**Modellvergleich und Optimierung von NLP-Verfahren zur Klassifikation kryptobezogener Reddit-Beiträge**

Digital Business University of Applied Sciences

Data Science & Business Analytics  
DMI01 Data Mining

Daniel Ambach

**Eingereicht von Dennis Reimer**

**Martrikelnummer: 190288**

**Datum: 16.04.2025**

**Zusammenfassung**  
Diese Arbeit untersucht den gezielten Einsatz von NLP-Modellen zur Sentimentanalyse kryptobezogener Reddit-Beiträge. Ziel ist es, verschiedene Klassifikationsansätze – von lexikonbasierten Methoden bis hin zu modernen Transformer-Architekturen – systematisch zu vergleichen und durch Fine-Tuning an die Besonderheiten informeller Social-Media-Texte anzupassen. Der Fokus liegt dabei auf der Unterscheidung zwischen Reddit-Posts und -Kommentaren sowie auf der Performance der Modelle in den Sentimentklassen *bullish*, *neutral* und *bearish*.

Die Analyse basiert auf einem hybriden Datensatz, bestehend aus eigenen gelabelten Reddit-Posts sowie einem öffentlich verfügbaren Kommentar-Datensatz aus einer aktuellen IEEE-Studie. Die methodische Umsetzung folgt dem CRISP-DM-Prozessmodell. Für die Modellierung kommen FinBERT und CryptoBERT als spezialisierte Transformer-Modelle zum Einsatz; VADER, TextBlob und weitere dienen als Vergleichsbaseline.

Die Ergebnisse zeigen: Während FinBERT auf Reddit-Posts nur moderate Verbesserungen erzielt (F1 = 0.53), erreicht CryptoBERT auf Kommentaren nach Fine-Tuning eine sehr hohe Klassifikationsgüte (F1 = 0.92). Die Arbeit verdeutlicht damit die zentrale Rolle domänenspezifischer Modellierung und Hyperparameteroptimierung für erfolgreiche Sentimentanalysen im Krypto-Umfeld.

Inhaltsverzeichnis

[1 Einleitung 2](#_Toc195696063)

[2 Theoretischer Hintergrund 3](#_Toc195696064)

[2.1 Grundlagen der Sentimentanalyse 3](#_Toc195696065)

[2.2 Klassifikationsansätze im Vergleich 4](#_Toc195696066)

[2.3 Data Mining und CRISP-DM-Prozess 5](#_Toc195696067)

[2.4 Bewertung von Klassifikationsmodellen 5](#_Toc195696068)

[3 Implementierung und Methodik 5](#_Toc195696069)

[3.1 Datenaufbereitung und Split 6](#_Toc195696070)

[3.2 Baseline- Vergleich der Modelle 6](#_Toc195696071)

[3.3 Modelloptimierung mit Custom- Trainer 7](#_Toc195696072)

[3.4 Vergleich verschiedener Lernrraten 7](#_Toc195696073)

[4 Ergebnisse und Diskussion 8](#_Toc195696074)

[4.1 Modellvergleich auf untrainierten Baselines 8](#_Toc195696075)

[4.2 Fine-Tuning Ergebnisse 9](#_Toc195696076)

[4.3 Lernratenvergleich 10](#_Toc195696077)

[4.4 Gesamtbewertung 11](#_Toc195696078)

[5 Fazit 11](#_Toc195696079)

[6 Literaturverzeichnis 12](#_Toc195696080)

[7 Abbildungsverzeichnis 13](#_Toc195696081)

1 Einleitung

Die Analyse öffentlicher Stimmungen auf sozialen Medien hat sich in den letzten Jahren zu einem zentralen Forschungsfeld im Bereich des Data Mining und der angewandten Künstlichen Intelligenz entwickelt. Besonders im Kontext von Finanzmärkten und Kryptowährungen versuchen Analyst:innen, aus Diskussionen auf Plattformen wie Reddit Rückschlüsse auf Marktbewegungen und Investorenstimmung zu ziehen. Solche datengetriebenen Methoden ergänzen traditionelle Finanzindikatoren und ermöglichen einen tieferen Einblick in kollektives Verhalten (Tan et al., 2016, S. 312–313).

Die Sentimentanalyse auf Reddit ist jedoch mit zahlreichen Herausforderungen verbunden: Kurztexte, Ironie, informelle Sprache, Meme-Kultur und der Mangel an Kontext erschweren die zuverlässige Klassifikation. Diese Eigenschaften führen dazu, dass einfache lexikonbasierte Methoden häufig an ihre Grenzen stoßen (Tan et al., 2016, S. 316–318). Aus diesem Grund kommen verstärkt transformerbasierte Sprachmodelle wie FinBERT oder CryptoBERT zum Einsatz, die in der Lage sind, Sprachbedeutung im Kontext zu modellieren (Huyen, 2022, S. 89–93).

Ziel dieser Arbeit ist es, verschiedene NLP-Modelle zur Sentimentanalyse auf Reddit-Daten systematisch zu vergleichen und anschließend gezielt zu optimieren. Berücksichtigt werden sowohl transformerbasierte Modelle (FinBERT, CryptoBERT, RoBERTa, DeBERTa) als auch lexikonbasierte Verfahren (VADER, TextBlob). Die Modelle werden auf zwei unterschiedlich strukturierten Texttypen getestet: Reddit-Posts und Reddit-Kommentaren. Aufbauend auf einer quantitativen Evaluation erfolgt ein kontrolliertes Fine-Tuning der leistungsfähigsten Modelle.

Die Arbeit orientiert sich am etablierten Prozessmodell CRISP-DM (Cross Industry Standard Process for Data Mining). Dieses strukturierte Vorgehensmodell gliedert Data-Mining-Projekte in sechs Phasen: Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation und Deployment (Bramer, 2007, S. 15–17). Der CRISP-DM-Ansatz stellt sicher, dass technische Umsetzungen stets im Kontext des Gesamtziels und der zugrunde liegenden Daten erfolgen. In dieser Arbeit wird jede Phase explizit benannt, begründet und dokumentiert.

Für die Analyse wurden zwei separate Datensätze herangezogen:

Reddit-Posts: Eigene Sammlung von ca. 750 Reddit-Beiträgen zum Thema Kryptowährungen, die mittels Web-Scraping extrahiert und manuell in die Klassen *bullish*, *neutral* und *bearish* gelabelt wurden.

Reddit-Kommentare: Ein öffentlich verfügbarer Datensatz aus der Studie „Sentiment Analysis of Crypto-Related Reddit Posts Using Transformer Models“ (IEEE, 2023), in dem die Autor:innen der Kommentare ihre Beiträge selbst klassifizieren. Die Datenbasis wird laufend aktualisiert und stellt eine saubere, domänenspezifische Grundlage für die Bewertung transformerbasierter Modelle dar.

Für die technische Umsetzung kamen Python sowie Bibliotheken wie transformers, datasets und scikit-learn zum Einsatz. Das Modelltraining und das Fine-Tuning wurden unter Verwendung GPU-beschleunigter Instanzen durchgeführt (Downey & Lang, 2024, Kap. 2).

Die Arbeit gliedert sich in fünf Kapitel. Kapitel 2 behandelt die theoretischen Grundlagen der Sentimentanalyse, stellt die eingesetzten Modelltypen vor und erläutert die verwendeten Bewertungsmethoden. In Kapitel 3 wird das methodische Vorgehen entlang des CRISP-DM-Prozesses beschrieben. Kapitel 4 dokumentiert die empirischen Ergebnisse, stellt diese grafisch dar und diskutiert zentrale Beobachtungen. Kapitel 5 schließt die Arbeit mit einer Zusammenfassung der Erkenntnisse und einem Ausblick auf mögliche Weiterentwicklungen ab.

2 Theoretischer Hintergrund

2.1 Grundlagen der Sentimentanalyse

Die Sentimentanalyse (auch Meinungsanalyse) ist ein Teilgebiet des Natural Language Processing (NLP) und beschäftigt sich mit der automatisierten Erkennung und Klassifikation subjektiver Meinungsäußerungen in Texten. Ziel ist es, Aussagen als *positiv*, *negativ* oder *neutral* einzuordnen. Anwendung findet Sentimentanalyse u. a. im E-Commerce, in der politischen Stimmungsanalyse oder im Finanzbereich zur Bewertung von Investorenstimmungen (Tan et al., 2016, S. 312–314).

Die Herausforderung bei der Sentimentanalyse besteht darin, dass Sprache mehrdeutig, ironisch, kontextabhängig oder implizit sein kann. Besonders in sozialen Medien wie Reddit sind Texte oft sehr kurz, verwenden Umgangssprache, Memes oder Sarkasmus. Diese Eigenschaften erschweren es, klare semantische Marker für Sentiment zu identifizieren (Tan et al., 2016, S. 316–318). Klassische regelbasierte Systeme stoßen hier schnell an ihre Grenzen, was den Einsatz lernender Systeme erforderlich macht.

2.2 Klassifikationsansätze im Vergleich

Für die Sentimentklassifikation kommen im Wesentlichen zwei methodisch unterschiedliche Ansätze zum Einsatz: lexikonbasierte Verfahren und transformerbasierte Klassifikationsmodelle auf Basis neuronaler Netze.

Lexikonbasierte Verfahren wie VADER (Valence Aware Dictionary and sEntiment Reasoner) oder TextBlob beruhen auf vordefinierten Wortlisten, in denen Begriffen bestimmte Polarisierungswerte zugewiesen sind. Der Gesamtsentimentwert eines Textes ergibt sich aus der Aggregation dieser Einzelwerte. VADER wurde speziell für Social-Media-Inhalte entwickelt und berücksichtigt neben Wortwerten auch Großschreibung, Emoticons und Satzzeichen. TextBlob hingegen arbeitet auf Basis eines einfachen Naive-Bayes-Modells mit linguistischer Vorverarbeitung.

Der Vorteil dieser Ansätze liegt in ihrer schnellen Implementierung und hohen Interpretierbarkeit. Ihr Nachteil ist jedoch das fehlende Kontextverständnis, wodurch insbesondere Ironie, Slang oder komplexe Satzstrukturen nicht adäquat erkannt werden (Tan et al., 2016, S. 320).

Transformer stellen eine moderne Klasse neuronaler Netze dar, die speziell für den Umgang mit sequentiellen Textdaten entwickelt wurden. Im Gegensatz zu klassischen rekurrenten Architekturen (z. B. LSTMs) arbeiten Transformer vollständig auf Basis sogenannter Self-Attention-Mechanismen, die es ermöglichen, die Bedeutung eines Wortes im Kontext des gesamten Satzes zu modellieren (Huyen, 2022, S. 89–93).

Die bekannteste Architektur ist BERT (Bidirectional Encoder Representations from Transformers), aus der zahlreiche Domänenmodelle abgeleitet wurden. Für diese Arbeit besonders relevant sind:

FinBERT: Ein auf Finanzkommunikation spezialisiertes BERT-Modell, trainiert auf SEC-Filings, Unternehmensberichten und Marktanalysen.

CryptoBERT (ElKulako): Eine Variante, die gezielt auf Diskussionen im Kryptokontext trainiert wurde.

RoBERTa und DeBERTa: Weiterentwicklungen der BERT-Architektur mit verbesserter Trainingsmethodik und interner Struktur, jedoch ohne spezifische Ausrichtung auf Finanztexte.

Transformerbasierte Modelle benötigen umfangreiche Rechenressourcen und größere Datenmengen, bieten dafür aber ein deutlich stärkeres semantisches und kontextuelles Verständnis von Sprache – ein zentraler Vorteil bei der Analyse von Reddit-Daten.

2.3 Data Mining und CRISP-DM-Prozess

Die in dieser Arbeit durchgeführten Schritte lassen sich vollständig in das CRISP-DM-Modell (Cross Industry Standard Process for Data Mining) einordnen – ein weit verbreitetes Framework zur Strukturierung datengetriebener Projekte (Bramer, 2007, S. 15–17). Es besteht aus sechs klar definierten Phasen:

1. Business Understanding: Problemdefinition, Zielsetzung
2. Data Understanding: Beschreibung, Exploration und Bewertung der Datenqualität
3. Data Preparation: Bereinigung, Vorverarbeitung, Labeling und Formatierung
4. Modeling: Auswahl geeigneter Modelle und Konfigurationsparameter
5. Evaluation: Modellvergleich, Interpretation der Metriken
6. Deployment: Anwendung oder Transfer der Ergebnisse (in dieser Arbeit nur theoretisch)

Der Vorteil von CRISP-DM liegt in der Trennung von Analyse und Umsetzung, wodurch sowohl technische Details als auch fachliche Zielsetzungen klar nachvollziehbar bleiben (Tan et al., 2016, S. 13; Bramer, 2007, S. 17). In dieser Arbeit bildet CRISP-DM die methodische Klammer, an der sich alle Kapitel orientieren.

2.4 Bewertung von Klassifikationsmodellen

Die Güte von Klassifikationsmodellen im Bereich der Sentimentanalyse wird typischerweise anhand mehrerer Metriken bewertet:

Accuracy: Anteil korrekter Vorhersagen insgesamt

F1-Score: harmonisches Mittel von Precision und Recall

Cohen’s Kappa: Maß für die Übereinstimmung zwischen Vorhersage und Ground Truth unter Berücksichtigung des Zufalls

Insbesondere bei unausgeglichenen Klassen (z. B. viele *neutral*, wenige *bearish*) ist der F1-Score (macro) die bevorzugte Metrik, da er alle Klassen gleich gewichtet. Genauigkeit alleine ist hier wenig aussagekräftig (Huyen, 2022, S. 209–210).

3 Implementierung und Methodik

Im Folgenden wird das methodische Vorgehen der Arbeit detailliert beschrieben. Die Struktur orientiert sich vollständig am CRISP-DM-Prozessmodell (Bramer, 2007, S. 15–17), das eine systematische Umsetzung datengetriebener Projekte ermöglicht.

3.1 Datenaufbereitung und Split

Nach dem Import der Datensätze wurden diese bereinigt, vereinheitlicht und in numerische Labels für die drei Sentimentklassen *bullish* (0), *neutral* (1) und *bearish* (2) überführt. Die Daten wurden anschließend in Trainings-, Validierungs- und Testdaten unterteilt. Dabei wurde explizit darauf geachtet, dass die Klassenverteilung auch in den Teilmengen erhalten bleibt (stratifizierter Split).

Die konkrete Aufteilung und das Verhältnis (z. B. 60 % Training, 20 % Validierung, 20 % Test) variierten je nach Datensatz leicht und wurden direkt im Notebook durchgeführt. Auf einen externen Split-Helfer wurde bewusst verzichtet, um die Kontrolle über die Aufteilung direkt im Hauptnotebook zu behalten.

3.2 Baseline- Vergleich der Modelle

Zur Evaluation wurden zunächst sechs verschiedene Modelle auf einem festen Testset verglichen. Die Modellklasse reichte von lexikonbasierten Verfahren (VADER, TextBlob) bis zu transformerbasierten Klassifikatoren (FinBERT, CryptoBERT, RoBERTa, DeBERTa). Die lexikonbasierten Modelle erhielten eine explizite Schwellenwertlogik zur Klassenzuweisung:

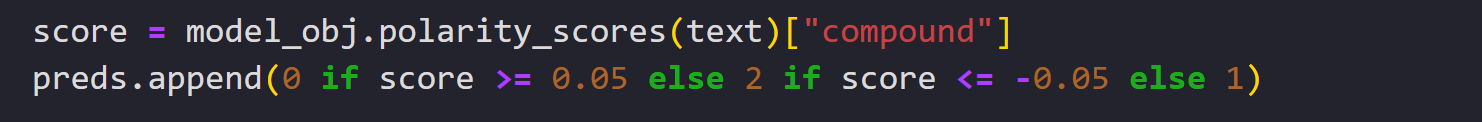


Abbildung 1: Klassifizierung eines Textes in bullish, neutral und bearish basierend auf dem compound-Score von Vader

Nach der Inferenz wurde für jedes Modell ein Klassifikationsreport generiert. Die F1-Scores der drei Klassen wurden extrahiert und zentral gespeichert:

Ein Bild, das Text, Screenshot, Schrift enthält.

KI-generierte Inhalte können fehlerhaft sein.

Abbildung 2: Funktion zur Extraktion der F1-Scores pro Klasse aus einem gespeicherten Klassifikationsreport

Die aufbereiteten Scores wurden später zur Erstellung der Balkendiagramme in Kapitel 4 verwendet.

3.3 Modelloptimierung mit Custom- Trainer

Für die zwei besten Modelle – FinBERT (Posts) und CryptoBERT (Kommentare) – wurde ein gezieltes Fine-Tuning durchgeführt. Dabei kam ein Custom-Trainer mit gewichteter Verlustfunktion zum Einsatz, um die unausgeglichene Klassenverteilung zu berücksichtigen:

Ein Bild, das Text, Screenshot, Schrift enthält.

KI-generierte Inhalte können fehlerhaft sein.

Abbildung 3: Implementierung eines gewichteten Custom- Trainers zur Berücksichtigung unbalacierter Klassen

Es wurde mit unterschiedlichen Gewichtungen experimentiert bis das beste Ergebnis gespeichert wurde.

3.4 Vergleich verschiedener Lernrraten

Ein zentrales Element der Optimierung war ein systematischer Vergleich unterschiedlicher Lernraten (2e-5 bis 1e-6). Für jede Rate wurde ein vollständiger Trainingslauf durchgeführt und die F1-Makro-Scores pro Epoche mitgeloggt.

Ein Bild, das Text, Screenshot, Software enthält.

KI-generierte Inhalte können fehlerhaft sein.

Abbildung 4: Trainingsschleife zum Vergleich verschiedener Lernraten mittels Trainingarguments in Hugging Face

Die daraus entstandenen Lernkurven sind in Kapitel 4 abgebildet und wurden zur Auswahl des besten Modells verwendet.

4 Ergebnisse und Diskussion

Ziel dieses Abschnitts ist die systematische Analyse und Bewertung der getesteten Modelle. Dabei wird zwischen den beiden Datentypen – Posts und Kommentare – unterschieden. Zusätzlich werden die Effekte des Fine-Tunings auf die Modellperformance untersucht.

4.1 Modellvergleich auf untrainierten Baselines

Abbildung 5 und 6 zeigen die F1-Scores der getesteten Modelle auf dem ursprünglichen, untrainierten Stand. Dabei wurde jeweils ein gemeinsames Testset verwendet. Auffällig ist, dass kein Modell alle drei Klassen konsistent gut erkennt.

* Bei Posts (Abb. 5) schnitten VADER und TextBlob insgesamt stabil ab, insbesondere bei *bearish*. Transformer-Modelle wie DeBERTa und RoBERTa hatten starke Einbrüche bei *bearish* (F1 teils < 0.1).

Ein Bild, das Text, Screenshot, Diagramm, Reihe enthält.

KI-generierte Inhalte können fehlerhaft sein.

Abbildung 5: F1-Scores der Ausgsangsmodelle auf Reddit-Posts

* Bei Kommentaren (Abb. 6) zeigte CryptoBERT auch ohne Fine-Tuning bereits eine sehr starke Performance (F1 bis zu 0.91 für *bullish*), während DeBERTa hier erneut nahezu vollständig versagte.

Ein Bild, das Text, Screenshot, Diagramm, Reihe enthält.

KI-generierte Inhalte können fehlerhaft sein.

Abbildung 6: F1-Scores der Ausgangsmodelle auf Reddit-Kommentaren

Diese Ergebnisse verdeutlichen zwei zentrale Punkte:

1. Die lexikonbasierten Modelle liefern bei kurzen Texten (*Kommentare*) keine belastbaren Ergebnisse.
2. Transformer-Modelle brauchen kontext- und domänenspezifisches Fine-Tuning, um ihr Potenzial auszuschöpfen.

4.2 Fine-Tuning Ergebnisse

Nach gezieltem Fine-Tuning auf jeweils den geeignetsten Modellen wurden deutliche Verbesserungen erzielt.

CryptoBERT wurde auf dem externen Datensatz mit 999 Kommentaren feinjustiert. Die Ergebnisse nach zehn Epochen und optimierter Lernrate (5e-6) sind in Tabelle 1 zusammengefasst:

* F1-Score (macro): 0.9199
* Cohen’s Kappa: 0.8814
* Sehr hohe Scores in allen drei Klassen: bullish (0.96), neutral (0.88), bearish (0.92)

Diese Ergebnisse übertreffen alle anderen Modelle deutlich. Besonders positiv fällt auf, dass auch *bearish* zuverlässig erkannt wird – was ohne Fine-Tuning nicht gelang.

FinBERT wurde auf einem eigenen Datensatz mit 298 Reddit-Posts trainiert. Trotz gezieltem Feintuning und Lernratenvergleich blieb die Modellgüte deutlich hinter CryptoBERT zurück:

* F1-Score (macro): 0.5329
* Cohen’s Kappa: 0.2895
* Klassenunterschiede: *bullish* (0.55), *neutral* (0.62), *bearish* (0.43)

Die moderate Performance kann auf die begrenzte Datenmenge (~750 Posts) zurückgeführt werden. Zudem enthalten Posts oft komplexere Satzstrukturen und weniger klare Stimmungsindikatoren als Kommentare.

4.3 Lernratenvergleich

Abbildung 3 und 4 zeigen die Lernkurven des F1-Scores (macro) auf dem Validierungsset für verschiedene Lernraten:

* Bei CryptoBERT (Abb. 3) zeigten sich klare Unterschiede. Die besten Ergebnisse wurden bei 5e-6 und 3e-6 erzielt, die auch langfristig stabil blieben. Höhere Raten (1e-5, 2e-5) führten zu instabilen Verläufen.

Ein Bild, das Text, Reihe, Diagramm, Zahl enthält.

KI-generierte Inhalte können fehlerhaft sein.

Abbildung 7: Lernkurven(F1 macro) für CryptoBERT bei verschiedenen Lernratenauf dem Kommentar-Datensatz

* Bei FinBERT (Abb. 4) war die Lernkurve weniger klar ausgeprägt. Zwar zeigten 3e-6 und 5e-6 ebenfalls stabile Resultate, jedoch ohne signifikante Trennung zu den schlechteren Raten.

Ein Bild, das Text, Reihe, Diagramm, Zahl enthält.

KI-generierte Inhalte können fehlerhaft sein.

Abbildung 8: Lernkurven (F1 macro) für FinBERT bei verschiedenen Lernraten auf dem Post-Datensatz

Diese Beobachtungen bestätigen die Annahme von Huyen (2022, S. 224), dass Transformer-Modelle bei kleinen Datensätzen besonders empfindlich auf Hyperparameter wie die Lernrate reagieren.

4.4 Gesamtbewertung

Die besten Ergebnisse wurden mit CryptoBERT nach Fine-Tuning erzielt. Die sehr hohe Klassifikationsgüte bei Kommentaren (F1 > 0.91) zeigt, dass domänenspezifisches Vorwissen und Fine-Tuning auf strukturell passenden Daten entscheidend sind. FinBERT hingegen blieb hinter den Erwartungen zurück – vermutlich primär aufgrund der limitierten Datenbasis.

Die Ergebnisse legen nahe, dass in zukünftigen Projekten der Fokus verstärkt auf:

* skalierbare Datenbeschaffung (mehr manuelle Labels oder automatisiertes Bootstrapping),
* und auf datengetriebene Hyperparameteroptimierung gelegt werden sollte.

5 Fazit

Ziel dieser Arbeit war es, verschiedene NLP-Modelle zur Sentimentanalyse von Reddit-Inhalten im Kryptokontext systematisch zu vergleichen und durch gezieltes Fine-Tuning zu optimieren. Die Ergebnisse zeigen deutlich: Während einfache lexikonbasierte Verfahren wie VADER und TextBlob bei stark verkürzten Texten wie Kommentaren unzureichende Ergebnisse liefern, können transformerbasierte Modelle wie FinBERT und CryptoBERT – insbesondere nach gezielter Anpassung – sehr leistungsfähige Klassifikatoren darstellen.

Der deutlichste Leistungsgewinn konnte bei CryptoBERT auf Reddit-Kommentaren erzielt werden. Nach gezieltem Fine-Tuning mit ausgewählten Lernraten erreichte das Modell einen macro-F1-Score von 0.92 bei gleichzeitig hoher Übereinstimmung (Cohen’s Kappa = 0.88). Dies zeigt, dass kontextbasiertes Training auf domänenspezifischen Texten der entscheidende Erfolgsfaktor für Sentimentmodelle in informellen Umgebungen wie Reddit ist.

Dagegen konnte FinBERT auf Reddit-Posts trotz aufwendiger Lernratenanalyse und Class-Weighting nur begrenzte Verbesserungen erzielen. Die Ursache liegt vor allem in der kleinen Datenbasis, die keine robuste Generalisierung ermöglicht. Auch Data Augmentation blieb ohne nennenswerte Wirkung – ein weiterer Hinweis darauf, dass Modellverbesserungen bei NLP stark datendependet sind.

5.1 Ausblick

Auf Basis der gewonnenen Erkenntnisse ergeben sich mehrere Perspektiven für weiterführende Forschung und Anwendung:

Größere Datenmengen: Der Aufbau eines umfangreicheren, manuell oder halbautomatisch gelabelten Post-Datensatzes würde das Finetuning-Potenzial erheblich steigern.

Semi-supervised Learning & Bootstrapping: Die Kombination kleiner gelabelter Datenmengen mit unlabelten Daten könnte mithilfe von Pseudo-Labeling oder Active Learning neue Wege eröffnen (vgl. Huyen, 2022, S. 225).

Explainable AI: Der Einsatz von Interpretationsmethoden wie LIME oder SHAP könnte helfen, Modellentscheidungen besser zu verstehen – insbesondere bei fehlklassifizierten *bearish*-Beispielen.

Abschließend lässt sich festhalten: Mit zunehmender Reife der Modelle und wachsender Datenverfügbarkeit stellt die Sentimentanalyse auf Reddit eine vielversprechende Methode dar, um kollektive Meinungsbilder zu erfassen – vorausgesetzt, die gewählten Modelle werden sorgfältig angepasst und datenseitig gestützt.

6 Literaturverzeichnis

Bramer, M. (2007). *Principles of data mining* (Vol. 180). London: Springer.

Caelen, O., & Blete, M. A. (2025). *Anwendungen mit GPT-4 und ChatGPT entwickeln: Intelligente Chatbots, Content-Generatoren und mehr erstellen*. o’Reilly Verlag.

Downey, A. B., & Lang, J. W. (2024). *Python lernen mit KI-Tools: Einstieg in die Programmierung mit KI-Unterstützung*. o’Reilly Verlag GmbH & Co. KG.

Huyen, C. (2022). *Designing machine learning systems: An iterative process for production-ready applications*. O’Reilly Media.

Tan, P.-N., Steinbach, M., & Kumar, V. (2016). *Introduction to data mining* (2nd ed.). Pearson Education India.

IEEE (2023). *Sentiment analysis of crypto-related Reddit posts using transformer models*. IEEE Xplore. <https://ieeexplore.ieee.org/document/10223689>

7 Abbildungsverzeichnis

Abb. 1: Klassifikation mit VADER auf Basis des compound-Scores.  
Abb. 2: Funktion zur Extraktion von F1-Scores aus Modellreports.  
Abb. 3: Implementierung eines gewichteten Custom-Trainers zur Berücksichtigung unbalancierter Klassen.  
Abb. 4: Trainingsschleife zum Vergleich verschiedener Lernraten mittels TrainingArguments.  
Abb. 5: F1-Scores der Ausgangsmodelle auf Reddit-Posts (ohne Fine-Tuning).  
Abb. 6: F1-Scores der Ausgangsmodelle auf Reddit-Kommentaren (ohne Fine-Tuning).  
Abb. 7: Lernkurven (F1 macro) für CryptoBERT bei verschiedenen Lernraten auf dem Kommentar-Datensatz.  
Abb. 8: Lernkurven (F1 macro) für FinBERT bei verschiedenen Lernraten auf dem Post-Datensatz.