

Natural Language Processing mit Transformern

Sprachanwendungen mit Hugging Face erstellen

DAS INHALTS- VERZEICHNIS

» Hier geht's
direkt
zum Buch

Vorwort	11
Einführung	15
1 Hallo Transformer	23
Das Encoder-Decoder-Framework	24
Der Attention-Mechanismus	26
Einsatz von Transfer Learning im NLP	28
Die Transformers-Bibliothek von Hugging Face: die Lücke schließen	32
Die Anwendungsmöglichkeiten von Transformern im Überblick.....	33
Textklassifizierung	33
Named Entity Recognition	34
Question Answering	35
Automatische Textzusammenfassung (Summarization)	36
Maschinelle Übersetzung (Translation)	37
Textgenerierung	37
Das Ökosystem von Hugging Face	38
Der Hugging Face Hub	39
Die Tokenizers-Bibliothek von Hugging Face	41
Die Datasets-Bibliothek von Hugging Face	41
Die Accelerate-Bibliothek von Hugging Face	42
Die größten Herausforderungen im Zusammenhang mit Transformer-Modellen	42
Zusammenfassung	44
2 Textklassifizierung	45
Der Datensatz	46
Ein erster Blick auf die Datasets-Bibliothek von Hugging Face	47
Dataset-Objekte in DataFrames überführen	50
Ein Blick auf die Verteilung der Kategorien	51
Wie lang sind unsere Tweets?	52

Vom Text zu Tokens	54
Tokenisierung auf der Ebene von Zeichen (Character Tokenization)	54
Tokenisierung auf der Ebene von Wörtern (Word Tokenization)	57
Tokenisierung auf der Ebene von Teilwörtern (Subword Tokenization)	58
Den gesamten Datensatz tokenisieren	60
Trainieren eines Textklassifikators	62
Transformer-Modelle als Feature-Extraktoren	64
Feintuning von Transformer-Modellen	72
Zusammenfassung	81
3 Die Anatomie von Transformer-Modellen	83
Die Transformer-Architektur	83
Der Encoder	86
Self-Attention	87
Die Feed-Forward-Schicht	98
Layer Normalization integrieren	99
Positional-Embeddings	100
Einen Head zur Klassifizierung hinzufügen	103
Der Decoder	104
Transformer-Modelle im Überblick	106
Die drei Entwicklungsstränge von Transformer-Modellen	107
Rein Encoder-basierte Transformer-Modelle	108
Rein Decoder-basierte Transformer-Modelle	110
Encoder-Decoder-basierte Transformer-Modelle	112
Zusammenfassung	114
4 Multilinguale Named Entity Recognition	115
Der Datensatz	116
Multilinguale Transformer-Modelle	120
Ein genauerer Blick auf die Tokenisierung	122
Die Tokenizer-Pipeline	122
Der SentencePiece-Tokenizer	124
Transformer-Modelle für die Named Entity Recognition	125
Der Aufbau der Model-Klasse der Transformers-Bibliothek	127
Bodies und Heads	127
Ein selbst definiertes Modell zur Klassifizierung von Tokens erstellen	128
Ein selbst definiertes Modell laden	130
Tokenisierung von Texten für die Named Entity Recognition	132
Qualitätsmaße	135

Feintuning eines XLM-RoBERTa-Modells	136
Fehleranalyse	138
Sprachenübergreifender Transfer	146
Wann ist ein Zero-Shot-Transfer sinnvoll?	147
Modelle für mehrere Sprachen gleichzeitig feintunen	149
Interaktion mit den Modell-Widgets	152
Zusammenfassung	153
5 Textgenerierung	155
Die Herausforderungen bei der Generierung von kohärenten Texten	157
Greedy-Search-Decodierung	159
Beam-Search-Decodierung	163
Sampling-Verfahren	167
Top-k- und Nucleus-Sampling	169
Welcher Ansatz zur Decodierung ist der beste?	173
Zusammenfassung	173
6 Automatische Textzusammenfassung (Summarization)	175
Der CNN/DailyMail-Datensatz	176
Pipelines für die automatische Textzusammenfassung	177
Ein einfacher Ansatz zur Textzusammenfassung	178
GPT-2	178
T5	178
BART	179
PEGASUS	180
Verschiedene Zusammenfassungen vergleichen	181
Evaluierung der Qualität von generierten Texten	182
BLEU	183
ROUGE	187
Evaluierung des PEGASUS-Modells auf dem CNN/DailyMail-Datensatz	190
Trainieren eines Modells zur Generierung von Zusammenfassungen	192
Das PEGASUS-Modell auf dem SAMSum-Datensatz evaluieren	193
Das PEGASUS-Modell feintunen	194
Zusammenfassungen von Dialogen erstellen	198
Zusammenfassung	199
7 Question Answering	201
Aufbau eines rezensionsbasierten QA-Systems	202
Der Datensatz	203
Antworten aus einem Text extrahieren	209
Die Haystack-Bibliothek zum Aufbau einer QA-Pipeline verwenden	217

Verbesserung unserer QA-Pipeline	226
Den Retriever evaluieren	226
Den Reader evaluieren	233
Domain Adaptation	235
Die gesamte QA-Pipeline evaluieren	240
Jenseits des extraktiven QA	241
Zusammenfassung	243
8 Effizientere Transformer-Modelle für die Produktion	247
Die Intentionserkennung als Fallstudie	248
Eine Benchmark-Klasse zur Beurteilung der Performance erstellen	250
Verkleinerung von Modellen mithilfe der Knowledge Distillation	255
Knowledge Distillation im Rahmen des Feintunings	256
Knowledge Distillation im Rahmen des Pretrainings	258
Eine Trainer-Klasse für die Knowledge Distillation erstellen	259
Ein geeignetes Modell als Ausgangspunkt für das Schüler-Modell wählen	260
Geeignete Hyperparameter mit Optuna finden	265
Unser destilliertes Modell im Vergleich	267
Beschleunigung von Modellen mithilfe der Quantisierung	268
Das quantisierte Modell im Vergleich	275
Optimierung der Inferenz mit ONNX und der ONNX Runtime	276
Erhöhung der Sparsität von Modellen mithilfe von Weight Pruning	282
Sparsität tiefer neuronaler Netze	283
Weight-Pruning-Methoden	283
Zusammenfassung	287
9 Ansätze bei wenigen bis keinen Labels	289
Erstellung eines GitHub-Issues-Tagger	291
Die Daten beschaffen	292
Die Daten vorbereiten	292
Trainingsdatensätze erstellen	297
Unterschiedlich große Trainingsdatensätze erstellen	299
Implementierung eines naiven Bayes-Klassifikators als Baseline	300
Ansätze, wenn keine gelabelten Daten vorliegen	303
Ansätze, wenn nur wenige gelabelte Daten zur Verfügung stehen	313
Datenaugmentierung	313
Embeddings als Nachschlagetabelle verwenden	316
Ein standardmäßiges Transformer-Modell feintunen	327
In-Context- und Few-Shot-Learning auf Basis von Prompts	330
Ungelabelte Daten nutzbar machen	331
Ein Sprachmodell feintunen	332
Einen Klassifikator feintunen	335

Fortgeschrittene Methoden	337
Zusammenfassung	339
10 Transformer-Modelle von Grund auf trainieren	341
Große Datensätze und wie sie beschafft werden können	342
Herausforderungen beim Aufbau eines großen Korpus	343
Einen eigenen Codedatensatz erstellen	346
Mit großen Datensätzen arbeiten	349
Datensätze zum Hugging Face Hub hinzufügen	352
Erstellung eines Tokenizers	354
Das Tokenizer-Modell	355
Die Leistung eines Tokenizers beurteilen	356
Ein Tokenizer für die Programmiersprache Python	357
Einen Tokenizer trainieren	362
Einen selbst erstellten Tokenizer auf dem Hub speichern	366
Ein Modell von Grund auf trainieren	367
Verschiedene Pretraining-Objectives im Überblick	367
Das Modell initialisieren	370
Den Dataloader implementieren	371
Die Trainingsschleife einrichten	374
Der Trainingslauf	382
Ergebnisse und Analyse	383
Zusammenfassung	388
11 Künftige Herausforderungen	389
Skalierung von Transformer-Modellen	389
Skalierungsgesetze	391
Herausforderungen bei der Skalierung	393
Attention Please! – Den Attention-Mechanismus effizienter gestalten	395
Sparse-Attention	396
Linearisierte Attention	398
Jenseits von Textdaten	399
Computer Vision	400
Tabellen	403
Multimodale Transformer	406
Speech-to-Text	406
Computer Vision und Text	409
Wie geht es weiter?	415
Index	417