HAW HAMBURG

MASTERTHESIS
Jonathan Wischhusen

Untersuchung von Active Learning Methoden für BERT-Modelle

FAKULTÄT TECHNIK UND INFORMATIK
Department Informatik

Faculty of Computer Science and Engineering
Department Computer Science

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN HAMBURG
Hamburg University of Applied Sciences

Jonathan Wischhusen

Untersuchung von Active Learning Methoden für BERT-Modelle

Masterarbeit eingereicht im Rahmen der Masterprüfung
im Studiengang *Master of Science Informatik*
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Olaf Zukunft
Zweitgutachter: Prof. Dr. Stefan Sarstedt

Eingereicht am: 23. Dezember 2021

Jonathan Wischhusen

Thema der Arbeit

Untersuchung von Active Learning Methoden für BERT-Modelle

Stichworte

Active Learning, BERT, Textklassifikation

Kurzzusammenfassung

Automatisierte Textklassifikation ist für viele praktische Anwendungen ein aussichtsreiches Analyse- und Moderationsinstrument. In der Praxis steht Textklassifikation jedoch oft teuren Annotationskosten und einem Ungleichgewicht der Klassen in den Trainingsdaten gegenüber. Active Learning beschreibt ein Paradigma, dass die Kosten für die Annotation signifikant senken kann, indem über mehrere Iterationen geeignete Daten gezielt annotiert werden. In Verbindung mit vortrainierten Sprachmodellen, wie BERT und seinen Variationen, ist Active Learning bisher wenig untersucht. Diese Arbeit untersucht verschiedene Active Learning Methoden für BERT-Modelle unter dem Problem der Mehr-Klassen-Textklassifikation. Der Fokus liegt auf Szenarien mit praxisnahen Startmengen und ihre Auswirkung für vielfältige Datensätze. Die Ergebnisse zeigen, dass Discriminate Active Learning im Umfeld der Untersuchung als einzige Methode über die Modelle und Daten hinweg signifikant besser ist als der Zufall. Andere Methoden sind in der Regel nicht besser als der Zufall, außer in einigen praxisnahen Situationen. Die Arbeit gewährt ebenfalls einen Einblick in den Einsatz des Menschen als Orakel. Durch eine Benutzerstudie wird beobachtet, dass für ein kleines Annotationsbudget Menschen eine konstante Leistung zeigen und Domänenwissen auch bei einfachen Kategorien hilfreich ist.

Jonathan Wischhusen

Title of Thesis

Study of Active Learning methods for BERT models

Keywords

Active Learning, BERT, text classification

Abstract

Automated text classification is a promising analysis and moderation tool for many practical applications. In practice, text classification often faces expensive annotation costs and class imbalance in the training data. Active Learning describes a paradigm that can significantly reduce annotation costs by selectively annotating the most suitable data over multiple iterations. In conjunction with pre-trained language models, such as BERT and its variations, Active Learning has been little studied. This work investigates Active Learning for BERT models under the problem of multi-class text classification. The focus is on scenarios with practical warmstart sets and their impact for diverse datasets. The results show that Discriminate Active Learning is the only method that significantly outperforms the random baseline across models and datasets in the setting of the study. Other methods are outperfoming the baseline only for some real-world scenarios. The work also provides insight into the use of humans as an oracle. A user study concludes that humans show consistent performance over a small annotation budget and that domain knowledge is helpful even for simple categories.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungen

AL Active Learning

ALBERT A Lite BERT

BERT Bidirectional Encoder Representations from Transformers

CNN Convolutional neural network

DAL Discriminative Active Learning

DistilBERT Distilled version of BERT

LSTM Long Short-Term Memory

MC Monte Carlo

MLM Masked Language Modeling

NLP Natural Language Processing

NSP Next Sentence Prediction

QBC Query By Committee

RNN Recurrent neural network

RoBERTa Robustly optimized BERT approach

SOP Sentence Order Prediction

1 Einleitung

1.1 Motivation

Die Klassifikation von Texten ist eine primäre Aufgabe beim Verarbeiten von natürlicher Sprache und hat eine große Bandbreite von Anwendungsfeldern. Als Teildisziplin der Informationsgewinnung aus Texten ist sie vor allem in den Bereichen interessant, die auf benutzergenerierten Inhalt zurückgreifen, wie zum Beispiel die Einordnung von Benutzererfahrungen, die Moderation von Benutzerbeiträgen oder für Stimmungsanalysen auf Social-Media-Plattformen.

Mit der Vorstellung von Bidirectional Encoder Representations from Transformers (BERT) [6] wurde eine Gruppe neuronaler Netze ins Leben gerufen, die in vieler Hinsicht den heutigen Maßstab beim Lösen von vielen Natural Language Processing (NLP) Aufgaben bilden [21, 26, 36]. Dadurch sind sie ein vielversprechender Ansatz für die Textklassifikation.

Die Leistungsfähigkeit der BERT-Modelle basiert, neben einem neuartigen Attention-Mechanismus [46], auf dem Transfer von Wissens. Die Modelle eignen sich abstraktes Wissen durch unüberwachtes Lernen einer Quelldomäne an. Im Fall von BERT ist dies ein möglichst großer Textkorpus, wie die Wikipedia Enzyklopädie, die einen umfassenden Querschnitt einer Sprache enthält. Das vortrainierte Modell kann daraufhin für eine Zielaufgabe für diese Sprache durch überwachtes Lernen fein abgestimmt werden. Zu der Popularität von BERT hat, neben der Leistungsfähigkeit, das Transfer Learning Prinzip beigetragen, wodurch aufwändig vortrainierte Modelle einer Sprache unkompliziert und mit verhältnismäßig geringem Rechenaufwand für weitere Aufgaben verwendet werden können [33]. Devlin u. a. [6] haben neben der Architektur des neuronalen Netzes ebenfalls einige dieser vortrainierten Modelle veröffentlicht, die sowohl in der Forschung als auch in der Praxis viel Anklang finden.

Obwohl BERT im Vergleich zu anderen Deep-Learning-Modellen, mit einer geringeren Trainingsmenge zurechtkommt [6], bleiben die Probleme des überwachten Lernens allgegenwärtig. Überwachtes Lernen umfasst zwei Aspekte, einerseits die Verfügbarkeit von tausenden Daten samt den dazugehörigen Klassen, andererseits die Fähigkeit, aus diesen Daten latente Merkmale zu lernen. Während die Forschung die Leistung der Modelle und deren Training stetig optimiert, wird oft vernachlässigt, mit welchem Aufwand vollständig annotierte Datensätze für die praktische Anwendung einhergehen. Einheitliche Datensätze sind zwar in der Forschung essentiell, da durch sie die Vergleichbarkeit der Modellperformance ermöglicht wird, allerdings haben sie relativ wenig Nutzen für individuelle Aufgaben und eigene Daten in der Praxis. Daten in großen Mengen zu annotieren, um einen Trainingsdatensatz aufzubauen, ist kostspielig und zeitaufwändig. Es können Experten für die Bestimmung der Zielklassen erforderlich sein, oder es werden redundante Daten annotiert, wodurch Arbeitszeit ineffizient genutzt wird. Hinzu kommt die Gefahr, dass die gewünschte Klasse in den erhobenen Daten unterrepräsentiert ist. Dies stellt eine nicht unerhebliche wirtschaftliche Hürde für die breite und praktische Nutzung von maschinellem Lernen für NLP-Aufgaben außerhalb den Forschungsdomänen und der englischen Sprache dar. Gildenblat [12] zufolge dreht sich die praktische Anwendung von überwachtem Lernen vor allem um die Möglichkeit Daten in Zukunft schnell und mit geringem Aufwand zu annotieren.

Active Learning (AL) ist ein Forschungsgebiet, die versucht Schwierigkeiten bei der Annotation von Daten zu lösen. Es wird davon ausgegangen, dass die Erhebung der Daten relativ einfach, der Prozess eine oder mehrere Klassen zuzuweisen jedoch kostspielig ist. Die Bestimmung der Klasse unstrukturierter Texte erfordert kognitive Arbeit, die in der Regel durch einen Menschen geleistet werden muss. Active Learning nimmt sich der Frage an, welche Datenpunkte zu der größten Verbesserung der Genauigkeit des Modells führen [40]. Dem Prinzip liegt die Annahme zugrunde, dass der Algorithmus selbst entscheiden kann, welche Daten für ihn und seine Aufgabe relevant und welche redundant sind. Kombiniert man diese Forschungsbereiche ergibt sich ein interessantes Bild für die praktische Klassifizierung von Texten.

Aktuelle Bestrebungen, wie das Forum 4.0 der Allianz der Hamburger Hochschulen für Informatik, unterstreichen den Bedarf nach einfacher Textklassifikation für die maschinelle Analyse, Aggregation und Visualisierung von Nutzerkommentaren.

In einer dieser Arbeit vorausgegangenen Untersuchung wurde die Annotation von Texten unter einem anderen Ansatz untersucht [49]. Es wurde erörtert, ob sich Stimmungsclus-

ter anhand der Ausgabevektoren eines vortrainierten BERT-Modells, allein anhand des Textaufbaus und den verewendeten Wörtern, erkennen lassen. Die These ist, dass negativ sowie positiv gestimmte Texte sich im Ausdruck ähnlich sind und Ausgabevektoren erzeugen, die durch verschiedene Clusterverfahren entsprechend ihrer Stimmung in homogene Gruppen unterteilt werden können. Durch anschließende Reduktion auf zwei Dimensionen erhält der Benutzer eine visuelle Darstellung der Cluster, die ein schnelles Verständnis der Verteilung erlauben und die Annotation der Gruppen vereinfacht. Die Untersuchung hat mit dem Basis Modell von BERT[1] keine generalisierbaren Ergebnisse im Hinblick auf die Stimmungsanalyse geliefert. Erst nach Feinabstimmung mit einer kleinen Teilmenge der Ausgangsdaten und ihrer Stimmungsklasse, deuten sich zusammenhängende Cluster an.

Im Gegensatz zu den unüberwachten Clusteralgorithmen der vorausgegangen Untersuchung und im Hinblick auf die Notwendigkeit von annotierten Daten für neuronale Netze, liegt das Augenmerk dieser Arbeit auf der Kombination des Active Learning Prinzips mit modernen neuronalen Netzen, die auf der BERT Architektur basieren.

1.2 Zielsetzung

Active Learning mag Vorteile bei speziellen Modellen und Einsatzbereiche bringen. Diese Vorteile lassen sich aber nicht über alle Modelle und Einsatzbereiche verallgemeinern [27]. Diese Arbeit untersucht verschiedene Active Learning Methoden in Kombination mit BERT-Modellen um ein allgemeines Bild für mehrklassige Textklassifikation zu erhalten. Die Performance der Methoden wird anhand verschiedene Daten in verschiedenen Szenarien gemessen. Dabei wird ein möglichst realitätsnaher Aufbau der Untersuchung verfolgt, der an die Untersuchung von Dor u. a. [7] angelehnt ist. Folgenden drei Fragestellungen wird nachgegangen:

RQ1. Ist Active Learning für BERT für das Problemfeld der Textklassifikation mit mehreren Klassen geeignet?

RQ2. Welche Auswirkungen hat die Qualität der initialen Annotationen auf die Performance des Modells?

[1] BERT-Base, Uncased: 12-layer, 768-hidden, 12-heads, 110M parameters

RQ3. Wie wirkt sich die Qualität der Datensätze, im Hinblick auf die Unterrepräsentation von Klassen, aus?

Die gestellten Fragen beziehen sich auf einen konzeptionellen Teil des Problems. Um Erkenntnisse der praktischen Machbarkeit von Active Learning zu erlangen, wird den Berechnungen eine Nutzerstudie über die Akzeptanz der Rolle des Menschen in einem Active Learning Szenario entgegengesetzt. Dabei wird die Frage betrachtet:

RQ4. Wie wirkt sich die repetitive kognitive Arbeit, die zur Bestimmung der Klassen notwendig ist, auf die Konzentration und Leistungsfähigkeit des Nutzers aus?

1.3 Aufbau

Die Arbeit ist folgendermaßen aufgebaut. Kapitel 2 startet mit der Beschreibung der Grundlagen auf die diese Arbeit Bezug nimmt. Kapitel 3 befasst sich mit der wissenschaftlichen Einordnung und wirft einen Blick auf relevante Arbeiten in dem Bereich. Der Aufbau der Untersuchung ist in Kapitel 4 aufgeführt. Dabei werden die Methodik, die Modelle, die Datensätze, die Ausgangssituation der Experiment, das Active Learning Framework, sowie eine Nutzerstudie dargestellt. Die Ergebnisse werden im Hinblick der Fragestellungen in Kapitel 5 beschrieben. Anschließend werden diese in Kapitel 6 diskutiert. Kapitel 7 fasst die Erkenntnisse der Arbeit zusammen und gibt einen Ausblick auf weiterführende Themen.

2 Grundlagen

In diesem Kapitel werden die grundlegenden Methoden und Konzepte im Hinblick auf die Fragestellungen beschrieben. Um Active Learning mit neuronalen Netzen zu verwenden, sind die untergeordneten Konzepte interessant. Zunächst werden die traditionellen Lernalgorithmen erläutert. Sie bilden die Grundlage für Transfer Learning und Active Learning, während Transfer Learning wiederum ein Kernelement der BERT-Modelle ist. Anschließend wird Active Learning ausführlich beschrieben. Es werden gängige Active Learning Methoden erläutert, die später für die Untersuchung in Kapitel 4 herangezogen werden. Anschließend wird die Herausforderungen der Adaption von Active Learning auf neuronale Netze wird dargestellt. Abschließend wird der Ursprung von BERT und seine Architektur eingegangen.

2.1 Algorithmen für maschinelles Lernen

Algorithmen für maschinelles Lernen treffen in der Regel Vorhersagen für unbekannte Daten mithilfe statistischer Modelle und neuronaler Netze. Diese werden auf einem Datensatz trainiert, sodass ein Modell die Muster der Eingabedaten adaptiert und auf zukünftige unbekannte Daten anwenden kann. Der Datensatz, der zum Trainieren herangezogen wird, bildet die Grundlage und bestimmt den späteren Rahmen der Vorhersagen des Modells. Es ist daher wichtig, dass die Trainingsdaten ausreichend divers und repräsentativ sind, damit ein Modell klare Entscheidungsgrenzen aufbauen kann. Die Trainingsalgorithmen versuchen dabei ein Gleichgewicht zwischen Optimierung, also der möglichst genauen Vorhersage der bekannten Daten und Generalisierung, der korrekten Vorhersage unbekannter Daten, zu erreichen.

2.1.1 Überwachtes Lernen

Überwachtes Lernen wird oft für Klassifizierungs- und Regressionsprobleme eingesetzt, wie beispielsweise die Stimmungsanalyse in Benutzerkiritken oder die Vorhersage des Verkaufsvolumens zu bestimmten Daten. Beim überwachten Lernen besteht das Ziel darin, Daten im Zusammenhang mit einer bestimmten Fragestellung zu betrachten.

Beim überwachten Lernen hat jeder Datenpunkt eine definierte Ausgabe, die zu Beginn des Trainings bekannt ist. Der Algorithmus bringt einem Modell eine Projektion bei, sodass zu den Eingabedaten möglichst genau die gewünschte Ausgabe vorhergesagt wird [32]. Die Parameter des Modells werden durch die Minimierung einer Verlustfunktion inkrementell angepasst, bis das Modell eine gewünschte Performance erreicht. Die erreichbare Performance eines Modells hängt, neben den Möglichkeiten des Modells selbst, stark von den zugrundeliegenden Daten und deren Anzahl ab. Damit ein Modell eine gewünschte Performance erreicht, müssen eine ausreichende Anzahl an korrekten Ein- und Ausgabedatenpaaren zur Verfügung stehen.

Die Bestimmung der Ausgabedaten, auch als Annotation, Label oder Klassen bezeichnet, stellt eine Herausforderung dar, da sie für gewöhnlich Domänenexperten erfordert, die zu jeder Eingabe die entsprechende Ausgabe festlegen. Hinzu kommen Faktoren wie Erfahrungen in dem Gebiet, Emotionen und persönlichen Umstände, die das Urteilsvermögen einer Person während des Erarbeitens der Ausgangsdaten beeinflussen. Daraus lässt sich ableiten, dass Annotationen einer einzelnen Person einem Bias unterliegen können und andere Fehler enthalten, was in einer falschen Bestimmung der Ausgabe resultiert. Die Integrität der Ausgabedaten kann gewährleistet werden, wenn mehrere Domänenexperten unabhängig die Ausgaben bestimmen. Die dadurch entstehende Redundanz minimiert die Fehler. Der Prozess wird dadurch jedoch teurer und zeitaufwändiger.

2.1.2 Unüberwachtes Lernen

Im Gegensatz zu überwachtem Lernen, sind beim unüberwachten Lernen keine Annotationen der Trainingsdaten erforderlich. Der Algorithmus ermittelt selbständig die Muster und Eigenschaften innerhalb der Daten, anstatt sie über festgelegte Klassen in Beziehung zu setzen. Durch die Freiheit die Eingabedaten selbständig in Relation zueinander zu setzen, können interessante und unerwartete Ergebnisse aufgedeckt werden, die für den Menschen nicht direkt ersichtlich scheinen. Unüberwachtes Lernen wird häufig zur

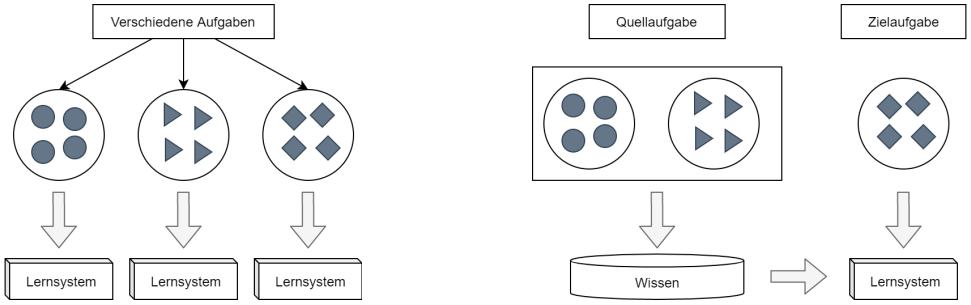

Abbildung 2.1: Gegenüberstellung der Lernprozesse von (a) traditionellem maschinellen Lernen und (b) Transfer Learning [33].

Bildung von Clustern, dem Finden von Gruppen innerhalb der Daten, und der Datenassoziation, dem Ausmachen der Regeln, die die Daten beschreiben, eingesetzt.

Ein Grundproblem beim unüberwachten Lernen besteht darin sicherzustellen, dass der Algorithmus Daten korrekt interpretiert, da die klare Zugehörigkeit einzelner Datenpunkte durch Klassen von außen nicht stattfindet. Die Interpretation der Ergebnisse kann dadurch schwieriger sein.

2.1.3 Transfer Learning

Transfer Learning beschreibt das Lernprinzip, bei dem zuvor gelerntes Wissen eines Modells übertragen wird [33]. Bei unüberwachtem und überwachtem Lernen sind die Quell- und Zieldomäne und die Quell- und Zielaufgabe gleich. Es existiert also ein Datensatz und ein Ziel, die das Modell beschreiben. Im Gegensatz dazu findet beim Transfer Learning eine Aufteilung in Quell- und Zieldomänen statt. Die Bereiche sind nicht mehr dieselben, sondern hängen lediglich miteinander zusammen. Das Modell wird zuerst mit den Daten einer Quelldomäne trainiert und anschließend mit denen der Zieldomäne.

Die beiden Verfahren unterscheiden sich im Hinblick auf das Lernverhalten, wie Tabelle 2.1 zeigt. Um verschiedene Aufgaben zu erfüllen, werden die klassischen Varianten, das über- und unüberwachte Lernen, pro Aufgabe von Grund auf neu trainiert. Bei dem Transfer von gelerntem Wissen hingegen wird versucht, einmal zuvor erlerntes Wissen für neue Aufgaben zu nutzen. Ein früherer Kenntnisstand wird auf einen ähnliche Zielbereich abgebildet. Ziel ist es einmal gelerntes Wissen möglichst effizient und mit wenig Verlust von Wissen für zusammenhängende Aufgaben bereitzustellen [33]. Dabei wird

Lernmethode		Quell- und Zieldomäne	Quell- und Zielaufgabe	Annotationen	
				Quelle	Ziel
Überwachtes Lernen		Gleich	Gleich	✓	✓
Unüberwachtes Lernen		Gleich	Gleich	-	-
Transfer Learning	induktiv	Gleich	Verschieden	✓/-	✓
	unüberwacht	Gleich	Verschieden	-	-
	transduktiv	Verschieden	Gleich	✓	✓

Tabelle 2.1: Beziehung zwischen traditionellem maschinellen Lernen und verschiedenen Transfer Learning Varianten.

einerseits Rechenaufwand minimiert und andererseits besteht die Möglichkeit die Leistung der Modelle zu erhöhen, da auf Kenntnisse aus einer zusammenhängenden Domäne zurückgegriffen werden kann. Ebenfalls fällt der Bedarf an neuen Trainingsdaten für die Abstimmung auf eine ähnliche Zielaufgabe wesentlich geringer aus.

Ein Nachteil von Transfer Learning ist der negative Transfer von Wissen, welcher auftritt, wenn das Modell mit Daten trainiert wird, die sich negativ auf die Performance auswirken [48]. Dazu zählen beispielsweise Datenpunkte die keinen Bezug zu der Quelldomäne des Modells haben.

Der Transfer Learning Ansatz für ein Problem ergibt sich daraus, ob die Daten aus der Quell- oder Zieldomäne annotiert sind. Danach richtet sich ebenfalls, ob überwachtes oder unüberwachtes Lernen für den Schritt genutzt werden kann. Tabelle 2.1 gibt Aufschluss über die verschiedenen Transfer Learning Varianten und ihren Anwendungsvoraussetzungen gegenüber traditionellem maschinellen Lernen.

2.2 Active Learning

Im Vergleich zu den vorherigen Lernansätzen existiert mit Active Leaning eine Methode, die den Aufwand Daten zu annotieren berücksichtigt [40]. Während unüberwachtes Lernen keine definierten Kategorien in der Klassifikation erlaubt und überwachtes Lernen das Vorhandensein aller Klassen im Datensatz voraussetzt, geht Active Learning einen Mittelweg, bei dem nur ein kleiner Teil des Datensatzes annotiert wird und der größere

Teil unbestimmt bleibt. Der Ansatz nutzt überwachtes Lernen auf einer Teilmenge der Daten, die inkrementell erweitert und annotiert wird.

Der Gedanke hinter Active Learning ist, dass der Algorithmus selbst bestimmen kann welche Daten er zum Lernen heranzieht, da nicht jeder Datensatz die gleiche Relevanz für das Modell hat. Davon verspricht man sich, bei einem geringeren Trainingsaufwand mit weniger annotierten Daten, schneller eine gleichwertige oder höhere Genauigkeit mit dem trainierten Modell zu erzielen als es mit überwachtem Lernen des komplett annotierten Datensatzes der Fall wäre [40].

Beim Active Learning nimmt der Algorithmus die Rolle eines Schülers ein, während ein Orakel, in der Regel ein Mensch, selten auch eine Maschine oder ein Computer, als Lehrer auftritt und erfragte Datenpunkt des Algorithmus mit der korrekten Klasse beantwortet. Durch ein menschliches Orakel wird die Human in the Loop Idee insofern angerissen, als dass die kognitive Fähigkeiten und das Domänenwissen des Menschen das Modell, durch Beantworten der erfragten Datenpunkte, indirekt beeinflusst.

Der Active Learning Algorithmus steht vor dem Problem, bei jeder Iteration herauszufinden welche Daten für ihn relevant sind, abhängig von den bereits bestimmten Daten in der Trainingsmenge. Dabei spielt einmal das vorhandene Senario sowie die Auswahlstrategie eine wichtige Rolle.

2.2.1 Szenarien

Die Art und Weise wie der Algorithmus das Orakel befragt wird allgemein als Active Learning Szenario bezeichnet.

Membership Query Synthesis

Bei diesem Ansatz synthetisiert der Algorithmus typischerweise fiktive Datenpunkte auf Grundlage der unbestimmten Daten [1]. Diese repräsentieren markante Eigenschaften, welche indirekt eine Gruppe der Eingabedaten beschreiben. Das Orakel weist den generierten Datenpunkten daraufhin ihre Klasse zu.

Bei dieser Methode entstehen Probleme, wenn das Orakel ein Mensch ist, da die synthetisierten Daten, die selbst nicht in den Eingabedaten vorkommen, oft nicht klar zu identifizieren sind. Nimmt man an, der Algorithmus trainiert ein Modell, das Handschrift

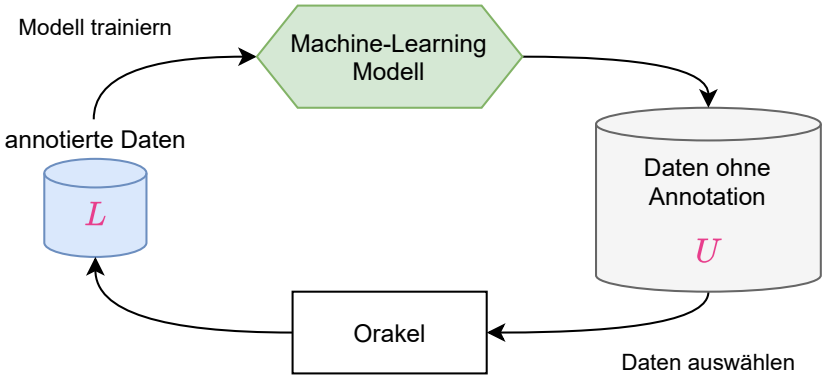

Abbildung 2.2: Pool basierter Active Learning Kreislauf [40].

erkennen soll, könnte dem Orakel eine Grafik vorgelegt werden, die aus Sicht des Algorithmus eindeutige Merkmale der Eingabedaten vereint, für den Menschen jedoch nach willkürlichen schwarzen Linien aussieht, die er keinem Zeichen zuordnen kann.

Andererseits ist diese Methode vielversprechend, wenn für das Klassifikationsproblem eine Maschine als Orakel eingesetzt wird, die für einen synthetischen fiktiven Datenpunkt die entsprechende Klasse durch ein Experiment bestimmen kann [18].

Stream-Based Selective Sampling

In diesem Szenario werden Datenpunkte nacheinander aus der Menge der Eingabedaten betrachtet. Das setzt voraus, dass das Auswählen von Datenpunkten aus dem Eingaberaum keine Kosten erzeugt. Dadurch das die Eingabemenge unklar ist, ist ebenfalls die Verteilung unbekannt. Der Algorithmus entscheidet also pro Datenpunkt, ob er dessen Klasse erfragt oder die Information darüber verwirft. Der Informationsgehalt kann unterschiedlich festgestellt werden, etwa durch einen berechneten Bereich im Eingaberaum, bei dem das Modell Schwierigkeiten hat, Klassen zu unterscheiden oder durch Auswahlstrategien (Abschnitt 2.2.2), die kein Wissen über andere Eingabedaten benötigen. Ein Ein Beispiel für dieses Szenario ist die Wortarterkennung (Part-of-speech-Tagging) [4].

Pool-based Sampling

Aus den Umständen der meisten Probleme, die durch maschinelles Lernen gelöst werden sollen, lässt sich dieses Szenario ableiten. Genauer gesagt sind es die Bereiche, bei denen

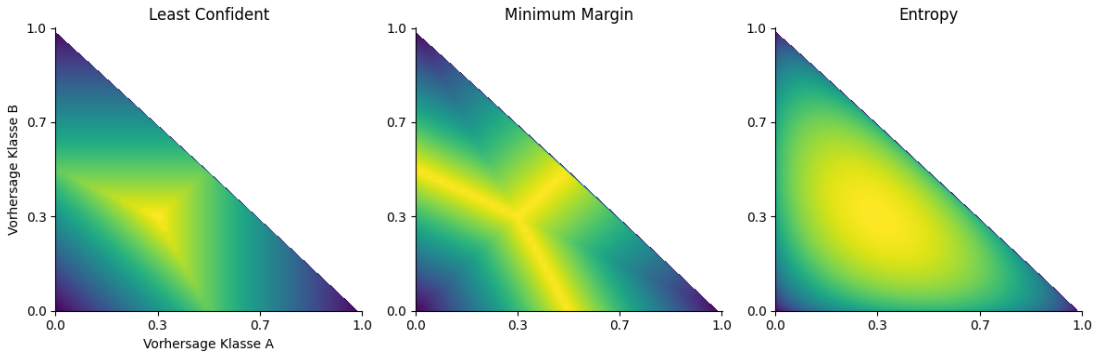

Abbildung 2.3: Zweidimensionale Darstellung der Unsicherheit des Uncertainty Samplings bei einem ternären Klassifikationsproblem.

unbestimmte Daten in großen Mengen und ohne viel Aufwand im Vorhinein gesammelt werden können und dadurch die Verteilung der Eingabedaten bekannt ist. Im Vergleich zum Stream-Based Selective Sampling, können so beim Pool-based Sampling die Eingabedaten in Relation zueinander betrachtet werden. Dabei werden ebenfalls Auswahlstrategien (Abschnitt 2.2.2) angewendet um den Informationsgehalt der Datenpunkte zu bestimmen. Die Datenpunkte mit dem größten Informationsgewinn werden dem Orakel vorgelegt. Abbildung 2.2 zeigt das typische Pool-based Sampling Szenario, bei dem die unbestimmten Eingabedaten durch einen Pool U und die annotierten Trainingsdaten den Pool L dargestellt werden.

2.2.2 Auswahlstrategien

Der Kern von Active Learning ist die Evaluation des Informationsgehalts der unbestimmten Datenpunkte. Ein einfacher und naiver Ansatz wäre es, die unbestimmten Datenpunkte auszuwählen, bei denen das Modell die falsche Klasse vorhersagt. Allerdings ist die Klasse unbekannt und der Lösungsansatz dadurch unmöglich. Es stellt sich also die Frage, wie die interessantesten Datenpunkte ermittelt werden können.

Uncertainty Sampling

Das verbreitetste Verfahren ist das Uncertainty Sampling. Potentielle Datenpunkte werden durch die Unsicherheit des Modells bestimmt. Die Unsicherheit wird aus der Wahrscheinlichkeitsverteilung der Ausgabe des Modells für jeden Datenpunkt abgeleitet [23].

Es gibt drei prominente Methoden diese zu bestimmen. In den folgenden Formen bezeichnet θ das Modell, y die Klassen und x die Eingabe.

Least Confident Je niedriger die größte Wahrscheinlichkeit in der kategorialen Verteilung der Ausgabe ist, desto größer ist die Unsicherheit des Modells.

$$\phi_{LC}(x) = 1 - \max_{y \in Y} P_\theta(y|x) \tag{2.1}$$

Minimum Margin Je dichter die beiden wahrscheinlichsten Klassen beieinander liegen, desto größer die Unsicherheit des Modells. Bei dieser Methode ist steht ein kleinerer Wert für größere Unsicherheit.

$$\begin{aligned} \phi_M(x) &= P_\theta(y_m|x) - P_\theta(y_n|x) \\ y_m &= \arg\max_{y \in Y} P_\theta(y|x) \\ y_n &= \arg\max_{y \in Y \setminus y_m} P_\theta(y|x) \end{aligned} \tag{2.2}$$

Shannon Entropy Für die Entropie [42] gilt, je ähnlicher die Wahrscheinlichkeiten der Klassen sind, desto größer die Unsicherheit des Modells. Diese Methode wird oft als Synonym für Uncertainty Sampling verstanden, da sie mit der Klassenanzahl skaliert und eine homogenere Gewichtung bei der Be1stimmung der Unsicherheit gegenüber den anderen beiden Formen aufweist.

$$\phi_{ENT}(x) = -\sum_{y \in Y} P_\theta(y|x) \log P_\theta(y|x) \tag{2.3}$$

Query By Committee

Die Samplingmethode Query By Committee (QBC) [41] nutzt ein Komitee aus Klassifikatoren. Das Komitee setzt sich aus verschiedenen Varianten des Klassifikators zusammen. Es können beispielsweise mehrere Modelle auf unterschiedlichen Bereichen der Eingabedaten trainiert werden, oder Varianten werden mit unterschiedlichen Parametern trainiert, sodass die resultierenden Klassifikatoren verschiedene Hypothesen gegenüber den Trainingsdaten vertreten (Abbildung 2.4). QBC klassifiziert die unbekannten Daten mit jedem Komiteemitglied und misst die Unstimmigkeit zwischen ihren Klassifizierungen,

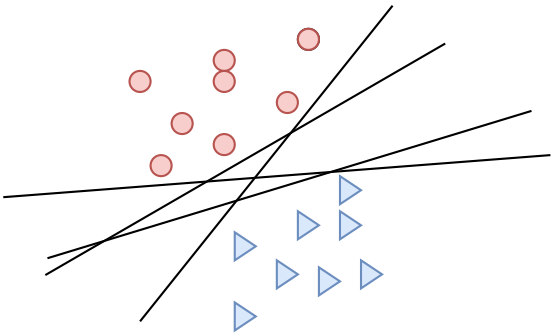

Abbildung 2.4: Entscheidungsgrenzen von Klassifikatorvarianten eines Komitees [40].

wodurch die Klassifizierungsvarianz approximiert wird. Die interessantesten Datenpunkte für die Befragung des Orakels sind jene, bei denen das Komitee sich am uneinigsten ist.

Für die Anwendung von QBC sind drei Fragestellungen interessant: Wie ist das Komitee zusammenstellt, wie wird die Uneinigkeit quantifiziert, um eine Strategie für die Auswahl der informativen Datenpunkte festzulegen, und wie werden die Antworten der einzelnen Modelle kombiniert, um ein robustes Ergebnis zu erhalten? In der Regel werden generische Ensemble-Lernalgorithmen für den Aufbau des Komitees verwendet. Query-by-bagging [2] oder Query-by-boosting [8] können verwendet werden, um schwache Klassifikatoren auf (gewichteten) zufälligen Variationen des Trainingsdatensatzes zu trainieren. Alternativ kann ein einzelnes Modell verwendet und viele Variationen davon abgeleitet werden, indem beispielsweise seine intrinsischen Parameter verändert werden [31].

Es gibt viele Heuristiken, um die Unstimmigkeit zwischen einem Klassifikatorensemble zu messen. Zu den den populärsten gehört die Vote Entropy [4] und die Kullback-Leibler Divergenz [20]. Mit [16] existiert eine moderne Methode, die Diversität und Repräsentativität miteinbezieht, um eine Vielfalt innerhalb zur Befragung ausgewählten Datenpunkte gewährleistet.

Im Vergleich zum Uncertainty Sampling bieten Ensemble-Samplingmethoden den Vorteil, durch verschiedene Ansichten der Eingabedaten, eine Unvoreingenommenheit gegenüber den Daten zu wahren. Während beim Uncertainty Sampling die Auswahl der Datenpunkte aus Sicht eines Modells getroffen wird, kann es vorkommen, dass dadurch informative Datenpunkte übersehen werden, die nicht im sichtbaren Bereich des Modells liegen.

Coreset

Eine weitere Methode ist der Coreset Ansatz [39]. Diese Methode gewichtet die Diversität der Trainingsdaten und nutzt keine Anhaltspunkte über die Unsicherheit des Modells. Es wird angenommen, dass der gesamte Datensatz durch eine Teilmenge, dem Coreset, repräsentiert werden kann, sodass zwischen einem Modell, dass auf dem gesamten Datensatz trainiert wird, und einem, das nur den Coreset kennt, kein Unterschied festgestellt werden kann.

Der Grundgedanke dabei ist, dass die Trainingsdaten die Vielfalt des Datensatzes annehmen sollen. Um die nicht annotierten Datenpunkte auszumachen, die in der Trainingsmenge noch nicht gut repräsentiert sind, wird in jedem Schritt das Coreset gesucht. Die Datenpunkte werden ausgewählt, die, wenn sie zum Trainingsset hinzugefügt würden, die Distanz zwischen den Datenpunkten aus der nicht annotierten Menge und denen aus dem Trainigset minimieren.

Die ideale Teilmenge zu finden ist NP-schwer [39], weshalb eine Greedy-Approximation angewendet wird, die nach der größten Distanz eines nicht annotierten Datenpunkts zu der Trainingsmenge sucht und diese Datenpunkte auswählt. Die Distanz wird in der Regel durch die Kosinusähnlichkeit der Ausgabevektoren des Modells ermittelt.

Monte Carlo Dropout

Der Monte Carlo (MC) Dropout Ansatz bietet ebenfalls einen Weg die Unsicherheit eines Modells zu bestimmen, weicht jedoch in der theoretischen Herleitung von dem Uncertainty Sampling ab. Die Bestimmung der Unsicherheit, wie sie beispielsweise das Uncertainty Sampling macht, also aus der Softmax-Ausgabe am Ende eines Modells, enthält kein verlässliches Maß über die Zuversicht des Modells, da die Vorhersagewahrscheinlichkeiten den bekannten Klassen gelten und keine Aussage getroffen wird, mit welcher Sicherheit diese das Modelle die Ausgabe getroffen hat [9].

Die Idee bei MC-Dropout leitet sich aus dem Ansatz der Bayesschen neuronalen Netze ab. Bayessche neuronale Netze stellen ihre Modellparameter durch Wahrscheinlichkeitsverteilungen dar [30]. Dies wirkt einerseits dem Overfitting beim Training entgegen, andererseits kann zu einem beliebigen Datenpunkt eine Wahrscheinlichkeitsverteilung über die bekannten Klassen ermittelt werden. Das Modell zieht, während der Vorhersage für eine Eingabe, aus der Wahrscheinlichkeitsverteilung seiner Parameter diskrete Werte für

die Berechnung der Klasse. Bei wiederholter Ausführung für dieselbe Eingabe können dadurch verschiedene Ausgaben entstehen. Betrachtet man die Ausgaben über mehrere Ausführungen, lässt sich eine Wahrscheinlichkeitsverteilung der Klassen erstellen. Tritt eine Klasse besonders oft auf, kann angenommen werden, dass sich das Modell in seiner Vorhersage sicher ist. Wenn sich die Ausgabe oft ändert ist die Unsicherheit für diesen Datenpunkt hoch.

Die Wahrscheinlichkeitstheorie von Bayes bietet mathematisch fundierte Instrumente, um ein Maß für die Modellunsicherheiten zu erhalten, allerdings sind sie in der Regel mit hohen Rechenaufwand verbunden. Sie skalieren sehr schlecht für Modelle mit vielen Parametern und sind aufgrund des enormen Rechenaufwands und Speicherbedarf außerhalb der Theorie für diese nicht praktikabel. Um dennoch die Vorteile zu nutzen, kann der Ansatz durch aktiven Dropout [45] während der Vorhersage bei Neuronalen Netzen approximiert werden, dem sogenannten MC-Dropout [9]. In der Praxis ist dies gleichbedeutend mit der Durchführung von n stochastischen Vorwärtsdurchläufen durch das Netz und der Mittelwertbildung der Ergebnisse.

Discriminative Active Learning

Discriminative Active Learning (DAL) ist eine Sampling Methode, bei der die Auswahl interessanter Daten in ein binäres Klassifizierungsproblem überführt wird [13]. Ähnlich dem Coreset Ansatz wird versucht die reale Verteilung des Datensatzes durch einen Bruchteile der Datenpunkte zu lernen. Die Datenpunkte aus der nicht annotierten Menge werden auswählt, sodass sich die annotierte Menge, also die Trainingsdaten, so schlecht wie möglich von der nicht annotierten Menge unterscheiden lässt. Grundlegend wird vorausgesetzt, dass genug Eingabedaten vorhanden sind um die wahre Verteilung zu repräsentieren.

Ein Perzeptron-Modell wird auf den Ausgabevektoren des lernenden Modells, also den erlernten Darstellungen der Eingabedaten, trainiert. Die eigentlichen Zielklassen der annotierten Daten werden ignoriert, stattdessen wird das binäre Problem zwischen dem Ursprung der Daten aufgestellt, gehört der Datenpunkt zu den bereits annotierten Daten oder zu der nicht annotierten Menge.

Die Datenpunkte aus der nicht annotierten Menge, für die das Modell die höchste Zugehörigkeit zu dieser vorhersagt, werden zur Bestimmung der eigentlichen Klasse an das Orakel übergeben.

2.2.3 Herausforderungen bei Neuronalen Netzen

Active Learning ist für viele ältere und überwiegend stochastischen Modelle, wie Support Vector Machines, Lineare Regression oder Markov Modelle weitreichend untersucht [40]. Viele Auswahlalgorithmen haben dort ihren Ursprung und sind für Szenarien entworfen, bei denen der informativste Datenpunkt gesucht wird, um diesem dem annotierten Pool hinzuzufügen und daraufhin ein neues Modell zu trainieren. Weder die Suche nach einem Datenpunkt, noch das Training des neuen Modelle ist in diesem Szenario besonders rechenintensiv, sodass die Active Learning Iteration in kurzer Zeit durchlaufen werden kann. Ersetzt man jedoch das Modell durch ein modernes neuronales Netz, wird schnell klar, dass die ursprüngliche Idee für den Active Learning Kreislauf nicht ohne weiteres anwendbar ist.

Neuronale Netze setzen vielen Trainingsdaten voraus, wodurch einerseits die Suche nach informativen Daten und andererseits das Training wesentlich rechenintensiver werden, was zur Folge hat, das das Orakel nur sporadisch befragt wird und ein großer Teil der Zeit von der Berechnung eingenommen wird. Um nicht nach jedem Datenpunkt die Kosten des Modelltrainings zu haben, ist es sinnvoll in einer Iteration mehrere Datenpunkte auszuwählen. Da Active Learning Szenarien oft den Menschen als Orakel einbeziehen, der diese Datenpunkte bestimmen soll, muss ein Grad gefunden werden, der einerseits die klassenbestimmende Seite, andererseits die Größe des Trainingssets, die sich auf die Konvergenz des Modelltrainigs auswirkt, berücksichtigt.

Batch Awareness

Neben der reinen Anzahl der Datenpunkte, die pro Iteration gefunden wird, ist es wichtig diese mit Sorgfalt auszuwählen. Ein naiver Ansatz wäre das Uncertainty Sampling zu nehmen, und statt wie üblich den unsichersten Punkt, die oberen n unsichersten Punkte auszuwählen. Ist das Modell bei einer Klasse unsicher, ist es wahrscheinlich, dass diese Auswahl redundante Datenpunkte umfasst, da das Modell ähnliche Daten als ähnlich unsicher ansieht.

Die Herausforderung ist es eine informative Gruppe an Datenpunkte zu finden, anstatt einzelner interessanter Datenpunkten, die als Kollektiv keinen großen Informationsgewinn haben. Dies wird als batch-aware oder batch-mode Active Learning [40, 12] bezeichnet. Die Aufgabe des Auswahlalgorithmuses für neuronale Netze ist es repräsentative und

diverse Mengen in jeder Iteration zu finden. Als repräsentativ gelten Datenpunkte die möglichst genau die Verteilung der gesamten Daten abbilden. Divers hingegen bedeutet, dass die Elemente in der Auswahl möglichst wenig redundant sind. Die Coreset Methode ist ein Beispiel für eine diverse Ausahlmethode, da bei jeder Iteration n verschiedene Datenpunkte ausgewählt werden, die nicht gut im Trainingsset repräsentiert sind und nicht im gleichen Bereich liegen.

Unsicherheit des Modells

Neuronale Netze für Klassifizierung und Regression haben kein reales Maß für ihre Unsicherheit in ihren Hervorsagen [39]. Dies wäre nicht nur für Active Learning, sondern auch für die meisten Gebiete in denen die Modelle eingesetzt werden von großem Wert [19, 11]. Uncertainty Sampling wird in der Literatur fälschlicherweise oft als Modellunsicherheit beschrieben, stellt in Wirklichkeit jedoch kein solches Maß dar.

In der Praxis wird die Unsicherheit eines Modells durch großzügige Datensätze komensiert, indem sämtliche Eingaben im Vorfeld abgedeckt werden. Dadurch, dass das Modell jede Möglichkeit bereits gesehen kann, kann angenommen werden, dass eine grundlegende Gewissheit in den Vorhersagen hat.

Es gibt jedoch auch Ansätze um die Unsicherheit eines Modell einzufangen, indem das neuronale Netz mit der Bayessche Statistik verknüpft wird [10, 9].

Inkompatibilität mit Auswahlstrategien

Manche Auswahlstrategien sind nicht auf vortrainierte Modelle anwendbar. Existiert ein vortrainierter Initialzustand eines Modells, wie beispielsweise bei BERT, ist die Architektur für diese Schichten vorgegeben und kann nicht nachträglich durch Schichten ausgetauscht werden, die für die Methode essentiell sind, wie jene die Wahrscheinlichkeitsverteilungen statt diskreter Werte annehmen.

Ein weiter Punkt bei neuronalen Netzen ist der Rechenaufwand im Vergleich zu älteren Active Learning Klassifikatoren. Während bei QBC ein Komitee aus den selben Modellen gebildet wird, ist dies für große neuronale Netze, wie BERT, nicht in annehmbarer Zeit umsetzbar.

Es gibt jedoch auch Ansätze wie DAL, die agnostisch vom Modell arbeiten und ein kleines eigenes Modell für die Bestimmung der Datenpunkte pflegen. Dieser Ansatz wird ebenfalls von [7] verfolgt und wird in dieser Arbeit eingesetzt.

2.3 Transformer

Mit der Vorstellung der Transformer Architektur von Vaswani u. a. [46] wurde eine neue Gruppe von neuronalen Netzen vorgestellt, die für den Umgang mit sequentiellen Daten ausgelegt ist und das Problem der maschinellen Sprachübersetzung (sequence to sequence) lösen soll. Die Architektur gehört zu den leistungsfähigsten auf diesem Gebiet und hat die etablierten Recurrent neural networks (RNNs) und Long Short-Term Memory (LSTM), die bisher für die Sprachübersetzung eingesetzt wurden, abgelöst.

Für die Übersetzung von Sprache kommen in der Regel Encoder- und Decoder-Einheiten zum Einsatz, die eine Eingabesequenz in einen Vektor kodieren, der allgemein als Sentence Embedding bezeichnet wird und aus diesem wiederum eine Ausgabesequenz mit dem Vokabular einer anderen Sprache generieren [3]. Die Idee dabei ist, dass der Sentence Embedding Vektor des Encoders die wahre Bedeutung des Satzes repräsentiert, losgelöst von dem ursprünglichen Vokabular. RNNs haben einen hohen Zeitaufwand, da Wörter aus der Eingabesequenz sequentiell über mehrere Zeitschritte, Wort für Wort verarbeitet werden. Hinzu kommen Schwächen bei der Erfassung der wahre Bedeutung der Worte, da sie den Kontext der Worte entweder nur von links nach rechts, oder bei bidirektionalen Netzen, auch von rechts nach links, separat lernen und anschließend die Ausgabe konkatenieren, der Kontext kann dadurch bestenfalls aus einer angenäherten Bidirektionalität abgeleitet werden. Da die Elemente einer Sequenz nacheinander verarbeitet werden, ist bei langen Sequenzen der Effekt nicht ungewöhnlich, dass die Information der Anfangselemnte der Sequenz mit der Sequenzlänge verschwinden und nur lokale Kontextbeziehungen erfasst werden.

Hier setzt die Transformer Architektur an, die auf eine simultane Verarbeitung der Eingabesequenz setzt und dadurch nicht nur schneller Ergebnisse berechnet, sondern durch die gleichzeitige Verarbeitung auch den Kontext aus beiden Richtungen im selben Verarbeitungsschritt der Sequenz betrachtet. Das führt zu einer Verbesserung beim Lernen des Kontextes, der eine Schlüsselrolle bei Sprachmodellen spielt, da Abhängigkeiten über lange Distanzen in der Sequenz genauso effektiv gelernt werden, wie die lokale Abhängigkeiten [46].

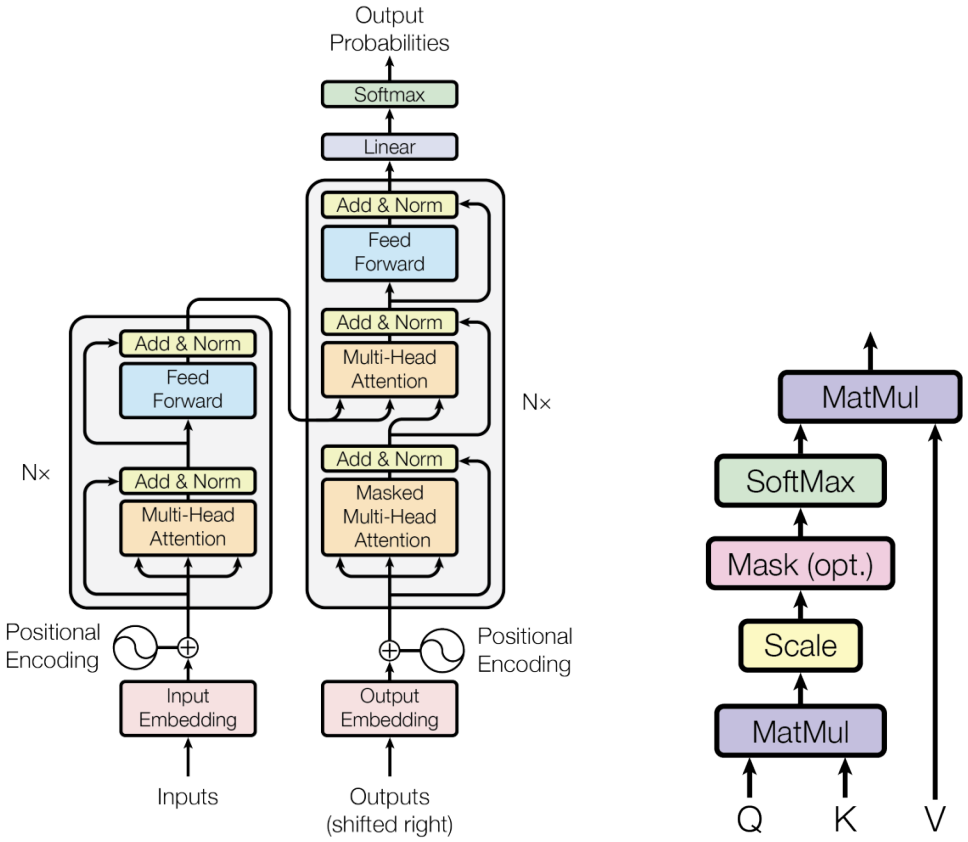

Abbildung 2.5: Transformer Architektur (a) und Blockansicht der Attention Funktion (b) aus der die Multi-Head Attention Blöcke bestehen [46].

Transformer bauen auf dem Aufmerksamkeitsmachanismus auf, der allgemein als Attention bezeichnet wird und verzichten auf die lineare rekurrente Struktur der älteren Modelle. Der Aufmerksamkeitsmachanismus ist die Kernkomponente der Transformer Modelle. Das Ziel ist es eine Sicht der Eingabedaten zu erhalten, bei der wichtige Elemente eine größere Beachtung erhalten. Eine Aufmerksamkeitsfunktion kann als Projektion von drei Vektoren auf eine Ausgabe beschrieben werden, wobei die drei Vektoren als query Q, keys K, und values V bezeichnet sind.

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{2.4}$$

Diese haben, je nachdem welches Problem das Modell löst, verschiedene Bedeutungen. Es existiert jedoch eine generelle Analogie zu Information Retrieval Systemen, bei denen eine Suchanfrage (query), gegen Metadaten (keys) abgeglichen wird und passende Ergebnisse (values) zurückgegeben werden. Die Softmax Funktion über Q und K ergibt eine Gewichtung für die Aufmerksamkeit der einzelnen Elemente in V. Tritt der Fall ein, bei dem alle Vektoren aus der selben Eingabesequenz berechnet werden, nennt man den Mechanismus Self-Attention.

Die Transformer Architektur setzt sich aus einem Encoder (Abbildung 2.5, linker Block) und einem Decoder (Abbildung 2.5, rechter Block) zusammen. Die Aufgaben sind aus konzeptioneller Sicht klar getrennt und lassen sich folgendermaßen formulieren: Der Encoder lernt das Vokabular und die Grammatik einer Sprache, sowie die Verwendung der Worte in ihrem Kontext. Der Decoder lernt die Projektion des Vokabulars der Quellsprache des Encoders auf eine Zielsprache. Die vom Encoder für jedes Wort berechneten Vektoren werden als Word Embedding bezeichnet und enthalten die Bedeutung des Wortes. Je ähnlicher sich zwei Wörter hinsichtlich ihres Kontextes sind, desto näher liegen ihre Vektoren zusammen. Die Vektoren werden durch den Decoder konsumiert und dienen als Grundlage für die Übersetzung in die Zielsprache. Beide Einheiten haben ein eigenes intrinsisches Verständnis davon, was Sprache ist und funktionieren dadurch auch getrennt voneinander. Während Transformer für die Übersetzung von Sprache ausgelegt sind, lassen sich die beiden Einheiten jeweils auf viele verschiedene Aufgaben aus dem Natural Language Processing Feld übertragen, wie Stimmungsanalysen, Textzusammenfassungen, oder die Beantwortung von Fragen. Das populäre Modell GPT-1 ist eine Aneinanderreihung von 12 Decoder-Einheiten aus der Transformer Architektur [35], während BERT eine Aneinanderreihung von 12 Encoder-Einheiten derselben Architektur ist [6].

2.3.1 BERT

Der Name entstammt dem Aufbau des Modells: Bidirectional Encoder Representations from Transformers. BERT verkörpert ein Sprachmodell, dass Eingaben in ihre Vektorrepräsentation kodiert, die einen kontextuellen Zusammenhang im Rahmen der zuvor gelernten Aufgabe und Sprache haben.

Das Training von BERT besteht aus zwei Phasen, dem Pretraining und dem Finetuning. Das Pretraining setzt sich aus zwei Aufgaben zusammen, die das Verständnis der Sprache

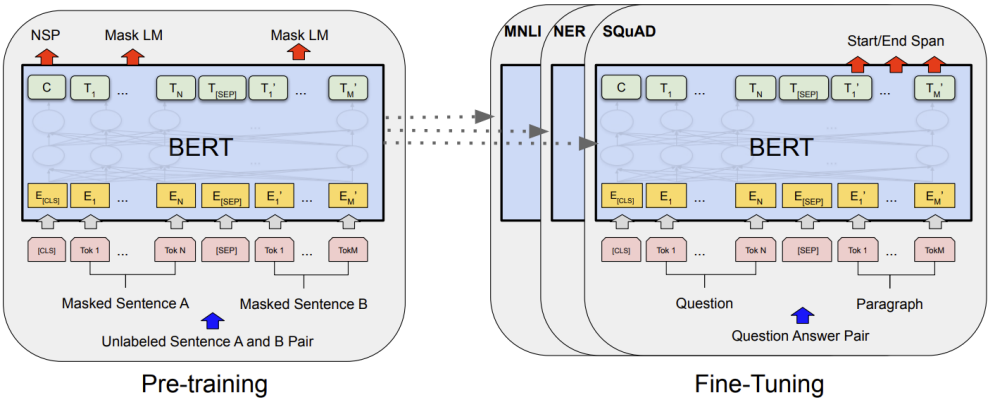

Abbildung 2.6: Transfer Learning bei BERT [6].

stützen. Die erste Aufgabe ist das Masked Language Modeling (MLM), bei dem es gilt fehlenden Worte in einem lückenhaften Eingabetext richtig vorherzusagen.

Das impliziert ein bidirektionales Verständnis des Kontextes des Wortes, das für die Wortlücke gesucht wird. Die zweite Aufgabe ist Next Sentence Prediction (NSP), bei der gelernt wird, welcher von zwei Sätzen vor dem anderen stehen sollte. Dies Hilft Abhängigkeiten zwischen Sätzen zu erkennen und ist wichtig für Aufgaben, wie beispielsweise das Beantworten von Fragen, bei der die oridnale Struktur wichtig ist. Beide Aufgaben werden simultan trainiert. In Abbildung 2.6 sind die beiden Aufgaben skizziert. Während die Ausgabe C für die Antwort auf die NSP-Aufgabe steht, wird für das MLM Problem eine vollverknüpfte Softmax-Schicht eingesetzt, Aufschluss über die vorhergesagten Wörter für die Lücken gibt. Abbildung 2.6 zeigt ebenfalls das Zusammenspiel des Pretrainings mit dem Finetuning eines BERT-Modells für spezifische Zielaufgaben. Die Parameter aus dem Pretraining dienen als Initialisierung für den Abstimmungsschritt, welche beide die selbe Architektur benutzen. Die spezifische Aufgabe wird durch Adaption der Ausgabeschicht, zum Beispiel durch einen vollverknüpfte Schicht, wie beim MLM Training, erreicht und kann dadurch für sämtliche NLP Aufgaben eingesetzt werden.

2.3.2 Varianten von BERT

Obwohl BERT neue Benchmarks setzt, gibt es bereits Weiterentwicklungen, die die immense Rechen- und Speicherkapazität, die BERT voraussetzt, reduzieren, und die Performance und das Pretraining optimieren. Zwei Varianten, die in Abschnitt 4 verwendet werden, sind im Folgenden beschrieben.

ALBERT

Eine Weiterentwicklung von BERT ist A Lite BERT (ALBERT) [21]. ALBERT nutzt die selbe Architektur wie BERT, unterscheidet sich jedoch in drei Punkten, die dazu führen, dass wesentlich weniger Parameter für das Modell benötigt werden. Obwohl die Speicherauslastung reduziert ist, bleibt der Rechenaufwand fast gleich, da die selben Rechenschritte in gleicher Häufigkeit, wie bei BERT, also für alle zwölf Encoder-Blöcke anfallen.

Die Autoren argumentieren, dass die WordPiece Embeddings des ersten Layers dazu bestimmt sind die kontextunabhängige Repräsentation der Worte zu lernen, während das Embedding der Hidden-Layer dazu bestimmt ist die kontextabhängige Repräsentation zu lernen.

ALBERT faktorisiert die Parameter der Word Embeddings, indem sie in zwei kleinere Matrizen zerlegt werden. Anstatt die One-Hot-Vektoren der Eingabesequenz direkt auf den Hidden-Layer zu projizieren, werden sie zunächst einen niedrigdimensionalen Einbettungsraum projiziert, der dann wiederum auf den Hidden-Layer projiziert wird.

Der nächste wesentliche Unterschied besteht darin, dass zwischen den Encoder-Blöcken die Layer-Parameter für jeden ähnlichen Block gemeinsam genutzt werden. Alle zwölf Blöcke nutzen also die selben Parameter, was in einer drastischen Verminderung der Parameter im Speicher sorgt. Das Teilen bewirkt zudem eine Stabilisierung des Neuronalen Netzes.

Der letzte Unterschied betrifft das Pretraning. Das NSP Problem wird in ein Sentence Order Prediction (SOP) Problem überführt, das satzübergreifende Kohärenz mit berücksichtigt. Aus Sicht des Pretrainings werden im gewissen Maße die Fragen aufgeworfen: Worum geht es in diesem Satz und wie hängen die Sätze zusammen?

DistilBERT

Distilled version of BERT (DistilBERT) ist eine destillierte Version von BERT [36], die dem Trend von immer größeren und rechenintensiveren Sprachmodellen entgegenwirkt.

Durch Wissensdestillation (knowledge distillation), eine Komprimierungstechnik für neuronale Netze [14], wird ein kleines Modell (Schüler) trainiert, um das Verhalten eines größeren Modells (Lehrer) zu reproduzieren. DistilBERT ist nach diesem Prinzip von

BERT abgeleitet, sodass es die gesamte Ausgangsverteilung des Lehrernetzes (sein Wissen) nachahmt.

Beim Training wird die Kreuzentropie-Verlustfunktion nicht wie üblich über die Klassen berechnet, sondern bezieht die die Ausgabe-Wahrscheinlichkeiten des Lehrers mit ein.

DistilBERT besitzt nur noch die Hälfte der Layer von BERT ohne große Einschnitte in der Performance. Anders als bei ALBERT berechnet DistilBERT wesentlich schneller ein Ergebnis, da viele Operationen durch die Abwesenheit von Layern wegfallen.

RoBERTa

Eine Weiterentwicklung von BERT ist Robustly optimized BERT approach (RoBERTa). Die Autoren argumentieren, dass BERT untertrainiert ist und viel Potential für Verbesserung bietet, wenn das Pretraining ausbaut wird [26].

RoBERTa unterscheidet sich gegenüber BERT im wesentlichen durch vier Punkte im Pretraining. RoBERTa ist mit einem immensen Trainingskorpus von 160GB gegenüber 16GB und auf längeren Sequenzen trainiert worden. Zusätzlich ist die NSP Zielsetzung auf zwei Wegen verändert, einerseits wird der Fehler beim Training nicht mehr berücksichtigt, andererseits wird eine, als Full-Sentence bezeichnete Methode angewandt, bei der die Eingabesequenzen solange mit den nächsten Sätzen aufgefüllt, bis die maximale Grenze von 512 Tokens erreicht sind. Für MLM wird eine dynamische Markierungen für die Sequenzen verwendet.

Während ALBERT und DistilBERT die Hardware-Voraussetzungen mindern, setzt RoBERTa auf die bestmögliche Performance von BERT.

3 Vergleichbare Arbeiten

Active Learning hat bisher seine Stärken vorwiegend bei klassischen Algorithmen für maschinelles Lernen gezeigt [40]. Die Adaption des Prinzips für neuronale Netze ist nicht trivial und hält einige Herausforderungen bereit. Obwohl dem Thema in der Vergangenheit keine große Aufmerksamkeit zuteil geworden ist, gewinnen die Forschungsbemühungen in diese Richtung an Fahrt und verbinden Active Learning mit Deep Learning für Klassifikationsaufgaben, wie der Überblick in [37] zeigt.

Bevor BERT an Popularität gewann, lag der Fokus von Active Learning mit neuronalen Netzen vorwiegend auf Convolutional neural networks (CNNs). In einer der ersten Untersuchungen, die Active Learning für CNNs verwenden, werden Auswahlmethoden vorschlagen, die auf Uncertainty Sampling basieren [47]. Andere Untersuchungen zeigen hingegen, dass Methoden auf Grundlage der Bayes'schen Unsicherheit vorteilhafter sein können [15, 10, 17], oder definieren die Auswahlfunktion durch ein Representativitätskriterium aus Sicht der Eingabedaten, wodurch Datenpunkte aus einem Vektorraum ausgewählt werden, sodass der komplette Datensatz geometrisch repräsentiert wird [39].

Neuere Veröffentlichungen haben den Nutzen von Active Learning für Textklassifikation bei neuronalen Netzen demonstriert [52, 44, 34, 27], nehmen jedoch keinen Bezug zu BERT-Modellen.

In [50] wird ein Ensembleansatz für BERT untersucht, bei dem verschiedene Active Learning Metriken verknüpft werden um Absichten zu klassifizieren (intent classification). Die Untersuchung geht jedoch nicht auf kleinere und unbalancierte Datensätze ein.

In [43] und [25] werden zwei besondere Varianten von BERT, BioBERT und BERT-CRF, adressiert, und für Named Entity Recognition (NER) und Sequenz-Tagging Aufgaben unter einer kleinen Sammlung von Active Learning Strategien untersucht. Es wird jedoch keine systematische Untersuchung für die Auswahlmethoden für BERT unter verschiedenen Einstellungen in verschiedenen Domänen durchgeführt.

In [38] werden Uncertainty Sampling Methoden in Bezug auf die Feinabstimmung von BERT-Modellen untersucht. Es wird eine ausführliche Evaluierung anhand von fünf Textklassifizierungs Benchmarks durchgeführt. Die Untersuchung umfasst sowohl binäre Probleme als auch Probleme mit mehren Klassen, beschränkt sich in der Untersuchung jedoch auf eine Auswahlstrategie.

In [28] werden verschiedene Textrepräsentation durch BERT ähnliche Modelle ausführlich im Active Learning Kontext untersucht. Allerdings werden Support Vector Machines als Klassifikatoren eingesetzt und der Fokus liegt auf einem binären Problem.

Eine aktuelle Untersuchung in [7] beschriebt gängige Active Learning Methoden im Zusammenspiel mit BERT und ihren Vorteilen gegenüber der zufälligen Auswahl. Obwohl BERT-Modelle mit geringeren Datensets für Downstream Aufgaben zurechtkommen, reduziert Active Learning den Bedarf an Annotationen und erleichtert dadurch die praktische Nutzung der Technologie. Gegenstand der Untersuchung von Dor u. a. [7] ist ein binäres Klassifikationsproblem, bei dem möglichst reale Szenarien nachgebildet sind, in denen die Ziel klasse unterrepräsentiert ist und die Ausgangsdaten der Active Learning Algorithmen geringfügig fehlerhafte Klassen besitzen. Die Autoren weisen in Ihrem Ausblick auf die Frage hin, wie sich Active Learning in Verbindung mit BERT für Multi-Klassen-Probleme schlägt.

Keine der bisherigen Arbeiten, mit Ausnahme von Dor u. a. [7], gehen systematisch auf unterschiedlich zusammengesetzte Startmengen für das Active Learning Szenario ein, oder untersuchen die praktische Annotation durch Benutzer.

4 Untersuchung

Dieses Kapitel beschreibt das Vorgehen und den Aufbau der Untersuchung im Hinblick auf die in Abschnitt 1.2 aufgestellten Forschungsfragen.

4.1 Datensätze

Um einen möglichst großen Querschnitt an Daten und Bereichen für das Active Learning abzudecken, werden folgende Datensätze für die Untersuchung herangezogen, die sich in ihrer Verteilung, der Klassenanzahl, der Sequenzlänge und in der Anzahl der Trainigsdaten voneinander unterscheiden:

IMDB Das Large Movie Review Dataset [29] ist eine Sammlung aus englischen Filmrezensionen aus der Internet Movie Database[1]. Der Datensatz wird in der Textklassifikation oft für die Stimmungsanalyse eingesetzt. Er besteht aus 50.000 Benutzerrezensionen, von denen jeweils 25.000 positiv und negativ sind. Der Datensatz besitzt zwei Klassen die gleich verteilt sind.

AG News AG's corpus of news articles[2] ist ein Datensatz, der sich aus über einer Million annotierter englischer Nachrichtenartikel aus mehr als 2000 Nachrichtenquellen zusammensetzt. Der in dieser Arbeit genutzte Datensatz basiert auf der Version von [51], einer Teilmenge des Originalen mit vier gleichverte ileten Nachrichtenkategorien.

TREC Experimental Data for Question Classification [24] der Text Retrieval Conference[3] setzt sich aus 5.952 englischen Fragen zusammen. Den Fragen ist jeweils eine grobe und eine genaue Klasse zugewiesen, sodass sich zwei Datensätze ergeben, je nachdem welche Klasseneinteilung herangezogen wird. Es gibt sechs grobe und 47 genaue

[1] https://www.imdb.com/
[2] http://groups.di.unipi.it/~gulli/AG_corpus_of_news_articles.html
[3] https://trec.nist.gov/

Datensatz	Klassen	Train	Test	Dev	∅ Sequenzlänge	Verteilung
IMDB	2	15000	3000	3000	266	50%
AG_NEWS	4	15000	3000	3000	50	25%
TREC	6	4900	500	552	11	17% (2%)[1]
ISEAR	7	5366	1534	766	25	14%
DBPEDIA	14	15000	3000	3000	62	7%
IMDB_IMB	2	12000	2400	2400	266	Siehe 4.2
AG_NEWS_IMB	4	12000	2400	2400	50	Siehe 4.2
TREC_FINE	47	4900	500	552	11	Siehe 4.2
ISEAR_IMB	7	4024	1150	574	25	Siehe 4.2
DBPEDIA_IMB	14	12000	2400	2400	62	Siehe 4.2

Tabelle 4.1: Übersicht der Datensätze und ihrer Eigenschaften.

Klassen, die die Art der Frage beschreiben, zum Beispiel ob eine Nummer oder eine Abkürzung erfragt wird. Die Verteilung der groben Klassen ist mit Ausnahme einer Klasse ausgeglichen, wohingegen die feinen Klassen verschieden stark vertreten sind und der Datensatz unausgeglichen ist.

DBPedia Der DBPedia ontology Datensatz [22] wurde durch Auswahl von 14 sich nicht überschneidenden Klassen aus DBpedia 2014[2] erstellt. Die Klassen sind gleich verteilt und beschreiben das Thema des Artikels. Die Texte sind auf Englisch verfasst. Der Datensatz wurde durch zufälliges Auswählen von 40.000 Trainings- und 5.000 Testdaten je Klasse erstellt.

ISEAR Der International Survey On Emotion Antecedents And Reactions [5] Datensatz enthält emotionale Aussagen, die von 1096 Teilnehmern mit unterschiedlichem kulturellem Hintergrund stammen. Den Aussagen sind sieben Emotionen zugeordnet. Die Klassen des Datensatzes sind gleich verteilt.

Eine Übersicht über die Datensätze ist in Tabelle 4.1 zu finden. Für die Untersuchung wird jeder Datensatz in Training-, Test- und Evaluation-set aufgeteilet. Große Datensätze werden analog zu der Untersuchung in [7] verkleinert, sodass die maximale Größe von 15.000/3.000/3.000 für Trainings-, Test- und Evaluierungsmenge eingehalten wird. Die Datenpunkte werden zufällig aus dem gesamten Datensatz ausgewählt und sind über

[1] Eine von sechs Klassen ist mit 2% vertreten, die anderen jeweils mit 17%.
[2] http://downloads.dbpedia.org/wiki-archive/data-set-2014.html

Klasse	IMDB_IMB	AG_NEWS_IMB	ISEAR_IMB	DBPEDIA_IMB	TREC
0	0.87	0.06	0.03	0.02	0.03
1	0.13	0.13	0.03	0.02	0.03
2	-	0.26	0.06	0.02	0.06
3	-	0.53	0.06	0.04	0.06
4	-	-	0.26	0.04	0.26
5	-	-	0.26	0.04	0.53
6	-	-	0.53	0.06	-
7	-	-	-	0.06	-
8	-	-	-	0.06	-
9	-	-	-	0.06	-
10	-	-	-	0.13	-
11	-	-	-	0.13	-
12	-	-	-	0.13	-
13	-	-	-	0.13	-

Tabelle 4.2: Klassenverteilung der unausgeglichenen Datensätze und der Startmenge des unausgeglichenen Szenarios.

einen *Seed* reproduzierbar. Sofern die ursprünglichen Datensätze eine Aufteilung in die verschiedenen Mengen haben, wird diese berücksichtigt. Die Datensätze in Tabelle 4.1 und 4.2 sind mit dem $Seed = 5$ generiert, der als Ausgang für die Untersuchung gilt.

Der Großteil der Datensätze hat eine ausgewogene Verteilung der Klassen und gilt somit als ausbalanciert. Ausbalancierte Datensätze spiegeln jedoch in der Praxis oft eine untergeordnete Rolle, weshalb zu den vorgestellten Datensätze jeweils eine künstliche unausgeglichene Version erstellt wird (Tabelle 4.1, untere Hälfte), mit Ausnahme von TREC, da dort die feinen Klassen unausbalanciert sind. Die unausgeglichenen Datensätze sind ihrem Original nachempfunden, jedoch ist ihre Klassenverteilung absichtlich verzerrt. Die Klassenverteilung der unausgeglichenen Datensatzversionen lassen sich in Tabelle 4.2 ablesen. Die Verteilung wurde anhand der Formel 4.1 für AG_NEWS_IMB bestimmt. n steht für die Anzahl der Klassen und k für den Klassenindex.

$$f(n,k) = \frac{1}{(2^n - 1)}k \tag{4.1}$$

Die Verteilungen der weiteren unausgeglichenen Datensätze basieren auf leichter Abwandlung der Formel. Tabelle 4.1 zeigt eine leichte Verminderung der Datenmengen in den Trainings-, Test- und Evaluierungsmengen. Dies resultiert aus der Beschaffenheit der originalen Datensätze, wenn die Verteilungsproportionen aus 4.2 nicht erfüllt werden können. Dies passiert, wenn eine Klasse häufiger vertreten sein soll, als sie ursprünglich im Datensatz enthalten ist. Durch eine verringerte Ausgangsmenge können die Proportionen eingehalten werden.

Sofern die Datensätze nicht aufbereitet sind, werden folgende Schritte zur Normalisierung der Texte angewendet:

- URLs werden entfernt.
- HTML-Tags werden entfernt.
- In eckigen Klammern eingeschlossene Wörter werden entfernt.
- Zeilenumbrüche, Sonder- und Satzzeichen werden durch den Regulären Ausdruck `[^a-zA-Z0-9_!.$~{}| n\]` entfernt.

4.2 Szenarien

Der Aufbau der Untersuchung entspricht dem pool-based Active Learning (Abschnitt 2.2.1). Diese Arbeit definiert fünf Szenarien, für jede Active Learning Strategie. Die Szenarien beeinflussen den Start des Active Learning Kreislaufs, indem sie die Eigenschaften der zu Beginn vorhandenen annotierten Daten festlegen. In [7] sind drei Szenarien definiert, die von idealen bis möglichst realistischen Bedingungen reichen. Für diese Arbeit sind folgende Szenarien definiert, die einen noch breiteren Einblick in den Start von Active Learning zeigen.

Ausgeglichen In dieser Ausgangssituation ist die Startmenge ausbalanciert. Jede Klasse ist gleich oft vertreten. Die Menge umfasst 100 Datenpunkte. Die Datenpunkte sind zufällig ausgewählt, bis sie eine gleiche Verteilung erreicht haben.

Unausgeglichen Hier ist die Startmenge nicht ausgeglichen. Es gibt ein Gefälle in der Verteilung der Klassen (Tabelle 4.2). Die Menge umfasst 200 Datenpunkte. Dies folgt der in [7] beschriebenen Anzahl, da weniger zu unbeständigen Trainigsdurchläufen führt. Die Datenpunkte sind innerhalb ihrer Gewichtung zufällig ausgewählt.

Klasse	IMDB	AG_NEWS	ISEAR	DBPEDIA	TREC
0	bad	president	angry	company	mean
1	good	team	disgusted	school	animal|dog
2	-	company	afraid	painter	abbreviation|stand.*for
3	-	software	guilty	baseball|hockey	who
4	-	-	joy	minister	many
5	-	-	sad	aircraft	where
6	-	-	ashamed	building	-
7	-	-	-	lake	-
8	-	-	-	village	-
9	-	-	-	snail	-
10	-	-	-	flowering	-
11	-	-	-	band	-
12	-	-	-	drama	-
13	-	-	-	book	-

Tabelle 4.3: Reguläre Ausdrücke für jeden Datensatz zur Bestimmung der Startmenge im Stichwort-Szenario.

Natürlich Die Verteilung der Klassen in der Startmenge entspricht der Verteilung des Datensatzes. Dieses Szenario ist eine Schnittmenge der beiden vorangegangenen Szenarien und stellt eine zufällig gezogene natürliche Menge als Startmenge da, die sich bei einem Datensatz ergeben würde, bei dem die reale Verteilung der Klassen nicht bekannt ist. Sofern der Datensatz ausbalanciert ist werden 100, ansonsten 200 Datenpunkte der Menge zugewiesen.

Fehlerhaft Dieses Szenario bildet schwach annotierte Klassen in der Startmenge ab. Ein künstliches Rauschen wird durch den Austausch der Klassen von 30% der Datenpunkte erzeugt. Dies bildet eine schnelle und kostengünstige Annotation der Startmenge ab, die als fehlerhaft angenommen werden kann. Ansonsten entspricht dieses Szenario dem natürlichen Szenario.

Stichwort Das Stichwort-Szenario stellt eine praxisnahe Annotation der Startmenge dar. Der Datensatz wird nach einem oder mehreren Stichwörtern durchsucht, woraufhin jedem Datenpunkt in der Suchergebnismenge die selbe Klasse zugewiesen wird, die Klasse die den Suchbegriffen entsprechen. Dadurch das Wörter nicht eindeutig zu einer Klasse gehören, sondern auch in Texten vorkommen, die eigentlich eine andere Klas-

se besitzen, beinhaltet diese Methode, ebenfalls schwach annotierte Datenpunkte, also teilweise fehlerhafte Klassenzuweisungen. Zusätzlich unterliegt die Startmenge einer gewissen Verzerrung, da nur Datenpunkte enthalten sind, die einen speziellen Suchbegriff enthalten. Bei diesem Szenario sind alle Klassen gleich oft vorhanden, die Startmenge ist also ausbalanciert, jedoch existieren zu der Klasse möglicherweise redundante Texte, da sie dem selben Suchbegriff gerecht werden. Die Startmenge umfasst 200 Datenpunkte. Tabelle 4.3 listet die Suchbegriffe für jeden Datensatz auf.

4.3 Auswahlstrategien

Für die Untersuchungen wird eine Auswahl moderner Active Learning Strategien betrachtet, die Uncertainty- , Ensemble- und Diversity-Sampling Ansätze umfassen. Jede Methode wählt jede Iteration 50 Datenpunkte aus. Random Sampling dient als Baseline. Zusätzlich wird das Optimum jedes Datensatzes und Modell durch überwachtes Lernen des Datensatzes dargestellt. Die Auswahlstrategien sind in Abschnitt 2.2.2 ausführlicher beschrieben. Sie orientieren sich, bis auf Expected Model Change, das laut Autor sehr ineffizient ist[1] und dem praktischen Ansatz widerspricht, an den Auswahlstrategien aus [7]. Im folgenden sind die konkreten Methoden zusammengefasst:

Coreset Wählt die Instanzen aus, die den Datensatz am besten im gelernten Repräsentationsraum abdecken. Der Greedy-Algorithmus aus [39] wird verwendet.

DAL Discriminative Active Learning optimiert die Trainingsmenge, sodass der gesamten Datensatz bestmöglich repräsentiert wird. Die Methode aus [13] wird verwendet.

Ensemble Wählt Instanzen, ähnlich der Uncertainty Methode, aus. Statt der Vorhersage des Modells selbst, wird jedoch der Mittelwert eines Ensembles aus 10 leichtgewichtigeren Perceptron-Modellen gebildet. Die Instanzen mit der größten Entropie werden dabei ausgewählt. Ein Ensemble aus BERT-Modellen zu bilden stellte einen unrealistischen Rechenaufwand dar, weshalb hier auf kleinere Modelle zurückgeriffen wird. Die Perzeption-Modelle bekommen die CLS-Vektoren[2] des Hauptmodells der Trainingsmenge als Eingabe und werden mit den bekannten Klassen für die Trainingsdaten trainiert.

[1] https://github.com/IBM/low-resource-text-classification-framework/issues/2

[2] Spezieller BERT Token, der dem Eingabetext vorangestellt wird und dessen Ausgabe als Sentence Encoding für Klassifikationsaufgaben genutzt wird.

Dropout Bei Monte Carlo Dropout werden zehn Vorhersagen der selben Daten gemittelt um die interessantesten Datenpunkte daraufhin durch die Entropie zu bestimmen. Nach [9] liegt der optimale Bereich zwischen 30 und 100 Vorwärtsdurchläufen, hinsichtlich des Rechenaufwands, wird der Wert jedoch niedriger gewählt.

Uncertainty Wählt die Instanzen aus, für die das Modell die größte Entropie in der Vorhersage hat. Dies wird als Maß der Unsicherheit verwendet und folgt dem Prinzip aus [23].

4.4 Modelle

Die Erstellung und das Pretraining von Grund auf eines gut funktionierenden Transformer-Modells erfordert eine Kombination aus langer Trainingszeit, spezieller und teurer Hardware und einer enormen Menge an Trainingsdaten, weshalb die veröffentlichte votrainierten Modelle herangezogen werden, die sich einer breiten Beliebtheit erfreuen und ihre Leistung in den Benchmarks der dazugehörigen Literatur bestätigt haben. Aus den verschiedenen Konfigurationen wird für diese Untersuchung jeweils auf die Basis-Version der Modelle zurückgegriffen, die kaum Leistungseinbußen im Vergleich zu ihren voll ausgebauten Versionen haben. Folgende konkrete Modelle werden für die Textklassifikation herangezogen, Tabelle 4.4 listet ihre Eigenschaften auf:

- **BERT** bert-base-uncased[1]
- **ALBERT** albert-base-v1[2]
- **DistilBERT** distilbert-base-uncased[3]
- **RoBERTa** roberta-base[4]

Für die Klassifikation von Texten ist jedem vortrainierten Modell ein vollverknüpfter Klassifikations-Layer angehängt, dessen Ausgänge der Anzahl der Klassen des jeweiligen Datensatzes (Tabelle 4.1) entsprechen.

Die Modellperformance wird anhand des makro-gemittelten F1-Scores berechnet. Der F1-Score bildet ein gängiges Maß für Textkategorisierungsprobleme und verknüpft die Recall-

[1] https://huggingface.co/bert-base-uncased
[2] https://huggingface.co/albert-base-v1
[3] https://huggingface.co/distilbert-base-uncased
[4] https://huggingface.co/roberta-base

	ALBERT	BERT	DistilBERT	RoBERTa
Encoder-Blöcke	12	12	6	12
Attention-Heads	12	12	12	12
Embedding-Dimension	128	768	768	768
Hidden-Dimension	768			
Sequenzlänge	512	512	512	512
Parameter	11 Mio.	109 Mio.	66 Mio.	124 Mio.
Groß-/Kleinschreibung	-	-	-	✓
Vokabular	30 Tsd.	30 Tsd.	30 Tsd.	50 Tsd.

Tabelle 4.4: Kennzahlen von BERT und den Variationen.

und Precision-Metrik, wodurch ein aussagekräftigerer Wert ermittelt werden kann, der ungleichverteilte Klassen stärker berücksichtigt, als zum Beispiel die Accuracy-Metrik. Als Grundlage dient die Konfusionsmatrix. Sie beinhaltet als Elemente die Anzahl an korrekten positiven (True Positives TP), korrekten negativen (True Negatives TN), falschen positiven (False Positives FP) und falschen negativen (False Negatives FN) Vorhersagen. Durch die Konfusionsmatrix lassen sich pro Klasse ablesen, wie viele der Vorhersagen für die positive Klasse und wie viele der Vorhersagen für die negative Klasse falsch sind. Aus der Konfusionsmatrix leiten sich Precision und Recall folgendermaßen ab:

$$Precision = \frac{TP}{TP + FP} \tag{4.2}$$

$$Recall = \frac{TP}{TP + FN} \tag{4.3}$$

Precision gewichtet demnach den Anteil an richtig vorhergesagten positiven Ergebnissen (TP) bezogen auf die Gesamtheit aller als positiv vorhergesagten Ergebnisse, wohingegen Recall den Anteil der korrekt als positiv klassifizierten Ergebnisse (TP) bezogen auf die Gesamtheit der tatsächlich positiven Ergebnisse gewichtet. Der F1-Score ist das harmonische Mittel aus Precision und Recall und wird als zusammenfassende Metrik verwendet:

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{4.4}$$

Obwohl die Konfusionsmatrix für mehrere Klassen aufgestellt werden kann, lassen sich Recall, Precision und F1-Score nur für eine binäre Klassifikation berechnen. Für Anwedung des F1-Scores bei Multi-Klassen-Problemen wird eine Mittlung der F1-Scores von jeder Klasse vorgenommen. Dies beschreibt den makro-gemittelten F1-Score, der jeder Klasse die selbe Gewichtung verleiht, wodurch auch Klassen mit wenig Datenpunkten in der Performance des Modells berücksichtigt werden. Der Makro-F1-Score ist beschrieben durch:

$$F1_{Macro} = \frac{1}{N} \sum_{n=1}^{N} F1 - Score_i \tag{4.5}$$

N beschreibt die Anzahl der Klassen im Datensatz. Neben dem makro-gemittelten F1-Score gibt es weitere Arten eine Metrik für mehrere Klassen aus der Konfusionsmatrix abzuleiten. Da diese Arbeit eine breite Menge an Datensätzen abdeckt und keine explizite Klasse stärker oder geringer beachtet werden soll, bildet die Makro-Mittlung die Grundlage für die Untersuchung.

4.4.1 Batchanalyse

Die von den Active Learning Methoden ausgewählten Datenpunkte werden analog zu [7] relativ zueinander verglichen. Dazu werden zwei Metriken herangezogen, die Aufschluss über die Diversität und Repräsentativität der Menge geben. Nachdem die Active Learning Auswahlstrategien in der ersten Iteration, also im ersten Durchlauf, 50 Datenpunkte ausgewählt haben werden die Eigenschaften dieser Menge folgendermaßen analysiert.

Diversität

Eine Auswahl von diversen Datenpunkten wirkt einem Bias des Modells entgegen, weshalb sie allgemein als konstruktiver gegenüber eine redundanten Auswahl im Active Learning Kontext gesehen wird. Zum Messen der Diversität wird die in [53] aufgestellte und in [7] abgewandelte Formel verwendet:

$$Diversity(B) = \left(\frac{1}{|U|} \sum_{x_i \in U} \min_{x_j \in B} distance(x_i, x_j) \right)^{-1} \tag{4.6}$$

Die Formel berechnet die Diversität einer Menge an ausgewählten Datenpunkten. x_i und x_j bezeichnen die jeweilige Vektorrepräsentation (CLS-Token) eines Textes, die von einem Modell erhalten werden, das mit der annotierten Menge L trainiert wurde. x_i ist eine Element aus der nicht annotierten Menge U und x_j aus der zu annotierenden Menge B. *distance* beichnet die Euklidische-Distanz zwischen den beiden Vektoren. [53] folgend wird die Gleichung durch den K-means Algorithmus approximiert, da die Berechnung der optimalen Lösung NP-schwer ist.

Repräsentativität

Im Gegensatz zur Diversität bezieht sich das Representativitätsmaß auf die bekannte Tatsache, dass insbesondere das Uncertainty Sampling dazu neigt, Ausreißer-Datenpunkte auszuwählen, welche die gesamte Datenverteilung nicht hinreichend repräsentieren. Durch die in [54] vorgestellte K-Nearest-Neighbor-Density Metrik, in der die Dichte eines Datenpunktes durch die durchschnittliche Distanz zwischen ihm und seinen k ähnlichsten Nachbarn in der nicht annotierten Menge quantifiziert wird, lässt sich eine Aussage über die Repräsentativität dieses Datenpunktes treffen. Analog zur Diversitätsmetrik, werden die Vektorrepräsentationen der Texte verwendet, die der Encoder-Teil eines Modells, das mit der annotierten Menge L trainiert ist, durch Interferenz liefert. Ein Datenpunkt der einen hohen Dichtewert hat, ist weniger wahrscheinlich ein Ausreißer. Damit für eine ausgewählte Menge die Repräsentativität berechnet werden kann, wird analog zu [7] die Repräsentativität als eins durch den Durchschnitt der KNN-Dichte, die über die Euklidischen Distanz ermittelt wird, zwischen den Elementen definiert. Für die Untersuchung ist $k = 10$. Die Representativität setzt sich aus foldenen Formeln zusammen, bei der $S(x)$ die k nächst ähnlichen Datenpunkte der nicht annotierten Menge zu x sind:

$$\begin{aligned} Representativeness(B) &= \left(\frac{1}{|B|} \sum_{x_i \in B} DS(x_i) \right)^{-1} \\ DS(x) &= \frac{1}{k} \sum_{s_i \in S(x)} distance(x, s_i) \end{aligned} \tag{4.7}$$

4.5 Implementierung

Für die Untersuchung werden die Datensätze, Auswahlstrategien, Modelle und Szenarien kreuzweise verknüpft und in einzelnen Experimenten aufgesetzt.

4.5.1 Hardware

GPU	GPU RAM	CPU	RAM
Tesla V100	16 GB HBM2	Intel Xeon Gold 5115	96 GB
Tesla V100s	32 GB HBM2	Intel Xeon Gold 5115	96 GB

Tabelle 4.5: Hardwaredaten der Containerkonfigurationen im ICC Cluster.

Die Untersuchungen werden in der Informatik Compute Cloud[1] durchgeführt. Tabelle 4.5 listet die verfügbare Hardware im Kubernetes-Cluster auf, auf der die Berechnungen stattfinden. Jedes Active Learning Experiment erhält einen Container mit einer GPU, 5 Kernen und 10 Threads, die mit 2,4 GHZ takten.

4.5.2 Hyperparameter

Die allgemeinen Einstellungen und Modellhyperparameter für die Experimente folgen der Untersuchung aus [7] und setzen sich wie folgt zusammen:

- **Iterationen = 5**, Anzahl an Active Learning Iterationen. Aufgrund der Anzahl der Experimente ist die Zahl gering gehalten.
- **Wiederholungen = 5**, Anzahl an Wiederholungen der Active Learning Durchläufe.
- **Seed = 5**, Startwert für die Reproduktion der Datensätze.
- **Sequenzlänge = 100**, Texte die mehr als 98 Tokens haben, werden nachfolgend abgeschnitten. Das erste und letzte Token werden durch die Modelle beansprucht.
- **Sample Size = 50**, in jeder Iteration wird diese Anzahl an Datenpunkte von den Active Learning Methoden ausgewählt und dem Orakel vorgelegt.
- **Epochen = 5**, jedes Modell wird bis zu fünf Epochen trainiert. Das Training endet früher, falls keine Verbesserung erkannt wird (EarlyStopping[2]).
- **Batch Size = 50**, der Wert wird bei jedem Modell verwendet.

[1] https://icc.informatik.haw-hamburg.de/
[2] https://www.tensorflow.org/api_docs/python/tf/keras/callbacks/EarlyStopping

4.5.3 Active Learning Framework

Für die Durchführung der Experimente ist im Rahmen dieser Arbeit ein Programm entwickelt worden, das verschiedene Active Learning Methoden für einen Datensatz und ein Modell durchführt und evaluiert. Der Ablauf ist durch den Algorithmus 1 dargestellt. Durch einen Startwert (Seed), lässt sich die Experimentumgebung reproduzieren. Zu Beginn jeden Durchlaufes wird ein vortrainiertes Modell, das bei der Initialisierung nicht vortrainierte Parameter per Zufall festlegt, gespeichert, sodass jede Sampling Methode das selbe Ausgansgmodell hat.

Algorithm 1: Active Learning Untersuchung mit verschiedenen Sampling Methoden

$seed \leftarrow$ Startwert
$n \leftarrow$ Iterationen
$k \leftarrow$ Anzahl auszuwählender Datenpunkten
$m_{init} \leftarrow$ Vortrainiertes Modell
$DS \leftarrow$ Datensatz
$Q \leftarrow$ Active Learning Sampling Methoden
$W \leftarrow$ Szenario (warmstart)

$train, test, dev \leftarrow$ generiere Trainings-, Test- und Validierungsset aus DS mit $seed$
$L_{start}, U_{start} \leftarrow$ initialisiere Pools nach $W(Train, seed)$
$m_{start} \leftarrow$ trainiere m_{init} mit L und dev
evaluiere m_{start} mit $test$

for $q \in Q$ **do**
    $L, U \leftarrow L_{start}, U_{start}$
    $m \leftarrow m_{start}$
    for $1...n$ **do**
        $ids \leftarrow$ wähle k interessante Datenpunkte durch $q(L, U, m, k)$ aus
        $L, U \leftarrow$ erfrage Klassen für ids durch Orakel
        $m \leftarrow$ trainiere m_{init} mit L und dev
        evaluiere m mit $test$
    end
end

Bei jedem Training wird das initiale vortrainierte Modell mit der gegenwärtigen Trainingsmenge L neu trainiert. Das Framework unterstützt die Sampling Methoden aus Abschnitt 4.3, die Modelle aus Abschnitt 4.4 und die Datensätze aus Abschnitt 4.1. Die Sampling Methoden basieren auf den open-source Methoden[1] aus [7].

[1] https://github.com/IBM/low-resource-text-classification-framework

Das Framework nutzt die Tensorflow Version der Hugging Face Bibliothek[1]. Das Training und die Interferenz der Modelle werden auf der GPU durchgeführt. Dies gilt ebenfalls für die Auswahlmethoden, die neuronale Netze nutzen. Jede Tensorflow Aktivität findet in einem vom Hauptprozess gekapselten Prozess statt, sodass der von Tensorflow allokierte GPU RAM freigeben wird, sobald das Modell nicht aktiv genutzt wird. Dies passiert auf Kosten der Zeit, die für die Interprozesskommunikation aufgewendet werden muss, garantiert jedoch ein sauberes Speichermanagement. Das hat zur Folge, dass Out of Memory Fehler beim Ausführen unterschiedlicher Modelle im selben Tensorflow Prozess präventiv verhindert werden.

4.6 Benutzerstudie

Der Benutzerstudie geht die Frage der Machbarkeit voraus, inwiefern der Mensch als Orakel in einem Active Learning Szenario funktioniert. Dazu wird das Framework in Abschnitt 4.5.3 angepasst, sodass ein Mensch die Klassen der ausgewählten Texte über eine Kommandozeile benatwortet. Die Kriterien, die zur Untersuchung der Machbarkeit herangezogen werden, sind:

- **Konzentration** Wie hoch ist die Fehlerrate der Klassen im Vergleich zu den originalen Klassen?
- **Geschwindigkeit** Wie viele Texte werden annotiert? Gibt es einen Leistungsabfall?

Anhand folgender Aufgabe werden die Kriterien der Machbarkeit beobachtet: Jeder Benutzer erhält die Aufgabe über fünf Active Learning Iterationen die ausgewählten Texte zu annotieren. Dabei kennt der Nutzer die möglichen Klassen. Die Texte werden in Teilmengen vorgesetzt, die 50 Elemente besitzen. Das Active Learning Szenario entspricht dem natürlichen Start aus Abschnitt 4.2. Als Auswahlmethode kommt Uncertainty Sampling zum Einsatz. Der zu annotierende Datensatz ist AG_NEWS (vier Klassen) und die sichtbaren Texte sind die unveränderten aus dem Datensatz ohne Vorverarbeitung, da sich dies negativ auf die schnelle Auffassung der Texte auswirkt, wenn beispielsweise Sonderzeichen oder bestimmte Satzzeichen fehlen. Die Benutzerstudie erlaubt einen stichprobenartigen praktischen Einblick in die Machbarkeit von Active Learning, hat jedoch nicht zum Ziel alle Facetten des Menschen in der Rolle des Orakels zu beleuchten.

[1] https://huggingface.co/

5 Ergebnisse

Dieses Kapitel beschreibt die Ergebnisse der Untersuchung in Abschnitt 5.1 und der Benutzerstudie in Abschnitt 5.2.

5.1 Active Learning Konstellationen

Insgesamt wurden 15.900 Active Learning Iterationen durchgeführt. Die Summe setzt sich aus den in Tabelle 5.1 gezeigten Konstellationen aus Szenario und Modell zusammen. Für jedes Tupel ist jeder Datensatz mit jeder Auswahlstrategie für fünf Active Learning Iterationen fünf mal durchgeführt worden[2]. Die Ergebnisse ergeben sich aus dem Durchschnitt der fünf Wiederholungen mit den Seeds: 6, 7, 8, 9 und 10. Während für das Hauptmodell BERT alle Szenarien berechnet wurden, liegt der Fokus für die weiteren Modelle aufgrund einer hohen Rechenzeit auf den beiden realitätsnahen Szenarien: Natürlich und Stichwort.

Szenario	ALBERT	BERT	DistilBERT	RoBERTa	F1-Scores
Ausgeglichen	-	Abb. A.1	-	-	Tab. A.1
Unausgeglichen	-	Abb. A.2	-	-	Tab. A.2
Natürlich	Abb. A.4	Abb. A.5	Abb. A.6	Abb. A.7	Tab. A.4
Fehlerhaft	-	Abb. A.3	-	-	Tab. A.3
Stichwort	Abb. A.8	Abb. A.9	Abb. A.10	Abb. A.11	Tab. A.5

Tabelle 5.1: Konstellation aus Szenarien und Modellen. Die Einträge verweisen auf die Darstellungen und Tabellen der Ergebnisse. Für leere Felder wurde keine Berechnung durchgeführt.

Für die Darstellung der Ergebnisse sind die makro-gemittelten F1-Scores der einzelnen Iterationen in Diagramme für jedes Szenario pro Datensatz zusammengefasst. Tabelle

[2]Das Stichwort-Szenario ist für TREC_FINE ungeeignet und nicht berechnet.

5.1 verweist auf die entsprechenden Diagramme im Anhang. Neben der Performance des Modells ist in den Diagrammen die Zufall-Baseline (grau, fein gestrichelt) und die Performance des passiv trainierten Modells[1] (grau, grob gestrichelt) eingetragen. Die nach der fünften Iteration erreichten F1-Scores sind in den Tabellen, auf jene die Tabelle 5.1 verweist, zu finden. Aufgrund der großen Anzahl befinden sich die entsprechenden Tabellen und Diagramme im Anhang. Abbildung 5.1 und 5.2 zeigen einen Ausschnitt aus dem natürlichen und dem Stichwort-Szenario von BERT und DistilBERT.

Die Auswahlmethoden zeigen in den Tabellen A.1-A.5, dass die zufällige Auswahl nach der fünften Iteration oft von mindestens einer Auswahlmethode übertroffen wird. Allerdings zeigen die Diagramme, dass sich keine Methode konsistenten über alle Modelle, Datensätze und Iterationen hinweg durchsetzen kann. Je nach Konstellation können sie sich einerseits deutlich von der Zufall-Baseline abheben, wie Abbildung A.2 bei IMDB_-IMB oder A.3 bei AG_NEWS_IMB zeigen. Andererseits liegt die Baseline oft gleichauf mit den Auswahlmethoden, wie unter anderem Abbildung 5.2 bei ISEAR_IMB zeigt. In Abbildung 5.2 bei DBPEDIA schlägt die Baseline sogar alle Methoden mit Ausnahme von DAL.

Für die Betrachtung des Multi-Klassen-Problems werden für die Auswertung die Datensätze mit mehr als zwei Klassen herangezogen. Nicht jedoch die TREC Datensätze, da diese durch die hohe Klassenanzahl beziehungsweise besonders kurze Sequenzlänge unter den Experimentparametern und fünf Iterationen keine aussagekräftige Entwicklung zeigen (Abbildungen: A.1-A.9, A.8, A.10).

Für die Bewertung der Leistung für das Multi-Klassen-Problem werden die p-Werte nach Wilcoxon für jede Active Learning Strategie zu den Szenarien gegenüber der Zufallsziehung berechnet. Durch die Mehrfachausführung des Experiments werden die Durchläufe mit der Bonferroni-Korrektur angepasst. Der p-Wert wird für eine Strategie S pro Szenario, nach dem Schema in [7], in erweiterter Form berechnet. Es werden alle F1-Score-Tupel (R_{mdik}, S_{mdik}) verglichen, die sich aus aus der Zufall-Baseline R, dem Modell m und dem Datensatz d, die für das Szenario genutzt werden, sowie der Iteration $i = (1...5)$ und der Wiederholung $k = (1...5)$ zusammensetzen.

Das Ergebnis ist in Tabelle 5.2 abgebildet und zeigt, dass die Discriminative Active Learning Methode gegenüber dem Zufall bei einem $\alpha = 0.05$ in den getesteten Szenarien über

[1] Das passive trainierte Modell ist auf dem gesamten Datensatz trainiert und stellt die best mögliche Performance unter selben Bedingungen dar, wenn alle Daten annotiert wären.

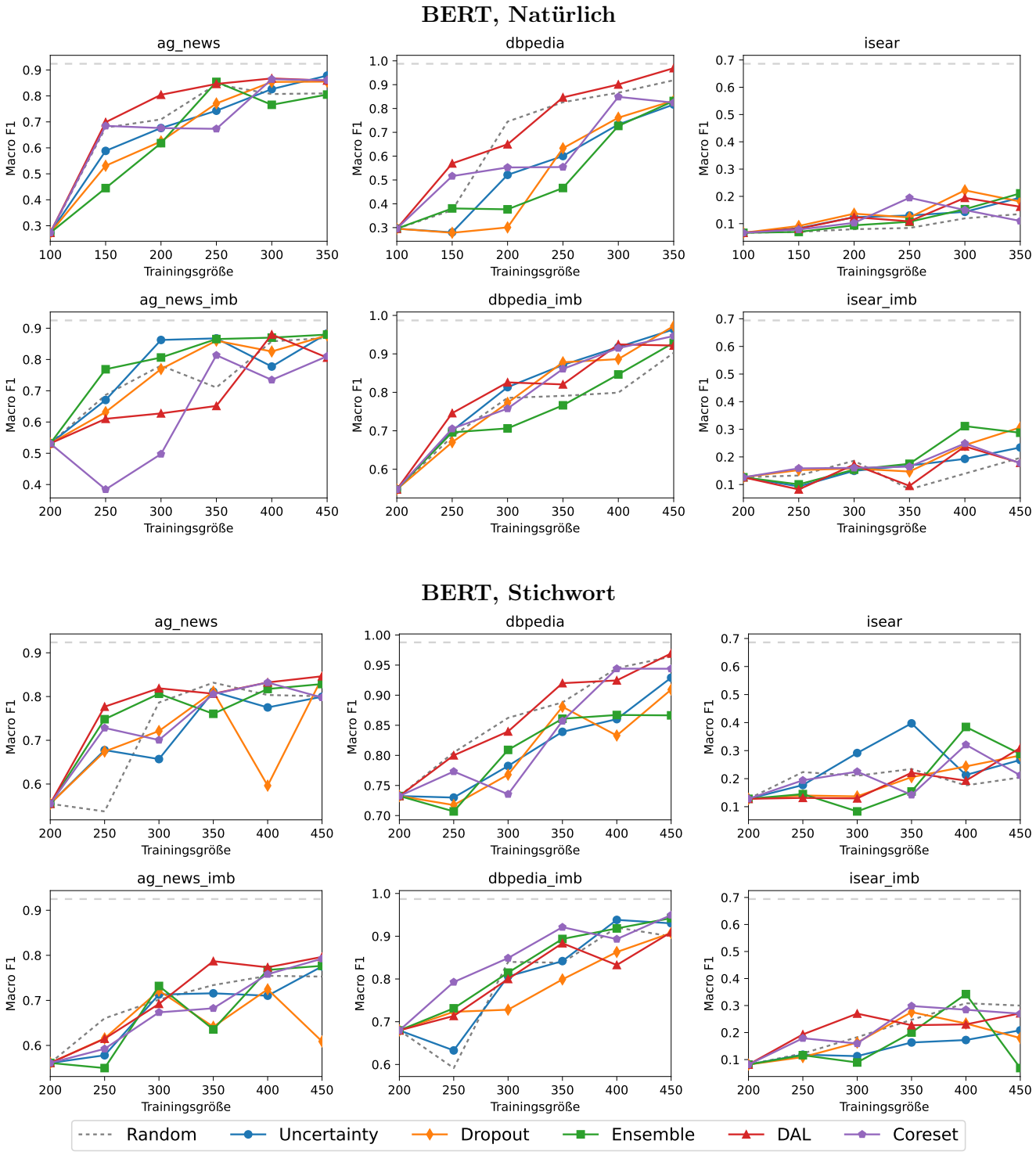

Abbildung 5.1: Active Learning Strategien für BERT bei einer natürlichen und stichwortbasierenden Startmenge für Multi-Klassen-Datensätze.

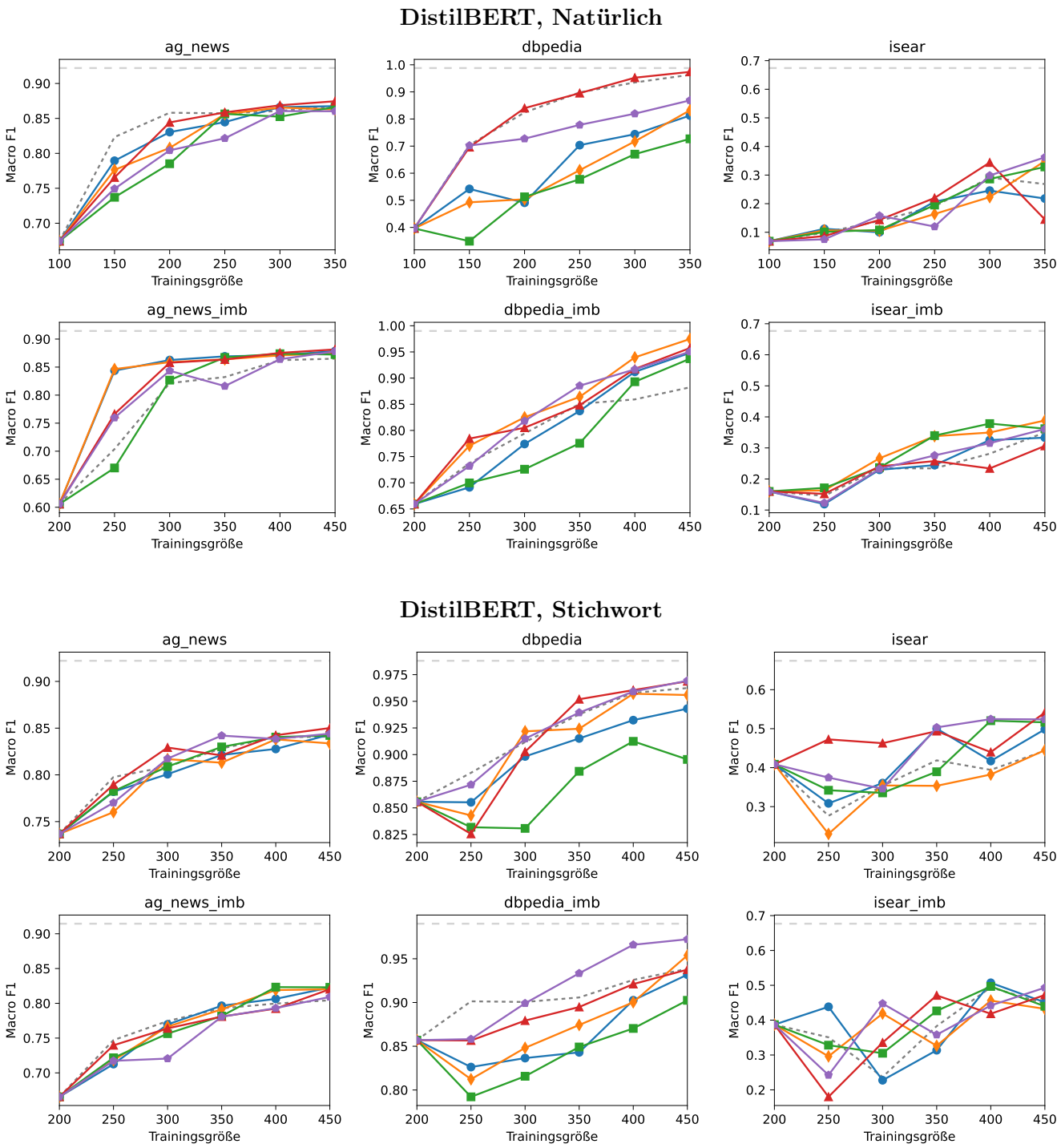

Abbildung 5.2: Active Learning Strategien für DistilBERT bei einer natürlichen und stichwortbasierenden Startmenge für Multi-Klassen-Datensätze.

	Natürlich	Fehlerhaft	Stichwort
Coreset	-	-	-
DAL	4.5×10^{-4}	2.2×10^{-3}	3.2×10^{-2}
Dropout	-	-	-
Ensemble	-	-	-
Uncertainty	-	-	-

Tabelle 5.2: P-Werte der Active Learning Strategien für natürliche, fehlerhafte und Stichwort-Szenarien gegenüber der zufälligen Auswahl. Leere Felder stehen für ein insignifikantes Ergebnis.

	Unausgeglichen		Natürlich		Fehlerhaft	
	D_{BAL}	D_{IMB}	D_{BAL}	D_{IMB}	D_{BAL}	D_{IMB}
Coreset	-	4.0×10^{-2}	-	-	-	-
Dropout	-	4.3×10^{-3}	-	-	-	-
Ensemble	-	-	-	-	-	-
Uncertainty	-	-	-	-	-	4.5×10^{-2}
DAL	5.4×10^{-5}	2.4×10^{-2}	-	3.2×10^{-2}	3.4×10^{-2}	4.8×10^{-3}

Tabelle 5.3: P-Werte der Active Learning Strategien für natürliche, unausgeglichene und fehlerhafte Startmengen unterteilt auf ausgeglichene und unausgeglichene Datensätze.

die verschiedenen Modelle hinweg signifikant besser ist. Coreset, Dropout und Uncertainty Sampling sind in nur bei Betrachtung einzelner Konstellationen besser.

Für die Betrachtung der ungleichverteilten Klassen in den Datensätzen ist in Tabelle 5.3 ein feingranularerer p-Wert für jede Auswahlstrategien in ausgeglichene Datensätze D_{BAL} und unausgeglichene D_{IMB} unterteilt. Nicht eingetragene Werte bedeuten keine signifikante Verbesserung gegenüber dem Zufall. Tabelle 5.3 zeigt, dass Active Learning Auswahlmethoden bei unausgeglichenen Datensätzen tendenziell signifikant besser sind als der Zufall, als bei ausgeglichenen Datensätzen. Während DAL im unausgeglichenen, natürlichen und fehlerhaften Szenario konsequent vor dem Zufall liegt, erreichen Coreset und Droput dies nur im unausgeglichenen Szenario und Uncertainty Sampling im fehlerhaften Szenario.

5.2 Benutzerstudie

Für die Benutzerstudie sind aufgrund der Coronapandemie eine kleine Anzahl von fünf Probanden ausgewählt worden. Die Personen sind im Durchschnitt 28 Jahre alt, davon eine weiblich und vier männlich. Drei Personen haben Berührungspunkte mit Themengebieten der Informatik. Zu Beginn wurde Active Learning und der Ablauf erläutert. Die Rolle als Orakel brachte keine Verständnisprobleme hervor.

In Abbildung 5.3 ist die durchschnittliche Zeit je Annotation und die Fehlerrate dargestellt. Die beobachtete mittlere Zeitspanne zum Annotieren von 50 Datenpunkte liegt bei 12 Minuten. Bei fortschreitenden Iteration nimmt die durchschnittliche Annotationszeit pro Datenpunkt geringfügig ab. Die Ausreißer bei 100 und 150 Datenpunkten liegen auf den Übergängen zwischen zwei Iterationen, bei denen einige Zeit auf auf das Modelltraining und die Auswahlmethoden gewartet werden muss. In der Zeit hat das Orakel keine Arbeit. Die beobachtete Fehlerrate steigt bis zum Ende der dritten Iteration an, fällt daraufhin stark ab und steigt wieder, und ähnelt dadurch einer Kippschwingung.

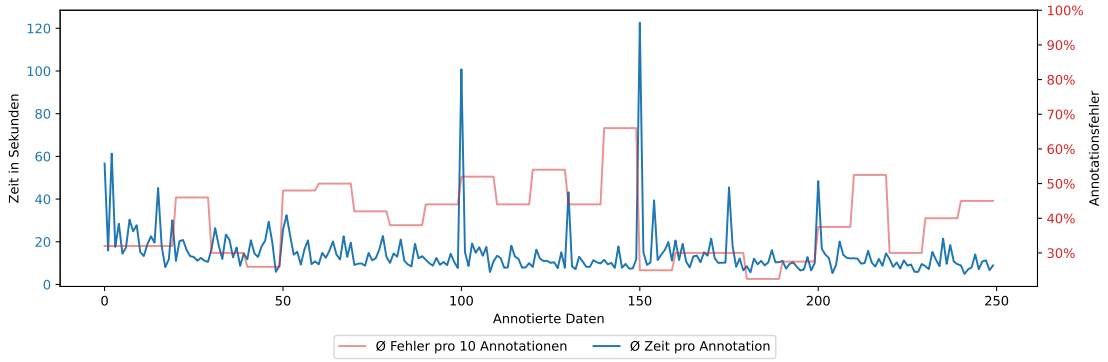

Abbildung 5.3: Durchschnittliche Zeit pro Annotation und die gemittelte Fehlerrate auf 10 Annotation.

In Abbildung 5.4 sind die wahren Klassen den annotierten Klassen des Benutzers in einer Konfusionsmatrix gegenübergestellt. Die Matrix zeigt, dass Sport und Science/Technology durchgängig richtig erkannt und selten anderen Kategorien zugeordnet wurden. Business und World sind über 30% mit der Kategorien Science/Technology verwechselt worden.

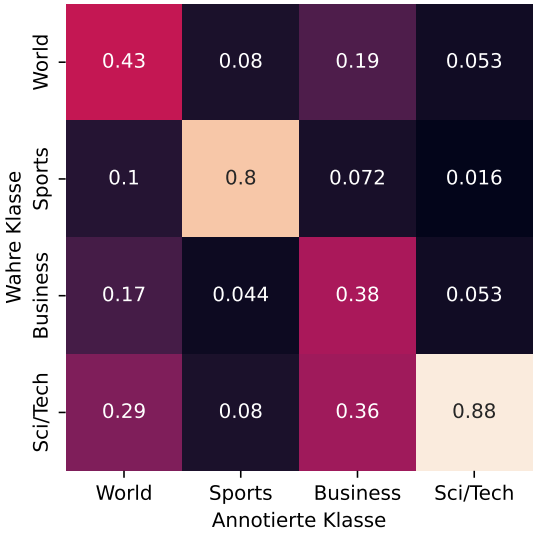

Abbildung 5.4: Konfusionsmatrix zwischen wahren und annotierten Klassen.

5.3 Laufzeit und Auswahlanalyse

Die durchschnittliche Laufzeit der Active Learning Methoden ist in Tabelle 5.4 aufgelistet. Die Laufzeit der Methoden bei ALBERT, DistilBERT und RoBERTa unterscheiden sich nur gering. Bei DistilBERT dauert Dropout im Schnitt 83 Sekunden aufgrund der kürzeren Interferenzdauer des kleineren Modells. Die Laufzeiten bilden nicht den kompletten Ablauf einer Iteration ab, sondern die Laufzeit einer Methode. Für eine Iteration kommen Training, Evaluation und das Orakel hinzu. Die kurze Laufzeit unterstützt den Einsatz in der Praxis.

Coreset	Dropout	Ensemble	Uncertainty	DAL
35	131	144	22	39

Tabelle 5.4: Durchschnittliche Laufzeit der Auswahlmethoden in Sekunden für BERT.

Die in [7] vorgestellte Analyse der ausgewählten Datenpunkte wird für die Untersuchung ebenfalls angewendet. Abbildung 5.5 zeigt die Diversität und Repräsentativität in den einzelnen Szenarien. Die Werte sind der Durchschnitt aus allen Wiederholungen und der jeweils ersten Active Learning Iteration, nachdem das Modell mit der Startmenge trainiert wurde und die ersten 50 Datenpunkte ausgewählt sind.

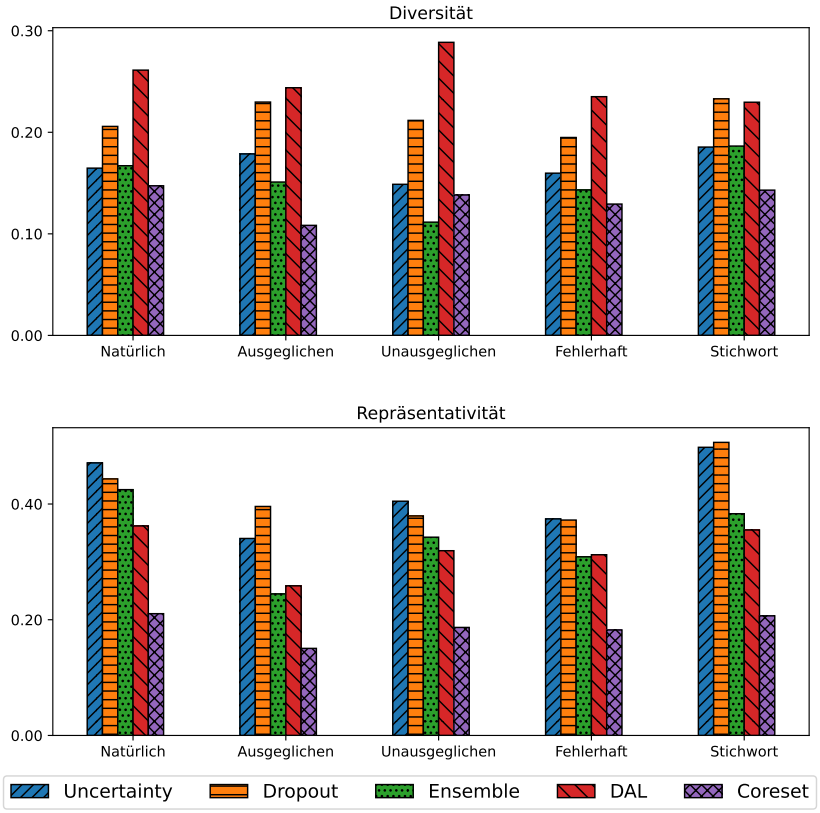

Abbildung 5.5: Diversität und Repräsentativität der Datenpunkte der Auswahlstrategien in jedem Szenario aller Modelle.

Für die Methoden Coreset und DAL geht man davon aus, dass sie die Diversität maximieren, weil sie Datenpunkte nicht einzeln sondern als geeignetste Gruppe auswählen. Allerdings erreicht nur DAL ein hohes Maß an Diversität in der Ausgewählten Menge, wie Abbildung 5.5 zeigt. Coreset bildet unerwartet das Schlusslicht. DAL hat als einzige Methode in der Untersuchung durchgehend gut performt, was den Schluss erlaubt, dass eine hohe Diversität durchaus ins Gewicht fallen kann.

Bei der Repräsentativität zeigt sich, das Coreset einen vergleichsweise niedrigen Wert hat, was bedeutet, dass der Algorithmus dazu neigt Ausreißer auszuwählen, die allgemein nicht viele andere Datenpunkte repräsentieren, wovon nach seiner Greedy-Implementation auszugehen ist. DAL steht den anderen Methoden nach, obwohl der Algorithmus ebenfalls repräsentative Eigenschaften fördert.

Auffallend ist, dass im Durchschnitt alle Methoden über die verschiedenen Szenarien nur geringe Abweichungen untereinander zeigen. Des Weiteren zeigt sich, dass Uncertainty- und Ensemble-Sampling, die jeweils auf dem Softmax des zuletzt mit der annotierten Menge trainierten Modells basieren, deutlich weniger Diversität aufweisen. Das entspricht der Tatsache, dass Datenpunkte der selben Klasse häufig in den ausgewählten Mengen auftauchen, da sie die selbe Unsicherheit des Modells betrifft.

6 Diskussion

Dieses Kapitel bezieht sich auf die zuvor vorgestellten Ergebnisse und diskutiert die Forschungsfragen chronologisch.

6.1 Active Learning für ein Multi-Klassen-Problem

Die in Abschnitt 5.1 beschriebenen Inkonsistenz der Auswahlstrategien decken sich mit [7] und [27], die ebenfalls eine Inkonsistenz der Methoden bei Klassifikationsaufgaben festgestellt haben.

Bezogen auf das Multi-Klassen-Problem dieser Untersuchung, können mehrere Gründe die Inkonsistenz begünstigen. Zum einen können die Hyperparameter für das Multi-Klassen-Problem nicht optimiert sein, sodass die Algorithmen auf Grundlage der wenigen Trainingsdaten stark variieren. Einige Modelle erreichen nicht die Performance des passiv gelernten Modells. Die Anzahl von fünf Iterationen ist einerseits nur aus praxisorientierter Sicht sinnvoll, da der Umfang für ein menschliches Orakel zumutbar bleibt, andererseits dient er der Vergleichbarkeit zu [7]. Es ist zu vermuten, dass die Epxerimentparameter restriktiv wirken. Die Datensätze, welche sowohl kurze Sequenzlängen als auch viele und ungleich verteilte Klassen besitzen, haben dadurch ein das Experiment übersteigendes Anforderungsprofil, das durch die vortrainierten Modelle nicht kompensiert werden kann.

Ein weiterer Grund für die Begünstigung der Inkonsistenz der Auswahlstrategien kann die ausgewählte Messmetrik des makro-gemittelten F1-Scores sein. Diese Messmetrik, die die Leistung streng bewertet, gibt allen Klassen unabhängig der Verteilung dasselbe Gewicht. Andere Messmetriken, wie der gewichtete oder der mikro-gemittelte F1-Score berücksichtigen hingegen eine Verteilung der Klassen.

Es ist dadurch möglich, dass ein Leistungsvorsprung der Modelle gegenüber den Baseline-Modellen auf Grundlage der verkleinerten Datensätze nicht gut widerspiegelt wird. Die

kleinen Trainingsdatensätze gewährleisten keine Generalisierung des bis zu sechs mal größeren Validierungssets.

Das universelle Training der Konstellationen kann zur Folge haben, dass die Auswahlstrategien entgegen der Theorie sich nicht vom Zufall abheben. Es ist zu erwarten, dass der Informationsgewinn der Auswahlstrategien gemessen am makro-gemittelten F1-Score deutlicher wird, wenn die Konstellationen individualisiert trainiert werden.

Tabelle 5.2 zeigt eine solche Abhebung über alle Konstellation bei DAL mit einem p-Wert unter 0,05 für das natürliche, fehlerhafte und Stichwort-Szenario. Das stützt den in [13] geäußerten agnostischen Ansatz der Autoren. Ebenso wird aus der Tabelle 5.2 deutlich, dass die weiteren getesteten Active Learning Methoden in den Szenarien nicht für ein beliebiges Problem funktionieren. Es lässt sich jedoch aufgrund der Diagramme vermuten, dass Active Learning bei einem Multi-Klassen-Problem eher in komplexeren Verhältnissen seinen Vorteil ausspielen kann. Dies deckt sich mit [7].

Im Hinblick auf RQ1 lässt sich sagen, dass das Mehr-Klassen-Problem gegenüber dem binären Problem eine deutliche komplexere Klassifikationsaufgabe darstellt, die mit den Parametern aus [7] in dieser Untersuchung nicht eindeutig beantwortet werden kann. Die signifikant bessere Performance von DAL in der eingeschränkten Umgebung lässt vermuten, dass bei optimierten Umgebungen bessere Performance bei Multi-Klassen-Problemen von allen untersuchten Methoden zu erwarten ist. Insbesondere dann, wenn ein Ungleichgewicht in den Klassen vorhanden ist.

6.2 Auswirkung der initialen Startmenge

In Hinblick auf RQ2 wird der Einfluss der initialen Startmenge auf den Experimenten beleuchtet. Dazu ist zum einen eine ausgeglichene Startmenge für unausgeglichene Datensätze (Abbildung A.1, zweite und vierte Reihe), zum anderen eine unausgeglichene Startmenge für ausgeglichene Datensätze (Abbildung A.2, erste und dritte Reihe) in den Experimenten verwendet worden. Hinzu kommen Startmengen aus dem fehlerhaften (Abbildung A.3) und Stichwort-Szenario (Abbildung A.9, A.8, A.11, A.9).

Eine Erkenntnis ist, dass bei jeder Startmenge die Modelle konvergieren. Abhängig ist dies von dem Datensatz und der Anzahl der Klassen, die eine Konvergenz hemmen, sofern die kleine Trainingsmenge nicht mehrere Klassen ausreichend abdecken kann. Bei ausgeglichenen Startmengen weist das initiale Modell in der Regel einen etwas höheren

F1-Score gegenüber den unausgeglichenen Startmengen auf. Diese Diskrepanz schrumpft jedoch in den darauf folgenden Iterationen.

Die Untersuchung legt nahe, dass für BERT und seine Variationen im Active Learning Kontext die Zusammensetzung der Startmenge keine große Auswirkung hat. Selbst eine Startmenge mit 30% fehlerhaft annotierten Daten wirkt sich langfristig nicht negativ aus, sofern der Datensatz für Active Learning geeignet ist. Geeignete Datensätze sind bestimmt durch wenige Klassen und eine ausreichend große Menge an Trainingsdaten. Die verschiedenen BERT-Modelle legen eine ähnliche Performance in jedem Szenario hin. Kleinere Unterschiede sind vor allem in den sehr komplexen Situationen zu beobachten, wie bei dem ISEAR und TREC Datensatz und ihren unausgeglichen Versionen. BERT und RoBERTa erreichen dort keine Konvergenz und resultieren in einem unbrauchbaren Modell. Im Gegensatz dazu ist bei DistilBERT und ALBERT eine Konvergenz im natürlichen und sitchwortbasierten Szenario angedeutet.

6.3 Unterrepräsentation von Klassen im Datensatz

Eine Hälfte der Datensätze der Untersuchung ist unausgeglichen. Bei Betrachtung der F1-Scores der Szenarien über die Modelle zeigt sich, wie in Abschnitt 6.1 geschlussfolgert, dass Active Learning tendenziell bei verzerrten Klassenverteilungen einen Vorteil gegenüber dem Zufall hat.

Dieser Trend lässt sich aus Tabelle 5.3 ablesen, in der die p-Werte für jede Auswahlstrategien in ausgeglichene Datensätze D_{BAL} und unausgeglichene D_{IMB} unterteilt sind. Nicht eingetragene Werte bedeuten keine signifikante Verbesserung gegenüber dem Zufall.

In den D_{IMB} Spalten werden mehr signifikante Verbesserung gegenüber dem Zufall erreicht, als in D_{BAL}. Zu RQ3 lässt sich schlussfolgern, dass Active Learning bei mehreren Klassen von einem Ungleichgewicht in der Regel profitiert. Bei ausgeglichenen Daten sind die Methoden selten signifikant besser als die Zufall-Baseline.

6.4 Der Mensch als Orakel

Die Benutzerstudie zeigt auf, dass die Benutzer Schwierigkeiten haben kurze Nachrichtentexte, die nah an Entscheidungsgrenzen liegen, ohne journalistisches Domänenwissen

korrekt zu annotieren. Die Textausschnitte werden ohne Kontext präsentiert und beziehen sich zum Teil auf regionale Ereignisse aus amerikanischen Nachrichten, zu denen keine Versuchsperson über ausreichendes Wissen verfügt. Die Konfusion bei der Science/Technology Klasse, die von den Personen über 30% als Business Klasse gedeutet wurde, lässt sich auf eine mangelhafte Definition der Klassen zurückführen. Die intuitive Vermutung ist bei sich überschneidenden Bereichen volatil und neigt daher zur zufälligen Auswahl einer der beiden passenden Kategorien.

Aus Abbildung 5.3 lässt sich schließen, dass der Benutzer über den Annotationszeitraum nicht durch Ermüdung eingeschränkt ist. Der geringe Abfall kann so gedeutet werden, dass die Benutzer die Aufgabe verinnerlicht haben und Entscheidungen nach einigen Wiederholungen schneller getroffen haben.

Die Fehlerrate über die Zeit wird als Indikator für die Konzentration herangezogen. Der Graph, der an an eine Kippschwingung erinnert, lässt vermuten, dass die Personen zu Beginn einer Iteration sehr konzentriert und motiviert sind möglichst wenig falsche Klassen zu verteilen. Durch die repetitive Aufgabe nimmt die Konzentration ab und folglich die Fehlerrate zu.

Eine weitere Vermutung bezieht sich auf die Auswahlmethode. Uncertainty Sampling wählt pro Iteration Datenpunkte mit hoher Redundanz aus, was zur Folge hat, dass eine unsichere Klasse häufig vertreten ist. Die zu annotierenden Datenpunkte sind sehr ähnlich zu einander, was es der Person erschwert diese korrekt zu klassifizieren, insbesondere wenn alle Datenpunkte an Entscheidungsgrenzen liegen.

Zu RQ4 lässt sich herleiten, dass diese Untersuchung als realistisches Active Learning Szenario im Hinblick auf ein menschliches Orakel umsetzbar ist. Es wird jedoch auch offen gelegt, dass die Annotation ohne Domänenwissen eine hohe Fehlerrate mitbringt. Des Weiteren wurden festgestellt, dass die Personen nach bestimmten Stichwörtern im Text suchen und bei großer Unsicherheit nach dem Ausschlussverfahren vorgehen.

7 Abschluss

Die Arbeit untersucht fünf Active Learning Methoden für verschiedene Multi-Klassen-Probleme für BERT und seine Variationen. Sie folgt dabei der systematischen Untersuchung des binären Problems aus [7]. Es sind fünf Szenarien definiert, die verschiedene Zustände der initialen Startmengen für Active Learning beschreiben. Die Szenarien umfassen ideale bis möglichst realitätsnahe Zustände. Damit wird die Diskrepanz zwischen Forschung und Praxis untersucht beziehungsweise überwunden.

Für einen Einblick in die praktische Anwendung von Active Learning wurde in einer Benutzerstudie der Mensch als Orakel eingesetzt. Die Untersuchung wurde für fünf Active Learning Iterationen mit AG_NEWS als Datensatz, BERT als Modell und Uncertainty Sampling als Auswahlmethode durchgeführt.

7.1 Fazit

Bei der Untersuchung der fünf Szenarien ergibt sich, dass Active Learning sich bei einem Multi-Klassen-Problem nur bedingt vom Zufall abheben kann. Die meisten Auswahlmethoden zeigen, dass kein konsequent besseres Ergebnis gegenüber der zufälligen Auswahl über die Modelle und Datensätze hinweg erzielt wird. Die Auswahlmethoden Coreset, Dropout und Uncertainty Sampling haben für Startmengen und Datensätzen mit ungleich verteilten Klassen einen messbaren positiven Einfluss auf das Ergebnis. Eine Ausnahme bildet Discriminate Active Learning, welches in fast allen untersuchten Experimenten signifikant besser abschneidet als die Zufall-Baseline.

Die Benutzerstudie zeigt auf, das bei dem genutzten AG_NEWS Datensatz, die Zuweisung von Kategorien ohne Leitfaden oder journalistischen Domänenwissen mit einer relativ hohen Fehlerrate bei sich überschneidenden Kategorien einhergeht. Ebenfalls zeigt sich, dass die Benutzer ohne Konzentrationsprobleme mit konstanter Geschwindigkeit kurze Texte mit einer durchschnittlichen Länge von 38 Wörtern über fünf Iterationen

problemlos annotieren können. Der Mensch als Orakel ist in diesem Active Learning Szenario plausibel.

7.2 Ausblick

Für die weitere Untersuchung von Active Learning für Multi-Klassen-Probleme bieten sich folgende Aspekte an. Die Arbeit ist den Untersuchungsparametern aus [7] gefolgt, die ein vereinfachtes Szenario abbilden. Die kleine Trainigsmenge nach fünf Iterationen ist für komplexe Datensätze mit vielen Klassen und besonders kurzen oder langen Sequenzlängen ungeeignet. Für komplexere Datensätze konvergiert die Performance des trainierten Modells nicht.

Für die weitere Forschung bietet sich das Auffinden nach geeigneten Experimentparametern an, wie der Anzahl der Iterationen und der vom Orakel zu annotierenden Daten. Ebenfalls kann ein Augenmerk auf das Modelltraining gelegt werden, sodass die Performance bei kleiner Trainigsmenge optimiert und Instabilität verhindert wird. Ein Ansatz wäre die zufällige Initialisation der nicht vortrainierten Parameter zu untersuchen.

Ebenfalls kann die Implementation der Auswahlmethoden sowie ihre Parameter optimiert werden. Coreset folgt in dieser Untersuchung beispielsweise der effizienteren Greedy-Version, die jedoch nicht dieselbe Leistung wie ihre robuste Implementation aufweist [39]. Ebenso setzt sich die Ensemble Auswahlmethode aus Effizienzgründen aus kleinen Perzeptron-Modellen zusammen und bildet kein Ensemble aus dem eigentlich trainierten Modell.

Eine ausgiebige Untersuchung der Parameter und deren Auswirkung auf die Performance eignet sich als weiterer interessanter Ansatz im Bereich von Active Learning für Multiklasse-Probleme.

Literaturverzeichnis

[1] Angluin, Dana: Queries and concept learning. In: *Machine learning* 2 (1988), Nr. 4, S. 319–342

[2] Breiman, Leo: Bagging predictors. In: *Machine learning* 24 (1996), Nr. 2, S. 123–140

[3] Britz, Denny: *Attention and Memory in Deep Learning and NLP*. Jan 2016. – URL https://www.wildml.com/2016/01/attention-and-memory-in-deep-learning-and-nlp/

[4] Dagan, Ido ; Engelson, Sean P.: Committee-based sampling for training probabilistic classifiers. In: *Machine Learning Proceedings 1995*. Elsevier, 1995, S. 150–157

[5] Dan-Glauser, Elise S. ; Scherer, Klaus R.: The difficulties in emotion regulation scale (DERS). In: *Swiss Journal of Psychology* (2012)

[6] Devlin, Jacob ; Chang, Ming-Wei ; Lee, Kenton ; Toutanova, Kristina: *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2019

[7] Dor, Liat E. ; Halfon, Alon ; Gera, Ariel ; Shnarch, Eyal ; Dankin, Lena ; Choshen, Leshem ; Danilevsky, Marina ; Aharonov, Ranit ; Katz, Yoav ; Slonim, Noam: Active Learning for BERT: An Empirical Study. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020, S. 7949–7962

[8] Freund, Yoav ; Schapire, Robert E.: A decision-theoretic generalization of on-line learning and an application to boosting. In: *Journal of computer and system sciences* 55 (1997), Nr. 1, S. 119–139

[9] Gal, Yarin ; Ghahramani, Zoubin: Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In: *international conference on machine learning* PMLR (Veranst.), 2016, S. 1050–1059

[10] Gal, Yarin ; Islam, Riashat ; Ghahramani, Zoubin: Deep bayesian active learning with image data. In: *International Conference on Machine Learning* PMLR (Veranst.), 2017, S. 1183–1192

[11] Ghahramani, Zoubin: Probabilistic machine learning and artificial intelligence. In: *Nature* 521 (2015), Nr. 7553, S. 452–459

[12] Gildenblat, Jacob: *Overview of Active Learning for Deep Learning*. Feb 2020. – URL https://jacobgil.github.io/deeplearning/activelearning

[13] Gissin, Daniel ; Shalev-Shwartz, Shai: Discriminative active learning. In: *arXiv preprint arXiv:1907.06347* (2019)

[14] Hinton, Geoffrey ; Vinyals, Oriol ; Dean, Jeff: Distilling the knowledge in a neural network. In: *arXiv preprint arXiv:1503.02531* (2015)

[15] Houlsby, Neil ; Huszár, Ferenc ; Ghahramani, Zoubin ; Lengyel, Máté: Bayesian active learning for classification and preference learning. In: *arXiv preprint arXiv:1112.5745* (2011)

[16] Kee, Seho ; Del Castillo, Enrique ; Runger, George: Query-by-committee improvement with diversity and density in batch active learning. In: *Information Sciences* 454 (2018), S. 401–418

[17] Kendall, Alex ; Gal, Yarin: What uncertainties do we need in bayesian deep learning for computer vision? In: *arXiv preprint arXiv:1703.04977* (2017)

[18] King, Ross D. ; Whelan, Kenneth E. ; Jones, Ffion M. ; Reiser, Philip G. ; Bryant, Christopher H. ; Muggleton, Stephen H. ; Kell, Douglas B. ; Oliver, Stephen G.: Functional genomic hypothesis generation and experimentation by a robot scientist. In: *Nature* 427 (2004), Nr. 6971, S. 247–252

[19] Krzywinski, Martin ; Altman, Naomi: Importance of being uncertain. In: *Nature methods* 10 (2013), Nr. 9, S. 809–811

[20] Kullback, Solomon ; Leibler, Richard A.: On information and sufficiency. In: *The annals of mathematical statistics* 22 (1951), Nr. 1, S. 79–86

[21] Lan, Zhenzhong ; Chen, Mingda ; Goodman, Sebastian ; Gimpel, Kevin ; Sharma, Piyush ; Soricut, Radu: Albert: A lite bert for self-supervised learning of language representations. In: *arXiv preprint arXiv:1909.11942* (2019)

[22] Lehmann, Jens ; Isele, Robert ; Jakob, Max ; Jentzsch, Anja ; Kontokostas, Dimitris ; Mendes, Pablo N. ; Hellmann, Sebastian ; Morsey, Mohamed ; Van Kleef, Patrick ; Auer, Sören u. a.: Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. In: *Semantic web* 6 (2015), Nr. 2, S. 167–195

[23] Lewis, David D. ; Gale, William A.: A sequential algorithm for training text classifiers. In: *SIGIR'94* Springer (Veranst.), 1994, S. 3–12

[24] Li, Xin ; Roth, Dan: Learning question classifiers. In: *COLING 2002: The 19th International Conference on Computational Linguistics*, 2002

[25] Liu, Mingyi ; Tu, Zhiying ; Wang, Zhongjie ; Xu, Xiaofei: LTP: a new active learning strategy for BERT-CRF based named entity recognition. In: *arXiv preprint arXiv:2001.02524* (2020)

[26] Liu, Yinhan ; Ott, Myle ; Goyal, Naman ; Du, Jingfei ; Joshi, Mandar ; Chen, Danqi ; Levy, Omer ; Lewis, Mike ; Zettlemoyer, Luke ; Stoyanov, Veselin: Roberta: A robustly optimized bert pretraining approach. In: *arXiv preprint arXiv:1907.11692* (2019)

[27] Lowell, David ; Lipton, Zachary C. ; Wallace, Byron C.: Practical obstacles to deploying active learning. In: *arXiv preprint arXiv:1807.04801* (2018)

[28] Lu, Jinghui ; MacNamee, Brian: Investigating the Effectiveness of Representations Based on Pretrained Transformer-based Language Models in Active Learning for Labelling Text Datasets. In: *arXiv preprint arXiv:2004.13138* (2020)

[29] Maas, Andrew L. ; Daly, Raymond E. ; Pham, Peter T. ; Huang, Dan ; Ng, Andrew Y. ; Potts, Christopher: Learning Word Vectors for Sentiment Analysis. In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Portland, Oregon, USA : Association for Computational Linguistics, June 2011, S. 142–150. – URL http://www.aclweb.org/anthology/P11-1015

[30] MacKay, David J.: A practical Bayesian framework for backpropagation networks. In: *Neural computation* 4 (1992), Nr. 3, S. 448–472

[31] McCallumzy, Andrew K. ; Nigamy, Kamal: Employing EM and pool-based active learning for text classification. In: *Proc. International Conference on Machine Learning (ICML)* Citeseer (Veranst.), 1998, S. 359–367

[32] Mohri, Mehryar ; Rostamizadeh, Afshin ; Talwalkar, Ameet: *Foundations of machine learning*. MIT press, 2018

[33] Pan, Sinno J. ; Yang, Qiang: A survey on transfer learning. In: *IEEE Transactions on knowledge and data engineering* 22 (2009), Nr. 10, S. 1345–1359

[34] Prabhu, Ameya ; Dognin, Charles ; Singh, Maneesh: Sampling bias in deep active classification: An empirical study. In: *arXiv preprint arXiv:1909.09389* (2019)

[35] Radford, Alec ; Narasimhan, Karthik ; Salimans, Tim ; Sutskever, Ilya: Improving language understanding by generative pre-training. (2018)

[36] Sanh, Victor ; Debut, Lysandre ; Chaumond, Julien ; Wolf, Thomas: DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. In: *arXiv preprint arXiv:1910.01108* (2019)

[37] Schröder, Christopher ; Niekler, Andreas: A survey of active learning for text classification using deep neural networks. In: *arXiv preprint arXiv:2008.07267* (2020)

[38] Schröder, Christopher ; Niekler, Andreas ; Potthast, Martin: Uncertainty-based Query Strategies for Active Learning with Transformers. In: *arXiv preprint arXiv:2107.05687* (2021)

[39] Sener, Ozan ; Savarese, Silvio: Active learning for convolutional neural networks: A core-set approach. In: *arXiv preprint arXiv:1708.00489* (2017)

[40] Settles, Burr: Active learning literature survey. (2009)

[41] Seung, H S. ; Opper, Manfred ; Sompolinsky, Haim: Query by committee. In: *Proceedings of the fifth annual workshop on Computational learning theory*, 1992, S. 287–294

[42] Shannon, Claude E.: A mathematical theory of communication. In: *The Bell system technical journal* 27 (1948), Nr. 3, S. 379–423

[43] Shelmanov, Artem ; Liventsev, Vadim ; Kireev, Danil ; Khromov, Nikita ; Panchenko, Alexander ; Fedulova, Irina ; Dylov, Dmitry V.: Active learning with deep pre-trained models for sequence tagging of clinical and biomedical texts. In: *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)* IEEE (Veranst.), 2019, S. 482–489

[44] Siddhant, Aditya ; Lipton, Zachary C.: Deep bayesian active learning for natural language processing: Results of a large-scale empirical study. In: *arXiv preprint arXiv:1808.05697* (2018)

[45] Srivastava, Nitish ; Hinton, Geoffrey ; Krizhevsky, Alex ; Sutskever, Ilya ; Salakhutdinov, Ruslan: Dropout: a simple way to prevent neural networks from overfitting. In: *The journal of machine learning research* 15 (2014), Nr. 1, S. 1929–1958

[46] Vaswani, Ashish ; Shazeer, Noam ; Parmar, Niki ; Uszkoreit, Jakob ; Jones, Llion ; Gomez, Aidan N. ; Kaiser, Łukasz ; Polosukhin, Illia: Attention is all you need. In: *Advances in neural information processing systems*, 2017, S. 5998–6008

[47] Wang, Keze ; Zhang, Dongyu ; Li, Ya ; Zhang, Ruimao ; Lin, Liang: Cost-effective active learning for deep image classification. In: *IEEE Transactions on Circuits and Systems for Video Technology* 27 (2016), Nr. 12, S. 2591–2600

[48] Wang, Zirui ; Dai, Zihang ; Póczos, Barnabás ; Carbonell, Jaime: Characterizing and avoiding negative transfer. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019

[49] Wischhusen, Jonathan: *Sentiment Clustering via BERT Encodings*. Juli 2020. – Projektarbeit

[50] Zhang, Leihan ; Zhang, Le: An ensemble deep active learning method for intent classification. In: *Proceedings of the 2019 3rd International Conference on Computer Science and Artificial Intelligence*, 2019, S. 107–111

[51] Zhang, Xiang ; Zhao, Junbo ; LeCun, Yann: Character-level convolutional networks for text classification. In: *Advances in neural information processing systems* 28 (2015), S. 649–657

[52] Zhang, Ye ; Lease, Matthew ; Wallace, Byron: Active discriminative text representation learning. In: *Proceedings of the AAAI Conference on Artificial Intelligence* Bd. 31, 2017

[53] Zhdanov, Fedor: Diverse mini-batch active learning. In: *arXiv preprint arXiv:1901.05954* (2019)

[54] Zhu, Jingbo ; Wang, Huizhen ; Yao, Tianshun ; Tsou, Benjamin K.: Active learning with sampling by uncertainty and density for word sense disambiguation

and text classification. In: *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, 2008, S. 1137–1144

A Anhang

Ausgeglichenes Szenario

Datensatz	Modell	Coreset	DAL	Dropout	Ensemble	Uncertainty	Zufall
AG_NEWS	BERT	0.724	0.803	0.728	**0.859**	0.844	0.829
AG_NEWS_IMB	BERT	**0.863**	0.839	**0.863**	0.703	0.762	0.852
DBPEDIA	BERT	0.837	**0.962**	0.885	0.763	0.804	0.957
DBPEDIA_IMB	BERT	0.838	**0.889**	0.873	0.715	0.837	0.794
IMDB	BERT	0.759	0.791	0.640	0.717	0.775	**0.783**
IMDB_IMB	BERT	0.676	0.701	0.702	0.669	0.664	**0.714**
ISEAR	BERT	0.155	0.122	0.200	**0.250**	0.190	0.144
ISEAR_IMB	BERT	0.080	0.087	0.140	0.170	0.078	**0.206**
TREC_COARSE	BERT	0.314	0.485	**0.580**	0.336	0.441	0.150
TREC_FINE	BERT	0.004	0.009	0.010	0.019	0.010	0.011

Tabelle A.1: Durchschnittlicher Makro-F1-Score nach der fünften Iteration aus fünf Wiederholungen für das ausgeglichene Szenario.

Unausgeglichenes Szenario

Datensatz	Modell	Coreset	DAL	Dropout	Ensemble	Uncertainty	Zufall
AG_NEWS	BERT	0.828	**0.884**	0.872	0.875	0.872	0.810
AG_NEWS_IMB	BERT	0.724	0.863	**0.880**	0.868	0.872	0.864
DBPEDIA	BERT	0.927	**0.984**	0.960	0.915	0.930	0.942
DBPEDIA_IMB	BERT	0.959	0.941	**0.961**	0.862	0.925	0.871
IMDB	BERT	0.700	0.749	0.766	**0.784**	0.765	0.731
IMDB_IMB	BERT	0.647	0.737	0.732	**0.742**	0.711	0.571
ISEAR	BERT	0.102	**0.166**	0.158	0.164	0.148	0.143
ISEAR_IMB	BERT	0.146	0.097	**0.178**	0.126	0.083	0.083
TREC_COARSE	BERT	0.402	**0.517**	0.335	0.410	0.489	0.457
TREC_FINE	BERT	0.012	0.009	0.024	**0.027**	0.026	0.023

Tabelle A.2: Durchschnittlicher Makro-F1-Score nach der fünften Iteration aus fünf Wiederholungen für das unausgeglichene Szenario.

Fehlerhaftes Szenario

Datensatz	Modell	Coreset	DAL	Dropout	Ensemble	Uncertainty	Zufall
AG_NEWS	BERT	0.826	0.837	0.842	**0.868**	0.852	0.804
AG_NEWS_IMB	BERT	0.804	**0.880**	0.864	0.838	0.852	0.669
DBPEDIA	BERT	0.851	0.927	0.719	0.797	0.871	**0.939**
DBPEDIA_IMB	BERT	0.910	**0.920**	0.897	0.862	0.916	0.850
IMDB	BERT	0.728	**0.768**	0.758	0.726	0.735	0.738
IMDB_IMB	BERT	0.606	0.747	**0.752**	0.728	0.751	0.650
ISEAR	BERT	0.237	**0.258**	0.218	0.085	0.204	0.110
ISEAR_IMB	BERT	0.152	0.183	0.190	**0.256**	0.157	0.198
TREC_COARSE	BERT	0.277	0.130	**0.309**	0.204	0.360	0.228
TREC_FINE	BERT	0.008	0.011	**0.029**	0.025	**0.029**	0.023

Tabelle A.3: Durchschnittlicher Makro-F1-Score nach der fünften Iteration aus fünf Wiederholungen für das fehlerhaftes Szenario.

Natürliches Szenario

Datensatz	Modell	Coreset	DAL	Dropout	Ensemble	Uncertainty	Zufall
AG_NEWS	ALBERT	0.834	0.856	0.818	0.846	0.777	0.854
	BERT	0.860	0.859	0.855	0.805	**0.878**	0.810
	DistilBERT	0.860	0.875	0.863	0.866	0.868	0.867
	RoBERTa	0.871	0.872	0.843	0.862	0.854	0.852
AG_NEWS_IMB	ALBERT	0.829	0.844	0.853	0.862	0.849	0.846
	BERT	0.810	0.806	0.877	0.880	**0.883**	0.869
	DistilBERT	0.878	0.882	0.875	0.872	0.879	0.865
	RoBERTa	0.879	0.871	0.871	0.873	0.878	0.874
DBPEDIA	ALBERT	0.893	**0.976**	0.874	0.777	0.911	0.969
	BERT	0.824	0.969	0.833	0.831	0.817	0.919
	DistilBERT	0.869	0.974	0.832	0.727	0.813	0.963
	RoBERTa	0.830	0.925	0.886	0.772	0.789	0.956
DBPEDIA_IMB	ALBERT	0.963	0.969	0.969	0.899	0.965	0.916
	BERT	0.946	0.922	0.972	0.929	0.964	0.903
	DistilBERT	0.950	0.957	**0.975**	0.937	0.948	0.882
	RoBERTa	0.941	0.946	0.951	0.892	0.959	0.947
IMDB	ALBERT	0.657	0.672	0.677	0.683	0.668	0.685
	BERT	0.734	0.795	0.764	0.725	0.768	0.769
	DistilBERT	0.676	0.797	0.758	0.795	0.784	0.792
	RoBERTa	0.801	**0.845**	0.772	0.814	0.736	0.839
IMDB_IMB	ALBERT	0.628	0.633	0.659	0.621	0.645	0.596
	BERT	0.662	0.709	0.738	0.737	0.705	0.691
	DistilBERT	0.658	0.749	0.757	0.747	0.748	0.715
	RoBERTa	0.777	0.788	0.789	**0.793**	0.786	0.745
ISEAR	ALBERT	0.194	0.228	0.248	0.284	0.318	0.257
	BERT	0.110	0.162	0.181	0.211	0.195	0.135
	DistilBERT	**0.362**	0.145	0.349	0.328	0.218	0.268
	RoBERTa	0.051	0.072	0.105	0.059	0.081	0.080
ISEAR_IMB	ALBERT	0.311	0.234	0.243	0.312	0.245	0.156
	BERT	0.177	0.178	0.307	0.287	0.234	0.197
	DistilBERT	0.361	0.306	**0.388**	0.362	0.333	0.350
	RoBERTa	0.106	0.072	0.127	0.106	0.131	0.094
TREC_COARSE	ALBERT	0.498	0.525	**0.629**	0.516	0.436	0.519
	BERT	0.336	0.241	0.418	0.174	0.260	0.255
	DistilBERT	0.216	0.360	0.375	0.436	0.404	0.255
	RoBERTa	0.097	0.281	0.149	0.331	0.253	0.181
TREC_FINE	ALBERT	0.047	0.038	**0.058**	0.053	0.047	0.046
	BERT	0.011	0.015	0.028	0.031	0.023	0.016
	DistilBERT	0.010	0.011	0.024	0.025	0.031	0.013
	RoBERTa	0.022	0.013	0.017	0.023	0.020	0.018

Tabelle A.4: Durchschnittlicher Makro-F1-Score nach der fünften Iteration aus fünf Wiederholungen für das natürliches-Szenario.

Stichwort-Szenario

Datensatz	Modell	Coreset	DAL	Dropout	Ensemble	Uncertainty	Zufall
AG_NEWS	ALBERT	0.821	0.828	0.831	0.828	0.817	0.828
	BERT	0.798	0.846	0.843	0.829	0.800	0.801
	DistilBERT	0.844	0.850	0.834	0.842	0.843	0.841
	RoBERTa	0.833	0.843	0.771	**0.853**	0.731	0.827
AG_NEWS_IMB	ALBERT	0.690	0.807	0.819	0.809	0.675	0.779
	BERT	0.792	0.796	0.609	0.777	0.775	0.753
	DistilBERT	0.809	0.820	0.820	**0.823**	**0.823**	0.804
	RoBERTa	0.672	0.818	0.803	0.819	0.755	0.791
DBPEDIA	ALBERT	0.954	0.963	0.921	0.932	0.931	0.962
	BERT	0.944	**0.969**	0.909	0.866	0.929	0.964
	DistilBERT	**0.969**	**0.969**	0.956	0.895	0.943	0.962
	RoBERTa	0.930	0.955	0.936	0.904	0.949	0.942
DBPEDIA_IMB	ALBERT	0.963	0.965	0.924	0.912	0.941	0.942
	BERT	0.949	0.909	0.908	0.942	0.930	0.900
	DistilBERT	**0.972**	0.938	0.954	0.902	0.932	0.938
	RoBERTa	0.955	0.943	0.934	0.919	0.924	0.921
IMDB	ALBERT	0.638	0.705	0.635	0.690	0.660	0.675
	BERT	0.697	0.676	0.729	0.719	0.670	0.766
	DistilBERT	0.769	0.783	0.772	0.774	0.776	0.770
	RoBERTa	0.776	**0.824**	0.806	0.811	0.723	0.822
IMDB_IMB	ALBERT	0.547	0.588	0.565	0.558	0.566	0.536
	BERT	0.491	0.676	0.573	0.607	0.546	0.583
	DistilBERT	0.575	0.667	0.694	0.681	0.705	0.677
	RoBERTa	0.543	**0.738**	0.626	0.720	0.719	0.683
ISEAR	ALBERT	0.417	0.206	0.295	0.326	0.392	0.310
	BERT	0.212	0.310	0.283	0.288	0.267	0.204
	DistilBERT	0.524	**0.541**	0.445	0.516	0.498	0.443
	RoBERTa	0.124	0.105	0.111	0.092	0.074	0.073
ISEAR_IMB	ALBERT	0.364	0.383	0.393	0.293	0.178	0.271
	BERT	0.270	0.271	0.180	0.069	0.208	0.300
	DistilBERT	**0.493**	0.472	0.432	0.440	0.451	0.448
	RoBERTa	0.169	0.108	0.188	0.096	0.090	0.077
TREC_COARSE	ALBERT	0.731	0.576	0.602	0.626	0.511	0.801
	BERT	0.232	0.547	0.419	0.469	0.296	0.621
	DistilBERT	0.657	0.680	0.679	**0.775**	0.762	0.666
	RoBERTa	0.190	0.284	0.217	0.309	0.415	0.184

Tabelle A.5: Durchschnittlicher Makro-F1-Score nach der fünften Iteration aus fünf Wiederholungen für das Stichwort-Szenario.

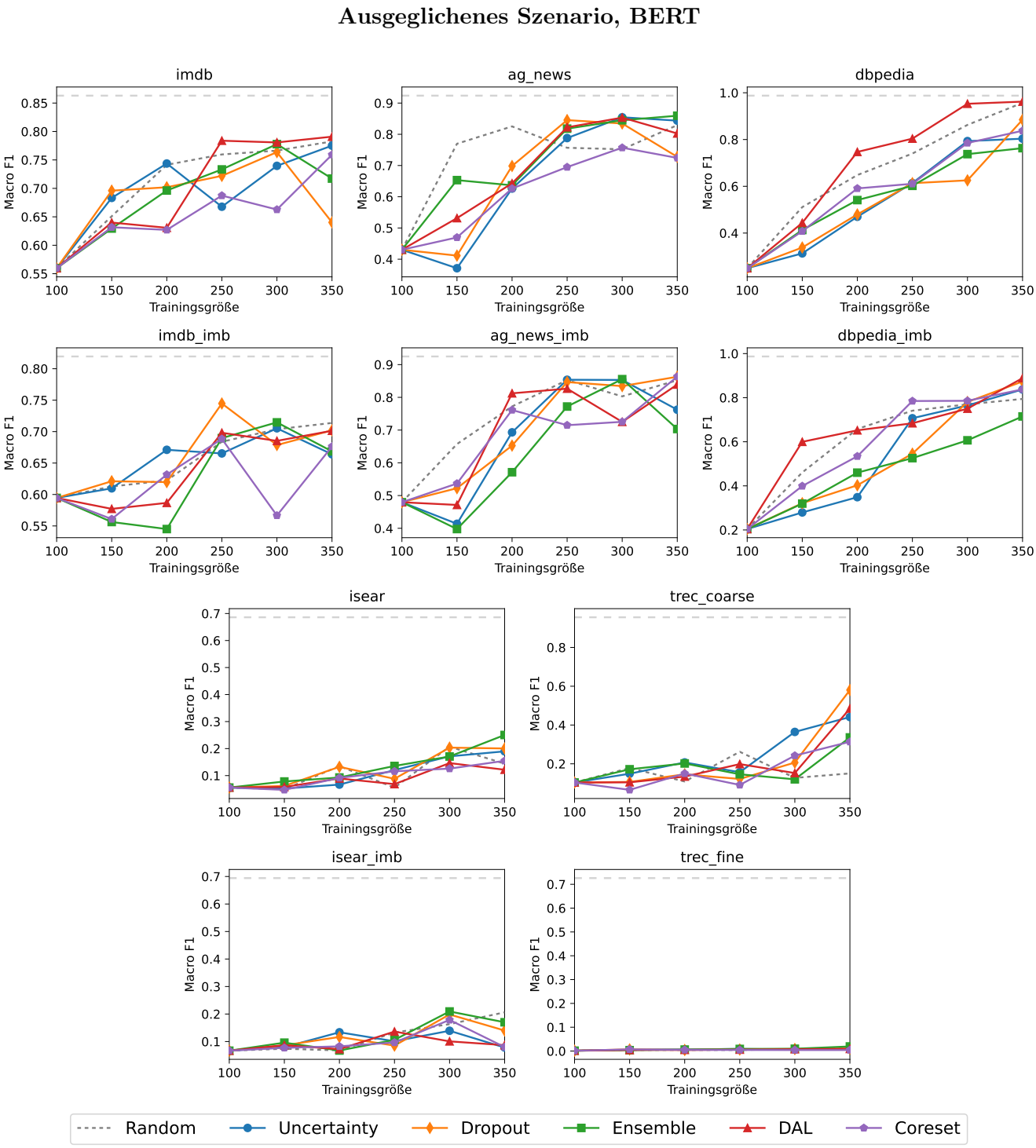

Abbildung A.1: Active Learning Strategien für BERT bei einer ausgeglichenen Startmenge.

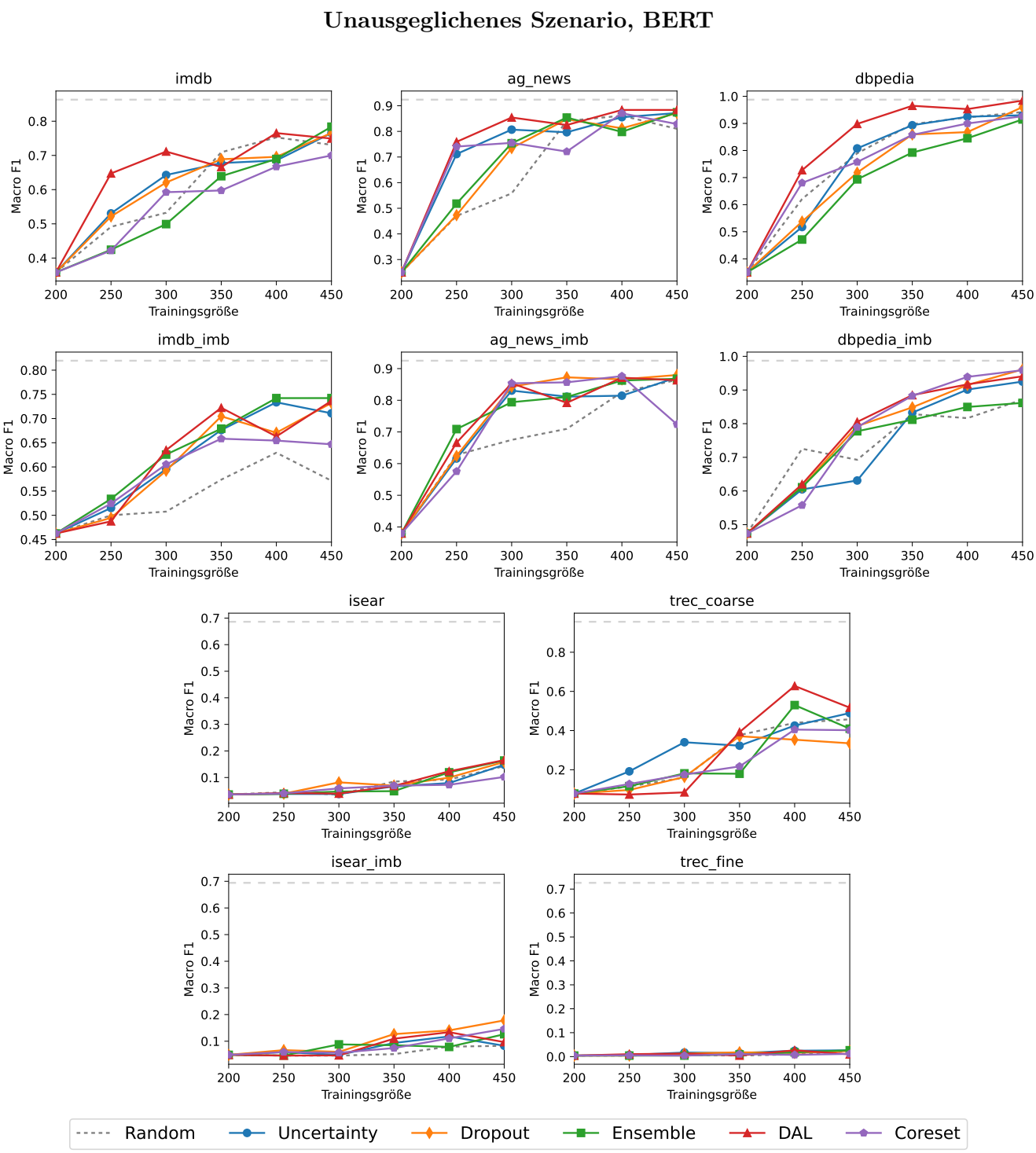

Abbildung A.2: Active Learning Strategien für BERT bei einer unausgeglichenen Startmenge.

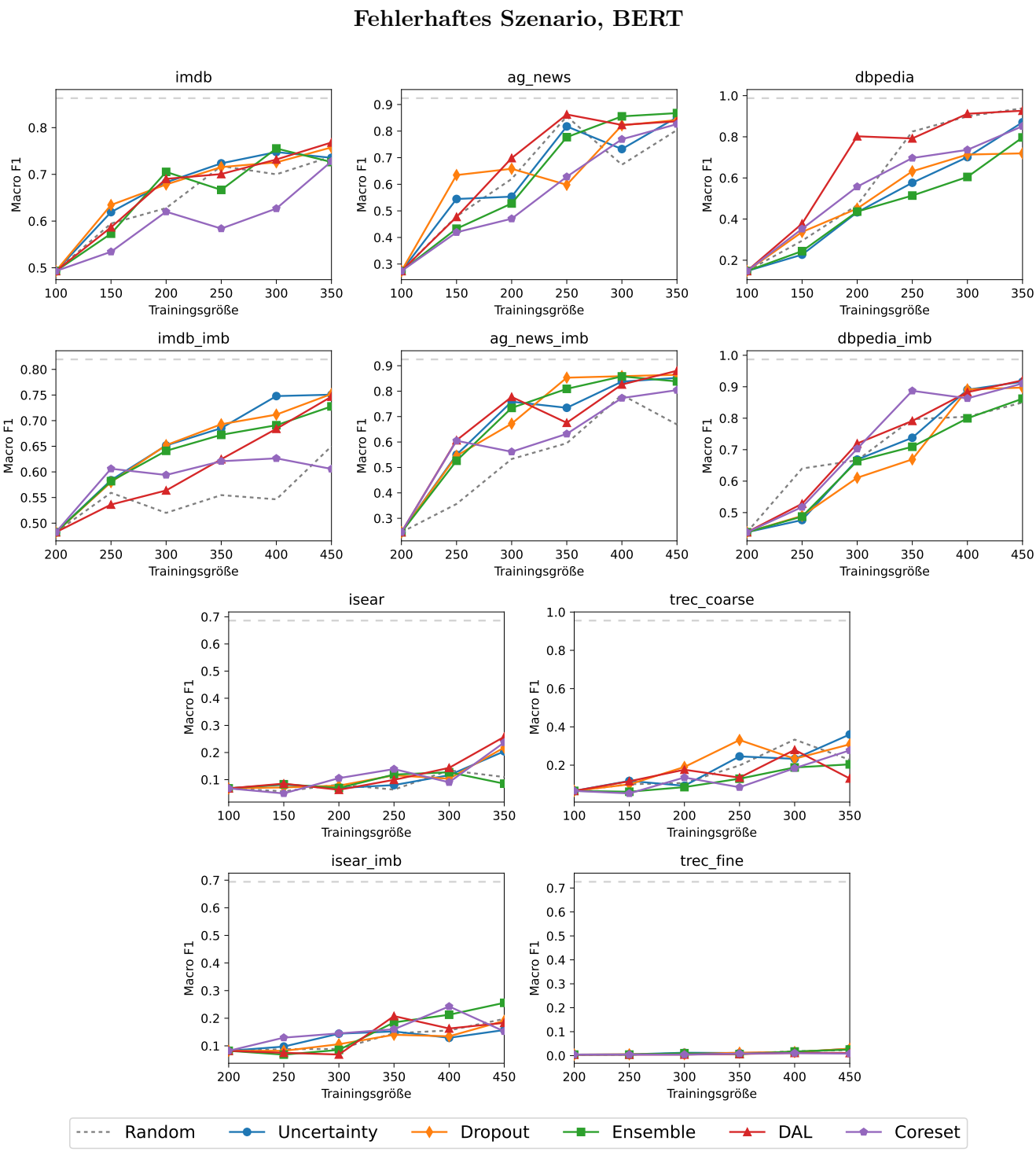

Abbildung A.3: Active Learning Strategien für BERT bei einer fehlerhaften Startmenge.

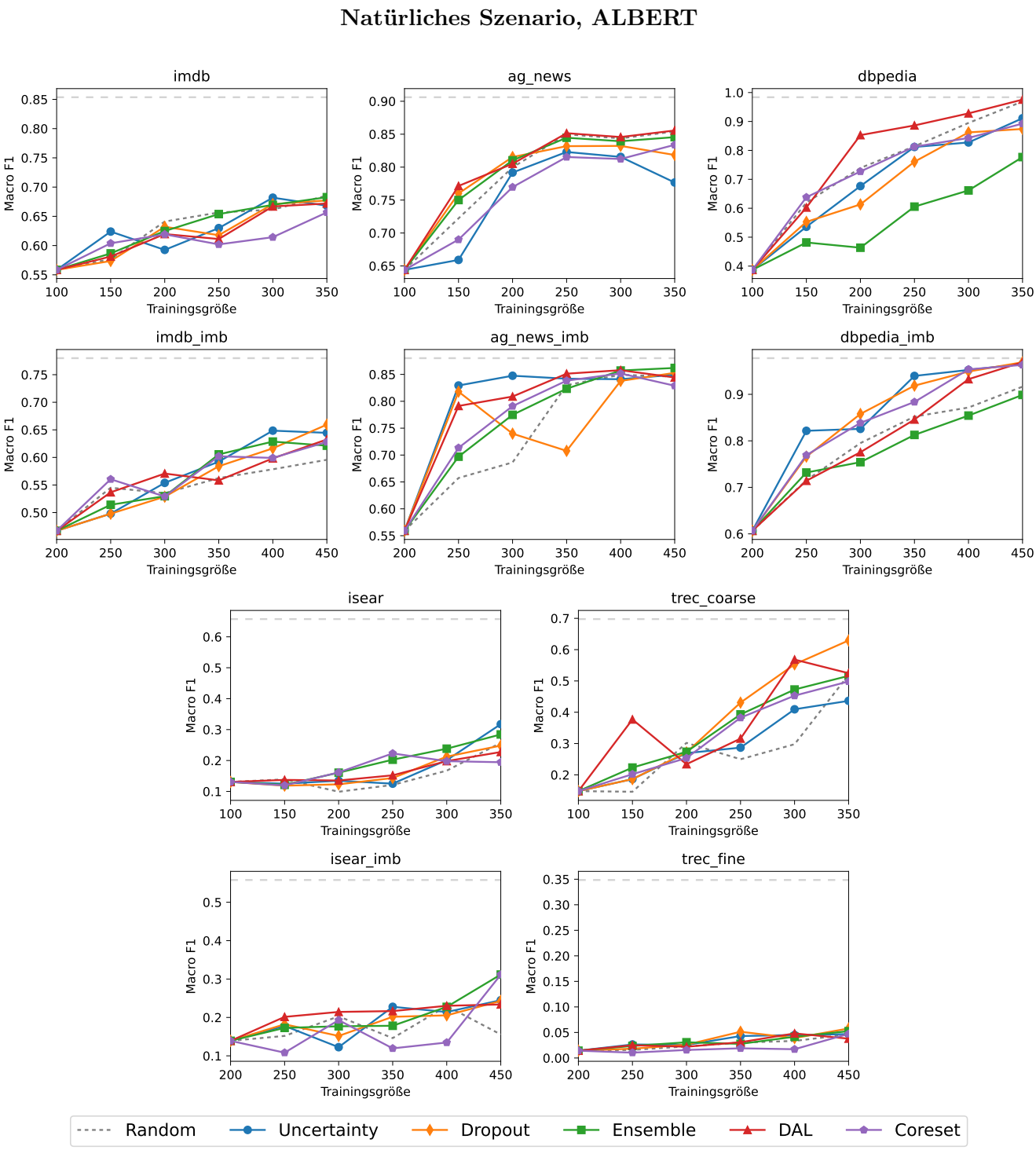

Abbildung A.4: Active Learning Strategien für ALBERT bei einer natürlichen Startmenge.

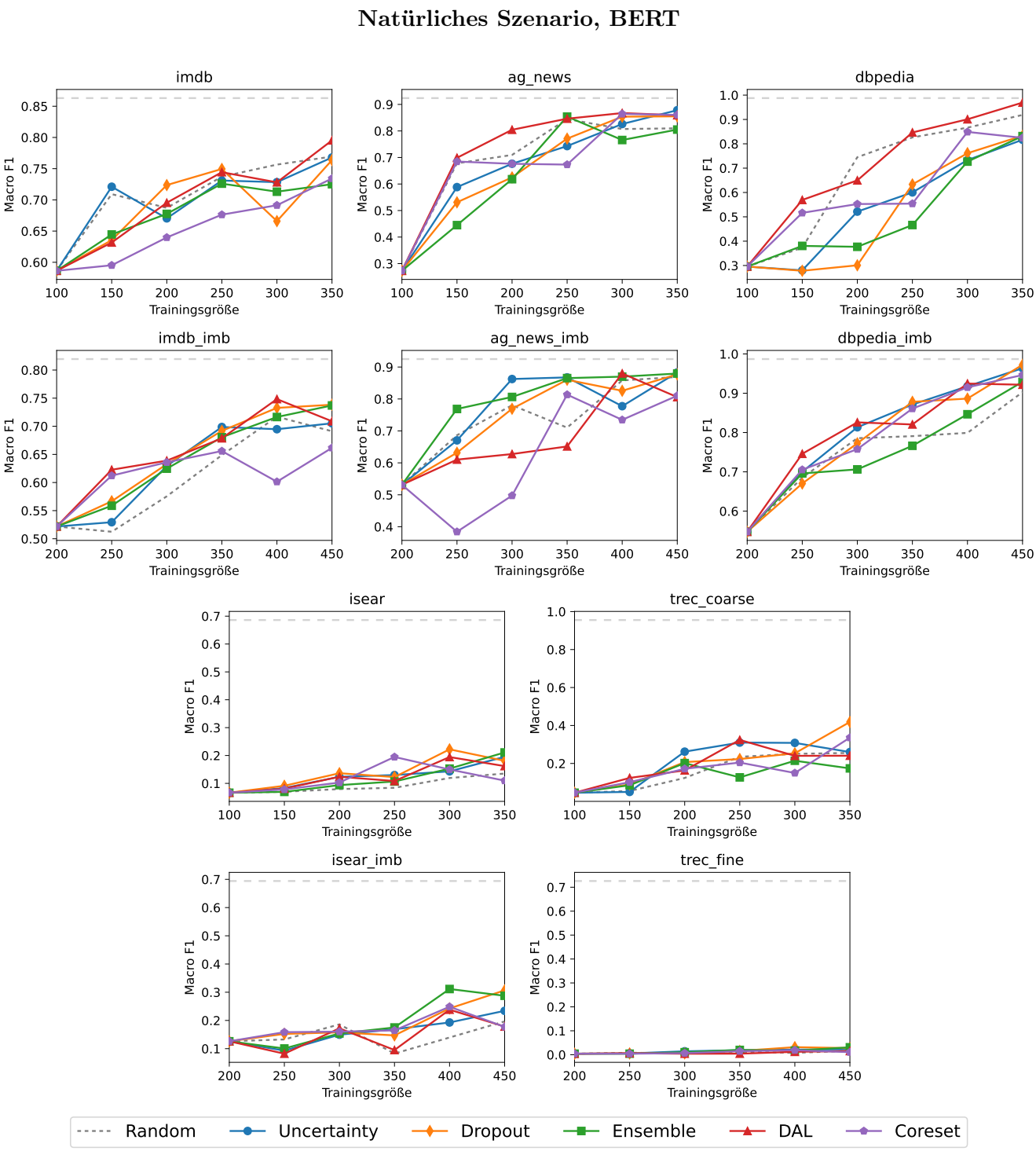

Abbildung A.5: Active Learning Strategien für BERT bei einer natürlichen Startmenge.

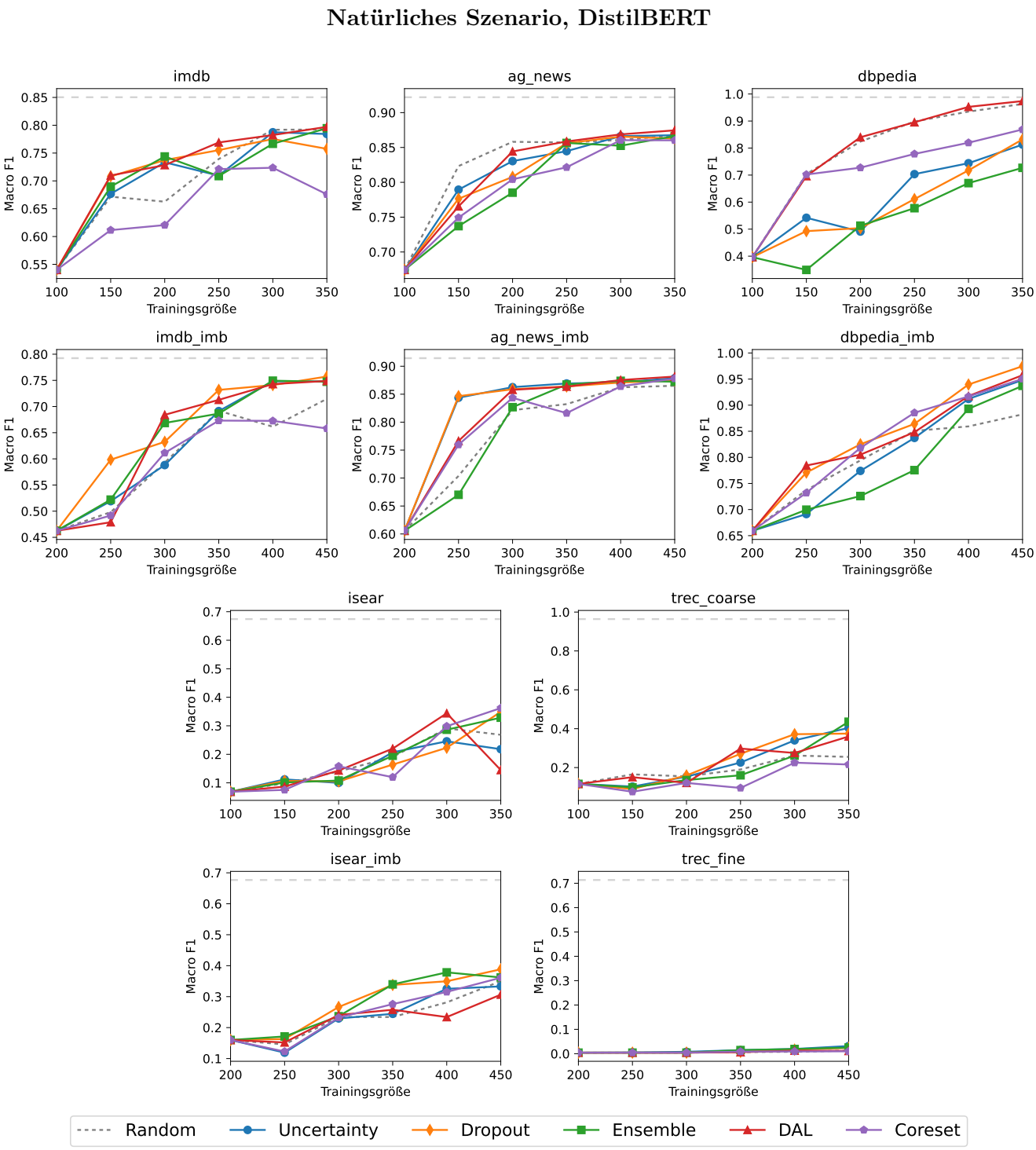

Abbildung A.6: Active Learning Strategien für DistilBERT bei einer natürlichen Startmenge.

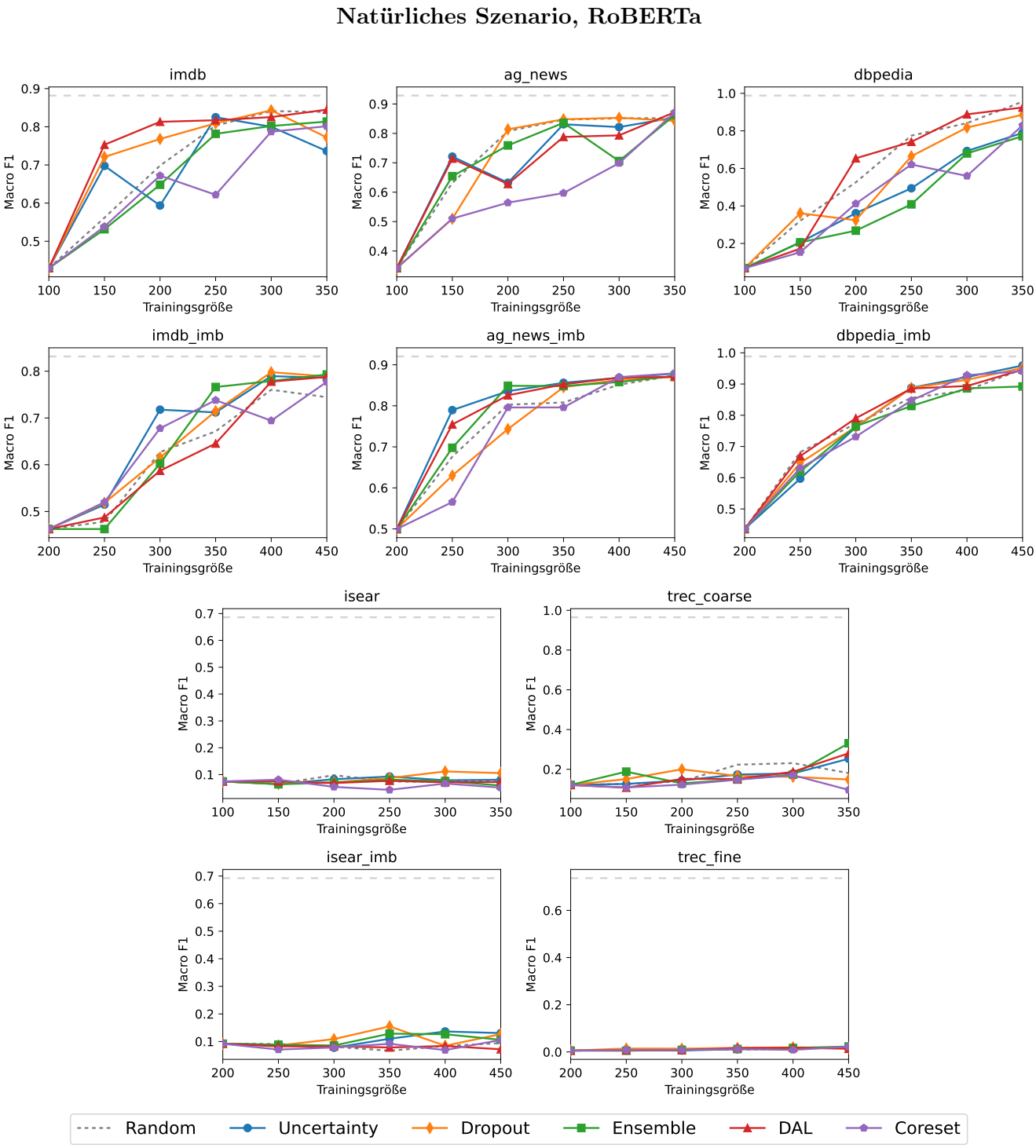

Abbildung A.7: Active Learning Strategien für RoBERTa bei einer natürlichen Startmenge.

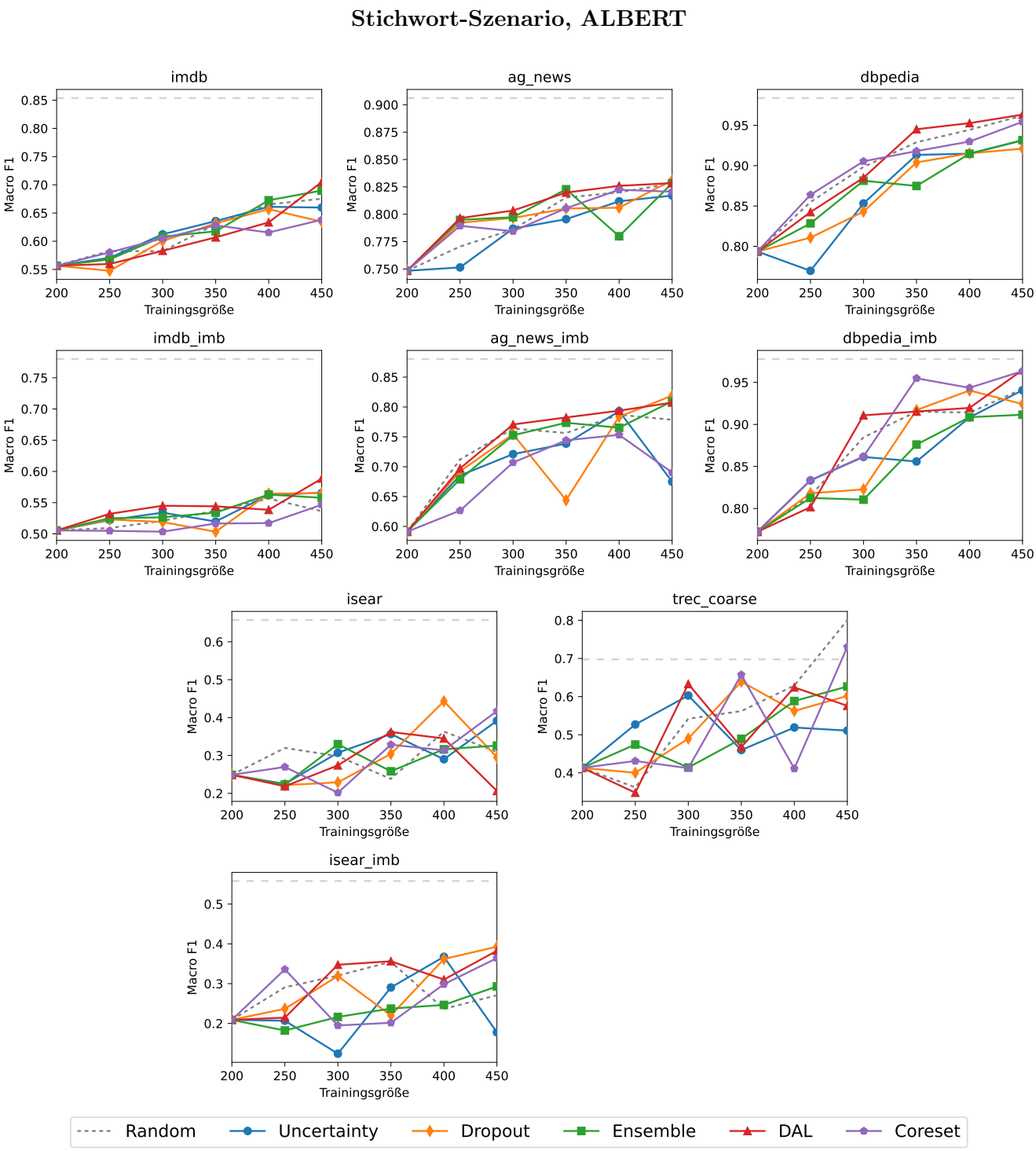

Abbildung A.8: Active Learning Strategien für ALBERT bei einer Stichwort-Startmenge.

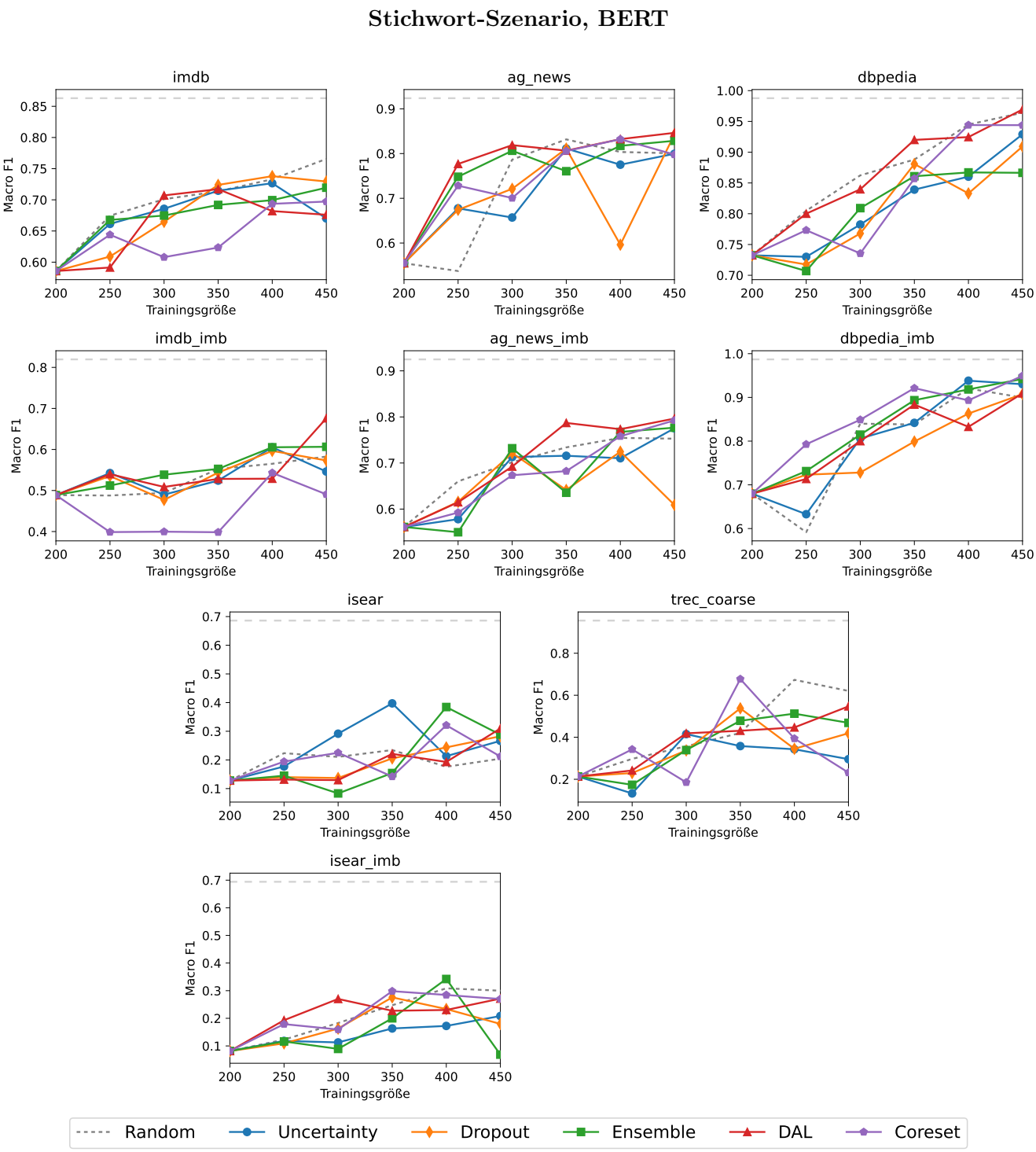

Abbildung A.9: Active Learning Strategien für BERT bei einer Stichwort-Startmenge.

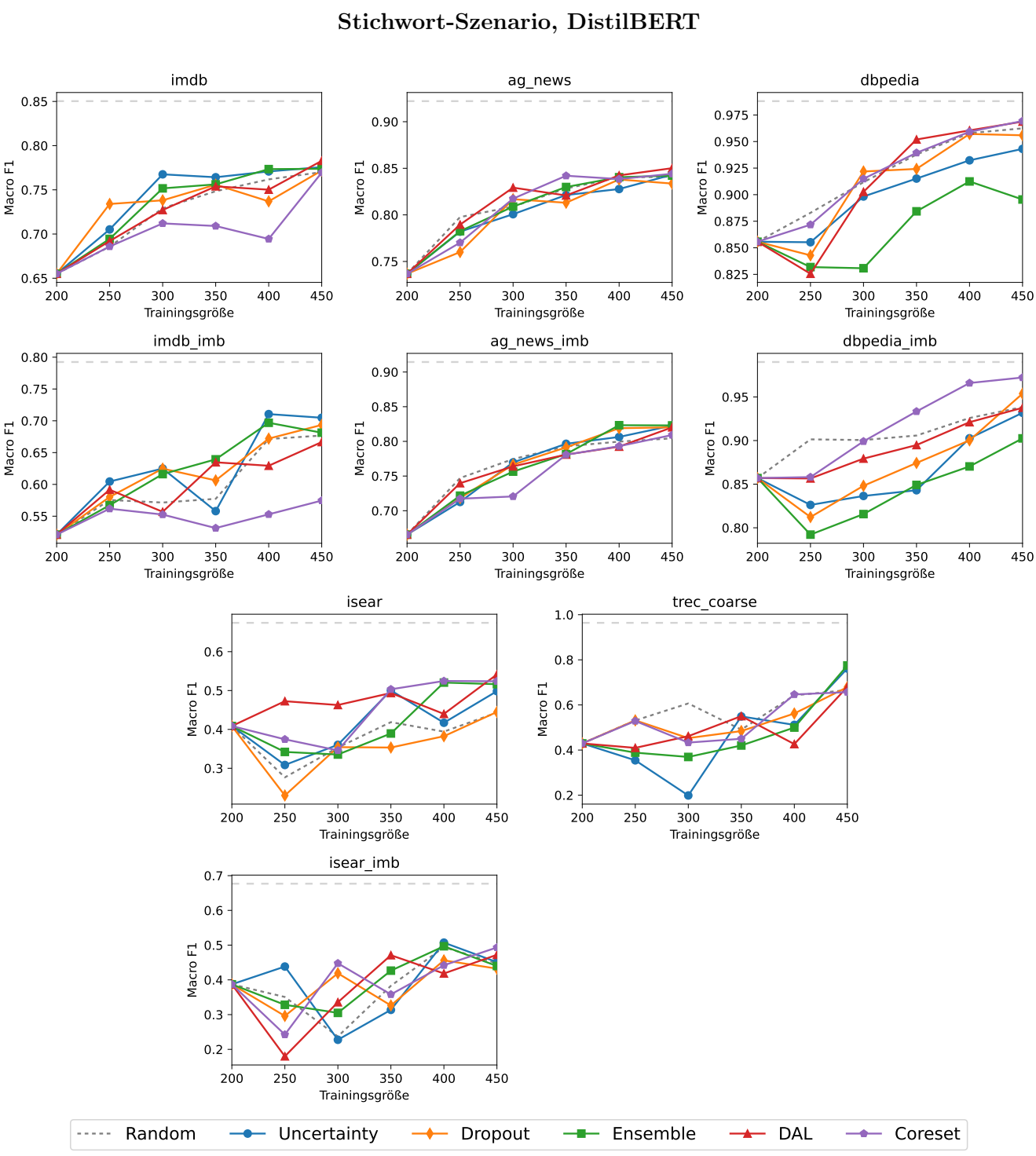

Abbildung A.10: Active Learning Strategien für DistilBERT bei einer Stichwort-Startmenge.

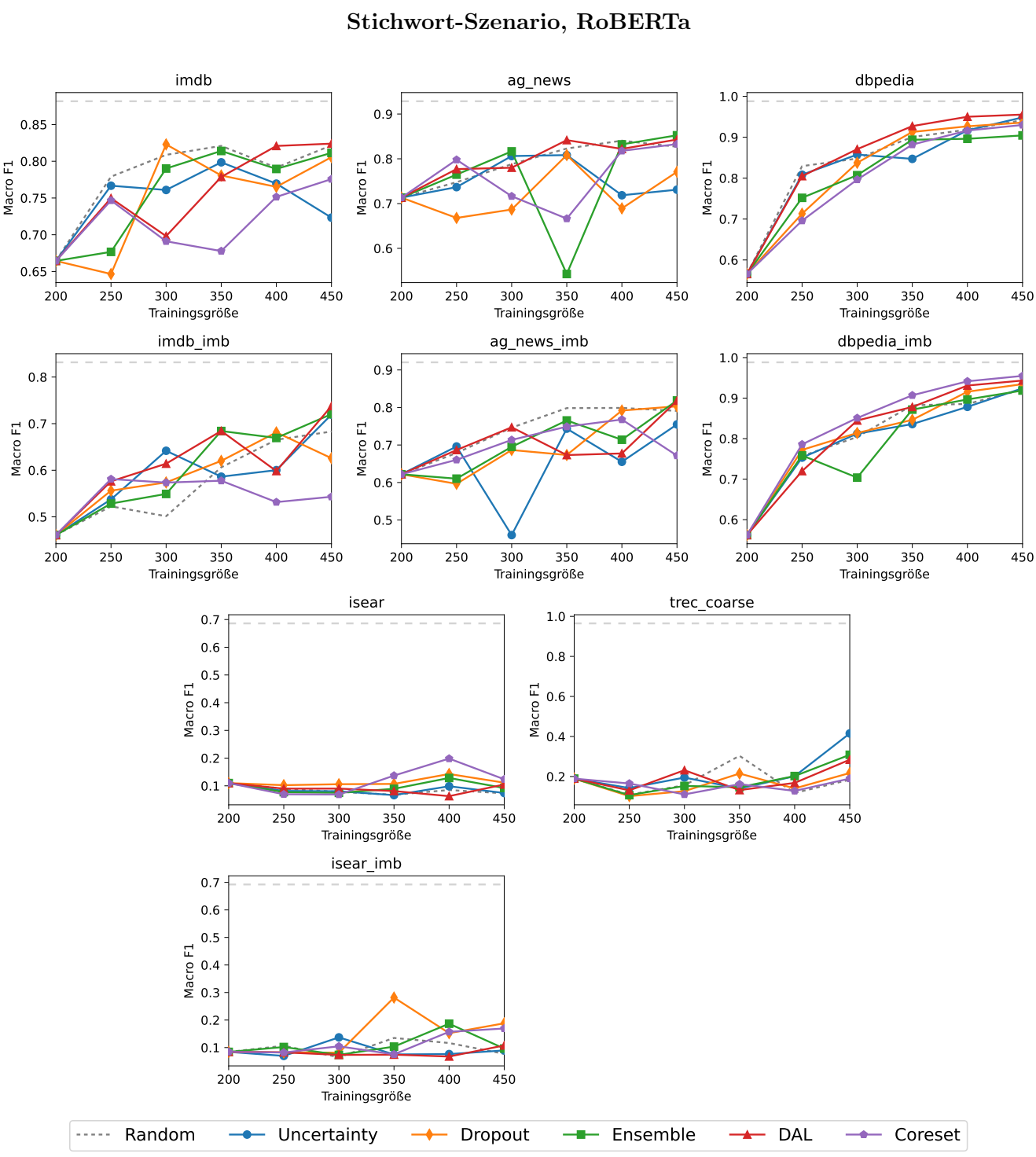

Abbildung A.11: Active Learning Strategien für RoBERTa bei einer Stichwort-Startmenge.

Erklärung zur selbstständigen Bearbeitung einer Abschlussarbeit

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich gemacht.

Ort	Datum	Unterschrift im Original