Attention is all you need

Marwin Härttrich
Laura Luckert
Ingo Ziegler

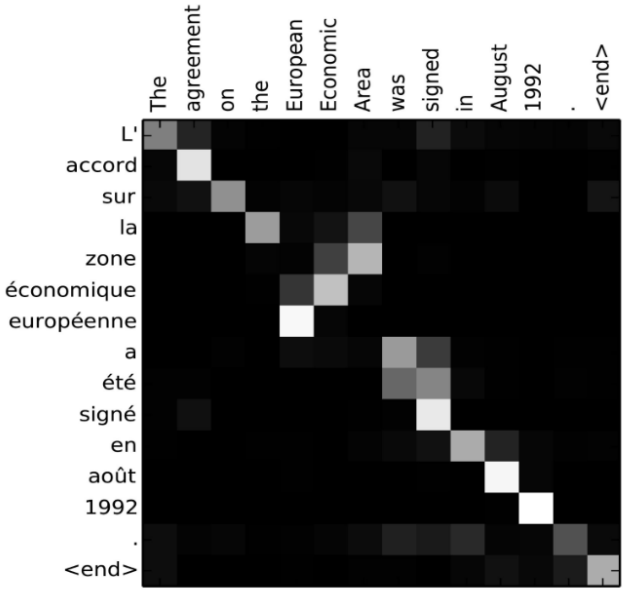
Text Classification & Clustering @CIS LMU, Dr. Stefan Langer
16.01.2023

Agenda

1. Motivation und Historie Attention
2. Soft Attention
3. Scaled Dot-Product Attention & Multi-Head Attention
4. Implementierung & Evaluation
5. Ergebnisse mit und ohne Attention
6. Fazit

Motivation und Historie Attention

Cross-Attention



Bahdanau et al., 2015[1]

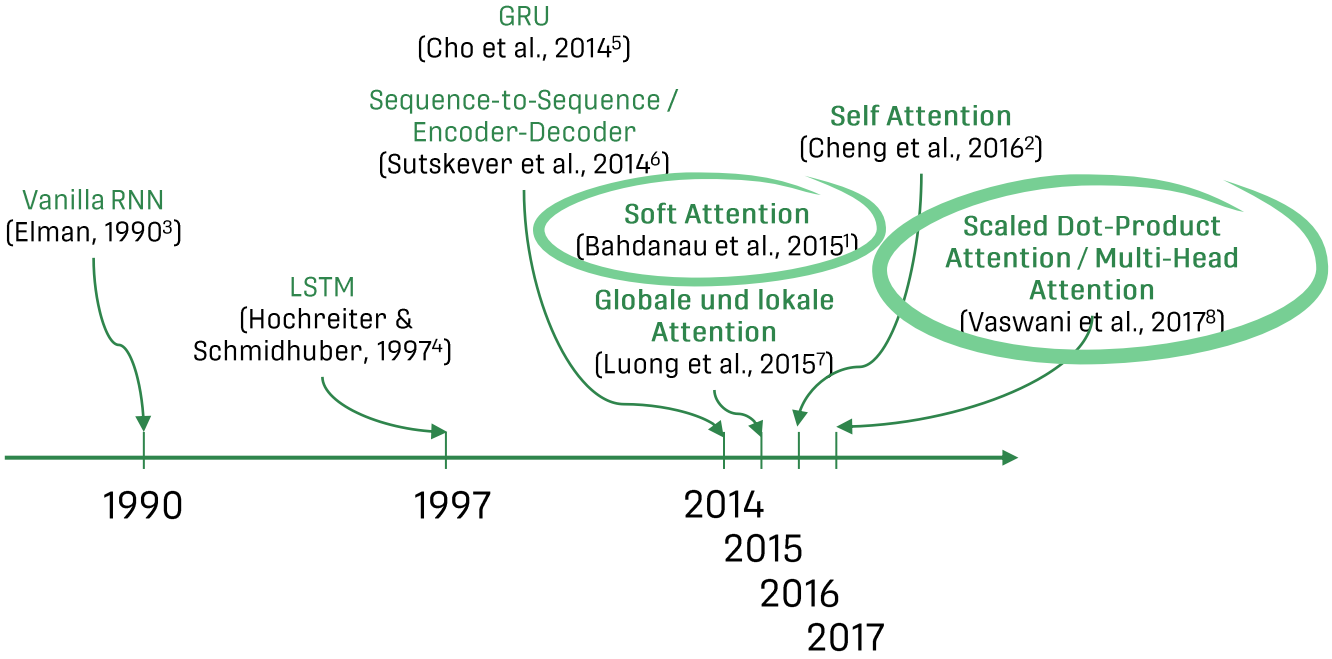
Self-Attention

The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .

Cheng et al., 2016[2]

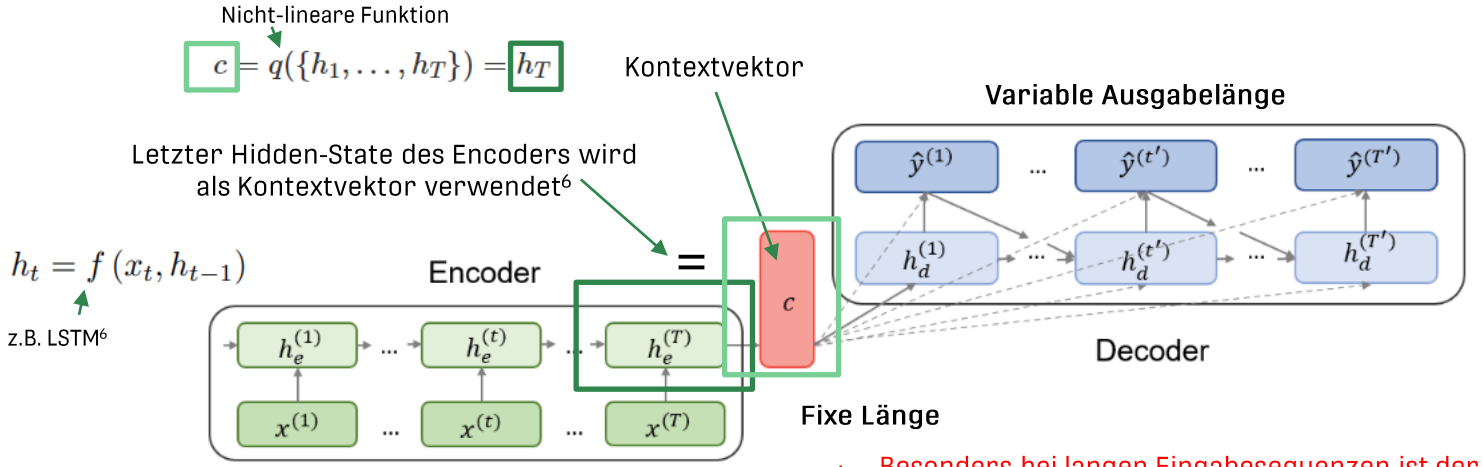
Motivation und Historie Attention

Attention im Zeitverlauf



Motivation für Attention

RNN-Encoder-Decoder



Besonders bei langen Eingabesequenzen ist der Kontextvektor nicht in der Lage, die relevante Information des Eingabetextes abzubilden
→ schlechtere Ergebnisse in der Ausgabe

Abb. 1: RNN-Encoder-Decoder und Kontextvektor[9]

Soft Attention[1]

Durch Attention wird aus einem Kontextvektor c fixer Länge, eine Sequenz an Kontextvektoren c_t variabler Länge

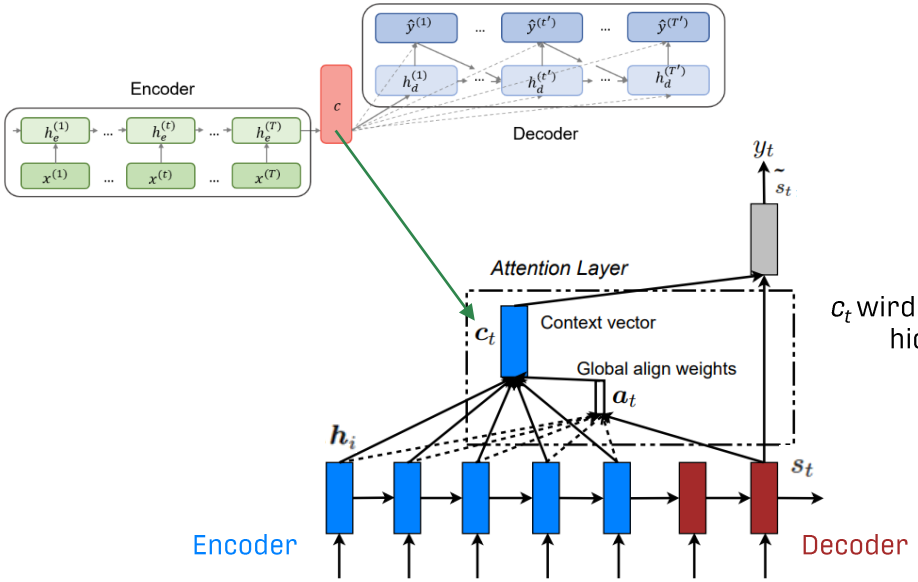


Abb. 2: Soft/Global Attention[7]

$$h_i = [\overrightarrow{h_i}; \overleftarrow{h_i}], i = 1, \ldots, n$$

Bidirektionalität:
h_i hat einen starken Fokus auf das i-te Eingabewort und dessen Umgebung

c_t wird als gewichtete Summe der hidden States berechnet

$$c_t = \sum_{i=1}^{T_x} \alpha_{t,i} h_i.$$

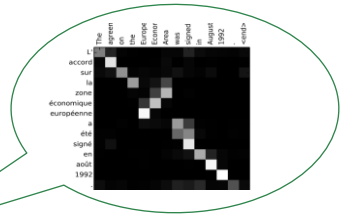
Der Gewichtungsfaktor $\alpha_{t,i}$ wird auch als Alignment Score bezeichnet

Soft Attention[1]

Decoder hidden State s_t

$$s_t = f(s_{t-1}, y_{t-1}, c_t)$$

s_{t-1} ist der hidden State des Decoders zum Zeitpunkt t-1



Alignment Score

$$\alpha_{t,i} = align(y_t, x_i) = \frac{exp(score(s_{t-1}, h_i))}{\sum_{i'=1}^{n} exp(score(s_{t-1}, h_{i'}))}$$

Softmax über eine Score-Funktion

Der Alignment Score modelliert, wie gut die Eingabe x_i und die Ausgabe y_t zueinander passen und weist das entsprechende Gewicht zu h_i zu

werden konkateniert

$$\text{score}(\boldsymbol{s}_t, \boldsymbol{h}_i) = \mathbf{v}_a^\top \tanh(\mathbf{W}_a[\boldsymbol{s}_{t-1}; \boldsymbol{h}_i])$$

Gewichte, die zusammen mit den anderen Modellparametern im Training gelernt werden

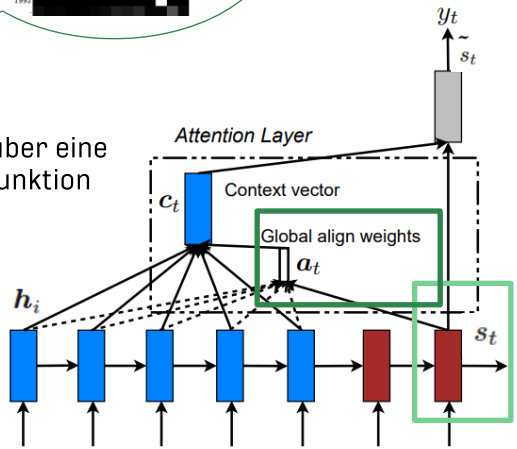


Abb. 2: Soft/Global Attention[7]

Von Soft Attention zur Scaled Dot-Product Attention

Soft Attention wird auch als Global Attention[7] bezeichnet, die Score-Funktion auch als Concat[7] oder Additive Attention[8]

Dot-Product in der Praxis wesentlich effizienter zu berechnen -> trotz ähnlicher theoretischer Komplexität

Warum **Scaled Dot-Product**? Für große *n* performt **additive Attention** besser -> für große *n* kann Softmax sehr kleine Gradienten haben

Name	Alignment Score Funktion	Quelle
Additive(*)	$\text{score}(\boldsymbol{s}_t, \boldsymbol{h}_i) = \mathbf{v}_a^\top \tanh(\mathbf{W}_a[\boldsymbol{s}_{t-1}; \boldsymbol{h}_i])$	Bahdanau2015[1]
Location-Base	$\alpha_{t,i} = \text{softmax}(\mathbf{W}_a \boldsymbol{s}_t)$ Note: This simplifies the softmax alignment to only depend on the target position.	Luong2015[7]
General	$\text{score}(\boldsymbol{s}_t, \boldsymbol{h}_i) = \boldsymbol{s}_t^\top \mathbf{W}_a \boldsymbol{h}_i$ where $\mathbf{W}_a$ is a trainable weight matrix in the attention layer.	Luong2015[7]
Dot-Product	$\text{score}(\boldsymbol{s}_t, \boldsymbol{h}_i) = \boldsymbol{s}_t^\top \boldsymbol{h}_i$	Luong2015[7]
Scaled Dot-Product(^)	$\text{score}(\boldsymbol{s}_t, \boldsymbol{h}_i) = \frac{\boldsymbol{s}_t^\top \boldsymbol{h}_i}{\sqrt{n}}$ Note: very similar to the dot-product attention except for a scaling factor; where n is the dimension of the source hidden state.	Vaswani2017[8]

Abb. 3: Alignment Score Funktionen[10]

Agenda

1. Motivation und Historie Attention
2. Soft Attention
3. **Scaled Dot-Product Attention & Multi-Head Attention**
4. Implementierung & Evaluation
5. Ergebnisse mit und ohne Attention
6. Fazit

Scaled Dot-Product Attention & Multi-Head Attention

Vorteile im Vergleich zu anderen Attention- Verfahren

- Verbesserte Trainingsstabilität
  - Skalierung reduziert Auftreten von Vanishing-Gradient
  - Skalierung reduziert Auftreten von Exploding-Gradient
- Effiziente Berechnung
  - Parallelisierung möglich
  - effiziente Berechnung auf GPUs
- Bessere Handhabung großer Dimensionen
  - Unempfindlich auf Input-Dimensionen

Dot-Product Bedeutung für Attention

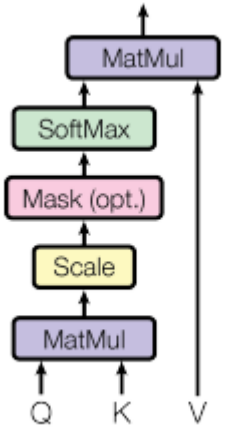
Dot-Product ist $a \cdot b = \sum_i a_i \cdot b_i$

- Eintragsweise Ähnlichkeit der Vektoren
- Ähnlichkeitsmaß

Matrixmultiplikation ergibt paarweise Ähnlichkeiten

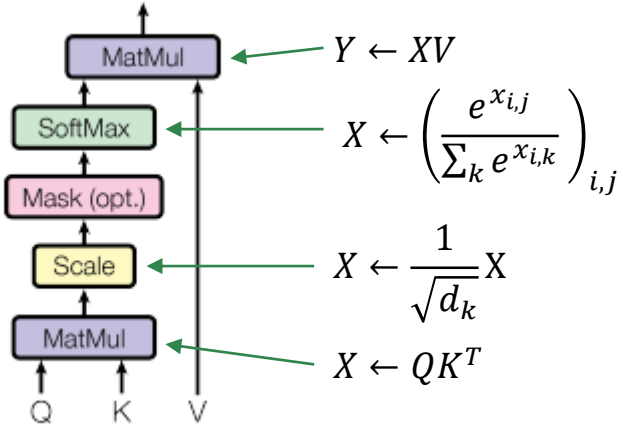
$$\begin{pmatrix} a_{11} & \cdots & a_{n1} \\ \vdots & \ddots & \vdots \\ a_{1m} & \cdots & a_{nm} \end{pmatrix} \cdot \left(\begin{array}{c|c|c} b_{11} & \cdots & b_{k1} \\ \vdots & \ddots & \vdots \\ b_{1n} & \cdots & b_{kn} \end{array}\right) = \begin{pmatrix} a_{\_1} \\ \vdots \\ a_{\_m} \end{pmatrix} (b_{1\_} \quad \cdots \quad b_{k\_}) = \begin{pmatrix} a_{1\_} \cdot b_{\_1} & \cdots & a_{1\_} \cdot b_{\_k} \\ \vdots & \ddots & \vdots \\ a_{m\_} \cdot b_{\_1} & \cdots & a_{m\_} \cdot b_{\_k} \end{pmatrix}$$

Scaled Dot-Product Attention[8]



- Eingabe **Q**ueries, **K**eys, **V**alues Matrizen
  $Q, K \in \mathbb{R}^{n \times d_k}$ und $V \in \mathbb{R}^{n \times d_v}$
- Ausgabe
  $X \in \mathbb{R}^{n \times n}, Y \in \mathbb{R}^{n \times d_v}$
- Mit n der Anzahl an gleichzeitig zu berechnenden Werten
- X paarweise Ähnlichkeiten von Queries und Keys
- Y Stellenweise Skalierung der Values um Ähnlichkeiten

Scaled Dot-Product Attention[8]



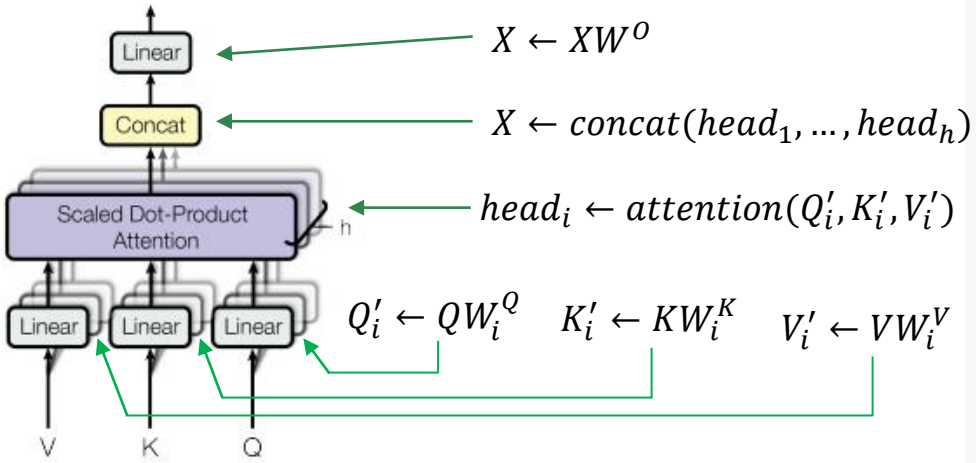

```python
class Attention(torch.nn.Module):
    def __init__(self, mask=None):
        super().__init__()
        self.softmax = torch.nn.Softmax(dim=1)
        self.mask = mask

    def forward(self, Q, K, V):
        d_k = K.size()[1]
        X = torch.mul(Q, torch.transpose(K, 0, 1))
        X = torch.mul(X, math.sqrt(d_k))
        if self.mask is not None:
          X = X.masked_fill(mask == 0, -9e15)
        X = self.softmax(X)
        Y = torch.mul(X, V)
        return X, Y
```

Angelehnt an Zhang, 2021[11]

$$attention(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Multi-Head Attention[8]




```python
class MultiHead(torch.nn.Module):
    def __init__(self, input_dim, embed_dim, num_heads, mask=None):
        super().__init__()
        assert embed_dim % num_heads == 0, "embed_dim % num_heads != 0"
        self.embed_dim = embed_dim
        self.num_heads = num_heads
        self.head_dim = embed_dim // num_heads

        self.q_proj = torch.nn.Linear(input_dim, emb_dim)
        self.k_proj = torch.nn.Linear(input_dim, emb_dim)
        self.v_proj = torch.nn.Linear(input_dim, emb_dim)
        self.attention = Attention(mask)
        self.o_proj = nn.Linear(embed_dim, embed_dim)

    def forward(self, Q, K, V, mask=None):
        QW, KW, VW = self.q_proj(Q), self.k_proj(K), self.v_proj(V)
        X, Y = self.attention(QW, KW, VW)
        X = self.o_proj(X)
        return X, Y
```

Angelehnt an Zhang, 2021[11]

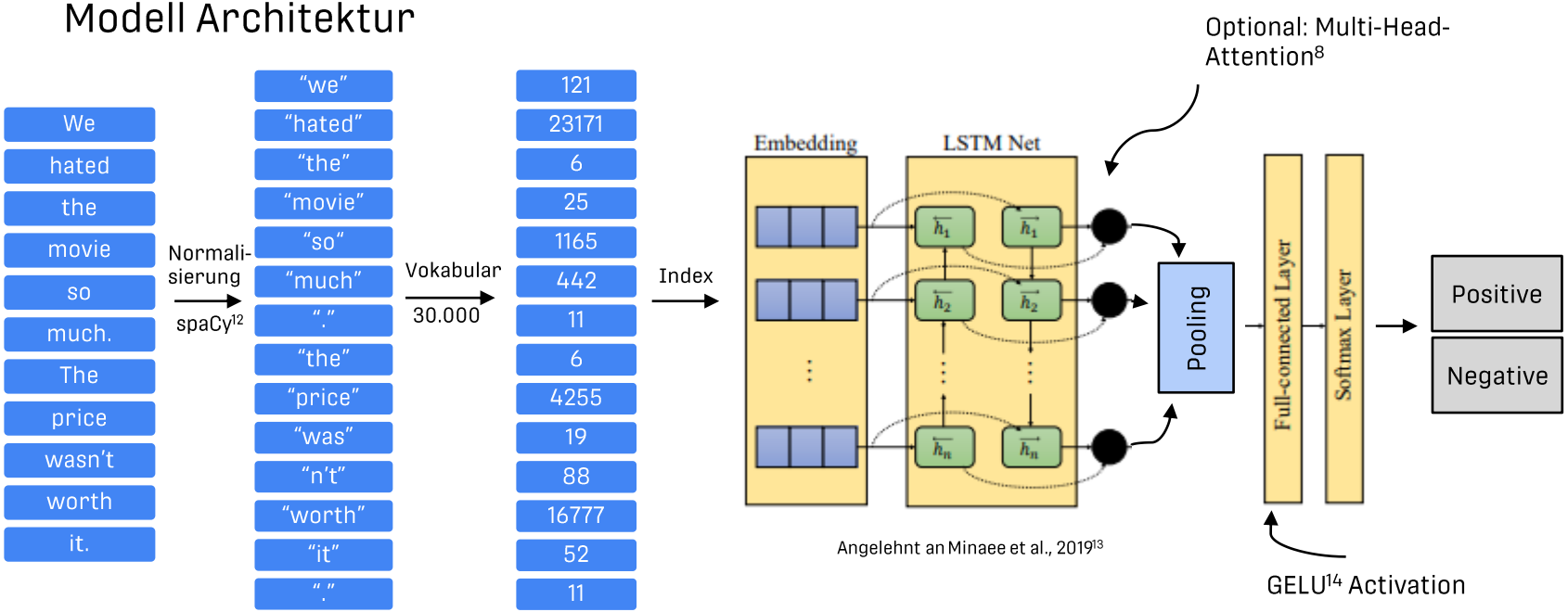
$$MultiHead(Q,K,V) \leftarrow concat(attention(QW_1^Q, KW_1^K, VW_1^V), \dots, attention(QW_h^Q, KW_h^K, VW_h^V))W^O$$

Agenda

1. Motivation und Historie Attention
2. Soft Attention
3. Scaled Dot-Product Attention & Multi-Head Attention
4. **Implementierung & Evaluation**
5. Ergebnisse mit und ohne Attention
6. Fazit

Implementierung & Evaluation

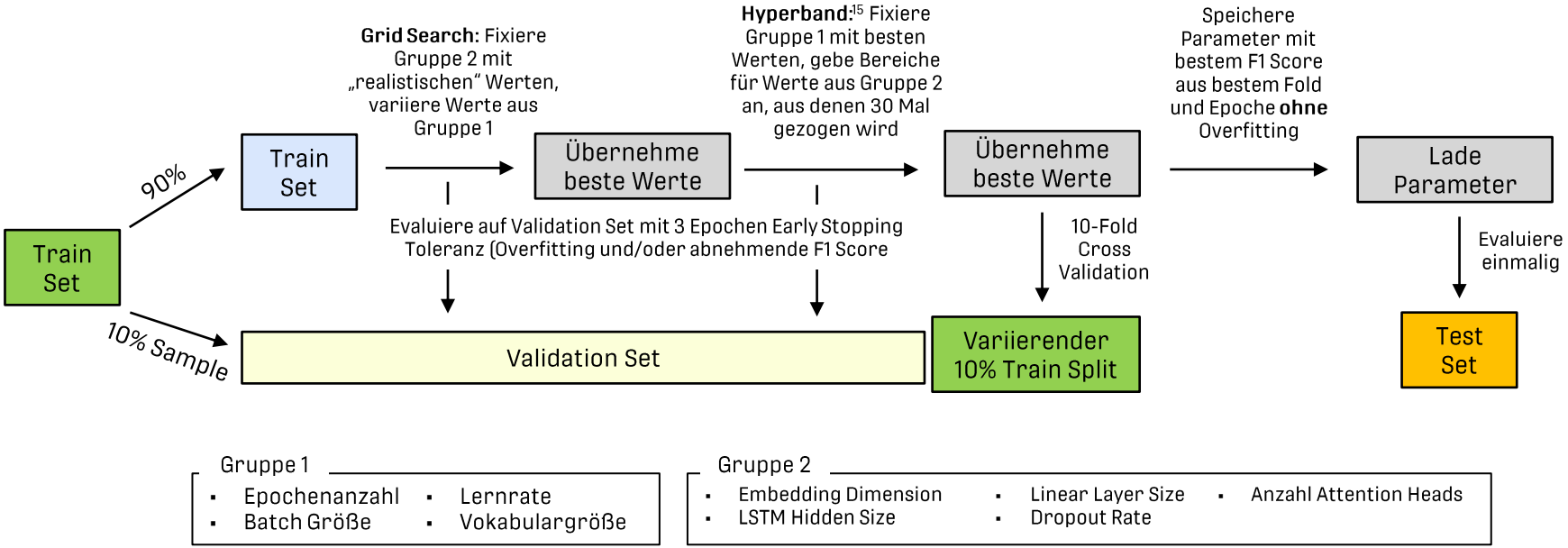
Modell Architektur



Angelehnt an Minaee et al., 2019[13]

Implementierung & Evaluation

Hyperparameter & Hyperparameter Suche



Implementierung & Evaluation

Reproduzierbarkeit

Software Details
- Neuronale Netze: PyTorch[16] v1.11.0
- Hyperparameter Tuning: Ray[17] v1.13.0
- Tokenisierung: spaCy[12] v3.4.0
- Evaluierung: Scikit-learn[18] v1.0.1

Hardware Details
- Training auf GPU (RTX 3090)

Loss und Optimierung
- (Binary) Cross Entropy Loss[19]
- Adam Optimizer[20]

```python
def seed_everything(seed: int):
    random.seed(seed)
    os.environ["PYTHONHASHSEED"] = str(seed)
    np.random.seed(seed)
    torch.manual_seed(seed)
    torch.cuda.manual_seed(seed)
    torch.backends.cudnn.deterministic = True
    torch.backends.cudnn.benchmark = True


seed_everything(1234)
```

Für alle Modelle und Hyperparametersuchen

Agenda

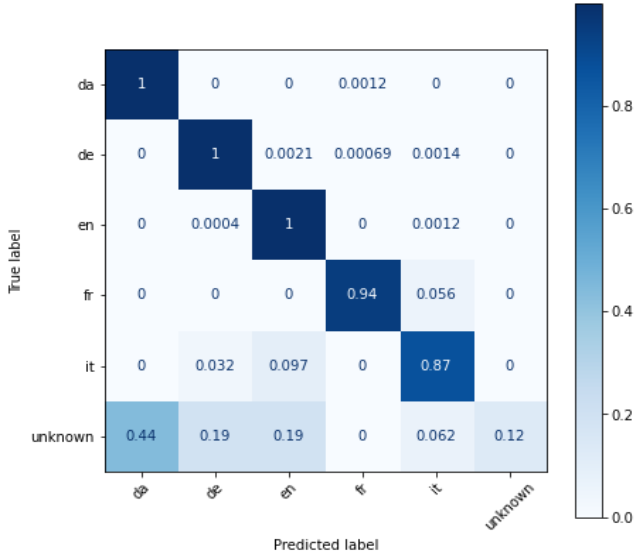
1. Motivation und Historie Attention
2. Soft Attention
3. Scaled Dot-Product Attention & Multi-Head Attention
4. Implementierung & Evaluation
5. **Ergebnisse mit und ohne Attention**
6. Fazit

Briefsammlung - Spracherkennung

Ohne Attention

	precision	recall	f1-score	support
en	0.9964	0.9984	0.9974	2517
de	0.9965	0.9958	0.9962	1443
da	0.9917	0.9988	0.9952	838
fr	0.9444	0.9444	0.9444	36
it	0.7714	0.8710	0.8182	31
unknown	1.0000	0.1250	0.2222	16
accuracy			0.9936	4881
macro avg	0.9501	0.8222	0.8289	4881
weighted avg	0.9938	0.9936	0.9926	4881

Standardabweichung: 0.451%



Gruppe 1

- Epochenanzahl = 15
- Batch Größe = 64
- Lernrate = 0.0001
- Vokabulargröße = 30000

Gruppe 2

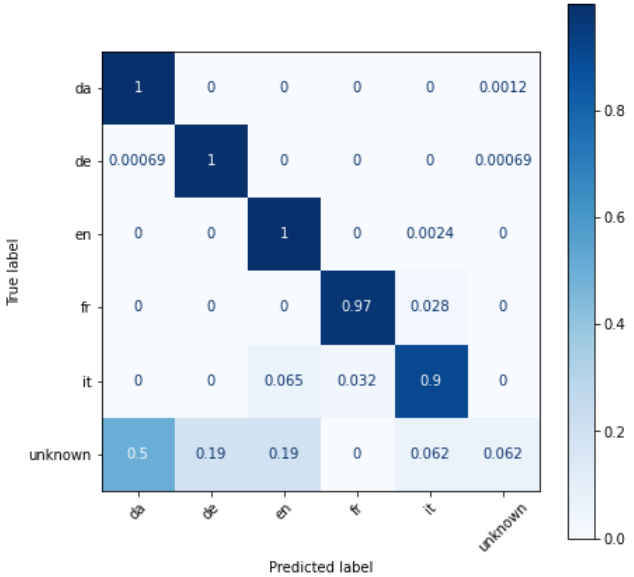
- Embedding Dimension = 65
- LSTM Hidden Size = 88
- Linear Layer Size = 72
- Dropout Rate = 0.12
- Anzahl Attention Heads = 0
- Anzahl Parameter = 2.1Mio.

Briefsammlung - Spracherkennung

Mit Attention

	precision	recall	f1-score	support
en	0.9980	0.9976	0.9978	2517
de	0.9979	0.9986	0.9983	1443
da	0.9894	0.9988	0.9941	838
fr	0.9722	0.9722	0.9722	36
it	0.7778	0.9032	0.8358	31
unknown	0.3333	0.0625	0.1053	16
accuracy			0.9943	4881
macro avg	0.8448	0.8222	0.8172	4881
weighted avg	0.9927	0.9943	0.9932	4881

Standardabweichung: 0.155%
Steigerung durch Attention: +0.07 p.p.



Gruppe 1

- Epochenanzahl = 15
- Batch Größe = 64
- Lernrate = 0.0001
- Vokabulargröße = 30000

Gruppe 2

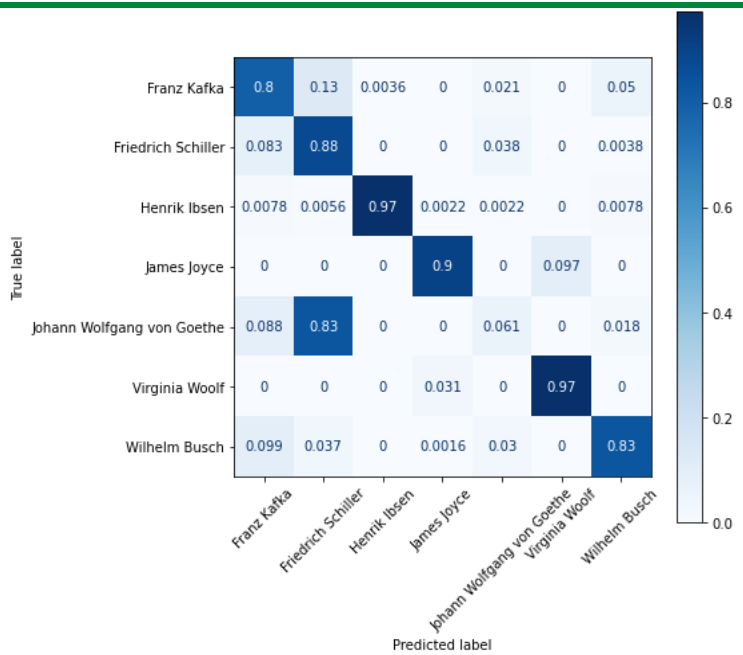
- Embedding Dimension = 80
- LSTM Hidden Size = 55
- Linear Layer Size = 60
- Dropout Rate = 0.19
- Anzahl Attention Heads = 1
- Anzahl Parameter = 2.5Mio.

Briefsammlung - Autoren

Ohne Attention

	precision	recall	f1-score	support
Virginia Woolf	0.9654	0.9690	0.9672	1901
Henrik Ibsen	0.9989	0.9744	0.9865	897
James Joyce	0.9086	0.9032	0.9059	682
Wilhelm Busch	0.9526	0.8325	0.8885	627
Franz Kafka	0.6677	0.7964	0.7264	280
Friedrich Schiller	0.4784	0.8759	0.6189	266
Johann Wolfgang von Goethe	0.2745	0.0614	0.1004	228
accuracy			0.8859	4881
macro avg	0.7494	0.7733	0.7419	4881
weighted avg	0.8861	0.8859	0.8788	4881

Standardabweichung: 2.682%



Gruppe 1

- Epochenanzahl = 20
- Batch Größe = 64
- Lernrate = 0.0001
- Vokabulargröße = 30000

Gruppe 2

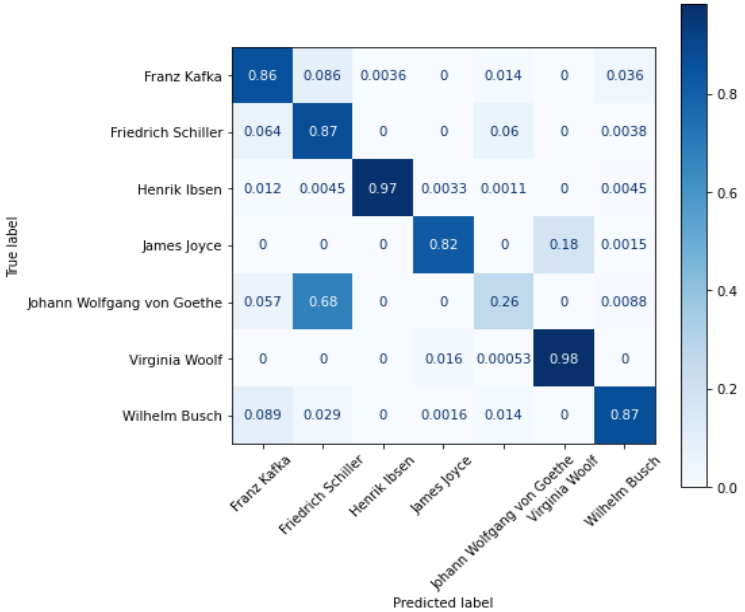
- Embedding Dimension=130
- LSTM Hidden Size = 277
- Linear Layer Size = 275
- Dropout Rate = 0.45
- Anzahl Attention Heads = 0
- Anzahl Parameter = 4.9Mio.

Briefsammlung - Autoren

Mit Attention

	precision	recall	f1-score	support
Virginia Woolf	0.9397	0.9837	0.9612	1901
Henrik Ibsen	0.9989	0.9744	0.9865	897
James Joyce	0.9429	0.8226	0.8786	682
Wilhelm Busch	0.9679	0.8660	0.9141	627
Franz Kafka	0.7130	0.8607	0.7799	280
Friedrich Schiller	0.5370	0.8722	0.6648	266
Johann Wolfgang von Goethe	0.6556	0.2588	0.3711	228
accuracy			0.8974	4881
macro avg	0.8221	0.8055	0.7937	4881
weighted avg	0.9064	0.8974	0.8941	4881

Standardabweichung: 4.59%
Steigerung durch Attention: +1.15 p.p.



Gruppe 1

- Epochenanzahl = 15
- Batch Größe = 64
- Lernrate = 0.0001
- Vokabulargröße = 30000

Gruppe 2

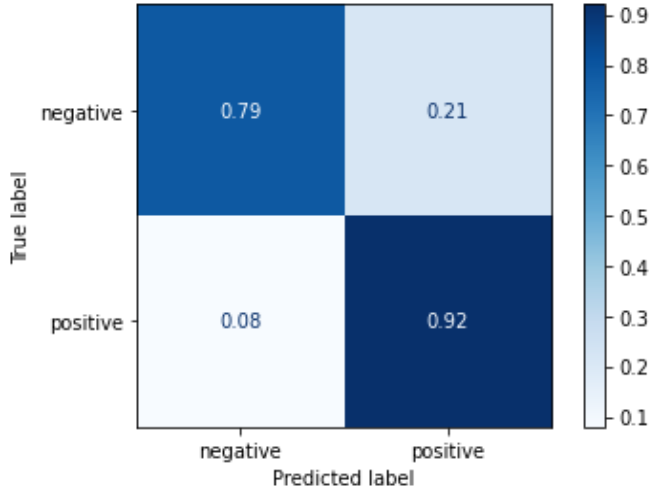
- Embedding Dimension=168
- LSTM Hidden Size = 282
- Linear Layer Size = 137
- Dropout Rate = 0.48
- Anzahl Attention Heads = 1
- Anzahl Parameter = 7.4Mio.

IMDB – Sentiment Analyse

Ohne Attention

	precision	recall	f1-score	support
negative	0.9077	0.7870	0.8430	4985
positive	0.8130	0.9204	0.8634	5015
accuracy			0.8539	10000
macro avg	0.8603	0.8537	0.8532	10000
weighted avg	0.8602	0.8539	0.8532	10000

Standardabweichung: 2.172%



Gruppe 1

- Epochenanzahl = 20
- Batch Größe = 64
- Lernrate = 0.0001
- Vokabulargröße = 30000

Gruppe 2

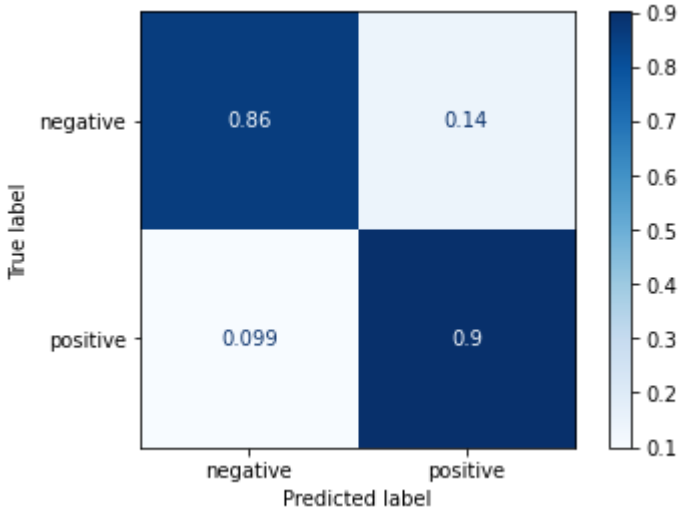
- Embedding Dimension=63
- LSTM Hidden Size = 266
- Linear Layer Size = 196
- Dropout Rate = 0.48
- Anzahl Attention Heads = 0
- Anzahl Parameter = 2.7Mio.

IMDB – Sentiment Analyse

Mit Attention

	precision	recall	f1-score	support
negative	0.8956	0.8570	0.8759	4985
positive	0.8637	0.9007	0.8818	5015
accuracy			0.8789	10000
macro avg	0.8796	0.8788	0.8788	10000
weighted avg	0.8796	0.8789	0.8788	10000

Standardabweichung: 1.579%
Steigerung durch Attention: +2.5 p.p.



Gruppe 1

- Epochenanzahl = 15
- Batch Größe = 64
- Lernrate = 0.0001
- Vokabulargröße = 30000

Gruppe 2

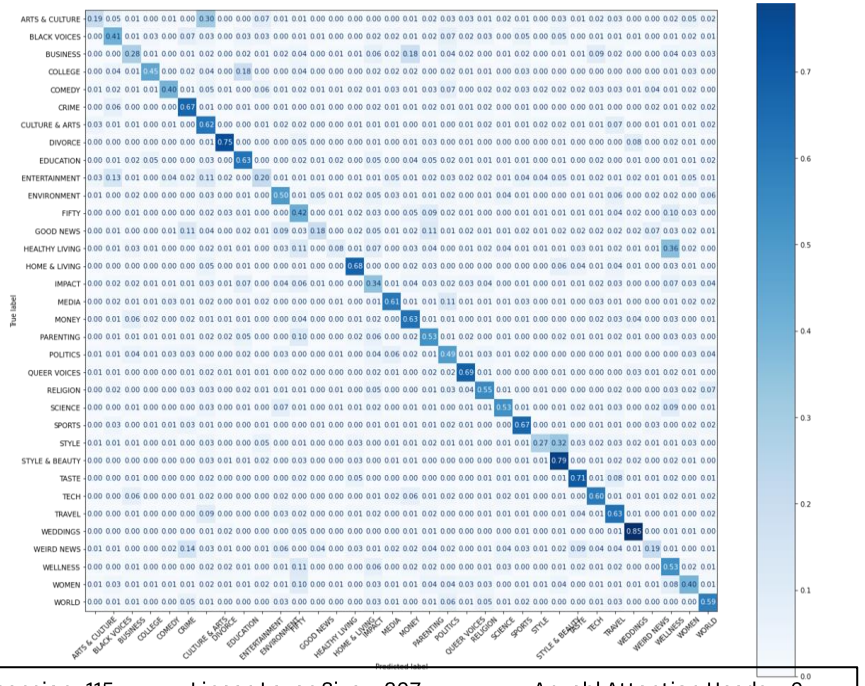
- Embedding Dimension=184
- LSTM Hidden Size = 256
- Linear Layer Size = 299
- Dropout Rate = 0.5
- Anzahl Attention Heads = 8
- Anzahl Parameter = 7.6Mio.

News – Überschriften & Beschreibung

Ohne Attention

POLITICS	0.4295	0.4857	0.4559	420
RELIGION	0.5931	0.5500	0.5708	440
MONEY	0.4607	0.6327	0.5332	324
FIFTY	0.2406	0.4212	0.3063	273
GOOD NEWS	0.4590	0.1836	0.2623	305
ARTS & CULTURE	0.4454	0.1879	0.2643	282
COLLEGE	0.6115	0.4528	0.5203	212
EDUCATION	0.4079	0.6263	0.4940	198
accuracy			0.5073	12824
macro avg	0.5115	0.5018	0.4879	12824
weighted avg	0.5171	0.5073	0.4942	12824

Standardabweichung: 1.785%



Gruppe 1

- Epochenanzahl = 16
- Batch Größe = 64
- Lernrate = 0.001
- Vokabulargröße = 30000

Gruppe 2

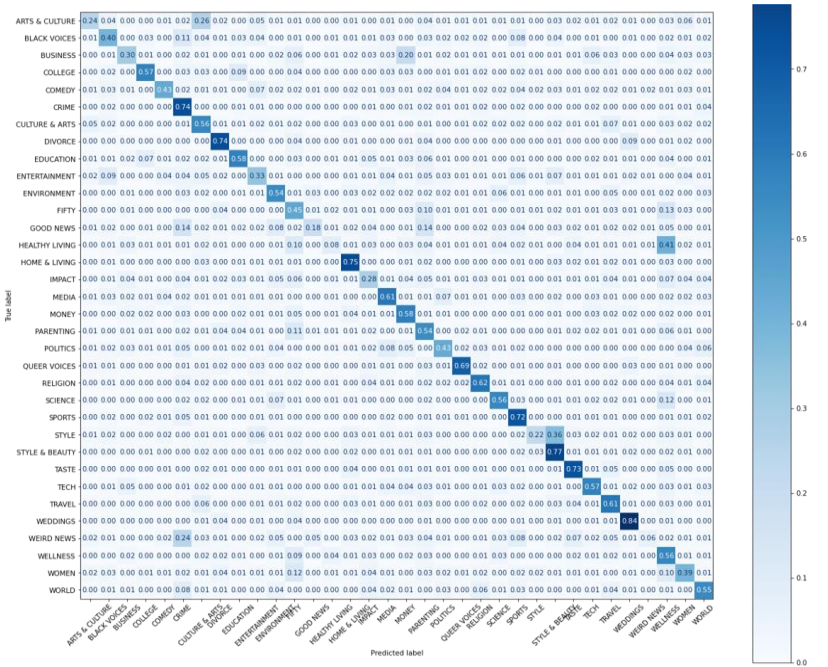
- Embedding Dimension=115
- LSTM Hidden Size = 265
- Linear Layer Size = 207
- Dropout Rate = 0.51
- Anzahl Attention Heads = 0
- Anzahl Parameter = 4.4Mio.

News – Überschriften & Beschreibung

Mit Attention

DIVORCE	0.6681	0.7446	0.7043	419
POLITICS	0.5471	0.4286	0.4806	420
RELIGION	0.6282	0.6182	0.6231	440
MONEY	0.4444	0.5802	0.5033	324
FIFTY	0.2515	0.4469	0.3219	273
GOOD NEWS	0.4870	0.1836	0.2667	305
ARTS & CULTURE	0.4662	0.2447	0.3209	282
COLLEGE	0.5874	0.5708	0.5789	212
EDUCATION	0.4914	0.5758	0.5302	198
accuracy			0.5104	12824
macro avg	0.5163	0.5064	0.4895	12824
weighted avg	0.5208	0.5104	0.4935	12824

Standardabweichung: 1.943%
Steigerung durch Attention: +0.31 p.p.



Gruppe 1

- Epochenanzahl = 15
- Batch Größe = 64
- Lernrate = 0.001
- Vokabulargröße = 30000

Gruppe 2

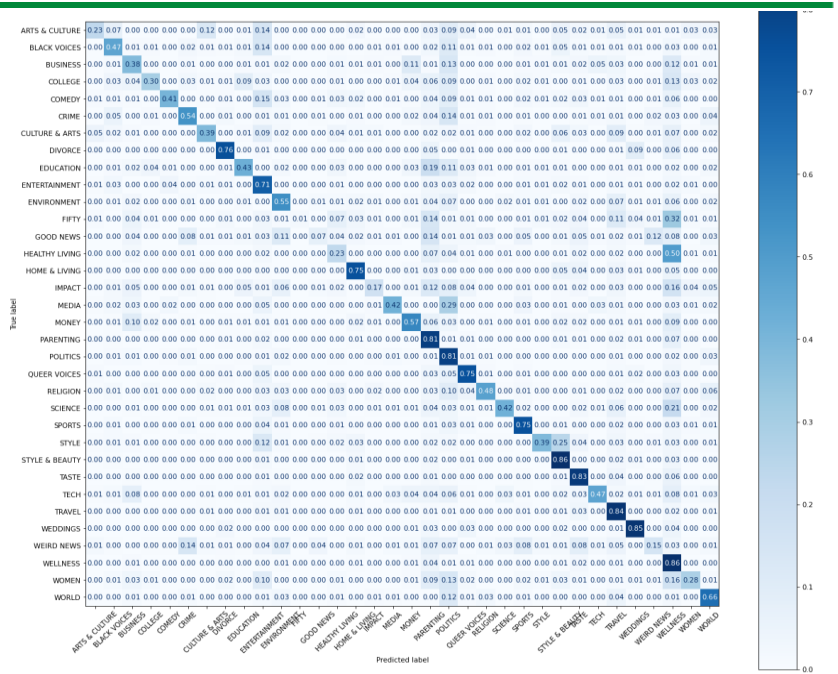
- Embedding Dimension=152
- LSTM Hidden Size = 171
- Linear Layer Size = 250
- Dropout Rate = 0.47
- Anzahl Attention Heads = 2
- Anzahl Parameter = 5.6Mio.

News Big – Überschriften & Beschreibung

Ohne Attention

RELIGION	0.5556	0.4795	0.5147	219
CULTURE & ARTS	0.5301	0.3860	0.4467	228
STYLE	0.6554	0.3865	0.4862	251
SCIENCE	0.6308	0.4184	0.5031	196
TECH	0.5287	0.4694	0.4973	196
MONEY	0.4183	0.5738	0.4839	183
FIFTY	1.0000	0.0147	0.0290	136
GOOD NEWS	0.2381	0.0719	0.1105	139
ARTS & CULTURE	0.4198	0.2313	0.2982	147
COLLEGE	0.4730	0.2991	0.3665	117
EDUCATION	0.3884	0.4273	0.4069	110
accuracy			0.6602	19972
macro avg	0.6016	0.5170	0.5283	19972
weighted avg	0.6542	0.6602	0.6410	19972

Standardabweichung: 0.507%



Gruppe 1

- Epochenanzahl = 13
- Batch Größe = 64
- Lernrate = 0.001
- Vokabulargröße = 30000

Gruppe 2

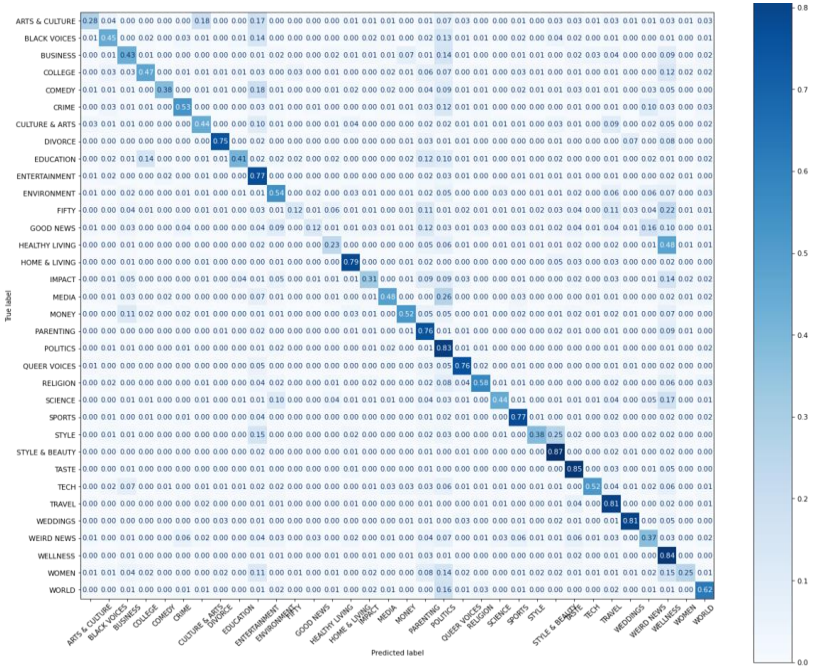
- Embedding Dimension=117
- LSTM Hidden Size = 230
- Linear Layer Size = 188
- Dropout Rate = 0.46
- Anzahl Attention Heads = 0
- Anzahl Parameter = 4.2Mio.

News Big – Überschriften & Beschreibung

Mit Attention

RELIGION	0.5100	0.5799	0.5427	219
CULTURE & ARTS	0.4410	0.4430	0.4420	228
STYLE	0.6419	0.3785	0.4762	251
SCIENCE	0.5959	0.4439	0.5088	196
TECH	0.5862	0.5204	0.5514	196
MONEY	0.4729	0.5246	0.4974	183
FIFTY	0.3137	0.1176	0.1711	136
GOOD NEWS	0.3404	0.1151	0.1720	139
ARTS & CULTURE	0.4316	0.2789	0.3388	147
COLLEGE	0.4198	0.4701	0.4435	117
EDUCATION	0.4688	0.4091	0.4369	110
accuracy			0.6725	19972
macro avg	0.5993	0.5439	0.5575	19972
weighted avg	0.6667	0.6725	0.6581	19972

Standardabweichung: 0.721%
Steigerung durch Attention: +1.23 p.p.



Gruppe 1

- Epochenanzahl = 23
- Batch Größe = 64
- Lernrate = 0.001
- Vokabulargröße = 30000

Gruppe 2

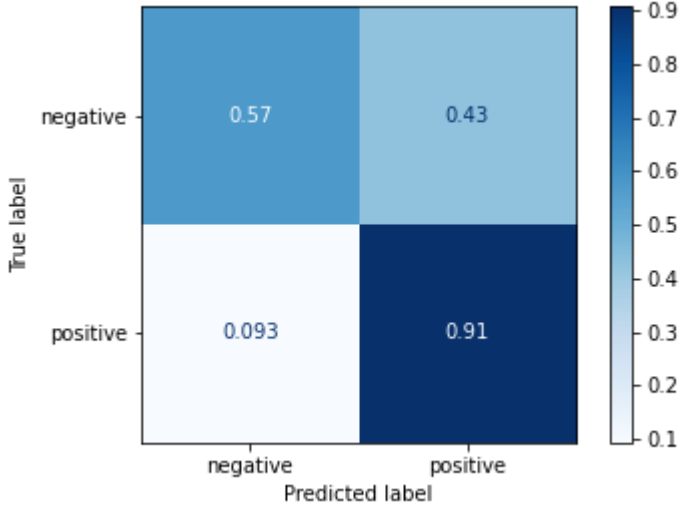
- Embedding Dimension=104
- LSTM Hidden Size = 344
- Linear Layer Size = 222
- Dropout Rate = 0.48
- Anzahl Attention Heads = 1
- Anzahl Parameter = 6.4Mio.

IMDB – Sentiment Analyse

Ohne Attention, absichtlich suboptimale Hyperparameter

	precision	recall	f1-score	support
negative	0.8602	0.5727	0.6876	4985
positive	0.6812	0.9075	0.7782	5015
accuracy			0.7406	10000
macro avg	0.7707	0.7401	0.7329	10000
weighted avg	0.7704	0.7406	0.7331	10000

Standardabweichung: 4.851%
Verlust durch suboptimale Parameter: -11.33 p.p.



Gruppe 1

- Epochenanzahl = 15
- Batch Größe = 64
- Lernrate = 0.0001
- Vokabulargröße = 30000

Gruppe 2

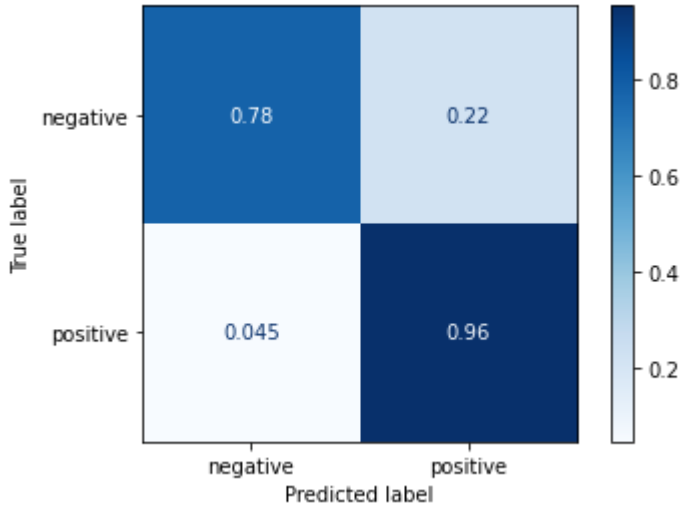
- Embedding Dimension=100
- LSTM Hidden Size = 256
- Linear Layer Size = 128
- Dropout Rate = 0.5
- Anzahl Attention Heads = 0
- Anzahl Parameter = 3.8Mio.

IMDB – Sentiment Analyse

Mit Attention, absichtlich suboptimale Hyperparameter

	precision	recall	f1-score	support
negative	0.9447	0.7785	0.8536	4985
positive	0.8126	0.9547	0.8780	5015
accuracy			0.8669	10000
macro avg	0.8787	0.8666	0.8658	10000
weighted avg	0.8785	0.8669	0.8658	10000

Standardabweichung: 3.08%
Steigerung durch Attention: +12.63 p.p.
Verlust durch suboptimale Parameter: -1.2 p.p.



Gruppe 1

- Epochenanzahl = 15
- Batch Größe = 64
- Lernrate = 0.0001
- Vokabulargröße = 30000

Gruppe 2

- Embedding Dimension=100
- LSTM Hidden Size = 256
- Linear Layer Size = 128
- Dropout Rate = 0.5
- Anzahl Attention Heads = 2
- Anzahl Parameter = 4.8Mio.

Agenda

1. Motivation und Historie Attention
2. Soft Attention
3. Scaled Dot-Product Attention & Multi-Head Attention
4. Implementierung & Evaluation
5. Ergebnisse mit und ohne Attention
6. **Fazit**

Fazit

Theorie

Scaled Dot-Product Attention …

- … berechnet ein Ähnlichkeitsmaß zwischen Queries und Keys und filtert damit die Values
- … skaliert durch Dot-Product deutlich besser als RNNs
- … ist unempfindlich gegenüber langen Eingabesequenzen
- … verbessert durch Skalierung über Dimension des Modells die numerische Stabilität

Praxis

Scaled Dot-Product Attention …

- … kann auch außerhalb von Transformer-Modellen eingesetzt werden
- … verbessert Performance in allen Experimenten im Vergleich zu Modellen ohne Attention
- … kann insbesondere dann vorteilhaft sein, wenn Vorhersagekapazität des Modells noch nicht erreicht ist

Literatur & Referenzen

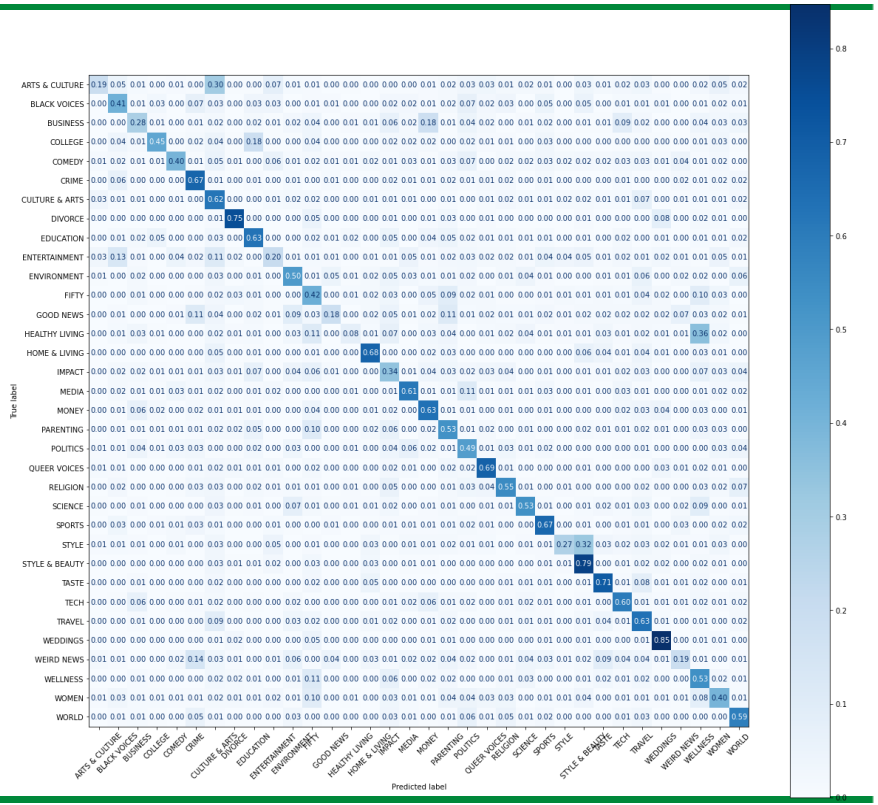
[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. CoRR, abs/1409.0473, 2014.
[2] Jianpeng Cheng, Li Dong, and Mirella Lapata. Long short-term memory-networks for machine reading. arXiv preprint arXiv:1601.06733, 2016.
[3] Elman, J. L. (1990). Finding structure in time. Cognitive Science, 14:179–211.
[4] Hochreiter, S. and Schmidhuber, J., 1997. Long short-term memory. Neural computation, 9(8), pp.1735-1780.
[5] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. CoRR, abs/1406.1078, 2014.
[6] Ilya Sutskever, Oriol Vinyals, and Quoc VV Le. Sequence to sequence learning with neural networks. In Advances in Neural Information Processing Systems, pages 3104–3112, 2014.
[7] Minh-Thang Luong, Quoc V. Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. Multi-task sequence to sequence learning. arXiv preprint arXiv:1511.06114, 2015.
[8] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł. and Polosukhin, I., 2017. Attention is all you need. Advances in neural information processing systems, 30.
[9] Wagner, J. 2020. Modern Approaches in Natural Language Processing. Retrieved at: https://slds-lmu.github.io/seminar_nlp_ss20/attention-and-self-attention-for-nlp.html#ref-luong2015effective
[10] Lilian Weng, 2018. Attention Visualization. Retrieved at: https://lilianweng.github.io/posts/2018-06-24-attention/
[11] Zhang, A., Lipton, Z.C., Li, M. and Smola, A.J., 2021. Dive into deep learning. arXiv preprint arXiv:2106.11342.
[12] Honnibal, M. & Montani, I., 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing.
[13] Minaee, S., Azimi, E. and Abdolrashidi, A., 2019. Deep-sentiment: Sentiment analysis using ensemble of cnn and bi-lstm models. arXiv preprint arXiv:1904.04206.
[14] Hendrycks, D. and Gimpel, K., 2016. Gaussian error linear units (gelus). arXiv preprint arXiv:1606.08415.
[15] Li, L., Jamieson, K., Rostamizadeh, A., Gonina, E., Ben-Tzur, J., Hardt, M., Recht, B. and Talwalkar, A., 2020. A system for massively parallel hyperparameter tuning. Proceedings of Machine Learning and Systems, 2, pp.230-246.
[16] Paszke, A. et al., 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In Advances in Neural Information Processing Systems 32. Curran Associates, Inc., pp. 8024–8035. Available at: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.
[17] Philipp Moritz, Robert Nishihara, Stephanie Wang, Alexey Tumanov, Richard Liaw, Eric Liang, Melih Elibol, Zongheng Yang, William Paul, Michael I Jordan, et al. Ray: A distributed framework for emerging {AI} applications. In 13th USENIX Symposium on Operating Systems Design and Implementation (OSDI 18), pages 561–577, 2018.
[18] Pedregosa, F. et al., 2011. Scikit-learn: Machine learning in Python. Journal of machine learning research, 12(Oct), pp.2825–2830.
[19] D.R. Cox. The Regression Analysis of binary sequences. Journal of the Royal Statistical Society: Series B (Methodological), 2(2), 1958. ISSN 00359246.
[20] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. CoRR, abs/1412.6980, 2015.

Anhang

Anhang

News – Überschriften & Beschreibung, ohne Attention

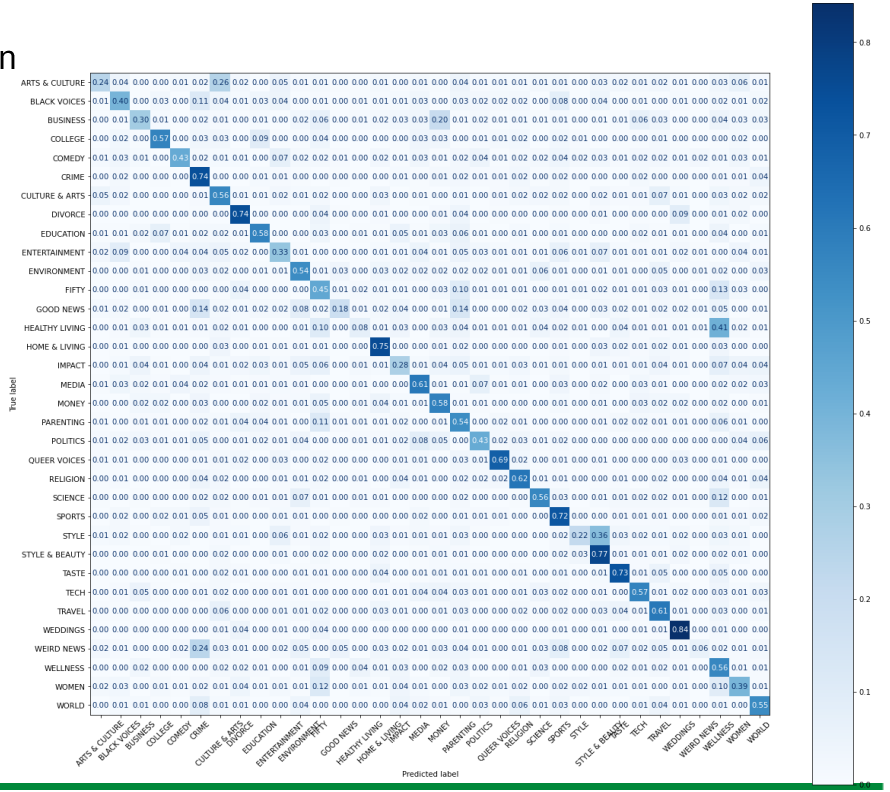
	precision	recall	f1-score	support
ENVIRONMENT	0.4621	0.4986	0.4797	355
BUSINESS	0.4163	0.2816	0.3359	380
HEALTHY LIVING	0.4531	0.0751	0.1289	386
HOME & LIVING	0.6805	0.6788	0.6796	386
ENTERTAINMENT	0.3594	0.2016	0.2583	387
STYLE & BEAUTY	0.4992	0.7903	0.6119	391
CULTURE & ARTS	0.3665	0.6215	0.4611	391
PARENTING	0.4304	0.5294	0.4748	391
BLACK VOICES	0.4485	0.4107	0.4288	392
MEDIA	0.6195	0.6101	0.6148	395
TASTE	0.6372	0.7078	0.6706	397
COMEDY	0.6490	0.3985	0.4938	399
WEDDINGS	0.7019	0.8475	0.7678	400
IMPACT	0.3135	0.3425	0.3274	400
WOMEN	0.4321	0.3975	0.4141	400
WORLD	0.5471	0.5891	0.5673	404
TRAVEL	0.4707	0.6346	0.5405	405
WELLNESS	0.3391	0.5332	0.4145	407
WEIRD NEWS	0.4171	0.1912	0.2622	408
CRIME	0.5433	0.6748	0.6020	409
SPORTS	0.6130	0.6683	0.6394	410
STYLE	0.6474	0.2712	0.3823	413
SCIENCE	0.6377	0.5314	0.5797	414
QUEER VOICES	0.6754	0.6867	0.6810	415
TECH	0.5828	0.6010	0.5917	416
DIVORCE	0.7990	0.7494	0.7734	419
POLITICS	0.4295	0.4857	0.4559	420
RELIGION	0.5931	0.5500	0.5708	440
MONEY	0.4607	0.6327	0.5332	324
FIFTY	0.2406	0.4212	0.3063	273
GOOD NEWS	0.4590	0.1836	0.2623	305
ARTS & CULTURE	0.4454	0.1879	0.2643	282
COLLEGE	0.6115	0.4528	0.5203	212
EDUCATION	0.4079	0.6263	0.4940	198
accuracy			0.5073	12824
macro avg	0.5115	0.5018	0.4879	12824
weighted avg	0.5171	0.5073	0.4942	12824



Anhang

News – Überschriften & Beschreibungen, mit Attention

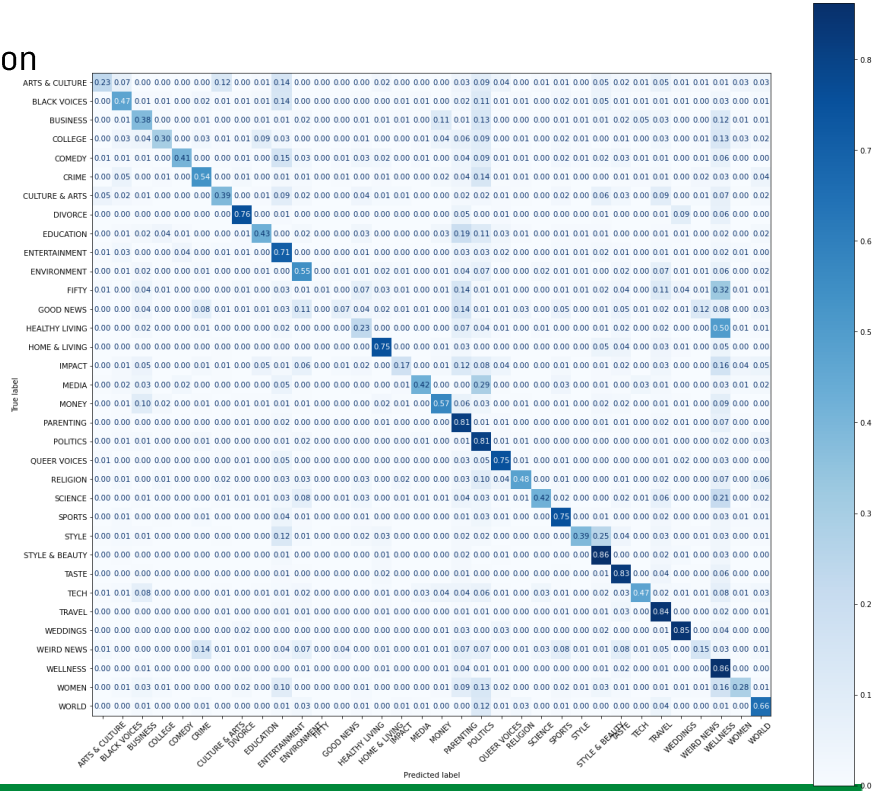
	precision	recall	f1-score	support
ENVIRONMENT	0.4923	0.5437	0.5167	355
BUSINESS	0.4978	0.3026	0.3764	380
HEALTHY LIVING	0.3118	0.0751	0.1211	386
HOME & LIVING	0.6263	0.7513	0.6832	386
ENTERTAINMENT	0.4175	0.3333	0.3707	387
STYLE & BEAUTY	0.4808	0.7698	0.5919	391
CULTURE & ARTS	0.4433	0.5601	0.4949	391
PARENTING	0.3862	0.5422	0.4511	391
BLACK VOICES	0.4845	0.3980	0.4370	392
MEDIA	0.5714	0.6076	0.5890	395
TASTE	0.6509	0.7280	0.6873	397
COMEDY	0.6920	0.4336	0.5331	399
WEDDINGS	0.6700	0.8425	0.7464	400
IMPACT	0.3741	0.2750	0.3170	400
WOMEN	0.4743	0.3925	0.4295	400
WORLD	0.5484	0.5470	0.5477	404
TRAVEL	0.5041	0.6074	0.5510	405
WELLNESS	0.2904	0.5602	0.3826	407
WEIRD NEWS	0.4600	0.0564	0.1004	408
CRIME	0.4246	0.7433	0.5404	409
SPORTS	0.5296	0.7195	0.6101	410
STYLE	0.6357	0.2155	0.3219	413
SCIENCE	0.6356	0.5604	0.5956	414
QUEER VOICES	0.7751	0.6892	0.7296	415
TECH	0.6046	0.5697	0.5866	416
DIVORCE	0.6681	0.7446	0.7043	419
POLITICS	0.5471	0.4286	0.4806	420
RELIGION	0.6282	0.6182	0.6231	440
MONEY	0.4444	0.5802	0.5033	324
FIFTY	0.2515	0.4469	0.3219	273
GOOD NEWS	0.4870	0.1836	0.2667	305
ARTS & CULTURE	0.4662	0.2447	0.3209	282
COLLEGE	0.5874	0.5708	0.5789	212
EDUCATION	0.4914	0.5758	0.5302	198
accuracy			0.5104	12824
macro avg	0.5163	0.5064	0.4895	12824
weighted avg	0.5208	0.5104	0.4935	12824



Anhang

News Big – Überschriften & Beschreibung, ohne Attention

	precision	recall	f1-score	support
POLITICS	0.7668	0.8068	0.7863	3215
WELLNESS	0.5625	0.8575	0.6794	1762
ENTERTAINMENT	0.6992	0.7075	0.7033	1600
PARENTING	0.6441	0.8146	0.7194	1262
TRAVEL	0.6664	0.8437	0.7446	947
STYLE & BEAUTY	0.7686	0.8623	0.8128	1017
WORLD	0.7192	0.6554	0.6858	891
TASTE	0.7230	0.8264	0.7712	818
HEALTHY LIVING	0.5338	0.2287	0.3202	656
QUEER VOICES	0.7228	0.7525	0.7373	610
BUSINESS	0.5231	0.3832	0.4424	561
COMEDY	0.7026	0.4072	0.5156	528
SPORTS	0.6889	0.7524	0.7193	521
BLACK VOICES	0.5711	0.4654	0.5129	492
HOME & LIVING	0.7206	0.7518	0.7358	415
ENVIRONMENT	0.4676	0.5471	0.5042	382
WEDDINGS	0.7367	0.8452	0.7872	394
WOMEN	0.5446	0.2835	0.3729	388
IMPACT	0.5179	0.1696	0.2555	342
DIVORCE	0.7638	0.7572	0.7605	346
CRIME	0.5417	0.5353	0.5385	340
MEDIA	0.6100	0.4192	0.4969	291
WEIRD NEWS	0.4211	0.1471	0.2180	272
RELIGION	0.5556	0.4795	0.5147	219
CULTURE & ARTS	0.5301	0.3860	0.4467	228
STYLE	0.6554	0.3865	0.4862	251
SCIENCE	0.6308	0.4184	0.5031	196
TECH	0.5287	0.4694	0.4973	196
MONEY	0.4183	0.5738	0.4839	183
FIFTY	1.0000	0.0147	0.0290	136
GOOD NEWS	0.2381	0.0719	0.1105	139
ARTS & CULTURE	0.4198	0.2313	0.2982	147
COLLEGE	0.4730	0.2991	0.3665	117
EDUCATION	0.3884	0.4273	0.4069	110
accuracy			0.6602	19972
macro avg	0.6016	0.5170	0.5283	19972
weighted avg	0.6542	0.6602	0.6410	19972



Anhang

News Big – Überschriften & Beschreibung, ohne Attention

	precision	recall	f1-score	support
POLITICS	0.7584	0.8252	0.7904	3215
WELLNESS	0.5840	0.8428	0.6899	1762
ENTERTAINMENT	0.6814	0.7725	0.7241	1600
PARENTING	0.6956	0.7567	0.7249	1262
TRAVEL	0.6925	0.8110	0.7471	947
STYLE & BEAUTY	0.7891	0.8722	0.8286	1017
WORLD	0.7670	0.6207	0.6861	891
TASTE	0.7318	0.8472	0.7853	818
HEALTHY LIVING	0.5808	0.2302	0.3297	656
QUEER VOICES	0.7500	0.7574	0.7537	610
BUSINESS	0.5638	0.4332	0.4899	561
COMEDY	0.7398	0.3769	0.4994	528
SPORTS	0.7408	0.7735	0.7568	521
BLACK VOICES	0.6105	0.4492	0.5176	492
HOME & LIVING	0.7338	0.7904	0.7610	415
ENVIRONMENT	0.4976	0.5445	0.5200	382
WEDDINGS	0.7935	0.8096	0.8015	394
WOMEN	0.6194	0.2474	0.3536	388
IMPACT	0.4693	0.3129	0.3754	342
DIVORCE	0.8156	0.7543	0.7838	346
CRIME	0.6228	0.5294	0.5723	340
MEDIA	0.5573	0.4845	0.5184	291
WEIRD NEWS	0.3607	0.3713	0.3659	272
RELIGION	0.5100	0.5799	0.5427	219
CULTURE & ARTS	0.4410	0.4430	0.4420	228
STYLE	0.6419	0.3785	0.4762	251
SCIENCE	0.5959	0.4439	0.5088	196
TECH	0.5862	0.5204	0.5514	196
MONEY	0.4729	0.5246	0.4974	183
FIFTY	0.3137	0.1176	0.1711	136
GOOD NEWS	0.3404	0.1151	0.1720	139
ARTS & CULTURE	0.4316	0.2789	0.3388	147
COLLEGE	0.4198	0.4701	0.4435	117
EDUCATION	0.4688	0.4091	0.4369	110
accuracy			0.6725	19972
macro avg	0.5993	0.5439	0.5575	19972
weighted avg	0.6667	0.6725	0.6581	19972