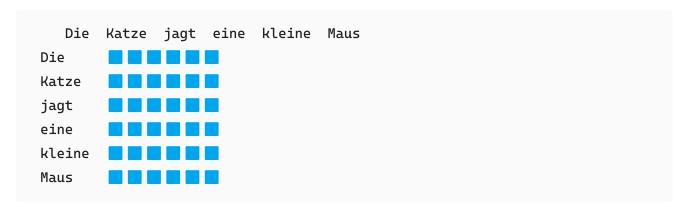


👉 Das Verb "jagt" verbindet "Katze" mit "Maus".

1.5 Kopf 5 – Langfristiger Kontext (Zusammenhang im ganzen Satz)

- Verteilt die Aufmerksamkeit gleichmäßig, um den gesamten Satzkontext zu behalten.
- Hilft z. B. dabei, das Hauptthema zu erkennen (eine Jagdszene).

Beispiel Attention-Matrix:



👉 Jedes Wort erhält Kontext zu allen anderen Wörtern.

1.6 Fazit

Jeder Attention-Kopf lernt eine andere Beziehung zwischen Wörtern:

- 1. **Grammatik** (Subjekt-Prädikat-Objekt)
- 2. Semantische Ähnlichkeit (Bedeutung von Wörtern)
- 3. Adjektiv- und Artikel-Beziehungen (Bestimmungswörter)
- 4. Handlungsfokus (Verb & beteiligte Akteure)
- 5. **Globaler Kontext** (Verbindung zwischen allen Wörtern)

Durch diese verschiedenen Blickwinkel wird das Modell viel leistungsfähiger, als wenn es nur eine einzige Gewichtungsmatrix hätte.

2 Code-Beispiel für eine echte Visualisierung

```
import torch
from transformers import BertTokenizer, BertModel
import matplotlib.pyplot as plt
import numpy as np

# Vortrainiertes BERT-Modell und Tokenizer laden
model_name = "bert-base-german-cased"
tokenizer = BertTokenizer.from_pretrained(model_name)
model = BertModel.from_pretrained(model_name, output_attentions=True)

# Beispiel-Satz
sentence = "Die Katze jagt eine kleine Maus."

# Tokenisierung und Modell-Eingabe vorbereiten
inputs = tokenizer(sentence, return_tensors="pt")
```

```
with torch.no_grad():
    outputs = model(**inputs)
# Attention-Matrizen extrahieren (Layer 0, weil es Multi-Head Attention auf
Token-Ebene ist)
attentions = outputs.attentions[0] # Erste Schicht
attentions = attentions.squeeze().numpy() # Entferne Batch-Dimension
# Tokens für Achsenlabels
tokens = tokenizer.convert_ids_to_tokens(inputs["input_ids"].squeeze())
# Visualisierung der Attention-Gewichte für 5 Köpfe
fig, axes = plt.subplots(1, 5, figsize=(20, 5))
for i in range(5): # Erste 5 Köpfe visualisieren
   ax = axes[i]
   ax.imshow(attentions[i], cmap="Blues")
   ax.set_xticks(range(len(tokens)))
   ax.set_yticks(range(len(tokens)))
   ax.set_xticklabels(tokens, rotation=45)
   ax.set_yticklabels(tokens)
   ax.set_title(f"Kopf {i+1}")
plt.tight_layout()
plt.show()
```

Visual

