Klassifikation mit Transformers

Seminar Klassifikation & Clustering

Ioannis Partalas
Jakob Murauer
Monica Riedler

16.01.2023

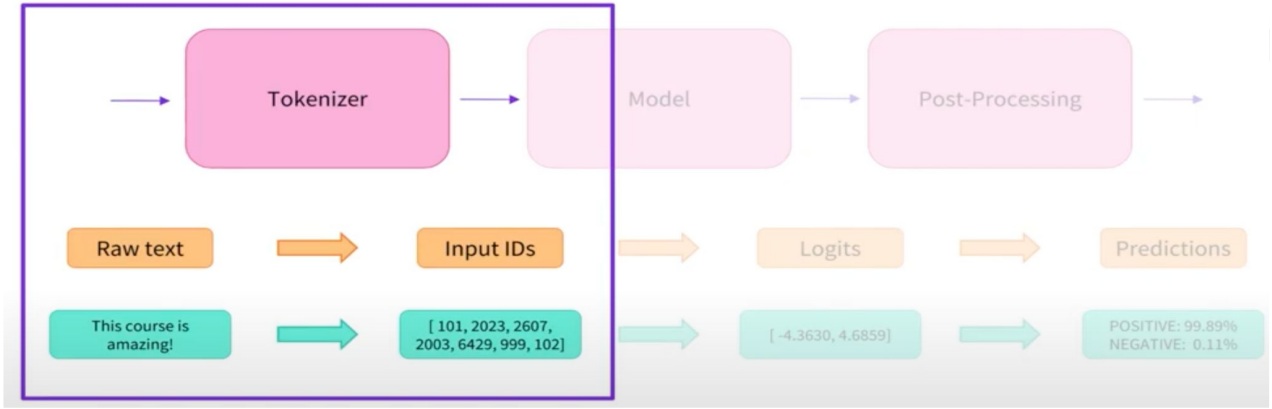
Gliederung

- Transformers Architektur
  - Tokenization
  - Embeddings
  - Positional Encoding
  - Encoder-Decoder
  - Self-Attention & Multi-Head Attention
  - Residuals and Layer Normalization
  - Decoder

- BERT
  - MLM
  - NSP
- RoBERTa
  - Verbesserungen zu BERT
- DistilBERT
  - Performanz, Größe

- Implementierung und Ergebnisse
- Fazit

Tokenization



Zerlegt eine Textkette in eine komprimierte Folge von Symbolen.

Stellt den Text numerisch dar, um Berechnungen damit durchzuführen.

Ergebnis → Vektor von Ganzzahlen, wobei jede Ganzzahl einen Teil des Textes darstellt.

Tokenization

- *Wortbasierte* Tokenizer
  - sehr umfangreiche Vokabulare
  - große Menge von Token außerhalb des Vokabulars
  - Bedeutungsverluste bei sehr ähnlichen Wörtern (z. B. dog & dogs)

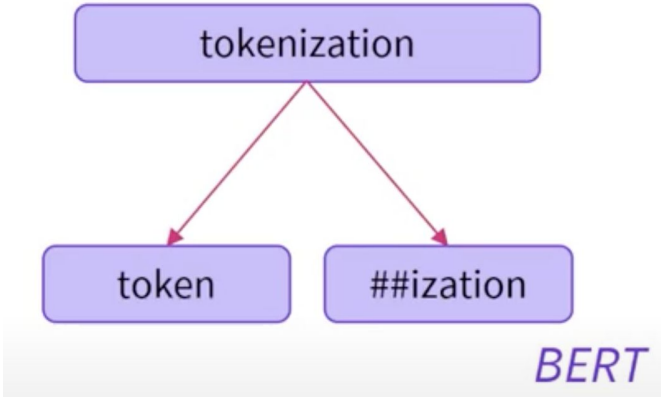
- *Zeichenbasierte* Tokenizer
  - sehr lange Sequenzen
  - weniger bedeutungsvolle einzelne Token

- *Teilwortbasierte (subword)* Tokenizer ermöglicht es dem Modell:
  - ein angemessenes Vokabular zu haben
  - gleichzeitig sinnvolle kontextunabhängige Darstellungen zu lernen
  - Wörter zu verarbeiten, die das Modell noch nie gesehen hat, indem es sie in bekannte Teilwörter zerlegt

Tokenization

Verschiedene Algorithmen zur Tokenisierung von Teilwörtern. Drei Haupttypen von Tokenizers, die in Transformers verwendet werden:

- Byte-Pair Encoding (e.g. GPT-2, RoBERTa)
- WordPiece (e.g. BERT, DistilBERT)
- SentencePiece + Unigram (e.g. XLM-R, T5)

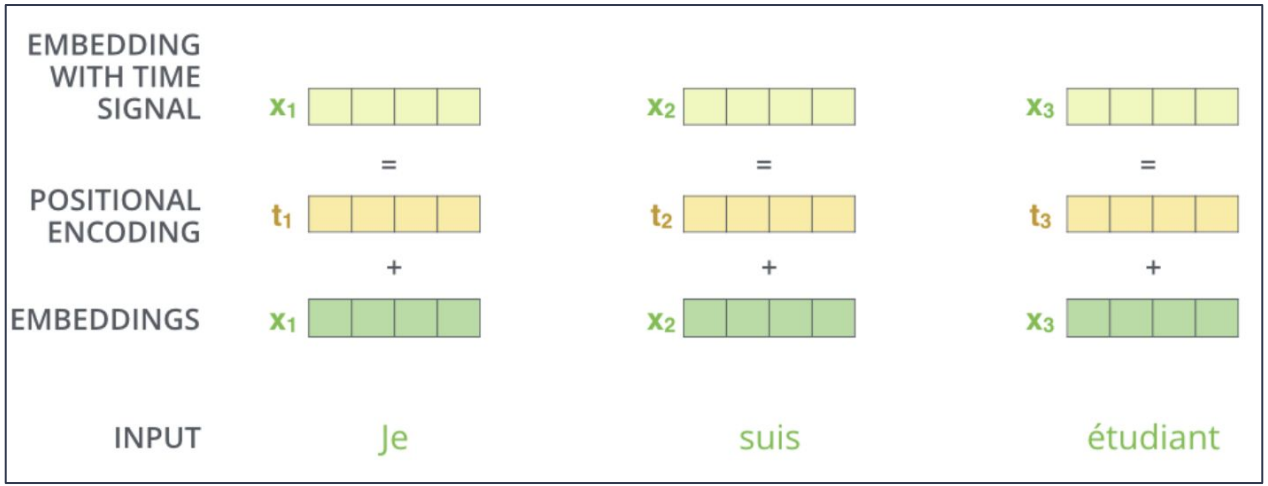


Embeddings

- Um die richtige Darstellung von Text zu lernen, wird jedes einzelne Token in der Sequenz durch eine Einbettung (embedding) in einen Vektor umgewandelt.
- Dies kann als eine Art “Layer” eines neuronalen Netzes betrachtet werden, da die Gewichte für die Einbettungen zusammen mit dem Rest des Transformer-Modells gelernt werden.
- Es enthält einen Vektor für jedes Wort im Vokabular, und diese Gewichte werden aus einer Normalverteilung N(0,1) initialisiert.

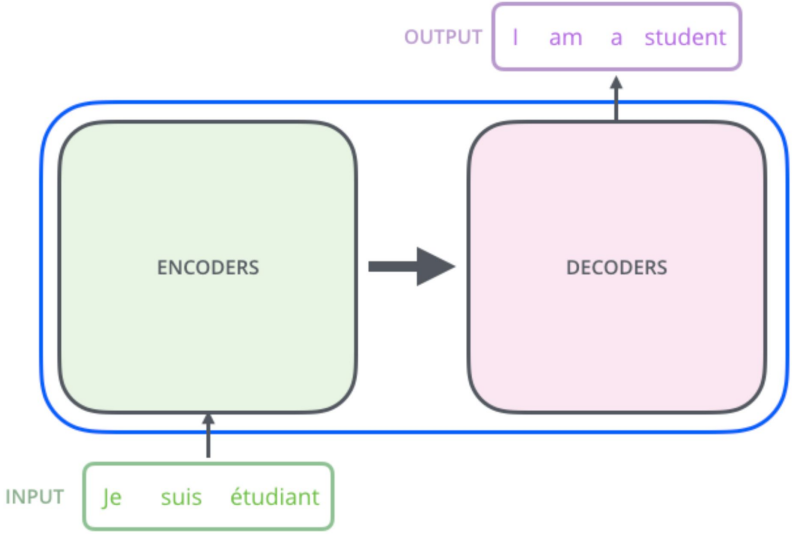
Positional Encoding

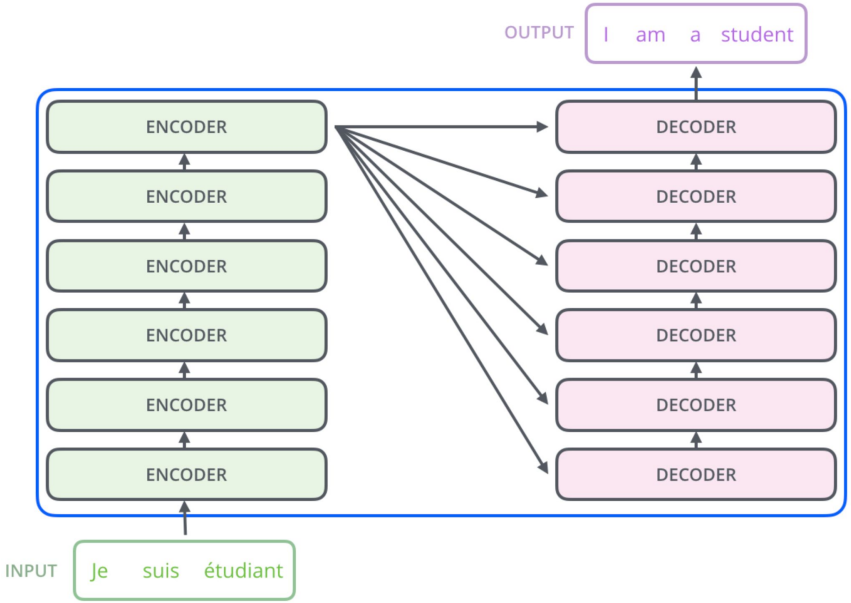
- Das Modell an sich hat keine Informationen über die relative Position der eingebetteten Token in einer Sequenz.
- Der Transformer fügt zu jeder Einbettung einen Vektor hinzu. Diese Vektoren folgen einem bestimmten Muster, das das Modell erlernt und das ihm hilft, die Position jedes Worts oder den Abstand zwischen verschiedenen Wörtern in der Sequenz zu bestimmen.



$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{model}})$$
$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{model}})$$

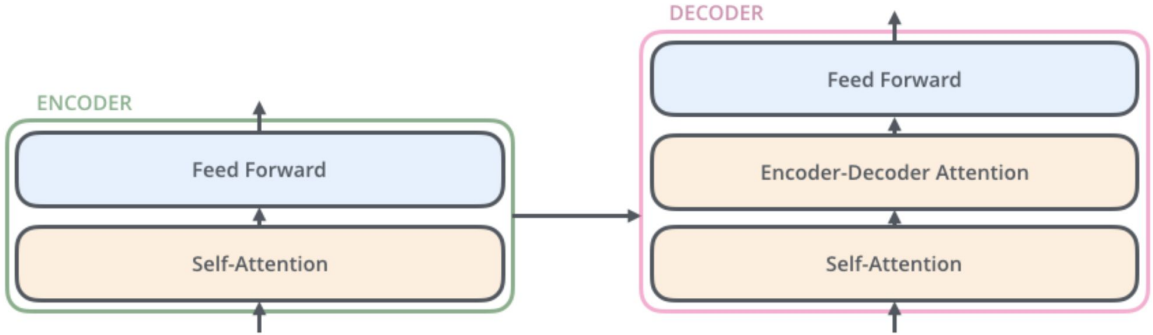
Encoder-Decoder





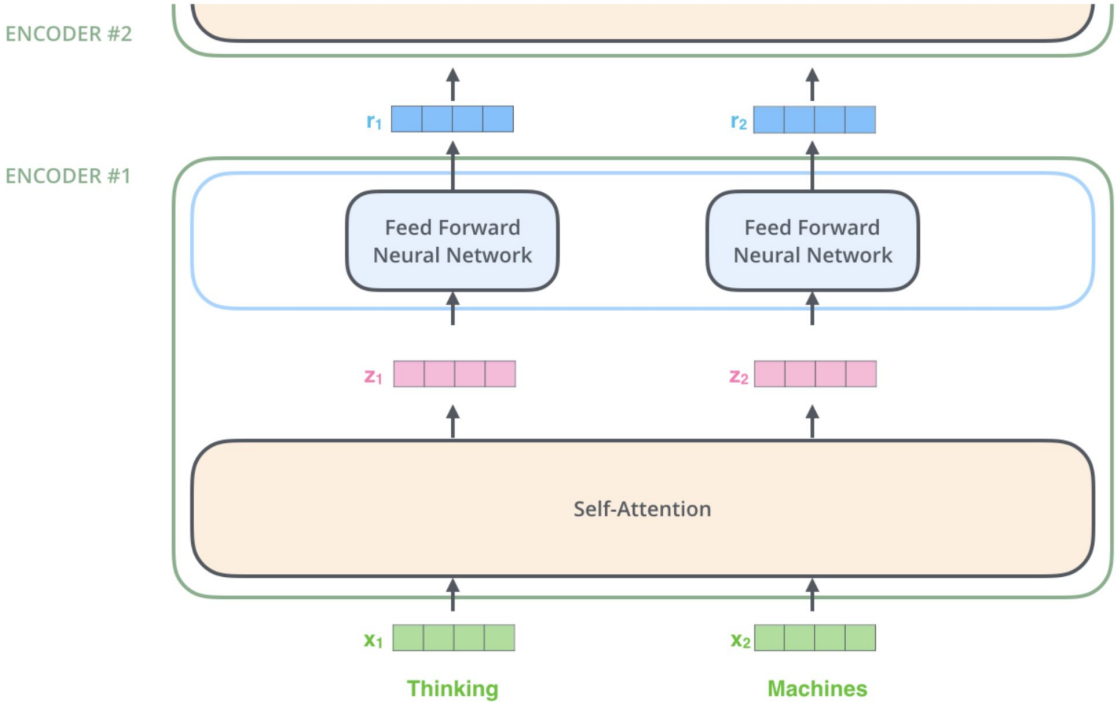
- Die Encoder-Komponente ist ein Stapel von Encoders. Die Decoder-Komponente ist ein Stapel von Decoders der gleichen Zahl.

Encoder-Decoder



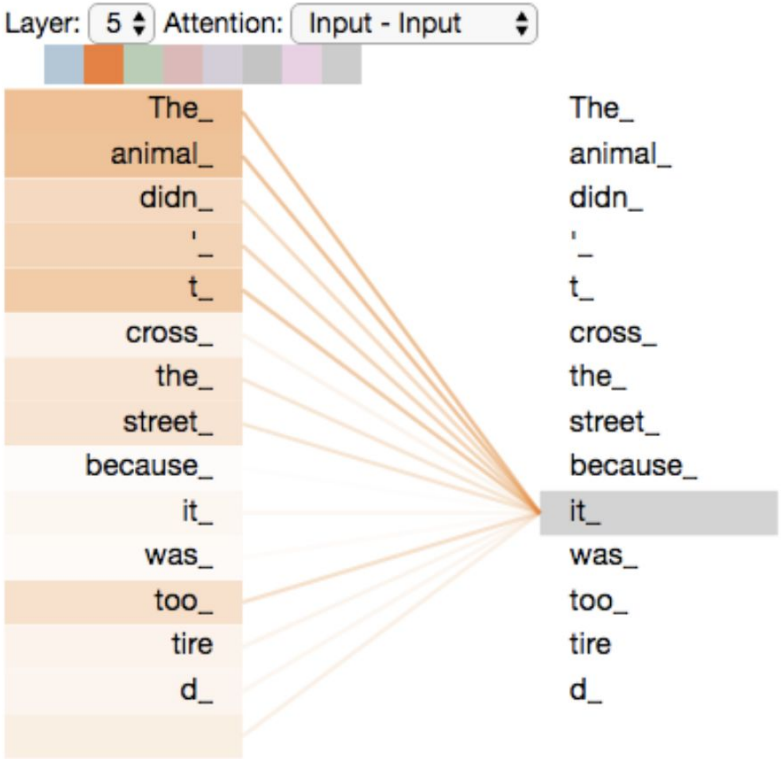
- Die Eingaben des Encoders fließen zunächst durch eine Self-Attention-Schicht, die dem Encoder hilft, andere Wörter im Eingabesatz zu betrachten, während er ein bestimmtes Wort kodiert.
- Die Ausgaben der Self-Attention-Schicht werden in ein neuronales Feed-Forward-Netzwerk eingespeist. Genau dasselbe Feed-Forward-Netzwerk wird unabhängig voneinander auf jeder Position angewendet.
- Der Decoder verfügt über diese beiden Schichten, aber dazwischen befindet sich eine Attention-Schicht, die dem Decoder hilft, sich auf relevante Teile des Eingabesatzes zu konzentrieren.

Encoder



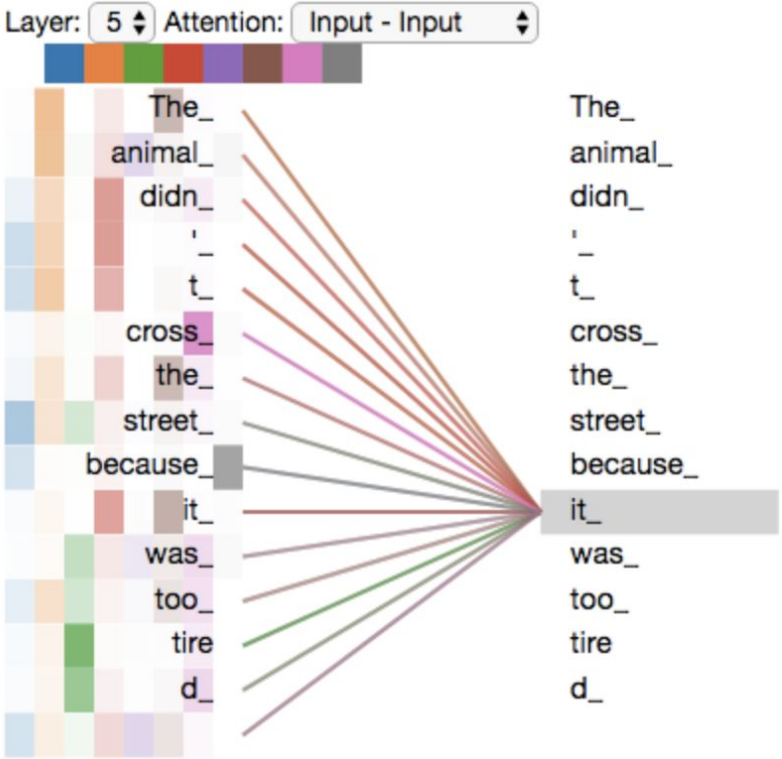
- Das Wort an jeder Position läuft seinen eigenen Pfad im Encoder durch.
- Zwischen diesen Pfaden in der Self-Attention-Schicht gibt es Abhängigkeiten.
- In der Feedforward-Schicht gibt es diese Abhängigkeiten jedoch nicht, so dass die verschiedenen Pfade parallel ausgeführt werden können, während sie durch die Feedforward-Schicht fließen.

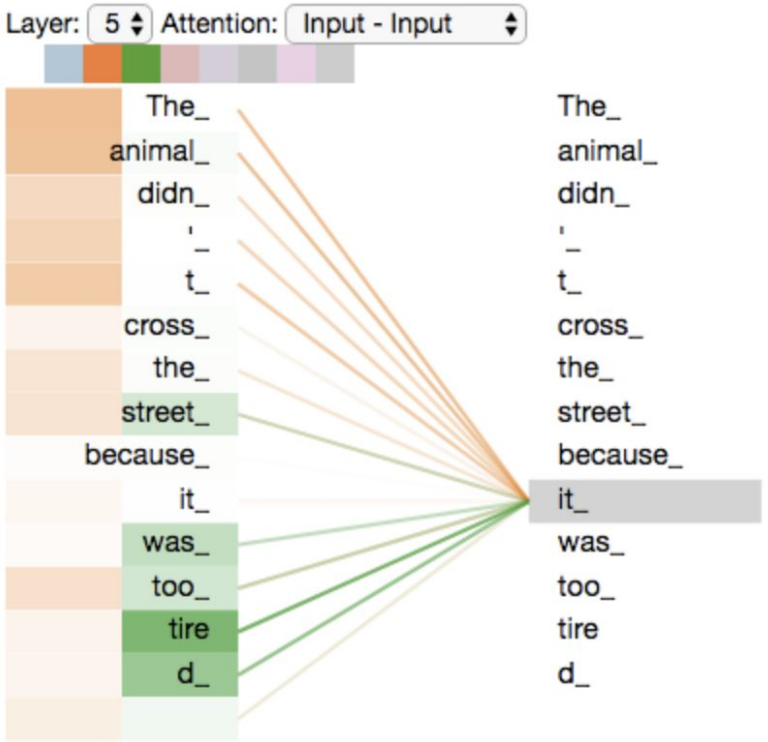
Self-Attention at a High Level



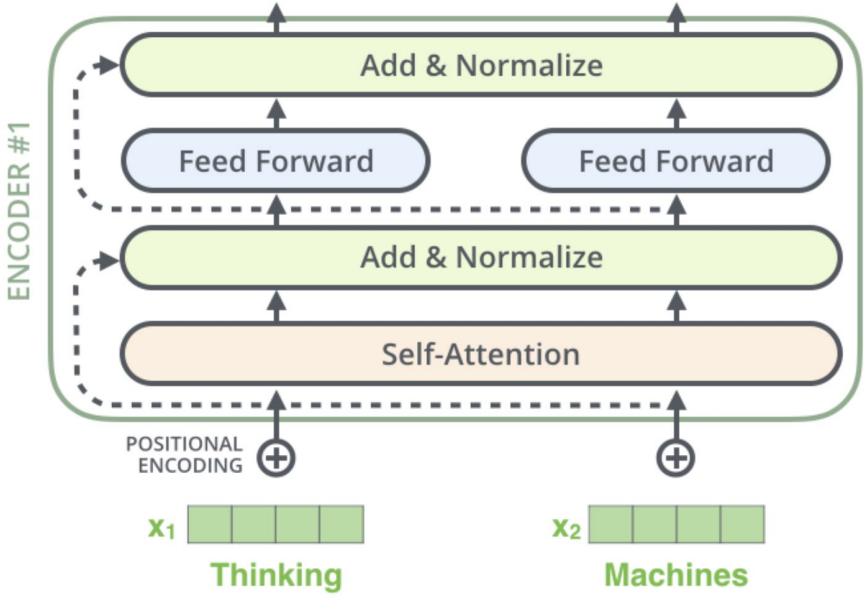
Self-Attention ist die Methode, die der Transformer verwendet, um das "Verständnis" anderer relevanter Wörter in das Wort, das wir gerade verarbeiten, zu integrieren.

Multihead-Attention



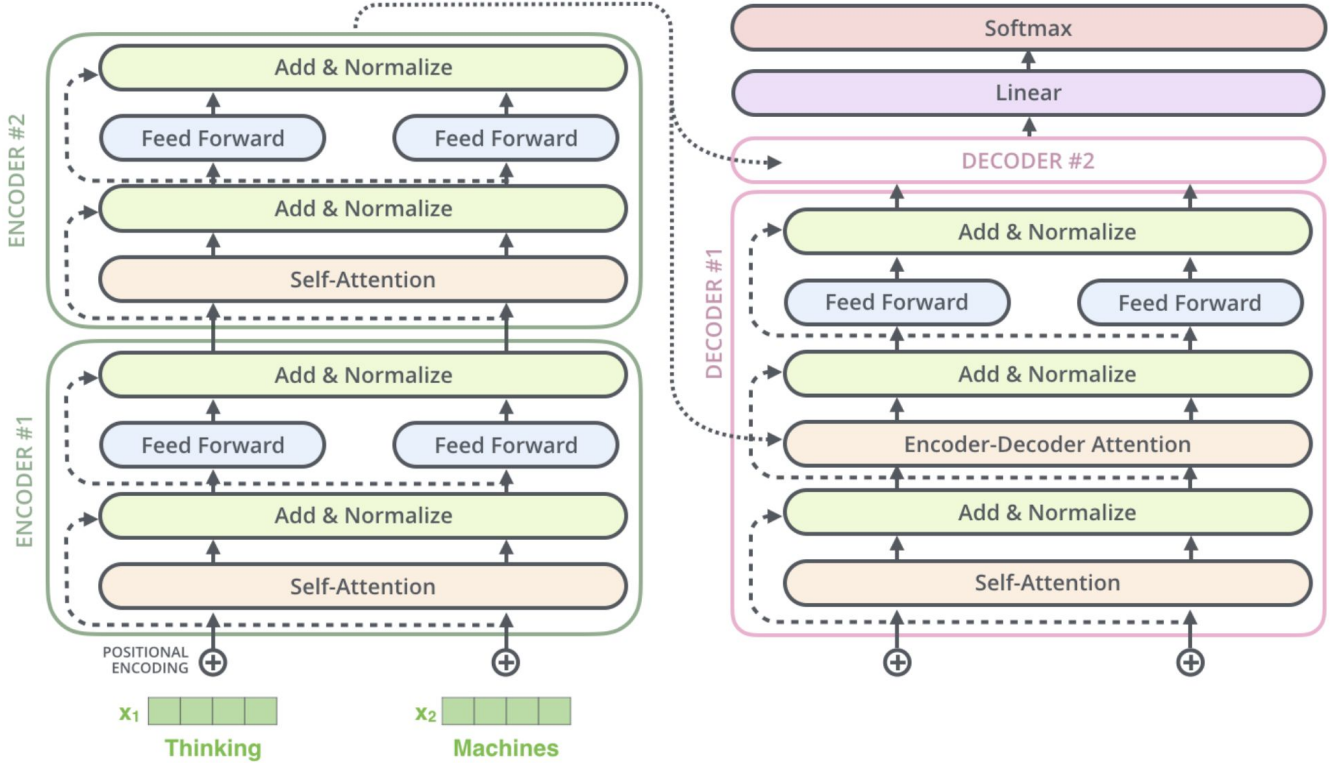


Residuals and Layer Normalization

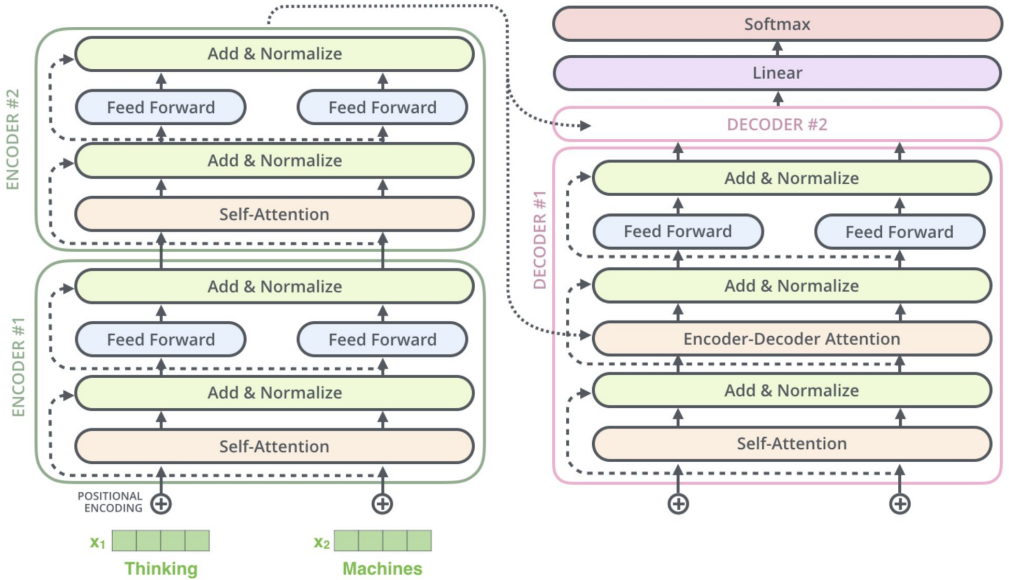


- Tragen dazu bei, das Problem des verschwindenden Gradienten (vanishing gradient) zu entschärfen.
- Der Self-Attention-Mechanismus erlaubt einen beliebigen Informationsfluss im Netzwerk und damit eine beliebige Permutation der Input-Token. Die Restverbindungen (residuals) "erinnern" die Repräsentation jedoch immer daran, wie der ursprüngliche Zustand war.
- In gewisser Weise bieten die Restverbindungen eine Garantie dafür, dass die kontextuellen Repräsentationen der eingegebenen Token auch wirklich die Token repräsentieren.

Residuals and Layer Normalization



Decoder



- Die Ausgabe des obersten Encoders wird von jedem Decoder in seiner **"Encoder-Decoder-Attention"-Schicht** verwendet, die dem Decoder hilft, sich auf geeignete Stellen in der Eingabesequenz zu konzentrieren.

- Im Decoder darf die **Self-Attention-Schicht** nur frühere Positionen in der Ausgabesequenz beachten. Dies geschieht durch Maskierung zukünftiger Positionen.

Decoder

Faustregel:

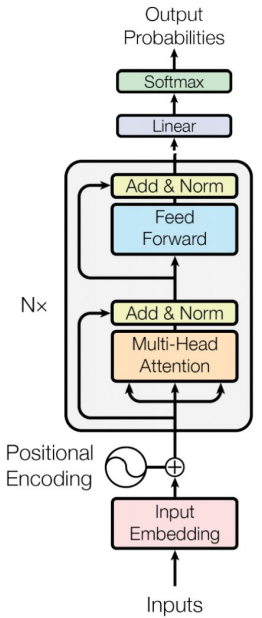
- Bidirektionaler Encoder: Satzklassifikation etc.
- Unidirektionaler Decoder: open-ended text generation etc.
- Encoder-Decoder: maschinelle Übersetzung etc.

BERT

Unsere Klassifikation wurde mit zwei auf BERT basierenden, vortrainierten Modellen (RoBERTa und DistilBERT) gemacht

BERT

- Bidirectional Encoder Representation from Transformer
- 2018 von Google veröffentlicht
- BERT Base:
  - 12 Encoder Blocks, Embedding Size: 768, Hidden Layers: 768, Attention Heads: 12
  - 110 M Parameter
- Training wurde mit dem BooksCorpus (800 Mio Wörter) und English Wikipedia (2,5 Mrd Wörter) → 13Gb
- 40 Epochen



Source: Vaswani et al. (2017)

BERT-Pretraining

Beim Pretraining von BERT gab es zwei Trainingsziele

- Masked Language Modelling (MLM)
- Next Sentence Prediction (NSP)

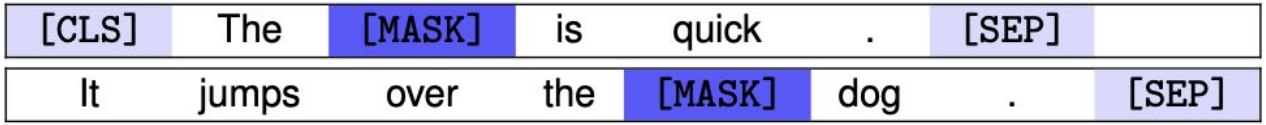
Masked Language Modeling (MLM)

- 15% der Tokens werden verwendet um ein Trainingsziel zu erreichen
- Von diesen 15%
  - werden 80% ersetzt durch einen [MASK] Token und müssen vorausgesagt werden
  - 10% werden ersetzt durch einen zufällig Token
  - die letzten 10% werden unverändert gelassen
- MLM kommt nur während des Pretraining vor, nicht beim Fine-Tuning

BERT-Pretraining

Next Sentence (Sequence) Prediction

- Beim Trainingsprozess werden dem Modell zwei Sätze (s1, s2) vorgelegt. Es soll bestimmen, ob s2 s1 folgt.
- 50% der Fälle ist es der echte nächste Satz, in den restlichen Fällen ist es ein zufällig Satz aus den Trainingsdaten



- [CLS] ist ein Token, der einen Klassifikationstask initialisieren soll
- [SEP] ist ein Token, der zwei Sequenzen separiert
- Output sollte dann circa so aussehen: (“fox”, “lazy”, True)

RoBERTa

- RoBERTa unterscheidet sich von BERT im Großteil in der Masking-Strategie und im Trainingsvolumen
- Es verwendet Dynamic Masking
    - Der Trainingskorpus wird 10-mal dupliziert und für jedes Duplikat werden unterschiedliche Tokens der Masking-Strategie unterworfen
    - RoBERTA sieht den Trainingskorpus in 10-verschiedenen Varianten 4-mal, da wieder mit 40 Epochen trainiert wird
- Kein Next Sequence Prediction
    - NSP schadet laut Autoren von RoBERTa der Performance des Modells
- Laut den Autoren von RoBERTa ist BERT stark “untertrainiert”
    - Es werden 160 Gb an Daten fürs Pretraining verwendet anstatt 13 Gb

Destillation von Modellen

Da viele Modelle groß und einen hohen Rechenaufwand haben, gibt es Motivation existierende Modelle zu Verkleinern, aber dabei so viel Leistung wie möglich zu behalten

Vorgehensweise:

- Wissenstransfer von “Soft-Targets” des “Lehrermodells”
  - Soft_Target = [0.01, 0.05, 0.03, 0.87, 0.04]; Hard_Target = [0, 0, 0, 1, 0]
  - Training mit Soft- und Hard-Target
- Einführen eines neuen Hyperparameter “T”, der das Ausmaß des Wissenstransfer der Soft-Target
  - $$q_i = \frac{\exp(z_i/T)}{\sum_j \exp(z_j/T)}$$
- Anpassen der Loss-Funktion
  - $$L = a \cdot L^{hard} + (1-a)L^{soft}$$

DistilBERT

- Halbe Anzahl von Layern im Vergleich zu BERT
- Circa halbe Größe von BERT
- Trotzdem 95% der Leistung beibehalten
- Gleiche Pretraining Daten
- Verbesserung von RoBERTa übernommen (bis auf mehr Trainingsdaten)
  - Kein NSP
  - Dynamisches Masked Language Modelling

Ergebnisse:

Model	**Score**	CoLA	MNLI	MRPC	QNLI	QQP
ELMo	68.7	44.1	68.6	76.6	71.1	86.2
BERT-base	79.5	56.3	86.7	88.6	91.8	89.6
DistilBERT	77.0	51.3	82.2	87.5	89.2	88.5

Source: Sanh et al. (2019)

Größe:

Model	# param. (Millions)	Inf. time (seconds)
ELMo	180	895
BERT-base	110	668
DistilBERT	66	410

Source: Sanh et al. (2019)

Implementierung

Modelle:

- **Briefsammlung:**
  - XLM-RoBERTa Base
- **Sentiment-Daten:**
  - RoBERTa Base
  - DistilBERT Base Uncased
- **News-Daten:**
  - RoBERTa Base
  - DistilBERT Base Uncased

Hyperparameter:

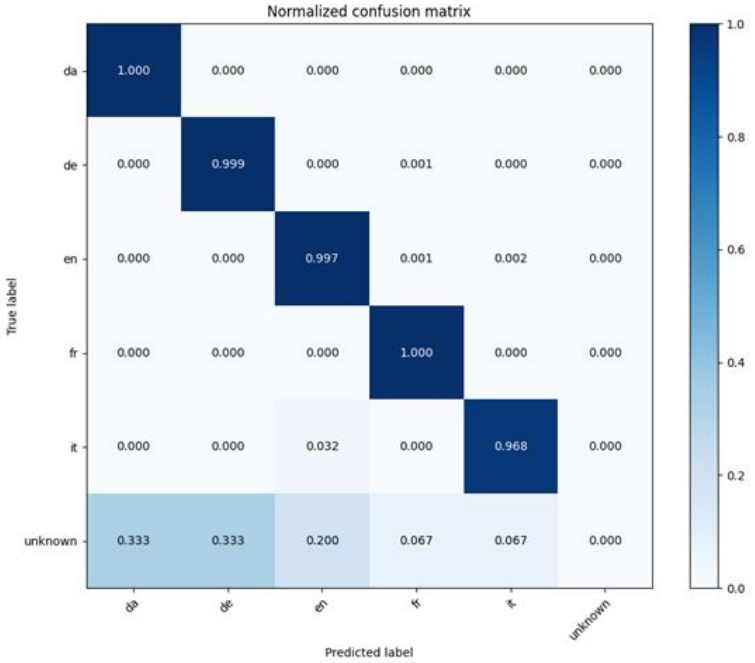
- **Briefsammlung:**
  - Epochs: 5
  - Batch Size: 8
  - Learning Rate: 5e-5, 5e-4
- **Sentiment-Daten:**
  - Epochs: 2, 4
  - Batch Size: 16, 32
  - Learning Rate: 3e-5, 5e-5
- **News-Daten:**
  - Epochs: 5
  - Batch Size: 32
  - Learning Rate: 5e-5

Ergebnisse: Sprachidentifizierung

Results:
- F-score (micro) 0.9951
- F-score (macro) 0.8062
- Accuracy 0.9951

By class:

	precision	recall	f1-score	support
en	0.9984	0.9972	0.9978	2517
de	0.9965	0.9993	0.9979	1436
da	0.9941	1.0000	0.9970	836
fr	0.8780	1.0000	0.9351	36
it	0.8571	0.9677	0.9091	31
unknown	0.0000	0.0000	0.0000	15
accuracy			0.9951	4871
macro avg	0.7874	0.8274	0.8062	4871
weighted avg	0.9922	0.9951	0.9936	4871

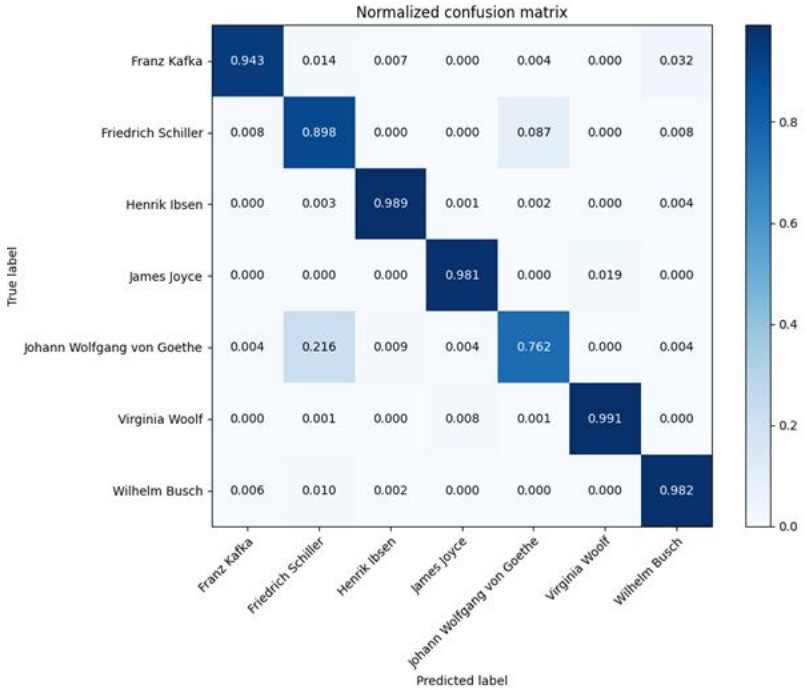


Ergebnisse: Autorenklassifikation

Results:
- F-score (micro) 0.9696
- F-score (macro) 0.9356
- Accuracy 0.9696

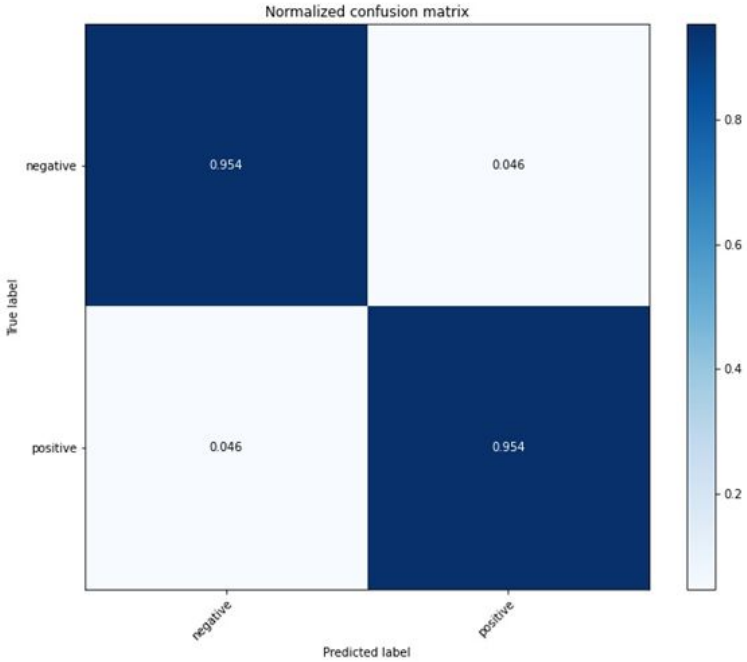
By class:

	precision	recall	f1-score	support
Virginia Woolf	0.9931	0.9911	0.9921	1901
Henrik Ibsen	0.9944	0.9888	0.9916	893
James Joyce	0.9752	0.9809	0.9781	682
Wilhelm Busch	0.9746	0.9824	0.9785	625
Friedrich Schiller	0.7900	0.8977	0.8404	264
Franz Kafka	0.9741	0.9427	0.9581	279
Johann Wolfgang von Goethe	0.8650	0.7621	0.8103	227
accuracy			0.9696	4871
macro avg	0.9381	0.9351	0.9356	4871
weighted avg	0.9704	0.9696	0.9697	4871



Ergebnisse: Sentimentanalyse mit RoBERTa

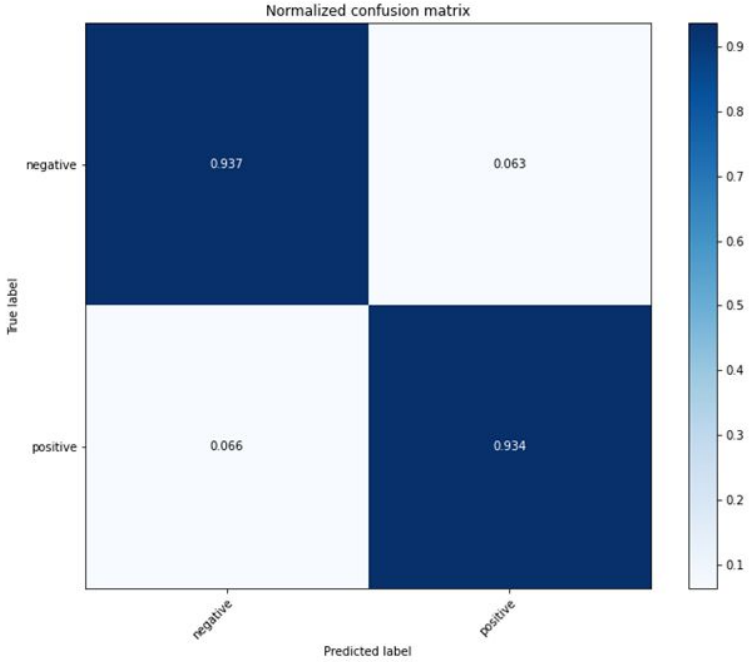
```
#Info:
Classifier: roberta-base
Labels: ['negative', 'positive']

#Counts:
Number of training data_records: 30000
Number of classified data_records: 10000

#Hyperparameters:
Epochs: 2
Batch size: 16
Learning rate: 5e-05

#Classification report:
              precision    recall  f1-score   support

    negative      0.953     0.954     0.954      4985
    positive      0.955     0.954     0.954      5015

    accuracy                        0.954     10000
   macro avg      0.954     0.954     0.954     10000
weighted avg      0.954     0.954     0.954     10000
```



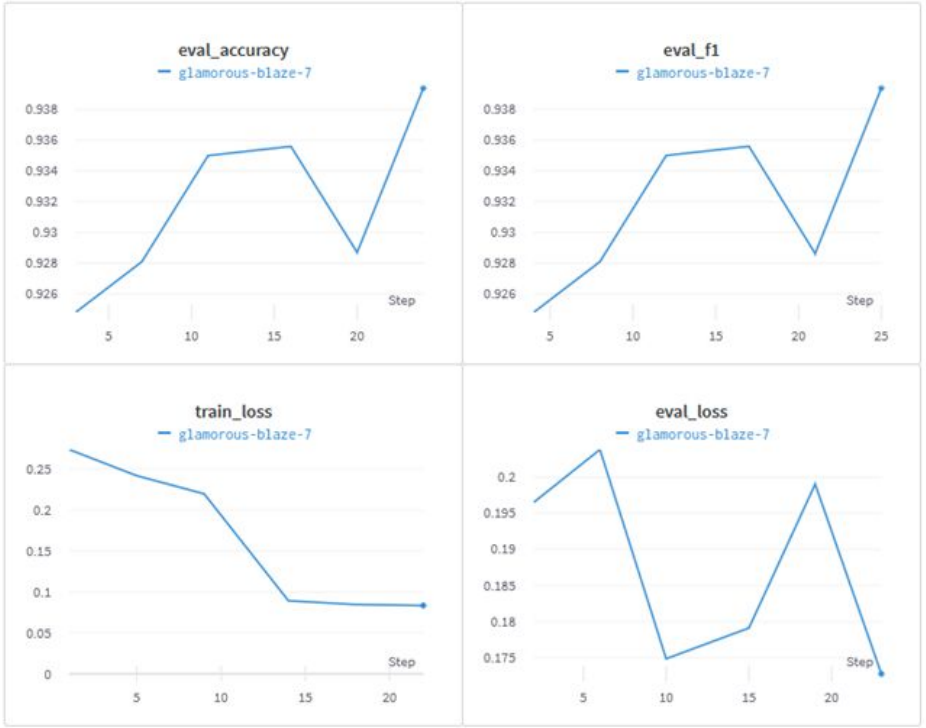
Ergebnisse: Sentimentanalyse mit RoBERTa

Ergebnisse: Sentimentanalyse mit DistilBERT

```
#Info:
Classifier: distilbert-base-uncased
Labels: ['negative', 'positive']

#Counts:
Number of training data_records: 30000
Number of classified data_records: 10000

#Hyperparameters:
Epochs: 2
Batch size: 16
Learning rate: 5e-05

#Classification report:
              precision    recall  f1-score   support

    negative      0.934     0.937     0.936      4985
    positive      0.938     0.934     0.936      5015

    accuracy                        0.936     10000
   macro avg      0.936     0.936     0.936     10000
weighted avg      0.936     0.936     0.936     10000
```



Ergebnisse: Sentimentanalyse mit DistilBERT



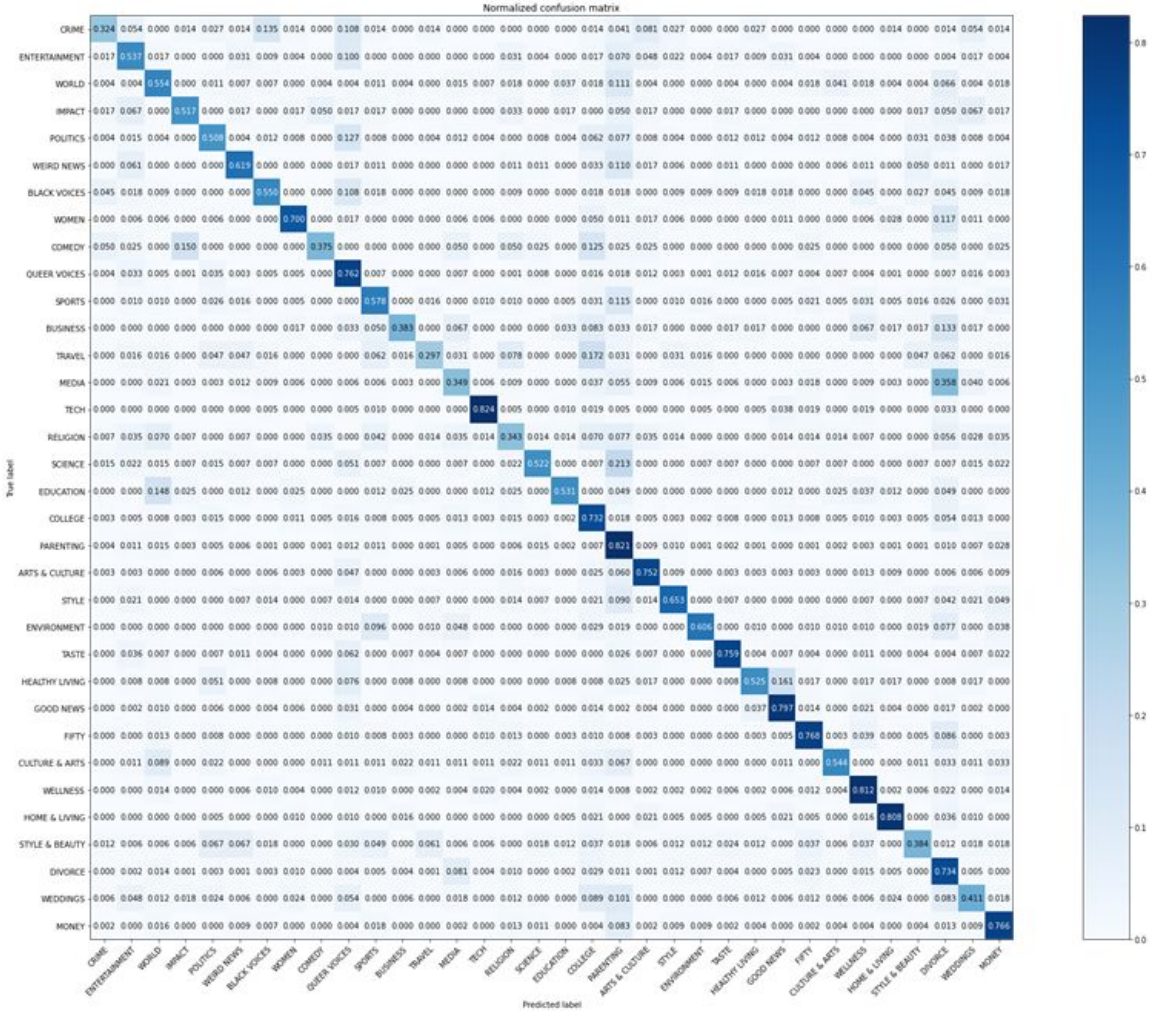
Ergebnisse: News mit RoBERTa

```
#Info:
Classifier: roberta-base
Labels: 34

#Hyperparameters:
Epochs: 5
Batch size: 32
Learning rate: 5e-5

#ShortResults
Loss:  1.1375532150268555
Accuracy:  0.6905977725982666
Top-3 accuracy:  0.8796435594558716
```

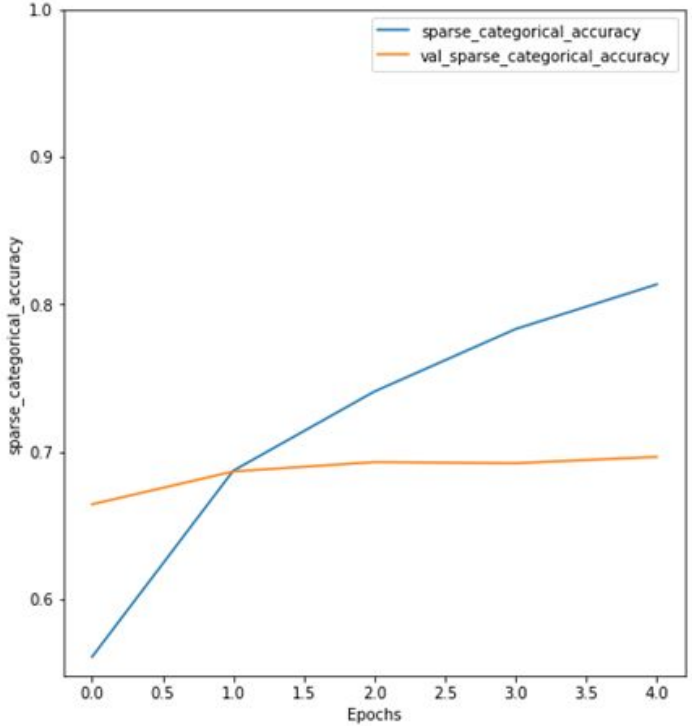
#CompleteResults

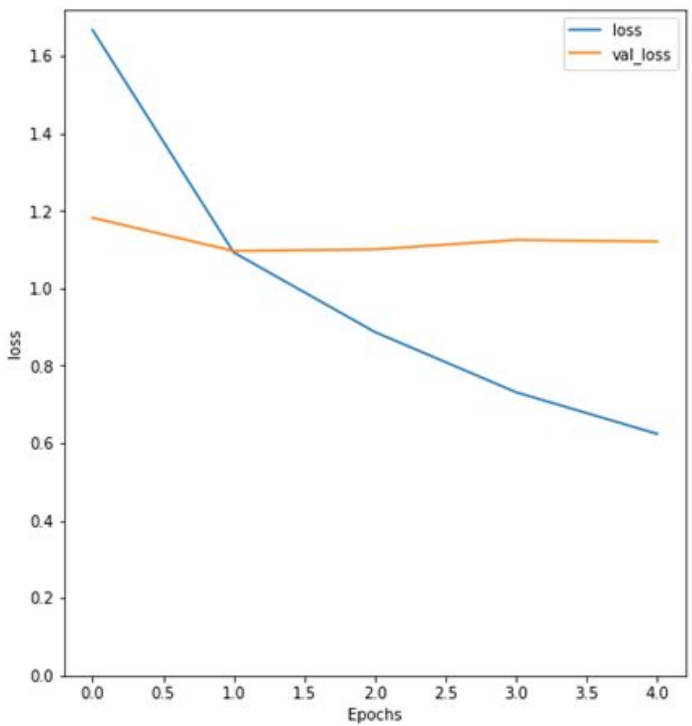
	precision	recall	f1-score	support
ARTS & CULTURE	0.42	0.32	0.37	74
BLACK VOICES	0.52	0.54	0.53	229
BUSINESS	0.54	0.55	0.55	271
COLLEGE	0.55	0.52	0.53	60
COMEDY	0.57	0.51	0.54	260
CRIME	0.65	0.62	0.63	181
CULTURE & ARTS	0.55	0.55	0.55	111
DIVORCE	0.75	0.70	0.72	180
EDUCATION	0.47	0.38	0.42	40
ENTERTAINMENT	0.73	0.76	0.74	765
ENVIRONMENT	0.53	0.58	0.55	192
FIFTY	0.49	0.38	0.43	60
GOOD NEWS	0.40	0.30	0.34	64
HEALTHY LIVING	0.45	0.35	0.39	327
HOME & LIVING	0.81	0.82	0.82	210
IMPACT	0.35	0.34	0.35	143
MEDIA	0.57	0.52	0.54	136
MONEY	0.58	0.53	0.55	81
PARENTING	0.68	0.73	0.71	613
POLITICS	0.80	0.82	0.81	1681
QUEER VOICES	0.75	0.75	0.75	318
RELIGION	0.60	0.65	0.63	144
SCIENCE	0.67	0.61	0.64	104
SPORTS	0.81	0.76	0.78	274
STYLE	0.53	0.53	0.53	118
STYLE & BEAUTY	0.83	0.80	0.81	483
TASTE	0.77	0.77	0.77	384
TECH	0.55	0.54	0.55	90
TRAVEL	0.79	0.81	0.80	504
WEDDINGS	0.83	0.81	0.82	193
WEIRD NEWS	0.58	0.38	0.46	164
WELLNESS	0.64	0.73	0.69	917
WOMEN	0.42	0.41	0.42	168
WORLD	0.74	0.77	0.75	448
accuracy			0.69	9987
macro avg	0.62	0.59	0.60	9987
weighted avg	0.69	0.69	0.69	9987

Ergebnisse: News mit RoBERTa



Ergebnisse: News mit RoBERTa





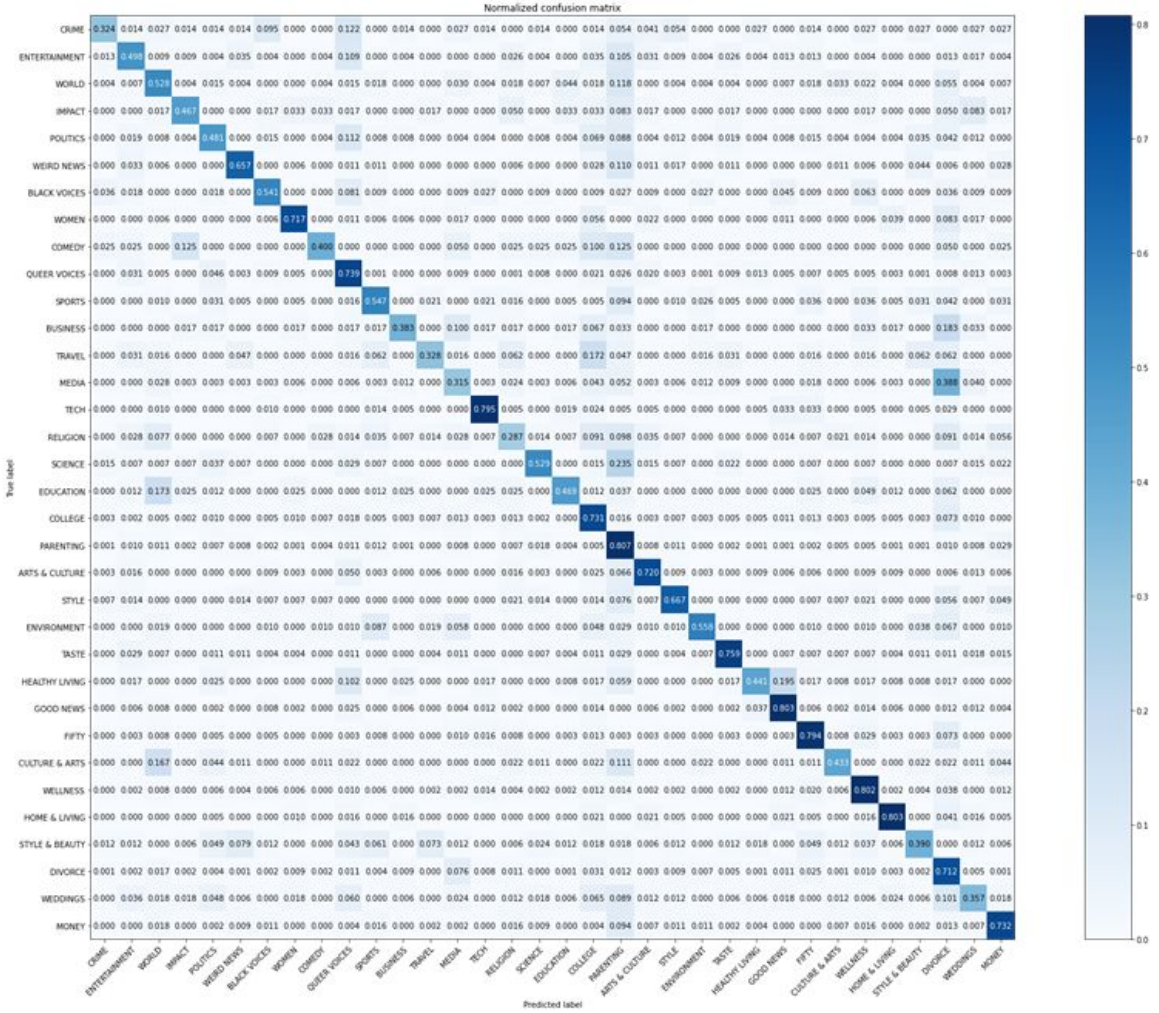
Ergebnisse: News mit DistilBERT

```
#Info:
Classifier: distilbert-base-uncased
Labels: 34

#Hyperparameters:
Epochs: 5
Batch size: 32
Learning rate: 5e-5

#ShortResults
Loss:  1.2675701379776
Accuracy:  0.6745769381523132
Top-3 accuracy:  0.8678281903266907
```

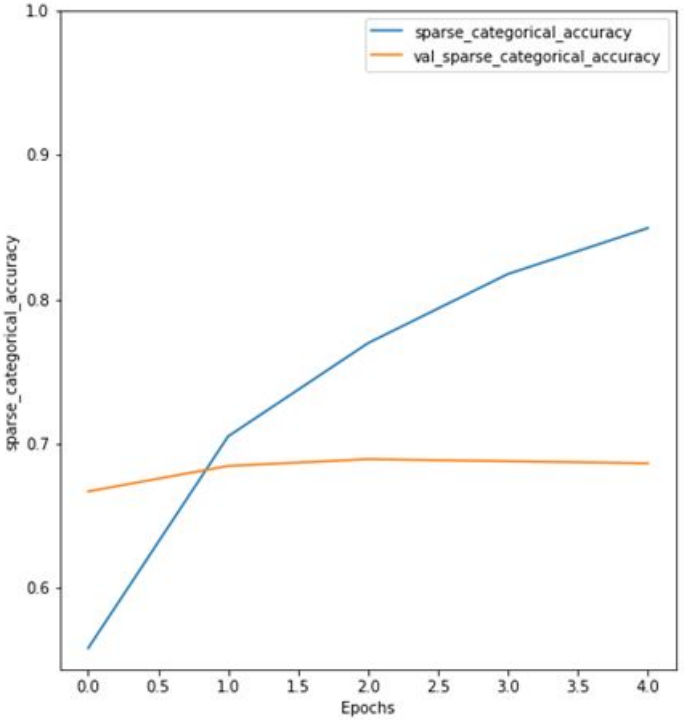
#CompleteResults

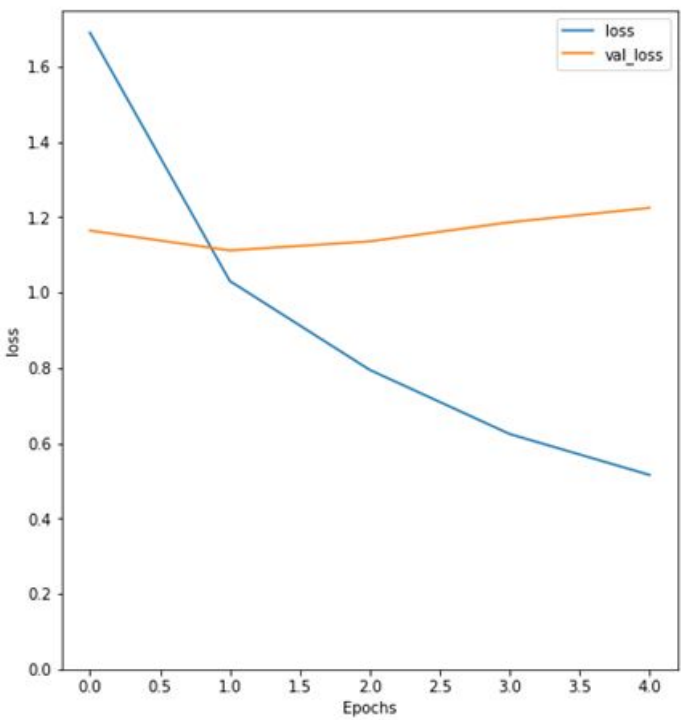
	precision	recall	f1-score	support
ARTS & CULTURE	0.56	0.32	0.41	74
BLACK VOICES	0.54	0.50	0.52	229
BUSINESS	0.52	0.53	0.52	271
COLLEGE	0.53	0.47	0.50	60
COMEDY	0.53	0.48	0.50	260
CRIME	0.67	0.66	0.66	181
CULTURE & ARTS	0.52	0.54	0.53	111
DIVORCE	0.76	0.72	0.74	180
EDUCATION	0.39	0.40	0.40	40
ENTERTAINMENT	0.73	0.74	0.74	765
ENVIRONMENT	0.54	0.55	0.54	192
FIFTY	0.40	0.38	0.39	60
GOOD NEWS	0.41	0.33	0.37	64
HEALTHY LIVING	0.41	0.31	0.36	327
HOME & LIVING	0.79	0.80	0.79	210
IMPACT	0.31	0.29	0.30	143
MEDIA	0.52	0.53	0.53	136
MONEY	0.49	0.47	0.48	81
PARENTING	0.67	0.73	0.70	613
POLITICS	0.78	0.81	0.80	1681
QUEER VOICES	0.74	0.72	0.73	318
RELIGION	0.59	0.67	0.62	144
SCIENCE	0.62	0.56	0.59	104
SPORTS	0.80	0.76	0.78	274
STYLE	0.53	0.44	0.48	118
STYLE & BEAUTY	0.82	0.80	0.81	483
TASTE	0.74	0.79	0.76	384
TECH	0.45	0.43	0.44	90
TRAVEL	0.78	0.80	0.79	504
WEDDINGS	0.81	0.80	0.81	193
WEIRD NEWS	0.55	0.39	0.46	164
WELLNESS	0.62	0.71	0.66	917
WOMEN	0.38	0.36	0.37	168
WORLD	0.75	0.73	0.74	448
accuracy			0.67	9987
macro avg	0.60	0.57	0.58	9987
weighted avg	0.67	0.67	0.67	9987

Ergebnisse: News mit DistilBERT



Ergebnisse: News mit RoBERTa





Fazit

- Die Verwendung großer vortrainierter Transformer-Modelle kann je nach Aufgabe deutliche Verbesserungen bringen.
- Transformer-Modelle weisen jedoch deutlich höhere Trainingszeiten auf (in unserem Fall je nach Datensatz und Modell zwischen 18 und 40 Minuten pro Epoche).
- Bei bereits “gelösten” Aufgaben wie z.B. Sprachidentifizierung lohnt es sich nicht, ein Transformer-Modell zu verwenden.
- Besonders bei Aufgaben mit Fokus auf die Semantik, können Transformer deutlich bessere Ergebnisse erzielen.

Literatur

- Alammar, J (2018). The Illustrated Transformer [Blog post]. Retrieved from *https://jalammar.github.io/illustrated-transformer/*
- Akbik, Alan, et al. "FLAIR: An easy-to-use framework for state-of-the-art NLP." Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics (demonstrations). 2019.
- Wolf, Thomas, et al. "Transformers: State-of-the-art natural language processing." Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations. 2020.
- Vaswani, Ashish et al. “Attention is All you Need.” ArXiv abs/1706.03762 (2017)
- Devlin, Jacob et al. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.” *ArXiv* abs/1810.04805 (2019)
- Sanh, Victor et al. “DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter.” *ArXiv* abs/1910.01108 (2019)
- Liu et al. “RoBERTa: A Robustly Optimized BERT Pretraining Approach.” https://openreview.net/forum?id=SyxS0T4tvS