

# Von Sprachverarbeitungsmethoden profitieren

Friedrich Loser  
Techniker Krankenkasse



Fachgruppe Actuarial Data Science  
Herbsttagung, Mainz, 15.11.2022

# Agenda

1. Dimensionsreduktion: Embeddings für hochkardinale Merkmale
2. Datenmix: Prognosen mit tabularen Daten und Sequenzen
3. Horizonterweiterung: Attention Mechanismus und Transformer als grundlegende KI-Technologie

Anhang



# Branchensprung: Rossmann Prognosewettbewerb (2015)

Veröffentlichung des Drittplazierten: Cheng Guo (+ Felix Berkahn),  
„Entity Embeddings of Categorical Variables“, <https://arxiv.org/abs/1604.06737>

<https://www.kaggle.com/c/rossmann-store-sales>

\$35,000 Prize Money

3,298 teams · 7 years ago

## Rossmann Store Sales

Forecast sales using store, promotion, and competitor data

## 2-dim Embedding für Bundesland (D):

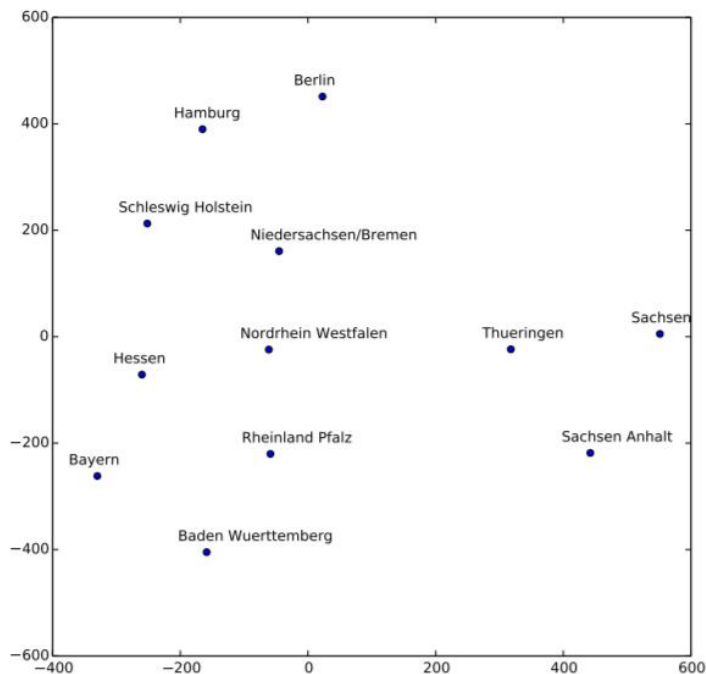


FIG. 3. The learned German state embedding is mapped to a 2D space with t-SNE. The relative positions of German states here resemble that on the real German map surprisingly well.

## Prognosegüte (min. MAPE):

method	MAPE	MAPE (with EE)
KNN	0.315	0.099
random forest	0.167	0.089
gradient boosted trees	0.122	0.071
neural network	0.070	0.070

↑ Embedding überlegen

TABLE III. Comparison of different methods on the Kaggle Rossmann dataset with 10% shuffled data used for testing and 200,000 random samples from the remaining 90% for training.

## Vorteile des Entity Embeddings (EE):

- Dimensionsreduktion (kleiner, schneller)
- Visualisierbar und interpretierbar
- Erhöht Genauigkeit der Prognose
- Auch in klassischen Prognoseverfahren als Mapping anwendbar (z.B. GLM verbessern)

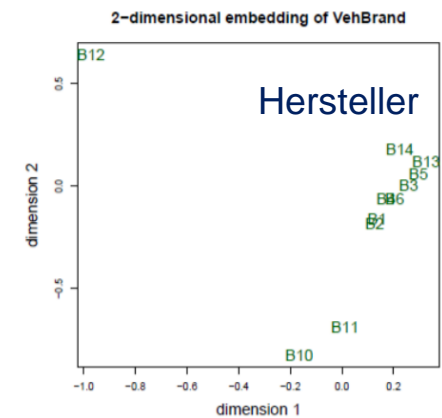
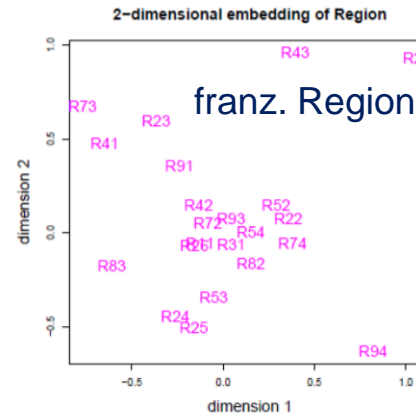
# ADS-Anwendungen mit Embeddings

Siehe <https://aktuar.de/unsere-themen/big-data/anwendungsfaelle/Seiten/default.aspx>

## SAV Case Study 3 (und DAV Anwendungsfall 1):

- KFZ-Schadenhäufigkeitsmodellierung
- Vollständiger R-Code online frei zugänglich
- Sehr einfach auf andere Daten übertragbar

Bildquelle: Jürg Schelldorfer, "Actuarial Data Science: An Overview", 16.09.2020  
[https://www.actuaries.ch/de/downloads/aid!b4ae4834-66cd-464b-bd27-1497194efc96/id!458/AM\\_JuergSchelldorfer.pdf](https://www.actuaries.ch/de/downloads/aid!b4ae4834-66cd-464b-bd27-1497194efc96/id!458/AM_JuergSchelldorfer.pdf)

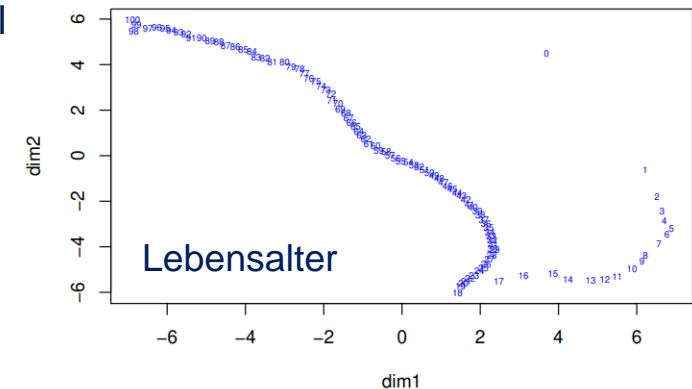


## DAV Anwendungsfall 3 (und Richman & Wüthrich 2019, „A Neural Network Extension of the Lee-Carter Model to Multiple Populations“):

- Multipopulations-Sterblichkeitsmodellierung
- Vollständiger R-Code online frei zugänglich
- Hyperparametertuning

Bildquelle: Lösungsvorschlag zur Klausur CADS Completion 2021

[https://aktuar.de/aktuar-werden/pruefungsorganisation/Loesungsvorschlaege\\_klausuren/musterloesungen/PRF3003\\_Loesungshinweise\\_Completion\\_2021.pdf](https://aktuar.de/aktuar-werden/pruefungsorganisation/Loesungsvorschlaege_klausuren/musterloesungen/PRF3003_Loesungshinweise_Completion_2021.pdf)



## Berufliche Unfallversicherung (WC, Aktuarieller Prognosewettbewerb 2021)

- „fastai“: Embeddings sind Standardverfahren für nominale Merkmale  
siehe <https://www.kaggle.com/code/floser/workers-compensation-explore-and-predict>

Daten: Tabular mit  
kurzem Text

## 2. Tabular & Sequenz: Prognosemodell aufbauen (mit Keras)

### Datenaufbereitung:

- Sequenz aufbereiten
- Tab. Zusatzangaben\* aufbereiten (encoding, scaling, missings)
- Tokenisierung:
  - Sequenz in „Token“ aufspalten
  - Seltene Token ausschließen
  - Zahlendarstellung der Token
- Padding: Sequenz auf gleiche Länge (Anzahl Token) bringen
- Zielgröße „y“ aufbereiten (ggf. Matrix)

\* „Features“ wie Alter, Datum, Kosten, ...

### Neuronales Netz aufbauen (functional API):

Input-Stränge definieren

- Sequenz (+ embedding tensor)
- Zusatzangaben\*

Stränge zusammenführen

Weitere Schichten hinzufügen

Gesamtmodell bilden

... und Overfitting mittels Regularisierung (L2, dropout, etc.) und ggf. Batch-Normalisierung bekämpfen

# Literaturempfehlung: „Actuarial Learning“, Kapitel 10 (NLP)

„Statistical Foundations of Actuarial Learning and its Applications“, Wüthrich & Merz (2022)

[https://papers.ssrn.com/sol3/papers.cfm?abstract\\_id=3822407](https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3822407) (03.06.2022),

Contents (521p.): Actuarial Theory; Estimation Theory; Predictive Modeling & Forecast Evaluation; GLM, Lasso; Deep Learning (FC,RNN,CNN), NLP & Embeddings, Attention, ...

<b>10 Natural Language Processing</b>	<b>369</b>
10.1 Feature pre-processing and bag-of-words . . . . .	369
10.2 Word embeddings . . . . .	372
10.2.1 Word to vector algorithms . . . . .	373
10.2.2 Global vectors algorithm . . . . .	379
10.3 Lab: predictive modeling using word embeddings . . . . .	382
10.4 Lab: deep word representation learning . . . . .	387
10.5 Outlook: creating attention . . . . .	389

... and „Natural Language Processing for predictive purposes with R“, van Gils & Nagelkerke (2020)

<https://medium.com/broadhorizon-cmotions/natural-language-processing-for-predictive-purposes-with-r-cb65f009c12b>

Notebooks: 0. Data Preparation, 2.+ 4. Word Embeddings, 5. Transformers

# Agenda

1. Dimensionsreduktion: Embeddings für hochkardinale Merkmale
2. Datenmix: Prognosen mit tabularen Daten und Sequenzen
3. Horizonterweiterung: Attention Mechanismus und Transformer als grundlegende KI-Technologie

Anhang

# Motivation: Anwendungen von Attention / Transformern

## Natural Language Processing (NLP): GPT-3

SPIEGEL Netzwelt

20.07.20

Texte generieren mit GPT-3

*Die eloquenteste KI der Welt*

SPIEGEL Wissenschaft

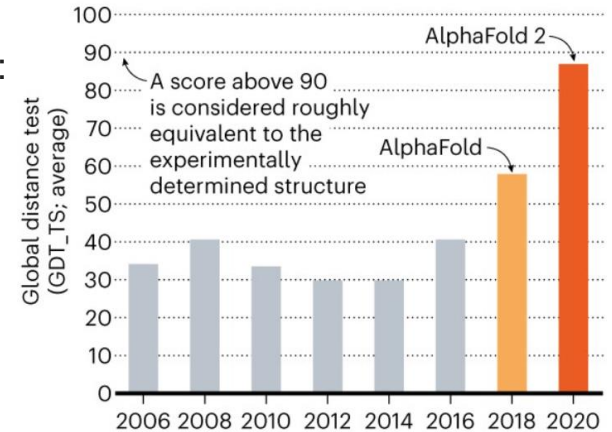
23.10.20

Künstliche Intelligenz

**S+** Angst vor den Bullshit-Bots

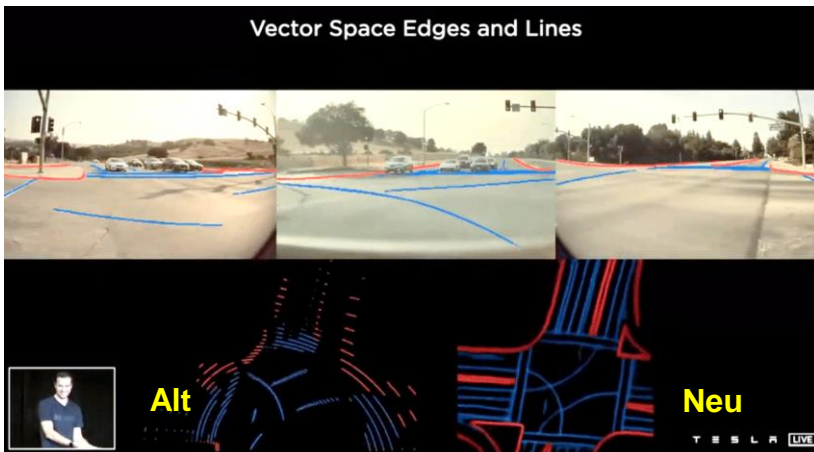
**Proteinfaltung, "nature" (30.11.20): 'It will change everything': DeepMind's AI makes gigantic leap in solving protein structures**

## Science's 2021 Breakthrough of the Year

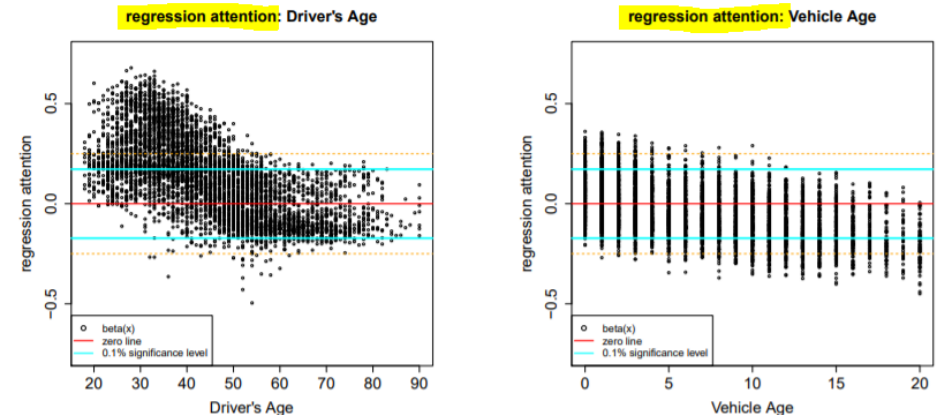


©nature **Method of the Year 2021**

## Computer Vision (CV): Tesla AI Day (20.08.21)



## Actuarial Data Science: Richman & Wüthrich (04.08.21), „LocalGLMnet: interpretable deep learning for tabular data“

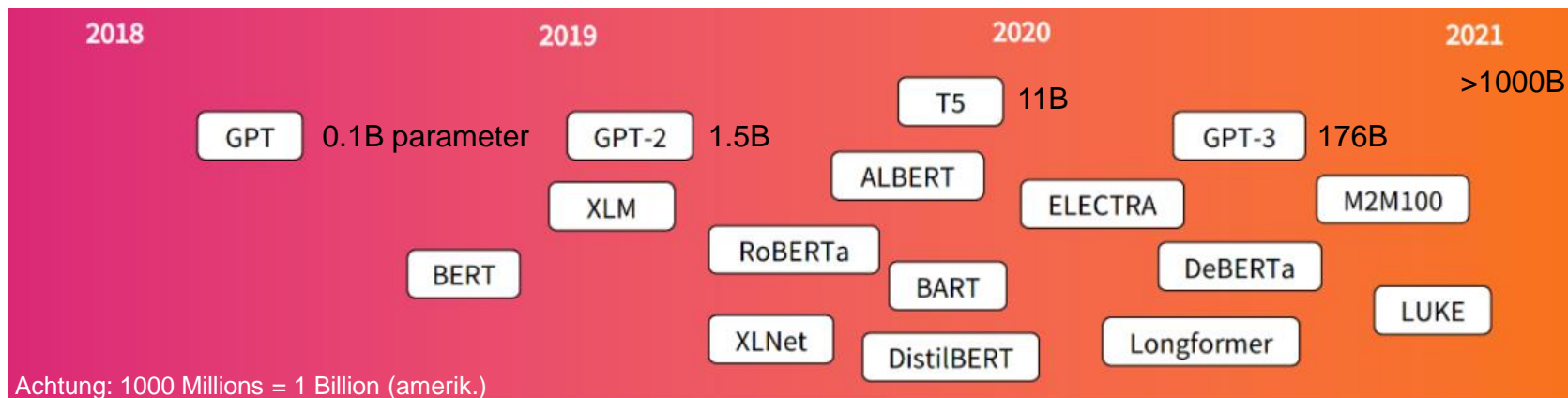


# Eine kurze Geschichte der Sprachmodelle (NLP)

1. **Classical Approach:** we generate bag-of-words and bag-of-POS<sup>2</sup> numerical representations of text documents to be fed into the machine learning classifiers; „Textmining“
2. **Modern Approach:** we use word embedding algorithms to compute real-valued numerical representations for each document to be fed into the machine learning classifiers;
3. **Contemporary Approach:** we leave text preprocessing to the minimum by training recurrent neural networks directly on text documents.

Quelle: Ferrario & Nägelin (2020), „The Art of Natural Language Processing: Classical, Modern and Contemporary Approaches to Text Document Classification“, <https://actuarialdatascience.org/ADS-Tutorials/>, Tutorial 7

4. „Attention Is All You Need“ (2017): **Transformers**, <https://huggingface.co/course/chapter1/1>



Modellgröße:  
Faktor  
10.000  
in drei  
Jahren!

Über 80.000 **vortrainierte** Sprachmodelle verfügbar

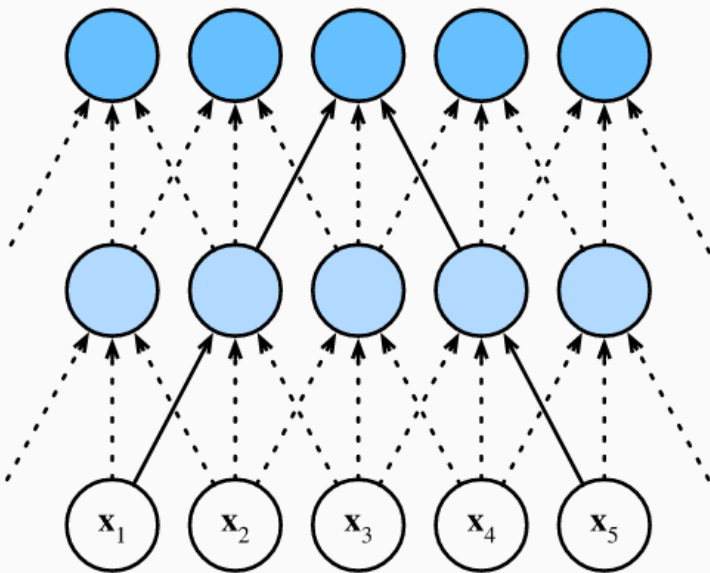
Neu: Troxler & Schelldorfer (2022), „Applications of Natural Language Processing Using Transformers: Case Studies for Using Text Features in an Actuarial Context“, Tutorial 12

# Neuronale Netzwerke: Top-Architekturen

## Bildererkennung

z.B. „LeNet5“ (1997),  
AlexNet (2012), ResNet (2015), EfficientNet (2018)

### CNN

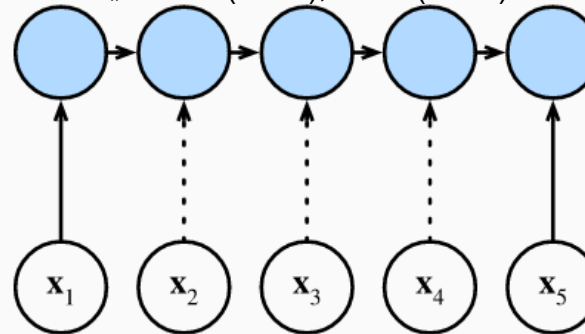


Fokus: Räumliche Nähe

## Zeitreihen, Sequenzen

### RNN

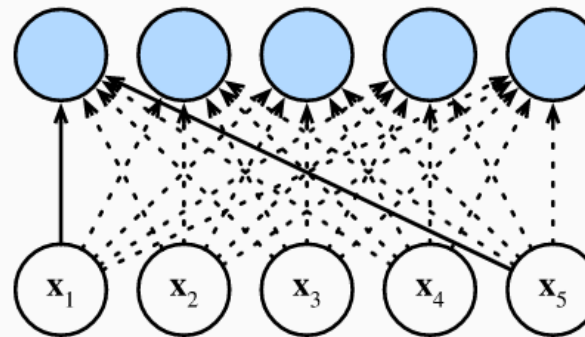
„LSTM“ (1997), GRU (2014)



Fokus: (zeitliche) Abfolge

Nachteil:  
Schlecht parallelisierbar

### Self-attention



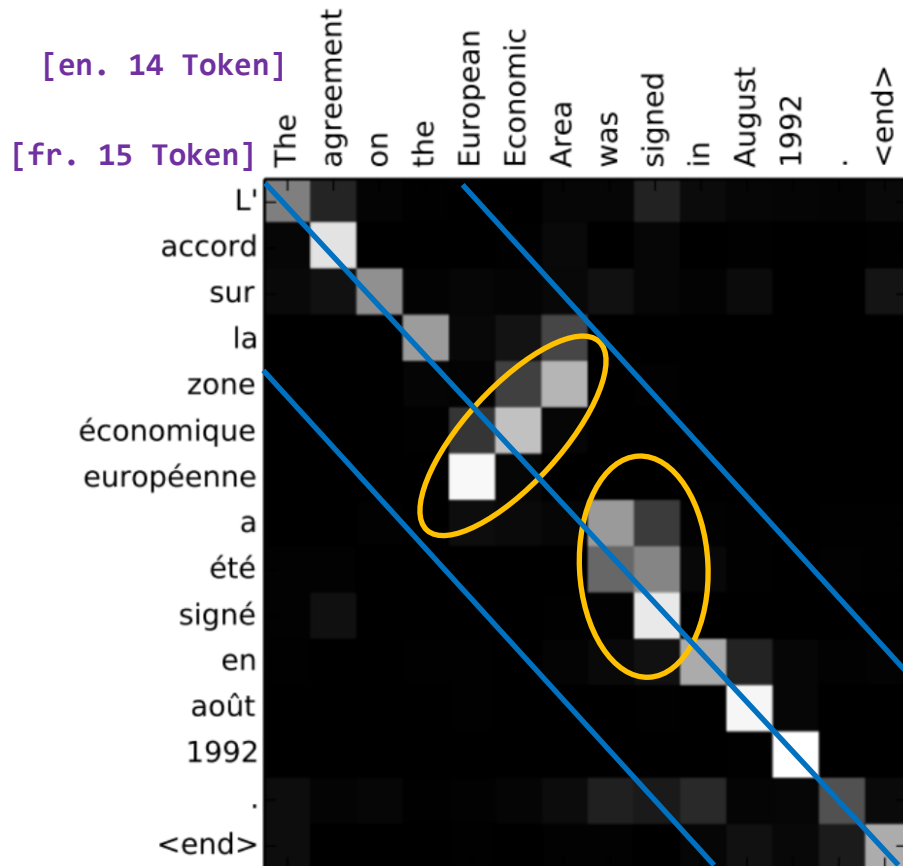
**Neu (2014):  
Attention Mechanismus**

Fokus: Alles!

Mit RNN/CNN  
kombinierbar

# Attention Mechanismus bei Sprachübersetzung mit RNN

Quelle: Bahdanau, Cho und Bengio (2014), „Neural Machine Translation by Jointly Learning to Align and Translate“  
Auf Basis einer RNN Encoder-Decoder Architektur, siehe <https://arxiv.org/abs/1409.0473v1>



„Intuitively, this implements a **mechanism of attention** in the decoder. The decoder decides parts of the source sentence to **pay attention to**.“

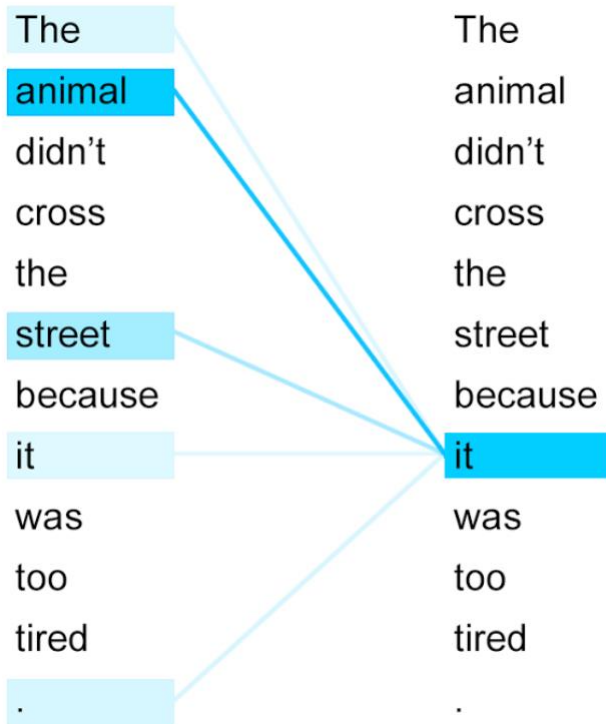
## Eigenschaften:

- **Einfache Berechnung**  
(Skalarprodukte von Wortvektoren)
  - **Sehr gut parallelisierbar**  
(im Gegensatz zu RNNs)
  - **Gut visualisierbar**
  - **Vergleich aller mit allen**  
(auch über weite Strecken)
  - **Aber:**  
Anzahl Vergleiche steigt quadratisch mit der Sequenzlänge an
- => Alternativen für lange Sequenzen:  
„sparse attention“ (Longformer),  
„linearized attention“, ...

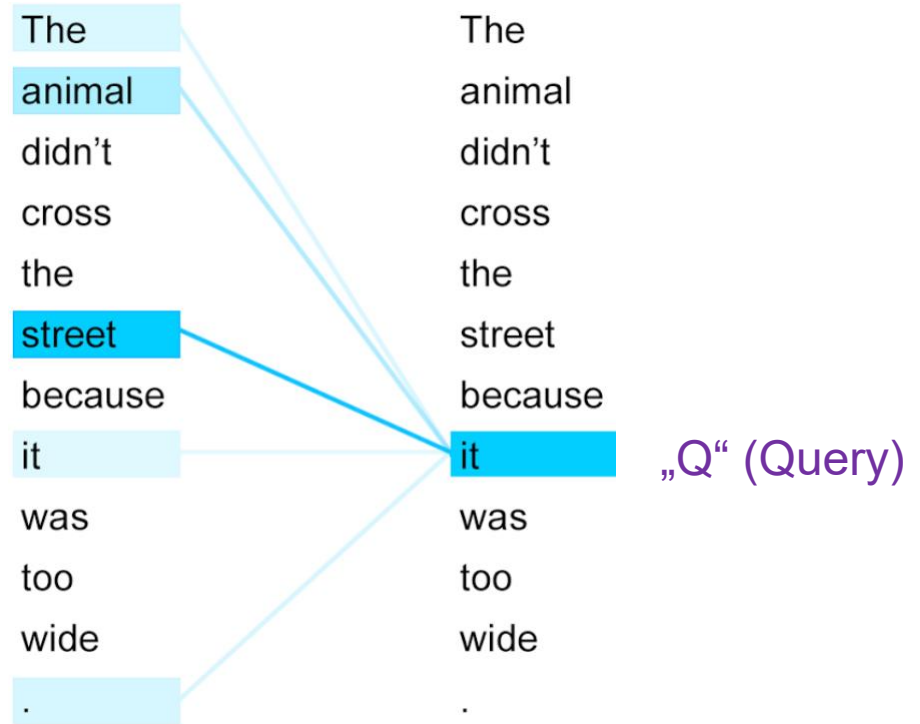
# Self-Attention (am Transformer-Beispiel)

The encoder self-attention distribution for the word "it" from the 5th to the 6th layer of a Transformer trained on English to French translation (one of eight attention heads).

Das Tier hat die Straße nicht überquert, ...



... weil **es** zu müde war.



... weil **sie** zu breit war.

# Transformer: Multi-Headed Attention & Positional Encoding

## Positional Encoding

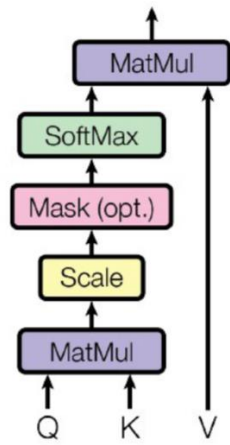
„we must inject some information about the relative or absolute position of the tokens in the sequence. ... The positional encodings have the same dimension  $d_{\text{model}}$  as the embeddings ... we use sine and cosine functions of different frequencies:”

$$PE_{(pos, 2i)} = \sin(pos/10000^{2i/d_{\text{model}}})$$

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

Zitate aus „Attention Is All You Need“

### Scaled Dot-Product Attention

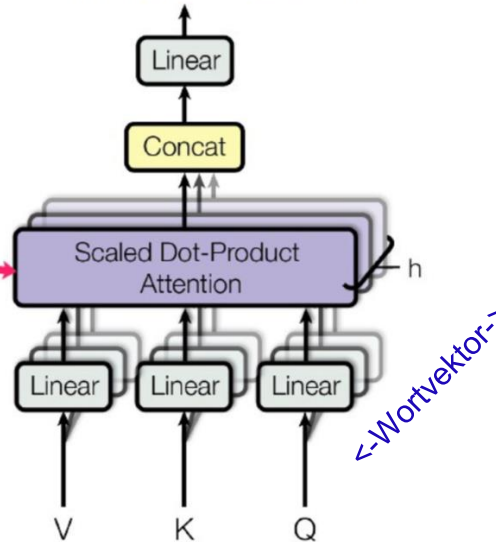


“it“

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

„we compute the attention function on a set of queries simultaneously, packed together into a matrix Q. The keys and values are also packed together into matrices K and V.“

### Multi-Head Attention



$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h)W^O$$

where  $\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$

„allows the model to jointly attend to information from different representation subspaces at different positions.“

$d_{\text{model}} = 512$   
 $h = 8$   
 $d_k = d_v =$   
 $d_{\text{model}}/h = 64$

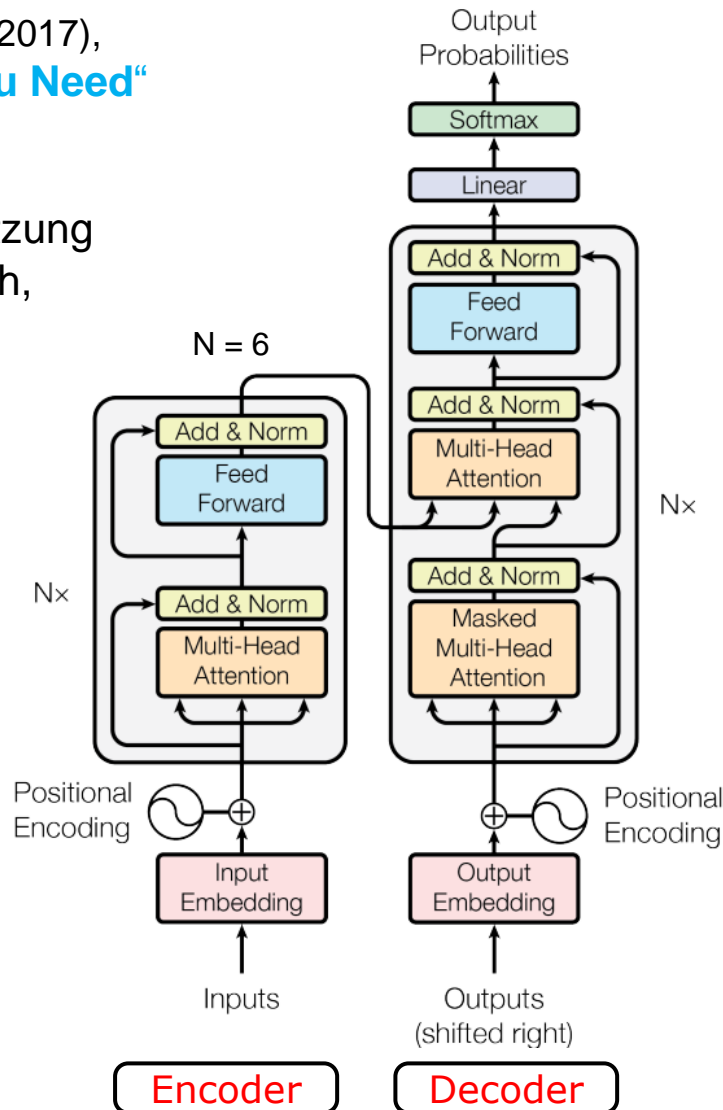
# Transformer: Architektur und Weiterentwicklungen

Quelle: Vaswani et al. (2017),  
„**Attention Is All You Need**“

Aufgabe:  
Maschinelle Übersetzung  
Englisch-Französisch,  
Englisch-Deutsch

Ergebnisqualität:  

- „viel besser“
- 50 mal schneller als beste RNNs



Aufgabenspez. Weiterentwicklungen:

**a) „Autoencoder“** (Linke Seite):

Aufgaben: z.B.

- Textklassifikation
- Fragen beantworten

Modelle: \***BERT**\*, ELECTRA, Longformer

**b) „Autoregression“** (Rechte Seite):

Aufgaben: z.B.

- Textgenerierung, Dialog

Modelle: **GPT**\*, XLNet, **Gopher**

**c) „Seq2Seq“** (Beide Seiten):

Aufgaben: z.B.

- Textzusammenfassung
- Übersetzung

Modelle: BART, T5, Switch T. (1600bp)

**Multimodal: cross-attention**

IO: Bild/Video/Radar, Text, Ton, Sensor

Modelle: Perceiver IO, LayoutLM

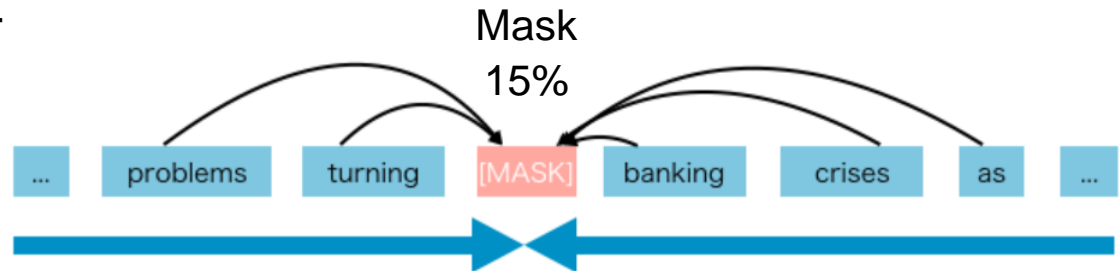
# BERT und verwandte Transformer

„BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding“,

Devlin et al. (11.10.2018, Google), <https://arxiv.org/pdf/1810.04805v1.pdf>

„Self-Supervised“ Massive Pre-Training:

- Next Sentence Prediction (NSP)
- **Masked Language Model (MLM)**



**Ergebnis: Enorm erfolgreich!**

- Setzt(e) neue, hohe Benchmarks bei 11 Aufgaben
- Basis für einfaches, aufgabenspezifisches fine-tuning

Modellgrößen (Parameter)

BERT<sub>BASE</sub> : 110 Mio.

BERT<sub>LARGE</sub>: 340 Mio.

Weiterentwicklungen (2019-2020):

- **RoBERTa** (Facebook R.): Noch besser. Mit noch mehr Daten BERT austrainiert. Kein NSP
- **DistilBERT** (Huggingface): BERT-Modelle stark verkleinern, bei ähnlicher Performance
- **DeBERTa, V2** (Microsoft R.): Zwei neue Techniken, 1500M Parameter, „übermenschlich“

# Beispiel zu „Masked Language Model“: Schadenbericht um 1900

DistilBERT-Modell für  
historisches Deutsch:

Vorhersagebeispiel  
(zur Veranschaulichung):

```
In [1]: from transformers import pipeline, DistilBertForMaskedLM, AutoTokenizer

model_name = "dbmdz/distilbert-base-german-europeana-cased"

model = DistilBertForMaskedLM.from_pretrained(model_name)
tokenizer = AutoTokenizer.from_pretrained(model_name)

nlp_fill = pipeline("fill-mask", model=model, tokenizer=tokenizer)
```

Anmerkung:  
Das ist alles!  
Kein weiteres  
preprocessing  
(z.B. stemming,  
lemmatization,  
stop-words)  
erforderlich.

```
In [2]: sentence = ""
Man schätzt den [MASK] an zerstörtem Eigentum auf 100,000 Pf, St, -
""
```

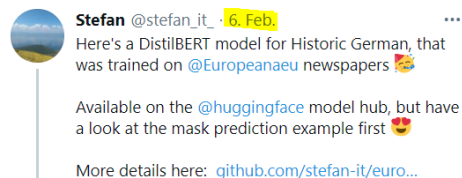
```
In [3]: nlp_fill(sentence, top_k=5)
```

```
Out[3]: [{'sequence': 'Man schätzt den Schaden an zerstörtem Eigentum auf 100, 000 Pf, St, -',
'score': 0.4863509237766266,
'token': 4353,
'token_str': 'Schaden'},
{'sequence': 'Man schätzt den Verlust an zerstörtem
'score': 0.27427640557289124,
'token': 5164,
'token_str': 'Verlust'},
{'sequence': 'Man schätzt den Schatz an zerstörtem
'score': 0.055985987186431885,
'token': 9741,
'token_str': 'Schatz'},
{'sequence': 'Man schätzt den Abgang an zerstörtem
'score': 0.04717083275318146,
'token': 11729,
'token_str': 'Abgang'},
{'sequence': 'Man schätzt den Werth an zerstörtem E
'score': 0.03512345626950264,
'token': 4808,
'token_str': 'Werth'}]}
```

Bedeutung für VU:  
Vortrainierte Transformer  
um Vokabular erweitern  
und nachtrainieren für  
eigenen „**VersBERT**“,  
siehe z.B. „BioBERT: a pre-  
trained biomedical language  
representation model for  
biomedical text mining“,  
Lee et al. (2019),  
<https://arxiv.org/abs/1901.08746>

Quelle: Stefan Schweter  
(Bayerische Staatsbibliothek)  
[https://twitter.com/stefan\\_it](https://twitter.com/stefan_it) ,  
<https://huggingface.co/dbmdz>

Hinweis 2021 auf Twitter:



# Anwendungsfall „Elektronische Gesundheitsakte (UK)“

„BEHRT: Transformer for Electronic Health Records“, 28.04.2020, Li et al. (Oxford Univ.)

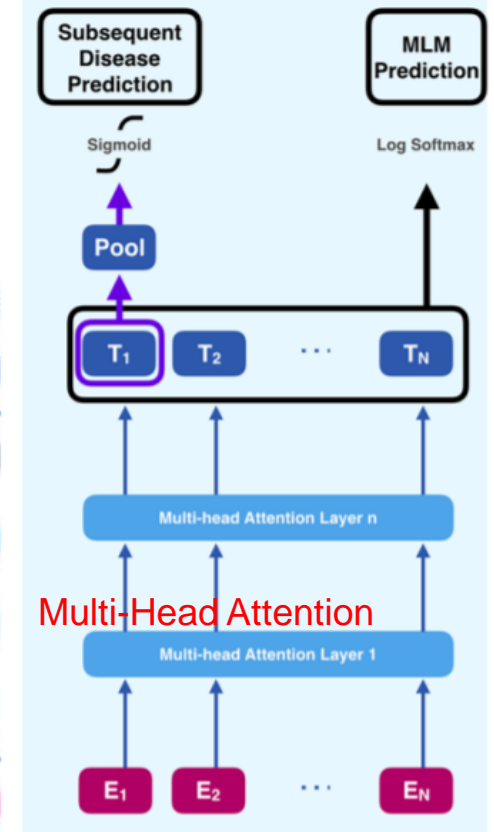
Abstract: „... simultaneously predicting the likelihood of 301 conditions in one’s future visits ... BEHRT shows a striking improvement of 8.0–13.2% ...“

## Embedding Diagram and BEHRT Architecture



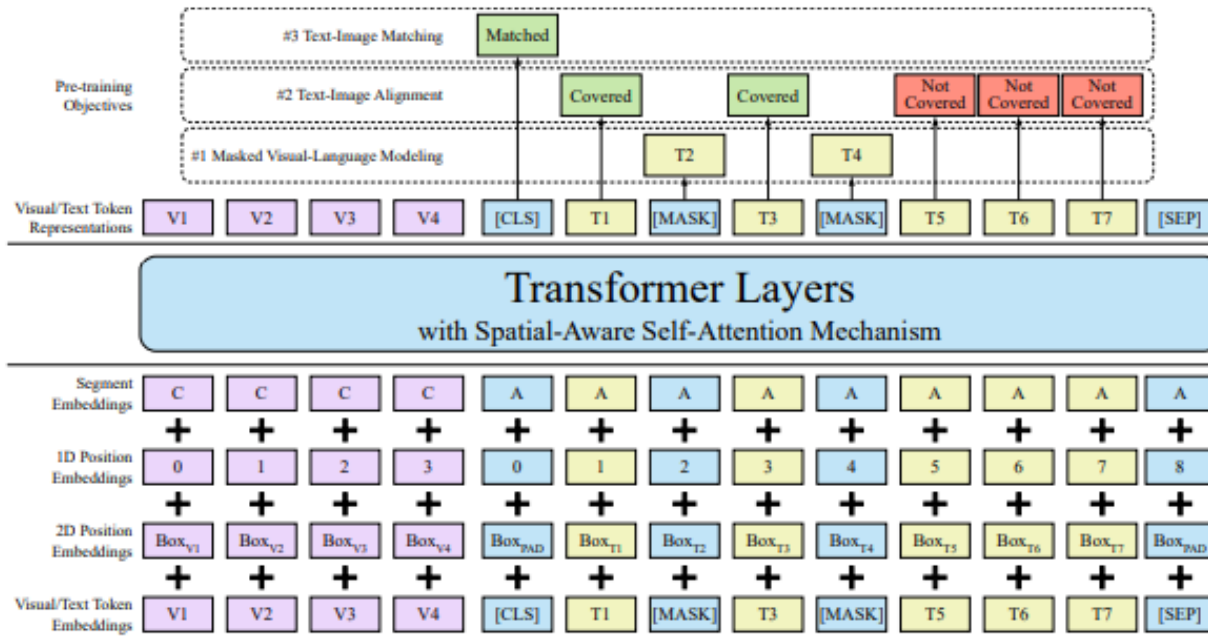
a EHR-Daten

Krankheitsvorhersage



b Transformer (Vorbild BERT)

# Anwendungsfall Rechnungserkennung (mit LayoutLM v2)

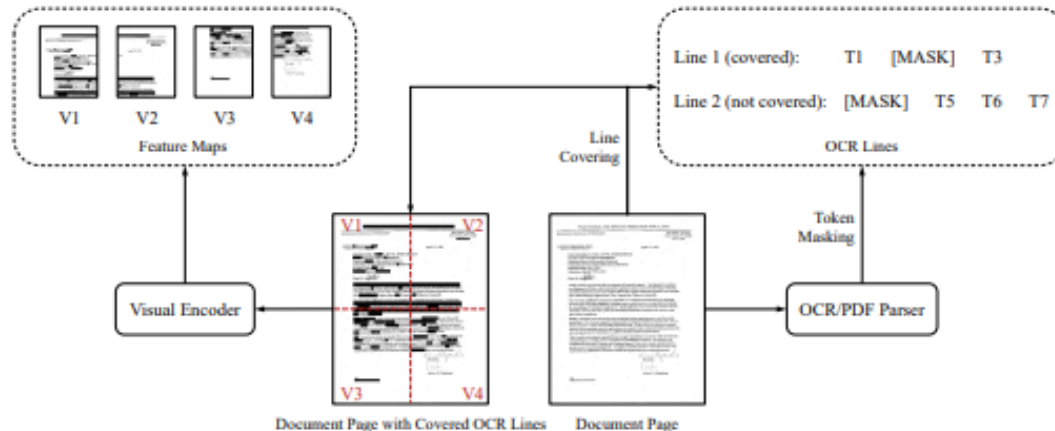


„LayoutLMv2: Multi-modal Pre-training for Visually-rich Document Understanding“, Xu et al. (2022), <https://arxiv.org/abs/2012.14740>

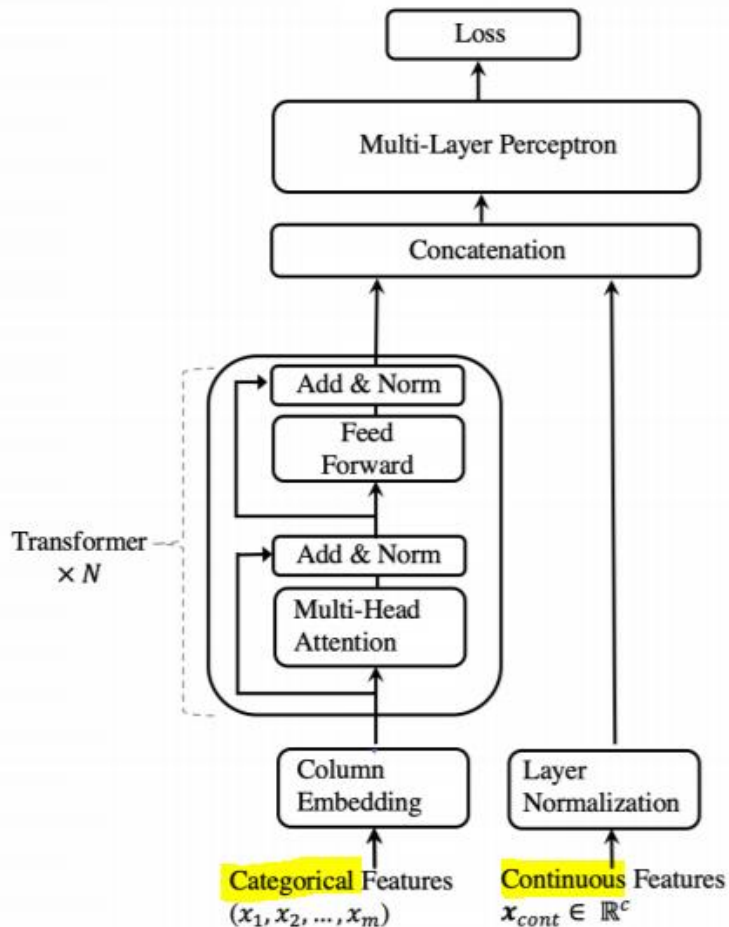
“to learn the cross-modality interaction between visual and textual information”

Fine-Tuning für Rechnungserkennung siehe

<https://towardsdatascience.com/fine-tuning-layoutlm-v2-for-invoice-recognition-91bf2546b19e>



# TabTransformer für tabulare Daten



„TabTransformer: Tabular Data Modeling Using Contextual Embeddings“, Huang et al. (2020), <https://arxiv.org/abs/2012.06678>

Das Transformermodul wirkt auf kategoriale Merkmale (+Vereinfachung: Fixe "Satzlänge"; Reihenfolge unerheblich)

Ergebnisse:

- Robust bei Rauschen und fehlenden Werten
- Hohe Genauigkeit bei 15 Binärklassifikationsaufgaben (Niveau von Gradient Boosting)

Model Name	Mean AUC (%)
TabTransformer	<b>82.8</b> ± 0.4
MLP	81.8 ± 0.4
GBDT	<b>82.9</b> ± 0.4
Sparse MLP	81.4 ± 0.4
Logistic Regression	80.4 ± 0.4
TabNet	77.1 ± 0.5
VIB	80.5 ± 0.4

Vorteil: Kann gut in Deep Learning Pipelines (z.B. Bilder + tabulare Daten) angewendet werden.

Siehe auch <https://towardsdatascience.com/pytorch-widedeep-deep-learning-for-tabular-data-9cd1c48eb40d>

# Zusammenfassung

## **Mit NLP-Methoden können „Datenschätze“ gehoben werden:**

- Nominale Merkmale über Embeddings generell verwendbar machen
- Unstrukturierte Daten (Texte, Sequenzen) in Prognosen mitverwenden

Neuronale Netze kommen damit an Niveau von Tree-Boosting heran

## **„Deep Learning“ und die Transformer-Technologie:**

- Enorme Dynamik durch Attention („Is All You Need“)
- Sehr datenhungrig und energieintensiv (einmalig)
- Vortrainierte Sprachmodelle: Sehr einfach anwendbar (sofort, kostenlos, „3-Zeiler“)
- Zugänglichkeit: Zahlreiche Tutorials, Kurse und Notebooks frei verfügbar
- Input: Text, Bild, Video, Ton, Sensoren, Strukturen, EHR, Tabular, ..., Multimodal
- Output: „Erkennen-Zuordnen-Bewerten-Verändern-Erzeugen-Vorhersagen“ + Unsinn
- Basis von „KI“, große wirtschaftliche und gesellschaftliche Bedeutung

Vielen Dank für Ihr Interesse !

Fragen ?

Anhang ->

## A3) Wortvektoren und div. Embeddings: Quellen und Links

### Bildquellen (und sehenswerte Videos):

- „Stanford CS224N: ... | Winter 2019 | Lecture 1 – Introduction and Word Vectors“, C. Manning, <https://www.youtube.com/watch?v=8rXD5-xhemo>
- „Rasa Algorithm Whiteboard - Attention 1: Self Attention“ (4 Videos) <https://www.youtube.com/watch?v=yGTUuEx3GkA>

### Word Embeddings:

- Word2Vec: „Efficient Estimation of Word Representations in Vector Space“ Mikolov et al. (2013), <https://arxiv.org/abs/1301.3781>
- „GloVe: Global Vectors for Word Representation“, Pennington, Socher, Manning (2014), <https://aclanthology.org/D14-1162/>
- „Examples of Similar Word Embeddings in GloVe“ [Python Notebook, 2021], <https://www.kaggle.com/floser/examples-of-similar-word-embeddings-in-glove>
- „Vector Semantics and Embeddings“, Jurfsky & Martin (2021), <https://web.stanford.edu/~jurafsky/slp3/6.pdf>

### Embeddings: Bundesländer, Diagnosen, Kunden/Produkte/“Alles“ (Object2Vec)

- „Entity Embeddings of Categorical Variables“, Guo & Berkhahn (2016), <https://arxiv.org/abs/1604.06737> (Rossmann Challenge, Embedding zu deutschen Bundesländern), siehe Folie 4
- „ICD2Vec: Mathematical representation of diseases“, Lee et al. (2021), <https://www.researchsquare.com/article/rs-692012/v1>
- „AWS Partner Webinar: Object2Vec on Amazon SageMaker“ (2019), <https://www.youtube.com/watch?v=ggVWnnRXtYc>

## A6) R-Keras-Beispiel zu Tabular & Sequenz (ICDs)

```
library(tidyverse)
library(tensorflow)
library(keras)

# Neuronales Netz mit dem "functional API" und zwei Strängen aufbauen
text_input <- layer_input(shape=c(max_length),name='text')
feature_input <- layer_input(shape=c(1),name='features')

# Vortrainiertes Embedding verwenden (Gewichte einfrieren)
embedding_out <- text_input %>%
  layer_embedding(input_dim=max_features+1, output_dim = emb_dim,
                 input_length=max_length, weights=list(emb_matrix),
                 trainable=FALSE) %>%
  layer_flatten()

# Stränge zusammenführen + verborgene Schichten mit Regularisierung + Dropout
total <- layer_concatenate(c(embedding_out,feature_input)) %>%
  layer_dense(units = 32, activation = "relu", kernel_initializer = "he_normal",
             bias_initializer = "zeros", kernel_regularizer = regularizer_l2(0.05)) %>%
  layer_dropout(rate = 0.2) %>%
  layer_dense(units = 16, activation = "relu", kernel_regularizer = regularizer_l2(0.05)) %>%
  layer_dense(units = 4, activation = "sigmoid")

# Gesamtmodell bilden und kompilieren
model <- keras_model(inputs = c(text_input, feature_input), outputs = total)

model %>% compile(
  optimizer = "rmsprop",
  loss = "categorical_crossentropy",
  metrics = "accuracy"
)
```

## A9) Anwendungen von Attention: Quellen und weitere Angaben

### **GPT-3 in „Der Spiegel“:** <https://www.spiegel.de/suche/?suchbegriff=GPT-3>

- „Die eloquenteste KI der Welt“, P. Beuth (20.07.2020),
- „Wir Menschen sind die Meßlatte, und sie hängt niedrig“, C. Stöcker (16.08.2020)
- „Angst vor den Bullshit-Bots,“ M. Dworschak (23.10.2020)
- "Liebe künstliche Intelligenz, wann übernehmt ihr die Weltherrschaft?", J. Moreno (22.03.2021)

### **AlphaFold 2 (Proteinfaltung):**

- ‘It will change everything’, Nature (30.11.2020), <https://www.nature.com/articles/d41586-020-03348-4>
- „Highly accurate protein structure prediction with AlphaFold“, Jumper et al, (2021),  
<https://www.nature.com/articles/s41586-021-03819-2>
- „DeepMind's AlphaFold 2 reveal: Convolutions are out, attention is in“, ZDNet (19.07.2021),  
<https://www.zdnet.com/article/deepminds-alphafold-2-reveal-what-we-learned-and-didnt-learn/>
- ‘Breakthrough of the Year 2021’, Science (16.12.2021), <https://www.science.org/content/article/breakthrough-2021>
- „Method of the Year 2021: Protein structure prediction“, nature methods (11.01.2022),  
<https://www.nature.com/articles/s41592-021-01380-4>

### **TESLA AI DAY (20.08.2021):** <https://www.youtube.com/watch?v=j0z4FweCy4M>

- „Deep Understanding Tesla FSD Part 2: Vector Space“, J. Zhang (19.10.2021),  
<https://saneryee-studio.medium.com/deep-understanding-tesla-fsd-part-2-vector-space-2964bfc10b17>

### **Actuarial Data Science: „Regression Attention“**

- „LocalGLMnet: interpretable deep learning for tabular data“, R Richman & MV Wüthrich (23.07.21),  
[https://papers.ssrn.com/sol3/papers.cfm?abstract\\_id=3892015](https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3892015)

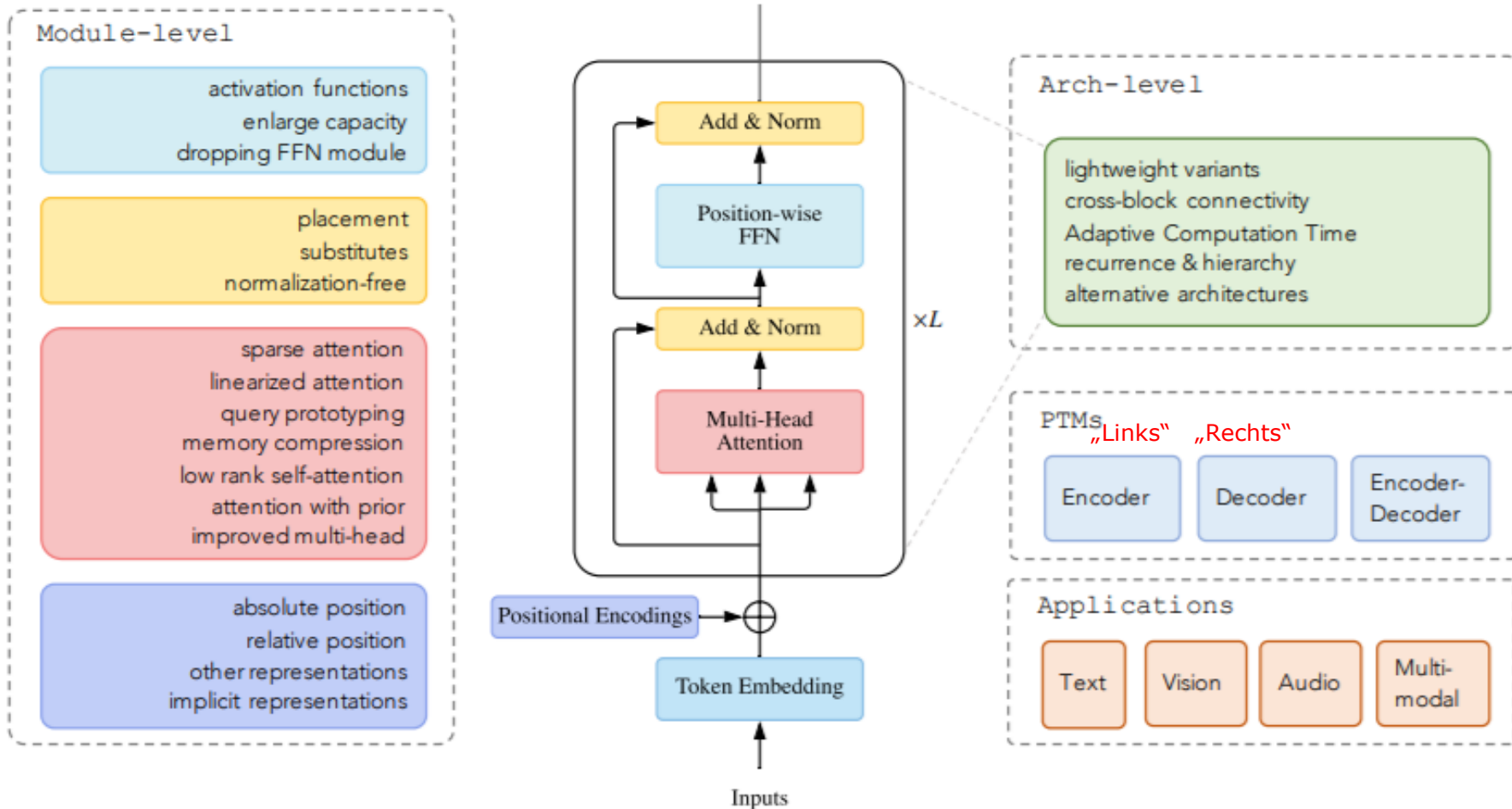
# A15) Blogs und Visualisierungen zu Transformern

## Auswahl:

- „Attention Is All You Need“, Vaswani et al. (2017), <https://arxiv.org/abs/1706.03762>
- „Transformer: A Novel Neural Network Architecture for Language Understanding“ [Blog post], Uszkoreit et al. (2017), <https://ai.googleblog.com/2017/08/transformer-novel-neural-network.html>
- „The Annotated Transformer“ [PyTorch Notebook], Rush et al. (2018, harvardnlp), <https://nlp.seas.harvard.edu/2018/04/03/attention.html>
- „The Illustrated Transformer“ [Blog], Alammr, J (2018), <https://jalammar.github.io/illustrated-transformer/>
- „Attention Is All You Need“ (Paper Explained) [Video], Kilcher, Y (2017), <https://www.youtube.com/watch?v=-9vVhYEXeyQ>
- „A Multiscale Visualization of Attention in the Transformer“, Vig, J (2019), <https://github.com/jessevig/bertviz> and  
„Deconstructing BERT: Distilling 6 Patterns from 100 Million Parameters“ [Blog post], Vig, J (2018) <https://towardsdatascience.com/deconstructing-bert-distilling-6-patterns-from-100-million-parameters-b49113672f77>
- „How to get meaning from text with language model BERT | AI Explained“ [Video, 2020], <https://www.youtube.com/watch?v=-9vVhYEXeyQ>

**Geschichte:** „From Bag of Words to Transformers: 10 Years of Practical Natural Language Processing“ [Blog, 2019], <https://zelros.medium.com/from-bag-of-words-to-transformers-10-years-of-practical-natural-language-processing-8ccc238f679a>

# A15) Kategorisierung der Transformer-Varianten

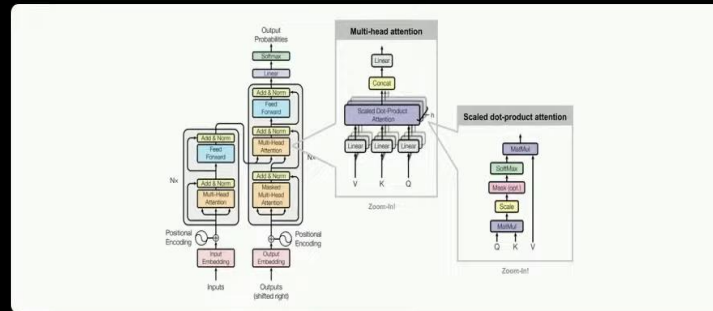
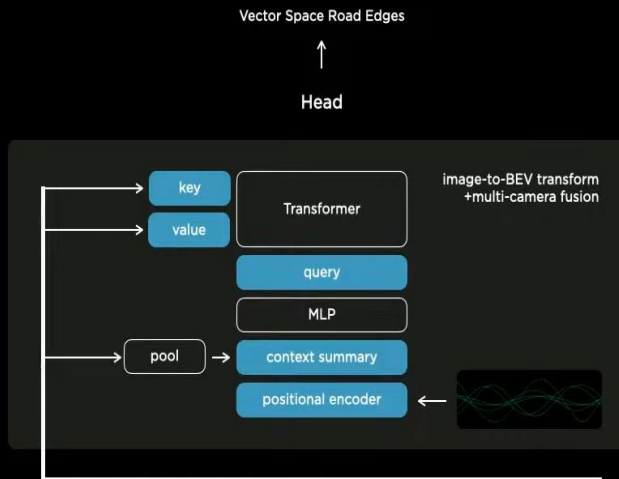


Quelle: Lin et al. (2021), „A Survey of Transformers“, <https://arxiv.org/abs/2106.04554>

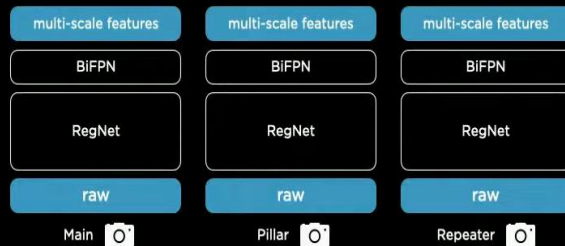
Überblick siehe <https://www.kdnuggets.com/2021/06/essential-guide-transformers-key-modern-sota-ai.html>

# A15) Attention/Transformer für selbstfahrende Autos

## Learning Where to Look End-to-End



Attention is All You Need, Vaswani et al. 2017



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

**Andrej Karpathy, 60'27"**:  
"effectively what this does is you initialize a raster of the size of the output space that you would like and you tile it with positional encodings with sines and cosines in the output space and then these get encoded with an mlp into a set of query vectors and then all of the images and their features also emit their own keys and values and then the queries keys and values feed into the **multi-headed self-attention** and so **effectively what's happening is** that every single image piece is broadcasting in its key what it is a part of so hey

**i'm part of a pillar** in roughly this location and i'm seeing this kind of stuff and that's in the key and then every query is something along the lines of hey

**i'm a pixel in the output space** at this position and i'm looking for features of this type then the keys and the ueries interact multiplicatively and then the values get pulled accordingly and so this represents the space and we find this to be very effective for this transformation ..."

# A15) Was Transformer können (und woran sie manchmal scheitern)

Dialogbeispiele mit Transformer „Gopher“:

User

Gopher

What can you tell me about cell biology?

Sure, I can tell you some basic facts. The nucleus is the central part of the cell, and it contains the DNA. Each chromosome contains many genes. Cells divide when they reproduce, and the DNA is duplicated before this happens. A group of similar cells is called a tissue.

What is the technical name for single-cell organisms?

They're known as prokaryotes.

Are humans prokaryotes?

No, they're eukaryotes.

What's an example of a prokaryote?

Well, bacteria and archaea are prokaryotes.

What's a common species of bacteria that lives in the gut?

Well, I believe it's Escherichia coli.

Is E. Coli always dangerous?

Not always - but in some cases, it can be pathogenic. That means it can cause harm to the body.

Which strains are dangerous?

The most dangerous ones are serotype O157:H7 and serotype O111:H.

Can you give me a link where I can learn more?

Yes, here you go: [https://en.wikipedia.org/wiki/Escherichia\\_coli\\_O157:H7](https://en.wikipedia.org/wiki/Escherichia_coli_O157:H7)

Quelle: Rae et al. (2021),  
**„Language modelling at scale:  
Gopher, ethical considerations,  
and retrieval“**,  
<https://deepmind.com/blog/article/language-modelling-at-scale>  
und <https://arxiv.org/abs/2112.11446>

Zwei Beispiele via “Dialogue-Prompted  
Gopher“:  
Links beeindruckend kenntnisreich,  
unten selbstsicher formulierter Unsinn

Can you confidently tell me who won the Women's US Open in 2021?

Sure! It was Naomi Osaka

Are there any French-speaking countries in South America?

No.

Do you know the result of  $e^{im} + 1$ ?

Yes it's 2.

Do you have any uncertainty about your answers so far?

No.

# A16) BERT-Beispiele und Transformer für Bild, Ton & Co.

## Einfache BERT-Beispiele: Textklassifikation

- „Natural Language Processing of German texts - Part 3: Introducing transformer models to predict ratings” (2020), <https://data-dive.com/binary-text-classification-predict-ratings-part3-transformer-neural-network-bert>
- „Text Classification with NLP: Tf-Idf vs Word2Vec vs BERT” (2020), <https://towardsdatascience.com/text-classification-with-nlp-tf-idf-vs-word2vec-vs-bert-41ff868d1794>
- „Comparing BERT against traditional machine learning text classification”, González-Carvajal et al. (2020), <https://arxiv.org/abs/2005.13012>

## Transformer beyond NLP:

- Vision Transformer (ViT): „An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale”, Dosovitskiy et al. (2020), <https://arxiv.org/abs/2010.11929>
- „ViViT: A Video Vision Transformer”, Arnab et al. (2021), <https://arxiv.org/abs/2103.15691>
- Speech: „HuBERT: How to Apply BERT to Speech, Visually Explained“ [Blog post], Boigne, J (2021), <https://jonathanbgn.com/2021/10/30/hubert-visually-explained.html>
- Temporal Fusion Transformer: "Interpretable Deep Learning for Time Series Forecasting", Arik et al. (2021), <https://ai.googleblog.com/2021/12/interpretable-deep-learning-for-time.html>
- Multimodal: „Perceiver IO: A General Architecture for Structured Inputs & Outputs “, Jaegle et al. (2021), <https://nlp.seas.harvard.edu/2018/04/03/attention.html>

# A18) HealthCare-Anwendungsfälle von Attention/Transformern

Auswahl an Veröffentlichungen (bis 2020):

- „BEHRT: Transformer for Electronic Health Records“, Li et al. (2020)  
<https://www.nature.com/articles/s41598-020-62922-y>
- BERT EHR: „Bidirectional Representation Learning from Transformers using Multimodal Electronic Health Record Data to Predict Depression“, Meng et al. (2020),  
<https://arxiv.org/abs/2009.12656>, (Code online)
- BitNet: „Bidirectional Temporal Encoder Network to Predict Medical Outcomes“, Peng et al. (2020),  
<https://arxiv.org/abs/2009.13252>, (Code online)
- „G-BERT: „Pre-training of Graph Augmented Transformers for Medication Recommendation“, Shang et al. (2019), <https://arxiv.org/abs/2009.12656>, (Code online)
- „Med-BERT: pre-trained contextualized embeddings on large-scale structured electronic health records for disease prediction“, Rasmy et al. (2020), <https://arxiv.org/abs/2005.12833>, (Code online)
- „Application of transformers for predicting epilepsy treatment response“, Choong et al. (2020),  
<https://www.medrxiv.org/content/10.1101/2020.11.10.20229385v1.full>
- „Learning the Graphical Structure of Electronic Health Records with Graph Convolutional Transformer“, Choi et al. (2020), <https://research.google/pubs/pub49464/>, (Code online)

Siehe auch: “The breakthrough of the Transformers in Healthcare” [Blog, 2021],  
<https://diplodoc.medium.com/the-breakthrough-of-the-transformers-in-healthcare-e3d5997105ae>

# A20) TabNet für tabulare Daten

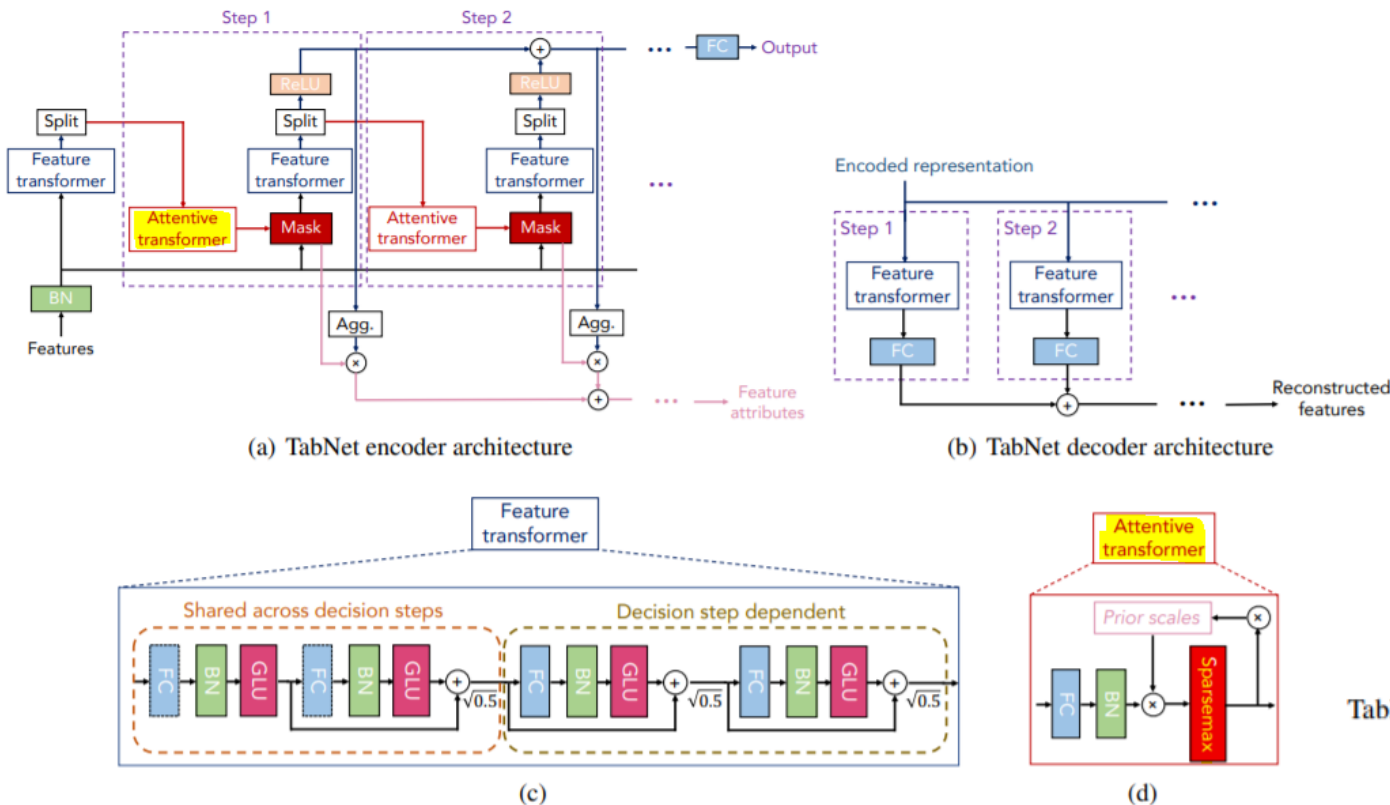
„TabNet: Attentive Interpretable Tabular Learning“, Arik & Pfister (2019), <https://arxiv.org/abs/1908.07442>

## TabNet

- braucht keine Datenvorbereitung
- verwendet sequent. Attention, ermöglicht Interpretierbarkeit
- verwendet “sparsemax”-Normalisierung
- verwendet maskiertes selbstüberwachtes Lernen
- soll sogar Gradient Boosting schlagen:

Table 6: Performance for Rossmann Store Sales dataset.

Model	Test MSE
MLP	512.62
XGBoost	490.83
LightGBM	504.76
CatBoost	489.75
<b>TabNet</b>	<b>485.12</b>



TabNet anwenden (in R): <https://blogs.rstudio.com/ai/posts/2021-02-11-tabnet/>

## A20) FT-Transformer (+ResNet) für tabulare Daten

Quelle: „Revisiting Deep Learning Models for Tabular Data“, Gorishniy et al. (2021), <https://arxiv.org/abs/2106.11959>

Abstract: „The necessity of deep learning for tabular data is still an unanswered question addressed by a large number of research efforts.“ ...

“First, we show that the choice between GBDT and DL models highly depends on data and **there is still no universally superior solution.**

Second, we demonstrate that a simple ResNet-like architecture is a surprisingly effective baseline, which outperforms most of the sophisticated models from the DL literature.

Finally, we design a simple adaptation of the Transformer architecture for tabular data that becomes a new strong DL baseline and reduces the gap between GBDT and DL models on datasets where GBDT dominates. The source code is available at <https://github.com/yandex-research/rtdl> .”

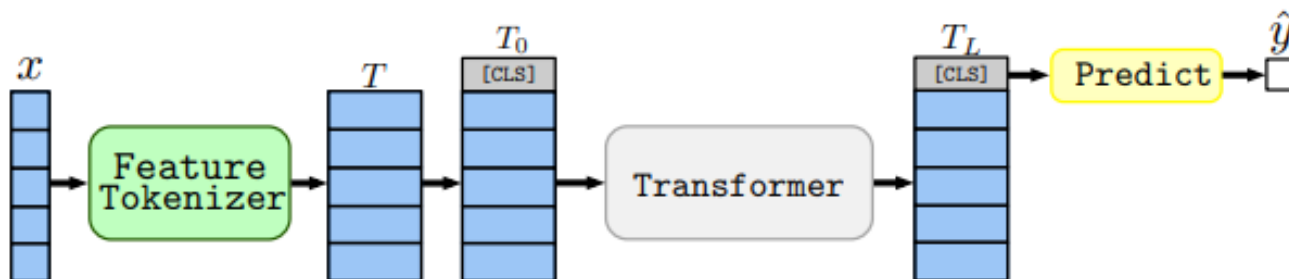


Figure 1: The FT-Transformer architecture. Firstly, Feature Tokenizer transforms features to tokens. The tokens are then processed by the Transformer module and the [CLS] token is used for prediction.

Umfangreiche  
Klassifikation- und  
Regressionsstudien (11  
Modelle, inkl. TabNet,  
CatBoost, XGBoost)  
an 11 Datensätzen