Martin-Luther-Universität Halle-Wittenberg
Naturwissenschaftliche Fakultät III
Studiengang Informatik

Multi-Task-Learning mit Transformer Modellen für die Ad-Hoc Suche mittels Information-Retrieval Axiomen

Bachelorarbeit

Adrien Klose

1. Gutachter: Prof. Dr. Matthias Hagen
2. Gutachter: M.Sc. Maik Fröbe

Datum der Abgabe: 15. November 2021

Erklärung

Hiermit versichere ich, dass ich diese Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Halle (Saale), 15. November 2021

. . . . . .
Adrien Klose

Zusammenfassung

Die Effektivität von Learning-to-Rank Ansätzen ergibt sich aus den Wechselwirkungen zwischen den induktiven Bias des Ansatzes und den verwendeten Trainingsdaten. Der induktive Bias ist eine Menge an getroffenen Hypothesen und bestimmt die Eignung des Ansatzes für eine Aufgabe. Die getroffenen Hypothesen definieren die Grenzen der erlernbaren Zusammenhänge aus den Trainingsdaten. Mit den wachsenden Anforderungen an Learning-to-Rank Ansätze werden auch stetig qualitativere und quantitativere Trainingsdaten und geeignetere Hypothesen benötigt. Information-Retrieval Axiome formalisieren als Hypothesen die erwarteten Eigenschaften von einem qualitativ hochwertigen Ranking und wurden bereits erfolgreich als Teil des induktiven Bias für die Regularisierung von Verlustfunktionen und die Konstruktion von Trainingsdaten eingesetzt. Wir kombinieren die Konstruktion von Trainingsdaten mit dem Einsatz von Information-Retrieval Axiomen als weitere getroffene Annahmen, indem wir mittels Multi-Task-Learning einem vortrainierten Transformer Modell das Ranken von Dokumenten mit der Bestimmung von Axiom-Präferenzen als Hilfsaufgabe antrainieren. Die Axiome fungieren als induktiver Bias in unserem Ansatz sowohl bei der Konstruktion der Trainingsdaten für die IR Axiom Aufgabe sowie gegen das Overfitting auf eine spezifische Aufgabe durch das gleichzeitige Trainieren des Transformer Modells auf die IR Axiom und Ranking Aufgabe. Durch das gleichzeitige Trainieren des Transformer Modells auf die IR Axiome und Ranking Aufgabe muss das Transformer Modell Repräsentationen für Anfragen und Dokumente bestimmen können, aus denen sowohl der Rank als auch das IR Axiom berechnet werden kann. Unsere Experimente auf verschiedenen Axiom-Kombinationen zum Multi-Task-Learning mit einer Ranking Aufgabe zeigen, dass das Multi-Task-Learning der Ranking Aufgabe mit den IR Axiomen REG und ANTI-REG als Hilfsaufgaben die Effektivität der Ranking Ergebnisse auf dem Test-Datensatz des TREC Deep Learning Track aus dem Jahr 2020 gegenüber einem alleinigen lernen der Ranking Aufgabe um im Mittel 5% erhöht.

Inhaltsverzeichnis

1 Einleitung **1**

2 Related Work **5**
2.1 Grundlagen Information-Retrieval 5
2.2 Transformer Modelle 9
2.3 Multi-Task-Learning 11
2.4 Grundlagen IR Axiome 12
2.5 Anwendungsbeispiele für IR Axiome 15

3 Information-Retrieval Axiome für Multi-Task Learning **18**
3.1 Multi-Task-Learning Paradigma 18
3.2 Verwendete IR Axiome und Anpassungen 20
3.2.1 Termhäufigkeits-Axiome 22
3.2.2 Längen-Normierungs-Axiome 24
3.2.3 Semantische-Ähnlichkeits-Axiome 24
3.2.4 Anfrage-Facetten-Axiome 25
3.2.5 Term-Nachbarschafts-Axiome 26
3.2.6 Retrieval-Score-Axiome 28

4 Veruchsaufbau **30**
4.1 Datensätze 30
4.2 Implementations-Details 32
4.3 Auswahl der Experimente 34

5 Evaluation **36**
5.1 Evaluationsmetriken 36
5.2 Experiment Ergebnisse 38

6 Fazit **56**

A Ergebnisse aller Experimente **58**
A.1 Ranking Baselines 58
A.2 Ergebnisse Single-Task-Setup 59
A.2.1 Ranking Aufgabe 59
A.2.2 Axiomen Aufgaben Eingabe Variante A 62
A.2.3 Axiomen Aufgaben Eingabe Variante B 73
A.3 Ergebnisse Experiment 1-A 84
A.4 Ergebnisse Experiment 1-B 96
A.5 Ergebnisse Experiment 2 108
A.6 Ergebnisse Experiment 3 111
A.7 Ergebnisse Experiment 4 114
A.8 Ergebnisse Experiment 5 117

Literaturverzeichnis **122**

Danksagung

Vor allem möchte ich mich bedanken bei M.Sc. Maik Fröbe und M.Sc. Alexander Bondarenko für die Grundlagen dieser Bachelorarbeit und die Unterstützung bei der Erstellung der Bachelorarbeit. Danken möchte ich auch den Autoren der Webis-Thesis-Vorlage für Ihre handliche und sehr gute Bachelorarbeit-Vorlage. Vielen Dank auch an Jonas Friedrich Tristan Lochmann und Daniel Wächtler, die mich durch das Studium begleitet haben und mir auch in dieser Bachelorarbeit mit Rat zur Seite standen. Zu guter Letzt möchte ich meiner Mutter Helgard Klose danken, die meine größte emotionale Stütze ist.

Kapitel 1

Einleitung

Mit mehr als 3.5 Milliarden Anfragen pro Tag an Google[1] ist die Verbesserung von Suchmaschinen nicht nur von ökonomischen Interesse für Konzerne, sondern auch von Interesse für die Menschheit als Ganzes. Mit der stetig wachsenden Menge an Daten, die global durch das Internet zur Verfügung gestellt werden, ist eine manuelle Suche nach Informationen ohne die Unterstützung einer Suchmaschine praktisch unmöglich. Forschungsergebnisse der letzten Jahre [15, 20, 42, 81] zeigen mit einer überwiegenden Mehrheit, dass Learning-to-Rank Ansätze effektivere Rankings als klassische Suchmodelle wie BM25 [67] erzielen. Mit der Entwicklung von Feature basierten Learning-to-Rank Ansätzen hinzu den aktuellen vortrainierten Transformer Modellen für immer effektivere Rankings zeigt sich nicht nur das Potenzial den induktiven Bias von Modellen stetig zu überdenken, sondern auch die wachsende Bedeutung von quantitativeren und qualitativeren Trainingsdaten, um die Grenzen des mit dem induktiven Bias erreichbaren auszureizen [51].

Die Erstellung von großen und qualitativ hochwertigen Trainingsdaten für Learning-to-Rank Ansätze aus selbst protokollierten Websuche-Click-Logs ist durch das gefestigte Monopol der Websuchanfragen durch wenige Konzerne[2] für unabhängige Gruppen und Organisationen nur äußerst eingeschränkt möglich. Google und weitere Konkurrenten wie Bing veröffentlichen aus rechtlichen Gründen nur selten und meist auch nur kleine anonymisierte Teile ihrer Suchanfrage-Logs, so dass die Forschung mit diesen wenigen Trainingsdaten und auf alternativen Wegen erstellten Trainingsdaten auskommen muss. Damit Learning-to-Rank Ansätzen, die auf vortrainierten Transformer Modellen basieren, mit weniger Relevanz-Trainingsdaten schneller und mit einem höheren Wert bezüglich der Effektivität konvergieren, untersuchen wir die Verbesserung

[1] `https://www.internetlivestats.com/google-search-statistics/#rate`

[2] `https://de.statista.com/statistik/daten/studie/222849/umfrage/marktanteile-der-suchmaschinen-weltweit/`

der Dokument- und Anfragerepräsentationen eines auf BERT [19] basierenden Learning-to-Rank Ansatzes mittels Multi-Task-Learning mit Information-Retrieval (IR) Axiom-Präferenzen als beliebig skalierbare Hilfsaufgabe. Unser Ziel ist es, die in BERT vortrainierte allgemeine Repräsentation von Sprache auf den Bereich des Information-Retrieval mit allgemeinen Repräsentationen von Sprache und Relevanz zu spezialisieren.

Das Paradigma vom Learning-to-Rank lässt sich am realitätsnahen Beispiel des Lernens, Informationen auf einer Wikipedia-Seite zu finden, verdeutlichen. Betrachten wir das Lernen vom Finden von Informationen auf einer Wikipedia-Seite als die individuellen Aufgaben die generelle Struktur der Webseite einzuschätzen, relevante Text-Abschnitt zu ranken und dann aus Text-Abschnitten die gesuchten Informationen zu extrahieren, dann werden wir deutlich langsamere und schlechtere Lernerfolge in den einzelnen Aufgaben erzielen, als wenn wir Versuchen Informationen über sprachliche und strukturelle Zusammenhänge, die uns beim allgemeinen Einschätzen der Struktur der Webseite und Ranken relevanter Passagen helfen, auf das Extrahieren von Informationen aus Text-Abschnitten anzuwenden und vice versa. Multi-Task-Learning als Technik wurde schon erfolgreich für Natural-Language-Understanding Aufgaben mittels eines Multi-Task-Deep-Neural-Networks (MT-DNN)[46] und die Erstellung des Text-To-Text Transfer Transformer (T5)[64] eingesetzt. Mit dem Multitask Unified Model[3] entwickelt Google derzeit einen Multi-Task-Learning Ansatz zur Verbesserung ihrer Suchmaschine, welcher komplexe Informationsbedürfnisse in Anfragen basierend auf T5 erkennt und geeignet beantwortet. Wir folgen in unserem Ansatz der Methodik von Liu et al. [46], so dass wir unsere Untersuchungen mit BERT [19] als Transformer Modell durchführen und die gewonnen Erkenntnisse versuchen auf das State of the Art Transformer Modell T5 in einer zukünftigen Arbeit zu übertragen.

Problematisch am Multi-Task-Learning mit verschiedenen Aufgaben, wie es Liu et al. [46] praktizieren, ist die Beschränkung der Skalierbarkeit des Setups durch die nur begrenzt zur Verfügung stehenden Trainingsdaten für jede einzelne Aufgabe. Es ist im Allgemeinen unklar, wie sich eine ungleichmäßige Skalierung der Menge an Trainingsdaten für die einzelnen Aufgaben auf die Effektiviät des Muti-Task-Learnings im Ganzen auswirkt [16]. Wir verwenden in unserem Ansatz neben der Ranking Aufgabe nur die IR Axiom-Präferenzen als Hilfsaufgabe und entscheiden uns bewusst gegen verwandte aber schlechter skalierbare Information-Retrieval Aufgaben wie Query Suggestion [1] und Reading Comprehension [53]. Der Einsatz von IR Axiomen zur Erstellung [23] und Verbesserung [4] von klassischen Suchmodellen ist bereits erprobt. Die direkte Anwendung von IR Axiomen auf Learnin-to-Rank Ansätze ist jedoch insofern

[3] `https://blog.google/products/search/introducing-mum/`

unmöglich, dass Learning-to-Rank Ansätze keine geschlossene mathematische Formel für die Bestimmung der Relevanz von Dokumenten haben und somit die Einhaltung von IR Axiomen durch eine Dekonstruktion der Relevanzfunktion nicht geprüft werden kann. Wir verwenden deshalb in unseren Ansatz die Definition von IR Axiomen als Hypothesen über geschlossen berechenbare Paarweise-Dokumentpräferenzen von Hagen et al. [34]. Die Menge an Trainingsdaten für die IR Axiom-Präferenz Aufgaben ist durch die Bestimmung eines Eintrages aus einer beliebigen Anfrage und zwei beliebigen Dokumenten mit einem minimalen Rechenaufwand skalierbar. Insgesamt wirken die IR Axiom-Präferenz Aufgaben im Multi-Task-Learning somit gegen das Overfitten auf eine spezifische Aufgabe [68] und trainieren dem gemeinsamen Encoder der Aufgaben eine Art und Weise an die Sprache von Dokumenten und Anfragen zu repräsentieren, die mehr Facetten von Relevanz abdeckt. Wir übertragen somit die Ergebnisse von Liu et al. [46], vortrainierte Transfomer Modelle und Multi-Task-Learning in Kombination im Bereich des Natural-Language-Understanding einzusetzen, unter Zuhilfenahme der Axiom-Präferenz Definitionen von Hagen et al. [34] auf den Bereich des Information-Retrieval und Vermeiden die Ungewissheit der Auswirkung ungleicher Skalierungen der Trainingsdaten auf die Effektivität des Multi-Task-Learning. Durch die beliebige Skalierung der IR Axiom-Präferenz Aufgaben ist es immer möglich, ein optimales Verhältnis der Trainingsverhältnisse bezüglich einer weiteren Information-Retrieval Aufgabe wie dem Ranken von Dokumenten zu bestimmen, die die Effektivität der weiteren Information-Retrieval Aufgabe optimiert.

Unser Learning-to-Rank Ansatz mit Multi-Task Learning mittels IR Axiomen hat somit vor allem die Spezialisierung des induktiven Bias des BERT Modells, von allgemeinen Repräsentationen für Sprache hinzu Repräsentationen für Dokumente, Anfragen und vielfältige Facetten der Relevanz zum Ziel. Da das BERT Modell gleichzeitig auf die Ranking und die IR Axiom-Präferenz Aufgaben trainiert wird, muss das BERT Modell in den gemeinsam verwendeten Encoding-Schichten Anfrage- und Dokument-Repräsentationen bestimmen, aus denen in aufgabenspezifischen Decoding-Schichten sowohl die Relevanz für das Ranking als auch die IR Axiom-Präferenz bestimmt werden kann. Dadurch das jede IR Axiom-Präferenz Aufgabe eine andere Facette von Relevanz repräsentiert, muss das BERT Modell jede einzelne dieser Facetten im gleichzeitigen Training erlernen, um dann am Ende des Trainings Repräsentationen für Dokumente und Anfragen erstellen zu können, aus denen jede der Facetten von Relevanz bestimmt werden kann. Wir untersuchen den Einfluss von 20 IR Axiomen auf die Ranking Aufgabe im Multi-Task-Learning, indem wir die Auswahl an IR Axiomen und das Verhältnis von Trainingsdaten für die IR Axiom Aufgaben zu Trainingsdaten für die Ranking Aufgabe variieren.

Wir trainieren unser BERT Modell auf dem Microsoft Machine Reading

Comprehension (MS MARCO) Passagen Datensatz [52] für die Ranking Aufgabe und die IR Axiom-Präferenz Aufgabe auf aus dem MS MARCO Passagen Datensatz automatisch erzeugten IR Axiom-Präferenzen. Der MS MARCO Passagen Datensatz ist aus dem MS MARCO Question Anwsering Datensatz entstanden, besteht aus 8.8 Millionen Passagen und bietet Anfragen mit positiven Relevanzlabels für Learning-to-Rank Ansätze. Die Effektivität unseres Ansatzes evaluieren wir mittels den Relevanzbewertungen vom National Institute of Standards and Technology (NIST) für den TREC Deep Learning Track aus dem Jahr 2019 (TREC19) und 2020 (TREC20) bezüglich NDCG@10, MRR@10 und P@1. Die von uns durchgeführten Experimente zeigen, dass eine Verwendung von 20 IR Axiom-Präferenz Aufgaben mit einer Ranking Aufgabe im Multi-Task-Learning im Trainingsdaten Verhältnis $1 : \cdots : 1$ die Effektivität der Ranking Aufgabe bezüglich dem Test-Datensatz von TREC19 und TREC20 um mehr als 10% senkt. Für die Auswahl der IR Axiom-Präferenz Aufgaben REG und ANTI-REG im Trainingsdaten Verhältnis 1:1:4 zur Ranking Aufgabe konnten wir zeigen, dass das Multi-Task-Learning die Effektivität der Ergebnisse für P@1 um 3%, NDCG@10 um 5% und MRR@10 um 7% auf TREC20 steigert. Wir werden unsere Untersuchung um weiterer Trainingsdatenverhältnisse, weiteren Axiom-Kombinationen, neue Axiome und einer Skalierung der Trainingsdaten für die IR Axiom-Präferenz und Ranking Aufgaben in zukünftigen Arbeiten erweitern.

Kapitel 2

Related Work

In den nachfolgenden Abschnitten dieses Kapitels betrachten wir für unsere Untersuchungen relevante grundlegende Begriffe und Entwicklungen aus dem Bereich des Information-Retrieval. Folgend betrachten wir Transformer Modelle und ausgewählte Ansätze zur Modifikation und Erweiterung von Transformer Modellen. Anschließend geben wir einen kurzen Überblick über Möglichkeiten des Multi-Task-Learnings als eine Technik des maschinellen Lernens und gehen auf einzelne Anwendungsbeispiele von Multi-Task-Learning im Bereich des Information-Retrieval und dem verwandten Bereich des Natural-Language-Processing ein. Danach betrachten wir das theoretische Fundament von Information-Retrieval Axiomen. Zum Schluss gehen wir auf Forschungsarbeiten ein, die den Einsatz von Information-Retrieval Axiome untersuchen.

2.1 Grundlagen Information-Retrieval

Am Anfang von allem menschlichen Handeln steht ein Bedürfnis.[1] Die Wissenschaft des Information-Retrieval beschäftigt sich mit Techniken, die das Erfüllen der Informationsbedürfnisse von Nutzern ermöglicht. Dabei formuliert der Nutzer sein Informationsbedürfnis in Form einer Suchanfrage aus, die aufgrund von einem Mangel an Verständnis des Nutzers über das eigene Informationsbedürfnis und der Mannigfaltigkeit von Sprache und Informationen nur in den einfachsten Fällen das Informationsbedürfnis nahezu perfekt widerspiegelt [6]. Um das Informationsbedürfnis des Nutzers zu befriedigen, wird ein Ranking an vermuteten relevanten Dokumenten erstellt und dem Nutzer dargeboten. Ein Dokument bezieht sich im Allgemeinen nicht nur auf Text-Dokumente, auf die wir uns nachfolgend beschränken, sondern auch auf Bilder, Videos, strukturierte und unstrukturierte Texte sowie alle vorstellbaren Kom-

[1] `https://www.nobelprize.org/prizes/literature/1950/russell/lecture/`

binationen. Damit ein Ranking an vermuteten relevanten Dokumenten erstellt werden kann, bedarf es einer Suchfunktion und einem Index. Eine Suchfunktion S bestimmt für eine Suchanfrage q und ein Dokument d aus dem Index einen nummerischen Wert, der die vermutete Relevanz des Dokumentes für die Suchanfrage wiedergibt [69]. Die Relevanz ist ein abstrakter Wert auf einer Ordinalskala, der angibt, in welcher Reihenfolge Dokumente das Informationsbedürfnis einer Suchanfrage voraussichtlich befriedigen. Der Index besteht aus den Repräsentationen von Dokumenten, die im Prozess der Indexierung entstehen und dienen dem effizienten finden von relevanten Dokumenten [69]. Die Repräsentation der Dokumente kann unter anderem Informationen zu den Termhäufigkeiten $tf(t, d)$ eines Terms t im Dokument, zu der Position eines Terms im Dokument sowie der Dokumentlänge enthalten und im Index abgespeichert werden. Information-Retrieval bezeichnet insgesamt einen Prozess zur Gewinnung von Daten zur Befriedigung eines Informationsbedürfnisses.

Da ein Informationsverlust vom Informationsbedürfnis zur Suchanfrage und vom Bedürfnis Informationen festzuhalten zur Erstellung der Repräsentation eines Dokument stattfindet, wurden verschiedenste Techniken entwickelt, um Auswirkungen des Informationsverlustes abzumildern. Eine Technik ist die Verwendung von Stoppwort-Listen. Stoppwörter sind die Wörter einer Sprache, die zumeist nur einen syntaktischen aber keinen oder nur einen sehr geringen semantischen Wert haben [7]. Es sind dadurch die in der Regel häufigsten Wörter (im Englischen or, the, a, to, be etc.) einer Sprache [78], die in fast allen Dokumenten vorkommen und damit als Stoppwörter klassifiziert werden. Werden die Stoppwörter als irrelevante Informationen während der Indexierung entfernt, ergibt sich nicht nur Speicher bezogen ein kleinerer Index, sondern auch in den meisten Fällen eine höhere Effektivität im Information-Retrieval [71]. Die Wahl der Stoppwort-Liste und die Entscheidung, ob Stoppwörter in jedem Fall entfernt werden sollten, ist durch Randfälle wie dem bekannten Zitat von Shakespear „to be or not to be“ oder den Film „The Matrix“ eine Abwägung zwischen mehr nutzbaren Informationen und einem kleineren effizienteren Index. Da Speicherplatz mittlerweile in quasi unbeschränkten Mengen zu moderaten Preisen zur Verfügung steht und die Effektivität des Information-Retrieval im Fokus steht, werden im kommerziellen Einsatz von Suchmaschinen die Stoppwörter bei der Indexierung nicht entfernt und für jede einzelne Suchanfrage erneut entschieden, ob diese potenzielle Stoppwörter enthält und ob sie für bessere Ergebnisse entfernt werden sollten oder nicht [73]. Eine weitere Technik, die Auswirkungen des Informationsverlustes weiter abzuschwächen, ist das Stemming von Wörtern [38]. Nehmen wir beispielhaft die Suchanfrage „bunte Häuser“ und gehen von einer Suchfunktion aus die nach exakten Wortübereinstimmungen sucht, dann würde ein Dokument welches „buntes Haus“ oder andere ähnliche Formen der Terme der Suchanfrage ent-

hält, als nicht relevant eingeschätzt werden. In den meisten Fällen sollen solche Dokumente trotz des nicht exakten Übereinstimmens der Terme aufgrund der Deklination und Konjugation von Wörtern dennoch als relevant bewertet werden. Deswegen werden alle Wörter der Dokumente und Suchanfragen mittels Stammformreduktion auf ihren Wortstamm abgebildet, so dass in unserem Beispiel „buntes“ und „Häuser“ auf „bunt“ und „Haus“ abgebildet wird. Der Wortstamm ist dabei nicht immer ein reell existierendes Wort. Ähnlich den Stoppwort-Listen bedeutet auch Stemming im Allgemeinen einen Informationsverlust, der in Randfällen wie der Abbildung von „fisher, fished, fishing“ auf „fish“ zu irrelevanteren Ergebnissen führen kann. Deshalb ist es üblich die Terme vor und nach dem Stemming im Index abzuspeichern und beide Formen bei der Suche geeignet zu kombinieren und somit eine differenziertere Suche auf dem Index mittels einem größeren Speicheraufwand zu ermöglichen. Wir verwenden in unseren Untersuchungen den Porter-Stemmer [60]. Seit seiner Einführung und weiteren kleineren Anpassungen hat sich der Porter-Stemmer als üblichster und gut funktionierender Algorithmus für das Stemming englischer Wörter etabliert [79]. Der Algorithmus hinter dem Porter-Stemmer basiert auf der Idee, dass sich die Suffixe in der englischen Sprache aus mehreren kürzeren Suffixen zusammensetzen. Zur Worstammbildung mittels dem Porter-Stemmer definiert dieser eine Rangfolge an Verkürzungsregeln, die solange erneut angewendet werden, bis keine einzige Verkürzungsregel mehr angewandt werden kann. Wenden wir den Porter-Stemmer auf das Wort „differently“ an erhalten wir mit den einzelnen Zwischenschritten die Ableitungsfolge „differently, differentli, different, differ“.

Welchen konkreten Einfluss Stoppwörter, Stemming und extra Informationen im Index haben, wird durch die eingesetzte Suchfunktion bestimmt. Wir unterscheiden Suchfunktionen in klassische Suchfunktionen und Learning-to-Rank Ansätze. Klassische Suchfunktionen wie BM25 [67] und Query Likelyhood [12] berechnen für ein Dokument-Suchanfrage-Paar in kurzer Zeit eine Relevanzbewertung anhand von vorberechenbaren Eigenschaften wie Termhäufigkeiten, Spezifität von Termen und Dokumentlängen und verzichten insbesondere auf jegliche Formen des maschinellen Lernen. BM25 ist eine Suchfunktion, die Dokumente und Suchanfragen als Bag-of-Words betrachtet, so dass nur Termstatistiken wie die Worthäufigkeit und nicht die Position oder semantische Ähnlichkeit von Termen in die Relevanzbewertung einfließen. Die Relevanzbewertung für ein Dokument gegeben eine Anfrage ergibt sich dann aus Termstatistiken über die Schnittmenge der Terme von Dokument und Anfrage und erlaubt es innerhalb kurzer Zeit ein Ranking von vermuteten relevanten Dokumenten aus den Posting-Listen des Index zu erstellen. Wie wichtig die Termhäufigkeit oder die Dokumentlänge für die Relevanzbewertung genau sind, kann in BM25 mittels Parameter kontrolliert werden.

Im Gegensatz zu klassischen Suchfunktionen werden Learning-to-Rank Ansätze meist nicht eingesetzt, um direkt ein Ranking aus dem Index für eine Suchanfrage zu konstruieren, da Learning-to-Rank Ansätze für ihre Relevanzbewertung in der Regel Dokumente als ganzes betrachten und nicht nur die Schnittmenge an Termen zwischen Dokument und Suchanfrage [44]. Damit Learning-to-Rank Ansätz trotz dieser Einschränkung in der Ad-Hoc Suchen, der Aufgabe ein Top-k Ranking aus einer Anfrage zu erzeugen, eingesetzt werden können, erfolgt die Erstellung eines initialen Rankings mittels klassischer Suchfunktionen. Ein initiales Ranking ist eine Vorauswahl an möglichen relevanten Dokumenten für ein Reranking durch Learning-to-Rank Ansätze [34]. Learning-to-Rank Ansätze nutzen stets Techniken des maschinellen Lernens und basieren auf Features oder neuronalen Netzwerken [44]. Da die Berechnung von komplexeren Features und neuronalen Modellen zeitaufwändig sind und nur Dokumentenweise durchgeführt werden können, ist die geeignete Wahl des initialen Rankings um so wichtiger. Nach dem initialen Ranking werden durch Learning-to-Rank Ansätze dem finalen Ranking keine weiteren Dokumente hinzugefügt, so dass ein schlechtes und zu kleines initiales Ranking die Effektivität des Learning-to-Rank Ansatzes massiv hemmen kann und bei einem zu großen initialen Ranking jedoch der Rechenaufwand zu groß werden kann. Sowohl Feature basierte Modelle als auch neuronale Modelle benötigen in der Regel quantitative und qualitativ hochwertige Trainingsdaten, die aus Suchanfrage-Dokument-Relevanzbewertungs Trippeln bestehen, um gute bis sehr gute Ergebnisse für zuvor ungesehene Suchanfragen zu erzielen [42, 81, 83]. Entsprechen die Trainingsdaten bezüglich Quantität und Qualität nicht den zuvor unbekannten Anforderungen eines neuronalen Modells, kann es bei den tausenden beziehungsweise millionen anzupassenden Parametern, die ein neuronales Modell besitzt, schnell zu Overfitting kommen, so dass das neuronale Modell Erkenntnisse von den gesehenen Trainingsdaten nicht auf ungesehene Suchanfragen generalisieren kann. Overfitting kann mit Domainwissen in Form von Regularizorn [82] und allgemeinen Techniken wie Dropout [72] abgeschwächt werden. Regularizor sind Strafterme, die der Verlustfunktion hinzugefügt werden und aus domänenspezifischem Wissen formuliert werden. Gegensätzlich zu Regularizorn, die weitere Annahmen über die Aufgabe definieren, nutzt Dropout die Größe der neuronalen Netzwerke aus. Dropout deaktiviert mit einer gewissen Wahrscheinlichkeit in jedem Trainingsschritt zufällige Knoten und deren Verbindungen im Netzwerk, so dass stets nur kleinere Teilnetzwerke trainiert werden. Da die Knoten zufällig weggelassen werden, ist die Bildung von Teilnetzen, die bestimmte Trainingsdaten auswendig lernen, deutlich erschwert.

2.2 Transformer Modelle

Aktuelle Learning-to-Rank Ansätze, die auf Convolutional-Neural-Networks (CNN) oder Recurrent-Neural-Networks (RNN) mit Encoder und Decoder [13, 77] basieren, bedeuten durch ihre Architektur einen erheblichen Mehraufwand im Training [75]. RNNs generieren im Training versteckte Zustände für jede Position eines Satz, die jeweils von allen ihren Vorgängern abhängen und somit nur sequentiell anstatt parallel abgearbeitet werden können. Insbesondere beim Training mit langen Sätzen beziehungsweise Dokumenten bedeutet die fehlende Möglichkeit der Parallelisierung eine erhebliche Verlängerung der Trainingsdauer. Bei CNNs ergibt sich ein ähnliches Problem bei längeren Sätzen, da die Anzahl notwendiger Operationen für die Bestimmung der versteckten Repräsentationen der Sätze mit dem Abstand zwischen je zwei Termen wächst und bei einem größeren Abstand zwischen zwei Termen nimmt ebenfalls die Qualität der Abbildung von Abhängigkeiten zwischen diesen ab [35, 75]. Vaswani et al. [75] entwickelten mit der hauptsächlich auf Self-Attention basierenden Transformer Architektur einen Ansatz, der besser parallelisierbar als RNNs ist und auch bei einem längeren Input nur konstant viele Schritte zur Bestimmung einer Repräsentation durchführt. Damit das Transformer Modell die Self-Attention in konstant vielen Schritten berechnen kann, muss zum Erstellungszeitpunkt des Models eine feste Länge für die Eingabesequenzen definiert werden. Wenn die maximale Länge von einer Eingabesequenz überschritten wird, muss die Eingabesequenz geeignet aufgeteilt werden oder auf die maximale Länge gekürzt werden, wodurch sich die Ausgaben des Transformer Modells verschlechtern [75]. Die Transformer nutzen die Encoder-Decoder Struktur, so dass für eine Eingabesequenz an Wörtern mittels dem Encoder basierend auf Self-Attention eine nummerische Zwischenrepräsentation bestimmt wird, aus der der Decoder Schrittweise eine Ausgabesequenz generiert. Der Decoder nutzt bei der Bestimmung des jeweils nächsten Ausgabezeichens immer das im zu vorigen Schritt generierte Ausgabezeichen.

Die Transformer Architektur wird im Bidirectional Encoder Reprasentations from Transformers (BERT) Modell wiederverwendet [19]. BERT ist ein auf Next-Sentence-Prediciton und Masked-Language-Model vortrainiertes Transformer Modell, welches durch das hinzufügen einer extra aufgabenspezifischen Output-Schicht und einem aufgabenspezifischen Trainingsprozess direkt für die spezifische Aufgabe verwendet werden kann. Die Next-Sentence-Predicition Aufgabe bereitet BERT auf den Einsatz für Sentence-Level Aufgaben wie der Bestimmung von Relationen zwischen zwei Sätzen vor. Die Masked-Language-Modeling Aufgabe hingegen ersetzt zufällige Wörter in der Eingabesequenz durch das Sonderzeichen Masked und versucht das ursprünglich an der Stelle gestandene Wort vorherzusagen und bereitet BERT auf den Einsatz für

Token-Level Aufgaben wie Named-Entity-Recognition vor. Im Gegensatz zu unidirektional vortrainierten Transformer Modellen [58, 62] wird BERT bidirektional vortrainiert, indem in der Masked-Language-Modeling Aufgabe alle Wörter vor und nach dem maskierten Wort für die Vorhersage verwendet werden. BERT erzielte auf verschiedenen Natural-Language-Processing Aufgaben State of the Art Ergebnisse [19] und erzielt auch kompetitive Ergebnisse als Learning-to-Rank Ansatz [14, 15, 61]. BERT konnte für Natural-Language-Inference Klassifikationsaufgaben durch die Anwendung von einem Stochastic Answer Network in der aufgabenspezifischen Ausgabeschicht weiter verbessert werden [45]. Das Stochastic Answer Network erzeugt die Ausgabe nicht in einem Schritt direkt aus der Repräsentation, sondern erzeugt iterativ mehrere Ausgaben aus der Repräsentation und einem Iterationszustand, der jeweils von dem vorherigen Iterationszustand und der Repräsentation abhängt. Für die finale Ausgabe wird vor dem Mitteln der Ausgaben nur im Training ein Dropout durchgeführt, so dass sich korrekte Ausgaben nicht nur aus einer bestimmten Iteration ergeben können.

BERT als ein vortrainiertes Modell ist durch das Training auf Eingaben, die aus einem oder zwei Sätzen bestehen, nicht direkt konzipiert für Aufgaben, die mehr als zwei Sätze als Eingabe verwenden. Eine universelle Möglichkeit ist es Eingaben mit mehr als drei Sätzen mittels einer Umformulierung der Aufgabe und dem Zusammenfassen mehrerer einzelner Eingabesätze auf zwei Eingabesequenzen zu reduzieren. Dieser Ansatz bedeutet einen Mehraufwand in der Anpassung der Aufgabe und führt zu einem Verlust der genauen Satzunterscheidung, die durch die ursprüngliche Trennung der Eingabesätze vorhanden war. Zwei alternative Ansätze im Umgang mit mehr als zwei Eingabesätzen sind BERTSUM [47] und duoBERT [54]. Der duoBERT Ansatz wurde für das Paarweise Ranking von Dokumenten eingeführt und nutzt das in BERT vordefinierte Sonderzeichen `[SEP]` zur Trennung von drei Eingabesätzen, obwohl BERT mit `[SEP]` ursprünglich nur auf die Trennung von zwei Sätzen trainiert wurde [19]. BERTSUM hingegen hat das bilden von einer Zusammenfassung aus mehreren Eingabesätzen, die aus einem Dokument stammen, zur Aufgabe und modifiziert dazu die Architektur von BERT. Ähnlich zu duoBERT trennt BERTSUM mehrere Eingabesätze durch die Verwendung von `[SEP]` und fügt dann noch zusätzlich das in BERT vorhandene Sonderzeichen `[CLS]` vor jedem Satz ein. Die Repräsentationen der extra eingefügten Sonderzeichen `[CLS]` aggregieren Informationen über den darauf folgenden Eingabesatz und ermöglichen zusammen mit dem Alternieren der BERT eigenen Segmentation-Embeddings für ursprünglich nur zwei Eingabesätze sowie extra hinzugefügten Inter-Sentence-Schichten das Lernen von hierarchischen Dokumentrepräsentationen [47]. Durch die extra `[CLS]` Sonderzeichen und den Inter-Sentence-Schichten kann BERTSUM somit Repräsentationen auf dem

Satz- und Dokument-Niveau bestimmen und auswerten.

2.3 Multi-Task-Learning

Multi-Task-Learning ist ein Teilgebiet des maschinellen Lernens, indem die Konstruktion von Methodiken zum steigern der Effektivität und Effizienz von Aufgaben des maschinellen Lernens mittels eines Informationsaustausches zwischen mindestens zwei Aufgaben im Fokus steht [16]. Für ein erfolgreiches Multi-Task-Learning ist nicht nur die Wahl geeigneter Aufgaben, um schädliche Interferenz und negativen Wissenstransfer zu verhindern, sondern auch die Wahl der konkreten Multi-Task-Learning Architektur von Bedeutung. Mögliche Architekturen erstrecken sich von kommunizierenden und untereinander abhängigen neuronalen Netzen, über vollständig gemeinsame Netz-Schichten hinzu hierarchisch strukturierten Aufgaben und Netzwerken und unterscheiden sich des weiteren im sequentiellen oder gleichzeitigen Training der Aufgaben sowie der Optimierungsentscheidung, ob alle Aufgabe als gleichwertig zu betrachten sind oder einzelne Aufgaben nur als Hilfsaufgaben für andere Primäraufgaben verwendet werden. Zusätzlich muss im Multi-Task-Learning noch die genaue Spezifikation jeder einzelnen Aufgabe getroffen werden, so dass sich insgesamt eine Vielzahl an zu treffenden Entscheidungen für ein Multi-Task-Learning Setup ergibt. Für einen tiefergehenden Einblick in die allgemeinen Techniken des Multi-Task-Learning empfehlen wir Crawshaw [16].

Nishida et al. [53] betrachten die Machine Reading at Scale (MRS) Aufgabe, für die gegeben eine Frage zunächst eine Menge an relevanten Passagen aus einem Korpus extrahiert wird und anschließend aus den Passagen ein Teiltext als Antwort auf die Frage entnommen wird (Reading-Comprehension). Nishida et al. [53] erstellen in ihrem Ansatz mittels exakten Term-Matching ein initiales Ranking an relevanten Passagen, Reranken dieses dann mittels eines Learning-to-Rank Ansatzes und extrahieren aus aus den relevantesten Passagen mittels Machine-Learning die Antwort. Sie trainieren für die MRS Aufgabe gleichzeitig ein Modell auf das Ranken von Passagen und das Finden von Antworten in Passagen, indem sich die Reading-Comprehension und Learning-to-Rank Aufgabe die Schichten zur Erstellung von Repräsentationen für Passagen und Fragen teilen und in einer aufgabenspezifischen Schicht eine Antwort beziehungsweise Relevanzbewertung erstellt wird. Mit ihrem Ansatz konnten sie State of the Art Ergebnisse auf $SQuAD_{full}$ erzielen und außerdem den gegenseitigen positiven Einfluss zwischen der Learning-to-Rank und Reading-Comprehension Aufgabe zeigen. Eine abgewandelte Herangehensweise an Multi-Task-Learning verwenden Ahmad et al. [1] für ihre Learning-to-Rank und Query-Suggestion Aufgabe für die Websuche. Die Learning-to-Rank

und Query-Suggestion Aufgabe teilen sich den Encoder für die Anfrage und den Session-Encoder, der eine Repräsentation für die aktuelle Sequenz an Anfragen erstellt. Mittels der Repräsentation der aktuellen Websuche-Session und dem für die Query-Suggestion Aufgabe spezifischen Anfrage Decoder wird ein Anfrage Vorschlag erstellt. Die Learning-to-Rank Aufgabe verwendet einen eigenen Dokument-Encoder und berechnet aus der Dokument- und Session-Repräsentation eine Relevanzbewertung für das Dokument. Dieser Ansatz von Ahmad et al. [1] erreicht kompetitive Ergebnisse für die Learning-to-Rank und Query-Suggestion Aufgabe und zeigt, dass Multi-Task-Learning auch für Aufgaben mit unterschiedlichen Eingabeformaten verwendet werden kann und Aufgaben, auch wenn sie nicht alle Encoding-Schichten teilen, voneinander lernen können. Mit ihren Untersuchungen im Bereich des Natural-Language-Understanding (NLU) zeigen Liu et al. [46], dass Multi-Task-Learning auch für mehr als zwei Aufgaben gleichzeitig eingesetzt werden kann. Durch den Einsatz von BERT mit aufgabenspezifischen Decodern für jede einzelne Aufgabe des Multi-Task-Learnings zeigen sie insbesondere, dass vortrainierte Transformer Modelle und Multi-Task-Learning mittels den vortrainierten Encoder gemeinsam eingesetzt werden können, um noch effektivere Ergebnisse als Multi-Task-Learning oder vortrainierte Modelle allein zu erzeugen. Sie erreichen auf acht der neun verwendeten NLU Aufgaben neue State of the Art Ergebnisse für den GLUE Benchmark und zeigen, dass Multi-Task-Learning in Kombination mit BERT auch bei einer Auswahl von nur 1% der Trainingsdaten kompetitive Ergebnisse erzielt.

2.4 Grundlagen IR Axiome

Ein Axiom ist eine „gültige Wahrheit, die keines Beweises bedarf“ [22]. Ein Axiomensystem als eine Menge möglichst überschneidungsfreier Axiome bildet die Möglichkeit, mittels weniger Annahmen weitere Zusammenhänge in einem Wissengebiet deduktiv zu schlussfolgern. Auch in der historisch betrachteten eher empirisch fundierten Wissenschaft des Information-Retrieval wurden Axiome für ein Axiomensystem definiert, um den abstrakten Begriff der Relevanz zwischen einer Anfrage und einem oder mehreren Dokumenten mathematisch zu formalisieren [23]. Die Motivation hinter IR Axiomen ist es Annahmen zu finden, die ein effektives Ranking charakterisieren, diese zu formalisieren und analytisch auf Suchfunktionen oder erzeugte Rankings anzuwenden [9]. Einer der ersten axiomatischen Ansätze im Information-Retrieval war von Mc Cune et al. [49], indem ein Suchmodell um Produktionsregeln aus dem Bereich der künstlichen Intelligenz erweitert wurde. In den folgenden Jahren formalisierte Van Rijsbergen [74] die bedingte Logik, die dem Information-

Retrieval zugrunde liegt, und Meghini et al. [50] formalisierte eine Beschreibungslogik, um Information-Retrieval Prozesse zu beschreiben. Die erste namentliche Verwendung von Axiomen im Information-Retrieval [34] datiert auf das Jahr 1994 [9] zurück. Bruza and Huibers [9] definieren ein Informationsfeld als eine Struktur und definieren anschließend über diese Struktur Axiome, mit dem Ziel einen theoretischen Rahmen zu bilden, um zwei Information-Retrieval Modelle nicht experimentell, sondern analytisch zu vergleichen.

In den vergangenen zwei Jahrzehnten gab es ein deutliches Wachstum an wissenschaftlichen Arbeiten in dem Bereich des Information-Retrieval mit einem Bezug zu Axiomen. Insbesondere die Formalisierung von Axiomen zur Beschreibung von guten Suchfunktionen hat sich weiterentwickelt. Die dabei entstandenen Axiome lassen sich nach den Facetten von Relevanz, auf der die jeweiligen Axiome basieren, wie folgt gruppieren [76]: Termhäufigkeit [23, 25, 26, 70] und untere Schranken der Termhäufigkeit [48], Dokumentlänge [25], Anfrage Facetten [33, 80, 85], semantische Ähnlichkeit [24], Term Nachbarschaft [34], Axiome für die Evaluation [3, 10], Axiome für die Beschreibung der Eigenschaften von Link-Graphen [2], Axiome für erlernte Ranking-Funktionen [17, 18], multi-kriterielle Relevanzbewertung [30], Nutzer-Feedback basiertes Ranking [84], Sprachmodell-Übersetzungs Axiome [39, 65] und Termabhängigkeiten [21]. Einen Überblick über weitere axiomatische Techniken im Information-Retrieval mit Verweisen zu relevanter Literatur findet sich auf der Webseite von Hui Fang.[2] Ein Hindernis für weiterführende wissenschaftliche Arbeiten, die Axiome von unterschiedlichen Autoren verwenden, ist, dass die Axiome zum einen keine feste Struktur bezüglich ihrer Definition haben und zum anderen die Axiome meistens auf ein bestimmtes Information-Retrieval Szenario angepasst definiert wurden [76]. Zuvor referenzierte Axiome und Axiom-Gruppen lassen sich gar nicht wie die Axiome für die Evaluation oder nur in modifizierter Variante wie Termhäufigkeits-Axiome in unserem Szenario der Ad-Hoc Suche anwenden, da diese Axiome entweder zu restriktiv sind, indem für die Anwendung beispielsweise Relevanzbewertungen benötigt werden oder die Anwendung des Axioms keine Aussage über eine Ranking-Präferenz erlaubt.

Das Termhäufigkeits-Axiom TFC1 [25] wurde ursprünglich für Anfragen, die nur aus einem Term bestehen und für zwei Dokumente, die die gleiche Länge haben, definiert. Eine solch restriktive Definition von IR Axiomen erlaubt es die getroffenen Annahmen und Beobachtungen, die einem IR Axiom zugrunde liegen, genauer mathematisch zu formalisieren. Betrachten wir TFC1, dann würde eine Formalisierung für Anfragen mit mehr als einen Term bedeuten, dass auch weitere Annahmen über die Bedeutung der Termhäufigkeiten von

[2] `https://www.eecis.udel.edu/~hfang/AX.html`

Phrasen und der Anfrageterme untereinander getroffen werden müssen. Da Bedingungen wie die gleiche Länge von Dokumenten und die Beschränkung auf 1-Term-Anfragen in der Ad-Hoc Suche und dem Reranken von Dokumenten nur in wenigen Fällen erfüllt werden,[3] müssen IR Axiome an den Einsatz in der Ad-Hoc Suche und dem Reranken von Dokumenten angepasst werden. Eine solche Anpassung der Definition von IR Axiomen für das Reranken von Dokumenten in der Ad-Hoc Suche, der auch wir weitestgehend folgen, stammt von Hagen et al. [34] und Völske et al. [76]. Hagen et al. [34] und Völske et al. [76] relaxieren die Bedingungen von IR Axiomen, damit diese öfter erfüllt werden und dennoch nicht komplett vernachlässigt werden und definieren die IR Axiom als Paarweise Dokumentpräferenz, so dass eine berechnete Paarweise Dokumentpräferenz direkt für das Reranking von Dokumenten eingesetzt werden kann. Außerdem findet eine Erweiterung der IR Axiome für beliebig Lange Anfragen statt. Ein Axiom A definieren wir demnach angelehnt an Völske et al. [76] als ein Tripel:

$$A = (Vorbedingung, Votumbedingung, Präferenz),$$

wobei die Vorbedingung und Votumbedingung Logik-Formeln über der Anfrage und den zwei Dokumenten sind, die sich zu Wahr oder Falsch auswerten lassen und die Präferenz eine prädikatenlogische Formel ist, deren Wahrheitswert auf einen numerischen Wert abgebildet wird. Die Notwendigkeit der Votumbedingung ergibt sich aus der notwendigen Modifikation von Axiomen [34, 76], die ursprünglich nur für 1-Term-Anfrage und 2-Term-Anfragen definiert wurden und nach der Anpassung auch mit Anfragen verwendet werden können, die aus mehr als ein oder zwei Termen bestehen. Gleiches gilt für die Berechnung von Axiom-Präferenz [76], so dass sich für Anfragen, die aus mehr als ein oder zwei Termem bestehen, die Präferenz als die Summe von Teilpräferenzen ergibt, die sich im einfachsten Fall wie die ursprüngliche Axiom-Präferenz über 1-Term-Anfragen und 2-Term-Anfragen bestimmen lassen.

Gegeben zwei Dokumente d_1 und d_2, eine Anfrage q und ein Axiom A definieren wir die Axiom-Präferenz analog zu Völske et al. [76] als eine Abbildung die entweder d_1 gegenüber d_2 bevorzugt $d_1 >_A d_2$ also wenn die $Präferenz$ und $Vorbedingung$ erfüllt ist, d_2 gegenüber d_1 bevorzugt $d_2 >_A d_1$ also wenn die $Präferenz/[d_1/d_2, d_2/d_1]$ und $Vorbedingung$ erfüllt ist oder sonst keines der beiden Dokumente präferiert.

$$Präferenz(q, d_1, d_2) = \begin{cases} 1 & d_1 >_A d_2 \\ -1 & d_1 <_A d_2 \\ 0 & sonst \end{cases}$$

[3] https://www.statista.com/statistics/269740/number-of-search-terms-in-internet-research-in-the-us/

Die berechnete Axiom-Präferenz entspricht dabei auch stets einer Ranking-Präferenz [25, 34], so dass gilt $Präferenz(q, d_1, d_2) = 1 \Rightarrow S(d_1, q) > S(d_2, q)$ und $Präferenz(q, d_1, d_2) = -1 \Rightarrow S(d_1, q) < S(d_2, q)$. Eine Präferenz von 0 bedeutet, dass das Axiom keinen ausreichenden Unterschied zwischen den Dokumenten feststellen konnte und bedeutet bezüglich der Ranking-Präferenz $S(d_1, q) \approx S(d_2, q)$.

2.5 Anwendungsbeispiele für IR Axiome

Der Einsatz von IR Axiomen erlaubt es Suchfunktionen auf die Einhaltung der Axiome zu analysieren und aus den Ergebnis Rückschlüsse über noch mögliche Verbesserungen der mathematischen Definition der Suchfunktion zum Erstellen effektiverer Rankings zu ziehen [26]. IR Axiome können nicht nur zur Analyse von Suchfunktionen, sondern auch zum direkten Reranken von Dokumenten verwendet werden [34]. Hagen et al. [34] demonstrieren die Möglichkeit die Effektivität der erstellten Rankings von klassischen Suchmodellen wie BM25, Terrier DPH und DirichletLM mittels IR Axiome-Präferenzen zu verbessern, indem sie in ihrem Ansatz zunächst mit dem jeweiligen Modell die Top-k Suchergebnisse bestimmen und anschließend nach den Präferenzen verschiedener Axiom-Kombinationen die Top-k Suchergebnisse reranken. Erwartbar und trotz dessen Interessant ist die Beobachtung, dass die einzelnen Axiome und Axiomkombinationen die Effektivität des Re-Rankings bezogen auf ein Suchmodell und auch zwischen unterschiedlichen Suchmodellen unterschiedlich beeinflussen.

Mit der Verbesserung der Effektivität der Suchergebnisse bei neuronalen Netzen mittels Axiomen als Regularizer beschäftigten sich in jüngster Vergangenheit Rosset et al. [68]. Sie formulieren dazu Termhäufigkeits-Axiome von Fang et al. [26] in Strafterme um, die sie der Verlustfunktion hinzufügen und betrachten zusätzliche künstlich erzeugte Dokumentpaare, die sich darauf fokussieren, ob ein bestimmtes Axiom derzeit vom Suchmodell eingehalten wird oder nicht. Ihre Ergebnisse zeigten nicht nur, dass die Effektivität der Suchergebnisse durch die axiomatischen Regularizer verbessert wird, sonder auch das die neuronalen IR-Modelle im Trainingsprozess schneller konvergieren. Insbesondere bei nur wenigen verfügbaren Trainingsdaten wirken axiomatische Regularizer positiv gegen Overfitting, indem sie den Adaptionsprozess für die im Verhältnis zu den verwendeten Trainingsdaten große Menge an anzupassenden Parametern moderieren.

Ebenfalls 2019 wurden die Untersuchungen zur Paarweisen-Kombination von Suchfunktionen-Scores mittels abgewandelter Information-Retrieval Axiome von Arora and Yates [5] veröffentlicht. Für die Ad-hoc Suche von Doku-

menten trainieren Arora and Yates [5] einen Learning-to-Rank Ansatz, der auf neun axiomatischen Features basiert. Der Learning-to-Rank Ansatz kombiniert in Abhängigkeit von der Anfrage und einem initialen Ranking der Top-k Dokumente die Scores zweier Suchfunktionen mit einer Wichtung durch die axiomatischen Features zu dem finalen Score. Ihre Tests auf beliebige Paare an klassischen Suchmodellen und neuronalen Modellen ergeben überwiegend bessere Ergebnisse gegenüber einer alleinstehenden Anwendung der jeweiligen Modelle und zeigen eine stärkere Verbesserung der Ergebnisse, wenn die zwei betrachteten Suchmodelle verschieden affin für bestimmte Axiome sind. Ihr Ansatz zeigt vor allem die Nützlichkeit verschiedene Axiom-Klassen wie Termhäufigkeits- und Term-Nachbarschafts-Axiome für die Untersuchung von Suchfunktionen gleichzeitig zu berücksichtigen und Suchmodelle als eine eigene pseudo Axiomklasse zu formulieren.

Eine andere Herangehensweise an die axiomatische Betrachtung von neuronalen Modellen zeigen Rennings et al. [66]. Für sie steht nicht die direkte Verbesserung von Suchergebnissen im Fokus, sondern die Erschaffung von Analysewerkzeugen für neuronale Modelle mit ihren großen Parameterräumen. Neuronale Modelle mit einer Anzahl von tausenden beziehungsweise millionen von Parametern können im Gegensatz zu klassischen Suchfunktionen wie BM25 nicht direkt auf die Einhaltung von Axiomen untersucht werden, da neuronale Modelle die Relevanz eines Dokumentes nicht mittels eines geschlossenen mathematischen Ausdrucks bestimmen. Sie erstellen aus diesem Grund diagnostische Datensätze aus unannotierten Datensätzen. Die neuen Datensätze sind dabei so konzipiert, dass die erwarteten Ranking-Ergebnisse der Einhaltung genau eines Axioms entsprechen. Durch die Anwendung von neuronalen Modellen auf die diagnostischen Datensätze schätzen sie somit ab inwiefern ein neuronales Modell die einem Axiom zugrundeliegende Heuristik erlernt hat. Sie zeigen, dass das Einhalten von Axiomen von neuronalen Modellen und die Effektivität des neuronalen Modells positiv zusammenhängen. Aufbauend auf den zu vorigen Untersuchungen von Rennings et al. [66] erweiterten Câmara and Hauff [11] die Auswahl der untersuchten Axiome und betrachteten nur BERT. Sie konnten keinen positiven Zusammenhang zwischen der Verbesserung der Effektivität gegenüber Query-Likelihood und dem Einhalten von Axiomen für BERT feststellen. Unklar ist noch, ob diese Ergebnisse auf den konkreten Test-Datensatz, die Erzeugung von Datensätzen für genau ein Axiom und keine Axiomkombinationen, eine ungenügende Beschreibungskraft der Axiome oder eine weitere noch unbekannte Ursache zurückzuführen ist.

Unser Ansatz greift Kernaspekte von Hagen et al. [34],Câmara and Hauff [11] und Rosset et al. [68] unter dem Paradigma des Multi-Task-Learning von Liu et al. [46] auf. Anstatt die Axiome direkt als Strafterme in die Verlustfunktion der Ranking-Aufgabe einzufügen oder ein direktes axiomatisches Reran-

king zu veranlassen, definieren wir neben der Ranking-Aufgabe noch Axiom-Aufgaben, die sich mit der Ranking-Aufgabe den BERT Encoder teilen. Den induktive Bias von BERT in Form der vortrainierten allgemeinen Repräsentation von Sprache spezialisieren wir durch die Ranking und IR Axiom-Präferenzen Aufgabe auf den Bereich IR.

Kapitel 3

Information-Retrieval Axiome für Multi-Task Learning

In diesem Kapitel betrachten wir das Multi-Task-Learning Paradigma im Kontext der IR Axiome-Präferenz Aufgaben und betrachten die in unserem Ansatz zur Anwendung kommenden modifizierten IR Axiome im Detail.

3.1 Multi-Task-Learning Paradigma

Das Multi-Task-Learning-Paradigma beruht auf der alltäglichen Beobachtung, dass die Fähigkeiten und das Wissen, das für verschiedene Aufgaben benötigt wird, sich in Teilen überschneidet und das Erkennen der Gemeinsamkeiten zum besseren bewältigen der einzelnen Aufgaben führt [8]. Betrachten wir das Lernen von neuen Sprachen. Ein Mensch, der bereits Deutsch und Englisch gelernt hat, wird deutlich leichter eine weitere europäische Sprache wie Französisch lernen können, weil sich die grundlegenden grammatikalischen Regeln und teils auch die Wortbildungen überschneiden. Schnellere und bessere Lernerfolge bezüglich dem Lernen multipler Sprachen sind auch zu beobachten, wenn die Sprachen nicht nacheinander, sondern simultan gelernt werden [27]. Die Verbesserung des Lernerfolges für einzelne Sprachen lässt sich jedoch kaum bis gar nicht mehr beobachten, wenn die Sprachen sehr starke Unterschiede wie Deutsch und die Tonsprache Mayathan aufweisen. Diese Form des Lernens mehrerer Sprachen nacheinander oder gleichzeitig wurde auch zur Erstellung von Multilingual-BERT [59] eingesetzt. Untersuchungen auf Multilingual-BERT zeigten, dass Zusammenhänge zwischen mehreren Sprachen effektiver gelernt werden, wenn die Sprachen lexikalische und strukturelle Überschneidungen aufweisen [59]. Gleichzeitig zeigt Multilingual-BERT bei dem Training

auf 104 Sprachen[1] für einzelne der Sprachen schlechtere Ergebnisse, als wenn BERT nur auf diese eine Sprache trainiert wurde. Aufgaben, die sich in ihren Grundzügen überschneiden, können bei einem gemeinsamen Training somit nicht nur die Effektivität der einzelnen Aufgaben steigern, sondern auch untereinander interferieren und zu einem negativen Wissenstransfer führen. Konzeptionell muss für Multi-Task-Learning eine Architektur bestimmt werden, die positiven Wissenstransfer fördert und negative Interferenzen vermeidet.

Für unser Multi-Task-Learning Setup mit einer Ranking Aufgabe im Fokus entschieden wir uns gegen verwandte Relevanz-Aufgaben wie Query Suggestion [1] und Reading Comprehension [53], da diese weitere Einschränkungen bezüglich der gleichmäßigen Skalierung auf insgesamt mehr Trainingsdaten für jede einzelne Aufgabe einführen würden. Für Paarweise IR Axiom-Präferenzen als eine Klassifikationsaufgabe können unbeschränkt viele Trainingsdaten mit einem geringen Mehraufwand bestimmt werden. Ein Trainingsdateneintrag lässt sich mittels einer simplen geschlossenen mathematischen Formel aus einer Anfrage und zwei Dokumenten bestimmen. Dokumente können in beliebiger Menge durch das Crawlen des Webs erzeugt werden [56] und Anfragen automatisch generiert werden [31]. Da die IR Axiom-Präferenz Aufgaben beliebig skalierbar und kombinierbar sind, können wir experimentell die besten Trainingsverhältnisse der einzelnen Aufgaben und einzusetzenden Axiom bestimmen, um einen positiven Wissenstransfer zu fördern und negative Interferenzen zu vermeiden. Eine analytische Bestimmung der optimalen Kombination an gemeinsam zu trainierenden Aufgaben und der einzusetzenden Trainingsverhältnisse ist eine aktuelle Forschungsfrage [16, 28], so dass die beliebige Skalierbarkeit von IR Axiom-Präferenzen umso wichtiger im Multi-Task-Learning ist, wenn die Effektivität und nicht die Trainings-Laufzeit im Vordergrund steht. IR Axiome sind insbesondere für Multi-Task-Learning mit einer Ranking Aufgabe geeignet, da IR Axiome ursprünglich für die Anwendung auf Suchfunktionen konzipiert wurden [25, 48, 70]. Jedes IR Axiome beschreibt eine spezifische Eigenschaft einer Suchfunktion, die sie erfüllen sollte, damit sich die Effektivität der erzeugten Rankings der Suchfunktion verbessert. Die IR Axiom-Präferenz Aufgaben beschreiben demnach wünschenswerte Eigenschaften einer Suchfunktion für das Ranken von genau zwei Dokumenten. Da jedes Axiom eine andere Eigenschaft spezifiziert und eine andere Facette an Relevanz abdeckt, sind IR Axiom-Präferenzen als automatisch erstellte Paarweise Relevanzbewertungen bezüglich einer Facette von Relevanz interpretierbar.

Bezüglich der Multi-Task-Learning Architektur folgen wir Liu et al. [46] und verwenden BERT für die gemeinsamen Encoding-Schichten im Multi-Task-Learning. Durch das Training von BERT auf die Masked-Language-Modeling

[1] `https://github.com/google-research/bert/blob/master/multilingual.md`

und Next-Sentence-Prediciton Aufgabe, besitzt jedes BERT Modell einen induktiven Bias für die allgemeine Darstellung von Sprache. Da die Trainingsdaten von BERT nicht auf einen bestimmten Wissens- und Themenbereich beschränkt sind [19], ist der induktive Bias von BERT auf eine möglichst allgemeine Repräsentation von Sprache ausgerichtet. Mittels Fine-Tuning auf frei wählbare Aufgaben, wird der induktive Bias von BERT, auf für die neue Aufgabe spezifische Repräsentationen von Sprache angepasst. Der Vorteil am Einsatz von mehreren IR Axiom-Präferenz Aufgaben gleichzeitig ist es den induktiven Bias von BERT auf die simultane Repräsentation mehrerer verschiedener Facetten von Relevanz anzupassen. Die Beschaffenheit der Trainingsdaten für die Ranking und IR Axiom-Präferenz Aufgaben aus Dokumenten und Anfragen passt die allgemeine Repräsentation von Sprache durch BERT auf spezifischere Repräsentationen für Anfragen, Dokumente und Facetten der Relevanz an. Ein auf BERT basierender Learning-to-Rank Ansatz erlernt durch das Multi-Task-Learning Setup mit IR Axiom-Präferenz Aufgaben nicht nur eine durch bestimmte Menschen gegebene Repräsentation von Relevanz, sondern auch theoretische Facetten der Relevanz, die aus der Beobachtung der Tendenzen einer größeren Menge an Menschen erstellt wurden. Insgesamt lernt das BERT Modell somit theoretisch Facetten der Relevanz und durch Menschen gegebene Relevanzbewertungen gleichzeitig zu repräsentieren und für die jeweils andere Aufgabe mitzuverwenden.

Unsere Forschungsfragen, die wir mit unseren Untersuchungen über Multi-Task-Learning mittels IR Axiom-Präferenzen und dem Ranken von Dokumenten beantworten wollen, sind angelehnt an Liu et al. [46] und lauten: 1. Welche IR Axiome sind durch BERT erlernbar. 2. Welche Auswahl an IR Axiom-Präferenz Aufgaben steigert in welchem Trainingsverhältnis zur Ranking Aufgabe, die Effizienz bezüglich notwendiger Trainingsschritte und die Effektivität der Ranking Aufgabe am meisten.

3.2 Verwendete IR Axiome und Anpassungen

IR Axiome wie TFC1 haben zu restriktive Vorbedingungen und können dadurch nicht direkt für die Ad-Hoc Suche und das Reranken eingesetzt werden. Beispielweise fordern Termhäufigkeits-Axiome wie TFC1 in ihrer ursprünglichen Definition [25], dass zwei zu vergleichende Dokumente die gleiche Länge haben, damit eine Präferenz bestimmt werden kann. Die Formalisierung und Begründung der dem IR Axiom zugrunde liegenden Annahmen und Beobachtungen wird durch besonders restriktive Vorbedingungen erleichtert. Restriktive Bedingungen wie die exakt gleiche Dokumentlänge lassen sich jedoch nur für einen sehr kleinen Anteil der Dokumente in einem Korpus erfüllen, so dass

das IR Axiom nur auf diesen sehr kleinen Teil der Dokumente angewandt werden könnten, um Paarweise Dokumentpräferenzen zu bestimmen. Aus diesem Grund folgen wir der Methodik von Hagen et al. [34] und relaxieren im Allgemeinen die Vorbedingungen, so dass ein approximativer Vergleich anstatt eines exakten Vergleiches vorgenommen wird. Wir führen dazu für approximative Vergleiche die folgende Notation ein:

$$a \approx_x b \Leftrightarrow \frac{|a-b|}{max\{a,b\}} \leq x$$
$$a >_x b \Leftrightarrow a > a * x + b$$

Wir verwenden die Implementierung und Anpassungen an IR Axiomen von Völske et al. [76][2] und nehmen an der Implementierung Kompatibilitätsänderungen für den Einsatz mit dem Pyserini-Toolkit vor. Die Eingabe für jedes IR Axiom sind eine beliebig lange Anfrage und zwei beliebig lange Dokumente. Nach Völske et al. [76] ergibt sich die IR Axiom-Präferenz als ein Tupel aus Vorbedingung und Präferenz. Die Vorbedingung ist eine prädikatenlogische Formel über eine Anfrage und zwei Dokumente, die sich zu Wahr oder Falsch auswerten lässt. Wenn die Vorbedingung Falsch ist, kann die Präferenz nach wie vor bestimmt werden, jedoch erlaubt diese dann keine Aussage über eine Paarweise Ranking-Präferenz der Dokumente [34, 76]. Die Präferenz ergibt sich aus den Teilkomponenten der Votumbedingung und der Teilpräferenz, die Votumbedingung eine und die Teilpräferenz zwei prädikatenlogische Formeln über eine Anfrage und zwei Dokumente sind. Die Votumbedingung und Teilpräferenz sind notwendig, um IR Axiome die für 1-Term-Anfragen oder 2-Termanfragen definiert wurden, auf Anfragen, die aus mehr als ein oder zwei Termen bestehen, zu verallgemeinern. Die Votumbedingung gibt an, ob für einen Term oder ein Paar an Termen die Teilpräferenz bestimmt wird. Aus der Summe über alle Teilpräferenz ergibt sich dann die Paarweise Dokumentpräferenz. IR Axiome wie TFC1 lassen sich direkt auf Anfragen, die aus mehreren Termen bestehen, verallgemeinern, so dass für diese IR Axiome die Votumbedingung und Teilpräferenz weggelassen werden und stattdessen zwei prädikatenlogische Formel über eine Anfrage und zwei Dokumente für die Präferenz angegeben wird. Die zwei prädikatenlogische Formel für die Präferenz und Teilpräferenz schließen sich gegenseitig aus. Betrachten wir beispielhaft die folgende prädikatenlogische Formel über die Termhäufigkeit für einen Term in zwei Dokumenten:

$$tf(t, d_1) > tf(t, d_2)$$

dann ist die Präferenz gleich 1 genau dann wenn die prädikatenlogische Formel zu Wahr ausgewertet wird. Die Präferenz -1 ergibt sich hingegen genau dann

[2] `https://github.com/webis-de/ICTIR-21`

wenn in der prädikatenlogische Formel, für wann die Präferenz gleich 1 ist, alle d_1 durch d_2 und d_2 durch d_1 substituiert werden und die so entstandene prädikatenlogische Formel zu Wahr ausgewertet wird. Wenn keine der beiden prädikatenlogische Formeln zu Wahr ausgewertet werden kann, dann ist die Präferenz gleich 0. Insgesamt ist eine IR Axiom-PrÃ¤ferenz fÃ¼r eine Anfrage und zwei Dokumente auch stets als eine Ranking-PrÃ¤ferenz zu interpretieren. Für die nachfolgenden IR Axiome geben wir stets eine der prädikatenlogischen Formeln für die Präferenz und Teilpräferenz an, insofern diese existiert, und die jeweils andere prädikatenlogischen Formel lässt sich wie zuvor beschrieben bilden.

3.2.1 Termhäufigkeits-Axiome

Termhäufigkeits-Axiome modellieren die Beziehung zwischen den Termhäufigkeiten in einem Dokument und dem Rang des Dokumentes.

TFC1

Dem IR Axiom TFC1 liegt die Beobachtung zu Grunde, dass ein Dokument d_1 gegenüber einem Dokument d_2 eher das Informationsbedürfnis einer Anfrage q befriedigt, wenn die Anfrageterme häufiger in d_1 als d_2 vorkommen. Basierend auf der ursprünglichen Definition von Fang et al. [25] wird TFC1 für Anfragen mit mehr als einem Term erweitert, indem die Summe über die einzelnen Termhäufigkeiten gebildet wird.

$$\begin{aligned} Vorbedingung :=& Länge(d_1) \approx_{10\%} Länge(d_2) \\ Präferenz :=& 1 \Leftrightarrow \sum_{t \in q} tf(t, d_1) >_{10\%} \sum_{t \in q} tf(t, d_2) \end{aligned}$$

TFC3

Das IR Axiom TFC3 berücksichtigt, dass zwischen zwei Dokumenten, deren Termhäufigkeit bezüglich der Anfrageterme mit gleichem Informationsgehalt (Term-Spezifität) gleich ist, dennoch das Dokument bevorzugt werden sollte, dass mehr verschiedene Anfrageterme enthält. Wir bestimmen die Term-Spezifität als inverse Dokumenthäufigkeit [69](IDF) des Terms. Für den Fall, dass eine Anfrage aus mehr als zwei Termen besteht, werden alle möglichen Anfrageterm-Paare mit approximativ gleicher Term-Spezifität gebildet und die

Teilpräferenzen für jedes solche Anfrageterm-Paar bestimmt [26].

$$\begin{aligned}
Vorbedingung :=& Länge(d_1) \approx_{10\%} Länge(d_2) \\
Votumbedingung :=& \lfloor 100 * idf(t_1) \rfloor \approx_{10\%} \lfloor 100 * idf(t_2) \rfloor \wedge \\
& tf(t_2, d_1) = 0 \wedge tf(t_1, d_2) \neq 0 \wedge tf(t_2, d_2) \neq 0 \\
Teilpräferenz :=& -1 \Leftrightarrow tf(t_1, d_1) = tf(t_1, d_2) + tf(t_2, d_2) \\
Präferenz :=& 1 \Leftrightarrow \sum Teilpräferenzen > 0
\end{aligned}$$

M-TDC

Das IR-Axiom M-TDC gibt an, dass wenn in zwei Dokumenten zwei Anfrageterme gleich oft vorkommen, dann das Dokument bevorzugt wird, welches den Anfrageterm mit der höheren Term-Spezifität enthält. Wie bei TFC3 werden alle möglichen Anfrageterm-Paare gebildet und die Teilpräferenz für jedes solche Anfrageterm-Paar bestimmtt [70].

$$\begin{aligned}
Vorbedingung :=& Länge(d_1) \approx_{10\%} Länge(d_2) \wedge \bigcup_{t \in q} tf(t, d_1) = tf(t, d_2) \\
Votumbedingung :=& idf(t_1) \geq idf(t_2) \wedge (tf(t_1, d_1) = tf(t_2, d_2) \wedge \\
& tf(t_2, d_1) = tf(t_1, d_2) \vee tf(t_1, q) > tf(t_2, q)) \\
Teilpräferenz :=& 1 \Leftrightarrow tf(t_1, d_1) > tf(t1, d_2) \\
Präferenz :=& 1 \Leftrightarrow \sum Teilpräferenzen > 0
\end{aligned}$$

LB1

Das IR Axiom LB1 vereint Aspekte von sowohl den Termhäufigkeits- als auch den Längen-Normierungs-Axiome, indem LB1 bei einem approximativ gleichen Retrieval-Score zweier Dokumente bezüglich BM25 (k1=0.9, b=0.4) das Dokument bevorzugt, welches einen Anfrageterm enthält, der nicht in dem anderen Dokumenten vorkommt. Es verhindert somit das Dokumente aufgrund ihrer größeren Dokumentlänge zu stark im Ranking benachteiligt werden. Im Unterschied zu der ursprünglichen Definition von Lv and Zhai [48] werden die Anfrageterme in der Implementation von Völske et al. [76] in einer zufälligen deterministischen Reihenfolge betrachtet und die Präferenz ergibt sich aus dem ersten Anfrageterm der eine der beiden prädikatenlogischen Formeln für die Präferenz erfüllt.

$$\begin{aligned}
Vorbedingung :=& score(q, d_1) \approx_{10\%} score(q, d_2) \\
Präferenz :=& 1 \Leftrightarrow \exists t \in q : t \in d_1 \wedge t \notin d_2
\end{aligned}$$

3.2.2 Längen-Normierungs-Axiome

Längen-Normierungs-Axiome haben modellieren die Beziehung zwischen der Dokumentlänge und dem Rang des Dokumentes.

LNC1

Mittels dem IR Axiom LNC1 werden bei gleich häufigen Vorkommen von Anfragetermen in zwei Dokumenten das Dokumente bevorzugt, welches kürzer ist. Die Berechnung von LNC1 wurde gegenüber der ursprünglichen Version von Fang et al. [25] wie folgt verallgemeinert:

$$\begin{aligned} Vorbedingung &:= \forall t \in q : tf(t, d_1) \approx_{10\%} tf(t, d_2) \\ Präferenz &:= 1 \Leftrightarrow Länge(d_1) < Länge(d_2) \end{aligned}$$

TF-LNC

Das IR Axiom TF-LNC verbindet Termhäufigkeits- und Längen-Normierungs-Axiome, indem das Längere von zwei Dokumenten bevorzugt wird, wenn nach dem Entfernen aller Anfrageterme aus den Dokumenten die Dokumente dieselbe Länge haben. Die ursprüngliche Definition von TF-LNC [25] wird für Anfragen mit mehr als einem Term verallgemeinert, indem für jeden einzelnen Anfrageterm die Teilpräferenz bestimmt wird.

$$\begin{aligned} Vorbedingung &:= Wahr \\ Votumbedingung &:= Wahr \\ Teilpräferenz &:= 1 \Leftrightarrow tf(t, d_1) > tf(t, d_2) \wedge \\ &\qquad Länge(d_1) = Länge(d_2) + tf(t, d_1) - tf(t, d_2) \\ Präferenz &:= 1 \Leftrightarrow \sum Teilpräferenzen > 0 \end{aligned}$$

3.2.3 Semantische-Ähnlichkeits-Axiome

Semantische-Ähnlichkeits-Axiome modellieren die semantische Beziehung zwischen den Dokumenttermen, den Anfragetermen und dem Rang des Dokumentes.

STMC1

Das IR Axiom STMC1 erweitert den syntaktischen Vergleich von Anfrage- und Dokumenttermen um einen semantischen Vergleich, der es erlaubt, Dokumente zu bevorzugen, die weniger syntaktisch gleiche Anfrageterme und dafür deutlich mehr semantisch ähnliche Terme enthalten. Die ursprüngliche Definition

von Fang and Zhai [24] wird für Anfragen und Dokumente mit mehr als einen Term erweitert, indem die semantische Ähnlichkeit zwischen jedem Anfrage-Dokument-Termpaar bestimmt wird und dann das arithmetische Mittel über die Termpaar-Werte bestimmt wird. Für die Bestimmung von semantischen Ähnlichkeiten zwischen zwei Termen ($sim(t_1, t_2)$) verwenden wir WordNet.[3]

$$Vorbedingung := Wahr$$
$$Präferenz := 1 \Leftrightarrow \frac{\sum_{t_1 \in q, t_2 \in d_1} sim(t_1, t_2)}{Länge(d_1)} > \frac{\sum_{t_1 \in q, t_2 \in d_2} sim(t_1, t_2)}{Länge(d_2)}$$

STMC2

Das IR Axiom STMC2 verhindert eine zu starke Wichtung der semantischen Ähnlichkeit, indem es berücksichtigt, dass Dokumente, die semantisch ähnliche Terme zu den Anfragetermen enthalten, nicht gegenüber Dokumenten, die den exakten Anfrageterm enthalten, bevorzugt werden. Die ursprüngliche Definition von Fang and Zhai [24] wird für Anfragen, die aus mehr als einem Term bestehen, so angepasst, dass zur Berechnung der Präferenz der Anfrageterm verwendet wird, der die höchste semantische Ähnlichkeit zu einem Term in einem der beiden Dokumente hat, der nicht in der Anfrage enthalten ist.

$$Vorbedingung := Wahr$$
$$Präferenz := 1 \Leftrightarrow \frac{Länge(d_2)}{Länge(d_1)} \approx_{20\%} \frac{tf(t_1, d_2)}{tf(t_2, d_1)}$$
$$\text{mit } t_1, t_2 := \underset{t_1 \in (d_1 \cup d_2) \setminus q, t_2 \in q}{\text{argmax}} \{sim(t_1, t_2)\}$$

3.2.4 Anfrage-Facetten-Axiome

Anfrage-Facetten-Axiome modellieren die Beziehung zwischen den Anfragetermen untereinander, den Dokumenttermen und dem Rang des Dokumentes.

REG und ANTI-REG

Das IR Axiom REG nimmt in Betracht, dass obwohl Anfragen aus einem Informationsbedürfnis entstehen, ein Informationsbedürfnis aus mehreren spezifischeren Facetten bestehen kann, die dann in einer Anfrage mittels mehrerer Anfrageterme zusammengefasst werden. Die Erweiterung zu der ursprünglichen Definition von Zheng and Fang [85] berücksichtigt, dass es in Anfragen stets eine Facette gibt, die zur Befriedigung des Informationsbedürfnisses am

[3] http://wordnet.princeton.edu/

wichtigsten ist. Für die Präferenz wird die Termhäufigkeit des Anfragetermes, der die höchste summierte semantische Ähnlichkeit zu allen anderen Anfragetermen aufweist, zwischen den zwei Dokumenten verglichen.

$$\begin{aligned} Vorbedingung :=& Wahr \\ Präferenz :=& 1 \Leftrightarrow tf(t_{max}, d_1) > tf(t_{max}, d_2) \\ & \text{mit } t_{max} := \underset{t_1 \in q}{\text{argmax}}\{\sum_{t_2 \in q} sim(t_1, t_2)\} \end{aligned}$$

Ähnlich zu REG ist das IR Axiom ANTI-REG definiert, indem hier der Anfrageterm verwendet wird, der die niedrigste summierte semantische Ähnlichkeit zu allen anderen Anfragetermen aufweist. Die zugrundeliegende Beobachtung ist eine ähnliche wie bei der Ergänzung von TF zu TF-IDF als Retrieval-Modell, indem nicht nur die prägnanteste Facette, sondern auch die spezifischste Facette berücksichtigt werden muss, um das Informationsbedürfnis adäquat zu erfüllen.

$$\begin{aligned} Vorbedingung :=& Wahr \\ Präferenz :=& 1 \Leftrightarrow tf(t_{max}, d_1) > tf(t_{max}, d_2) \\ & \text{mit } t_{max} := \underset{t_1 \in q}{\text{argmin}}\{\sum_{t_2 \in q} sim(t_1, t_2)\} \end{aligned}$$

DIV

Mittels dem IR Axiom DIV wird ein Dokument gegenüber einem anderen Dokument präferiert, wenn das Dokument unähnlicher zur Anfrage ist. Die zugrundeliegende Idee ist, dass ein weniger relevantes Dokument, welches eine andere Facette der Anfrage als ein relevanteres Dokument beantwortet, dann besser für die Befriedigung des Informationsbedürfnis ist, wenn die Facette des relevanteren Dokumentes bereits durch ein noch relevanteres Dokument abgedeckt wird. Diese Diversifikation der Ergebnis-Dokumente wurde ursprünglich von Gollapudi and Sharma [33] axiomatisch formalisiert und wird modifiziert, indem die Ähnlichkeit von Anfrage und Dokument mittels dem Jaccard-Koeffizient [12] bestimmt wird.

$$\begin{aligned} Vorbedingung :=& Wahr \\ Präferenz :=& 1 \Leftrightarrow J(q, d_1) < J(q, d_2) \end{aligned}$$

3.2.5 Term-Nachbarschafts-Axiome

Term-Nachbarschafts-Axiome modellieren die Beziehung zwischen dem Abstand zwischen Anfragetermen in Dokumenten und dem Rang des Dokumentes.

PROX1

Das IR Axiom PROX1 untersucht in welchem von zwei Dokumenten die mittlere summierte Distanz zwischen je zwei Anfragetermen am kleinsten ist und bevorzugt dieses Dokument [34]. Die mittlere Distanz zwischen allen Anfragetermpaaren in einem Dokument wird definiert als:

$$\pi(q,d) := \frac{1}{|P|} \sum_{(i,j)\in P} \delta(d,i,j),$$

wobei $P := \{(i,j)|i,j \in q, i < j\}$ alle möglichen Anfragetermpaare beschreibt und $\delta(d,i,j)$ die mittlere Anzahl an Wörtern zwischen allen Vorkommen des Termpaares t_i und t_j im Dokument d bestimmt.

$$\begin{aligned} Vorbedingung :=& q \cap d_1 = q \cap d_2 \wedge Länge(q) > 1 \\ Präferenz :=& 1 \Leftrightarrow \pi(q,d_1) < \pi(q,d_2) \end{aligned}$$

PROX2

Mittels dem IR Axiom PROX2 werden Dokumente bevorzugt, bei denen die Anfrageterme zum ersten Mal weiter vorne vorkommen als in anderen Dokumenten. Die zugrunde liegende Idee ist, dass ein Nutzer möglichst frühzeitig die Anfrageterme in einem Dokument wiederfinden möchte [34].

$$\begin{aligned} Vorbedingung :=& q \cap d_1 = q \cap d_2 \wedge Länge(q) > 1 \\ Präferenz :=& 1 \Leftrightarrow \sum_{t\in q\cap d_1\cap d_2} first(t,d_1) < \sum_{t\in q\cap d_1\cap d_2} first(t,d_2) \end{aligned}$$

PROX3

Das IR Axiom PROX3 erweitert die Annahme von PROX2, indem es zwischen zwei Dokumenten die Position des ersten Vorkommens der gesamten Anfrage als eine Phrase vergleicht [34]. Sollte die Phrase nicht im Dokument enthalten sein, wird der Wert des erstens Vorkommens auf ∞ gesetzt.

$$\begin{aligned} Vorbedingung :=& q \cap d_1 = q \cap d_2 \wedge Länge(q) > 1 \\ Präferenz :=& 1 \Leftrightarrow first(q,d_1) < first(q,d_2) \end{aligned}$$

PROX4

In der Regel wird ein Dokument nicht die gesamte Anfrage als eine unmittelbare Phrase, sondern als mehrere versetzte Teilphrasen enthalten. Hagen et al. [34] definieren aus diesem Grund das IR Axiom PROX4 mittels:

$$\omega : (d,q) \rightarrow (a,b)$$

mit a ist die Anzahl an nicht Anfragtermen in der nächsten Gruppierung aller Anfrageterme im Dokument und b bestimmt die Häufigkeit einer solchen Gruppierung a.

$$\begin{aligned} Vorbedingung :=& q \cap d_1 \cap d_2 = q \wedge Länge(q) > 1 \\ Präferenz :=& 1 \Leftrightarrow \omega(d_1, q) < \omega(d_2, q) \end{aligned}$$

PROX5

Das IR Axiom PROX5 greift PROX4 auf und macht es robuster gegen Ausreißer, indem PROX5 für jedes Vorkommen eines Anfragetermes im Dokument die Breite der nächsten Gruppierung um den Anfragterm bestimmt und die Mittelwerte für zwei Dokumente vergleicht [34].

$$\begin{aligned} Vorbedingung :=& q \cap d_1 \cap d_2 = q \wedge Länge(q) > 1 \\ Präferenz :=& 1 \Leftrightarrow \overline{s}(d_1, q) < \overline{s}(d_2, q) \end{aligned}$$

3.2.6 Retrieval-Score-Axiome

Retrieval-Score-Axiome modellieren die Beziehung zwischen dem Retrieval-Score und dem Rang des Dokumentes. Ähnlich dem IR Axiom ORIG von Hagen et al. [34] ermöglichen es die Retrieval-Score Axiome, dem ursprünglichen Ranking einen Einfluss auf das finale Ranking zu haben. Wir betrachten Retrieval-Score Axiome für die Suchfunktionen TF, TF-IDF, BM25 ($k_1 = 0.9$, $b = 0.4$) und QL($\mu = 1000$) [44].

$$\begin{aligned} Vorbedingung :=& Wahr \\ Präferenz :=& 1 \Leftrightarrow score(q, d_1) > score(q, d_2) \end{aligned}$$

Tabelle 3.1: Zusammenfassung für alle Axiome welche der Eigenschaften Termhäufigkeit (Term Fr.), Term-Spezifität (Term Sp.), Termposition (Term Pos.), Retrieval-Score (Score), Dokumentlänge (Doc. Le.) und semantische Ähnlichkeit (Sem. Sim.) BERT in der Repräsentation abbilden muss, damit das Axiom „erlernt“ wurde. Aufgrund der Beschaffenheit der einzelnen Axiome implizieren sich diese teils untereinander. Die Eigenschaft muss erlernt werden ✔, muss nicht erlernt werden ✖ oder wird durch eine andere Eigenschaft impliziert (✔).

Axiom	**Term Fr.**	**Term Sp.**	**Term Pos.**	**Score**	**Doc. Le.**	**Sem. Sim.**
TFC1	✔	✖	✖	✖	✔	✖
TFC3	✔	✔	✖	✖	✔	✖
M-TDC	✔	✔	✖	✖	✔	✖
LNC1	✔	✖	✖	✖	✔	✖
TF-LNC	✔	✖	✖	✖	✔	✖
DIV	✔	✖	✖	✖	✖	✖
REG	✔	✖	✖	✖	✖	✔
ANTI-REG	✔	✖	✖	✖	✖	✔
STMC2	✔	✖	✖	✖	✔	✔
STMC1	✖	✖	✖	✖	✔	✔
PROX1	(✔)	✖	✔	✖	✖	✖
PROX2	(✔)	✖	✔	✖	✖	✖
PROX3	(✔)	✖	✔	✖	✖	✖
PROX4	(✔)	✖	✔	✖	✖	✖
PROX5	(✔)	✖	✔	✖	✖	✖
RS-TF	(✔)	✖	✖	✔	(✔)	✖
LB1	(✔)	(✔)	✖	✔	(✔)	✖
RS-TF-IDF	(✔)	(✔)	✖	✔	(✔)	✖
RS-BM25	(✔)	(✔)	✖	✔	(✔)	✖
RS-QL	(✔)	(✔)	✖	✔	(✔)	✖

Kapitel 4

Veruchsaufbau

In diesem Abschnitt gehen wir auf die von uns eingesetzten und erstellten Test- und Trainingsdatensätze im Detail ein und beschreiben für eine bessere Reproduzierbarkeit und Erweiterbarkeit unseres Ansatzes Details unserer Implementation. Außerdem geben wir einen Überblick über die Hypothesen, die unsere Auswahl an durchgeführten Experimenten aus den vielfältig möglichen Experimenten motivierten.

4.1 Datensätze

Der MS MARCO Passagen Datensatz [52] ist aus dem MS MARCO Question Anwsering Datensatz [52] entstanden. Mit rund 8.8 Millionen verschiedenen Passagen, die aus rund 3.5 Millionen durch Bing indexierte Webseiten extrahiert wurden, und rund 1 Million anonymisierter Bing Suchanfragen, die im Schnitt eine durch Menschen annotierte Antwort-Passage haben, wurde der MS MARCO Datensatz zum Trainieren von Learning-to-Rank Ansätzen veröffentlicht. Nach einer Leerzeichen-Tokenisierung bestehen die Passagen im Mittel aus rund 56 Token und die längste Passage aus 362 Token. Da die Passagen nicht sehr lang sind, können sie ohne eine Zerteilung oder dem Abschneiden der letzten Token für Learning-to-Rank Ansätze wie BERT eingesetzt werden, die aus Speicher- und Laufzeit-Gründen eine Beschränkung der Eingabe auf 512 Token vornehmen.

Für den direkten Einsatz von Learning-to-Rank Ansätzen stehen im Github des MS MARCO Passagen Datensatzes[1] 400 Millionen Trainings-Tripel aus einer Anfrage, einer relevanten Passage und einer vermuteten nicht relevanten Passage zum Download bereit. Da zu jeder Anfrage nur relevante Passagen von Menschen annotiert wurden, gibt es keine durch Menschen verifizierte nicht

[1] `https://github.com/microsoft/MSMARCO-Passage-Ranking`

relevanten Passagen zu Anfragen. Wir wählen zur Erstellung der Trainingsdaten für unsere Ranking Aufgabe aus den 400 Millionen Trainings-Tripeln zufällige Tripel aus. Wir spalten jedes zufällig gezogenen Tripel in zwei Tripel aus Anfrage, Passage und ein entsprechendes Relevanzlabel auf, so dass die Trainingsdaten der Ranking Aufgabe immer gleich viele Trainingsdaten für relevante und vermutete nicht relevante Passagen enthält. Im Github des MS MARCO Passagen Datensatzes werden auch rund 530.000 Trainings-Anfragen, 60.000 Entwicklungs-Anfragen und 60.000 Evaluations-Anfragen zum Vergleichen von Learning-to-Rank Ansätzen untereinander bereitgestellt. Der MS MARCO Passagen Datensatz wurde im TREC Deep Learning Track aus den Jahr 2019 und 2020 verwendet, so dass jeweils 200 weitere Anfragen zur freien Verwendung verfügbar sind. Wir verwenden die jeweils 200 Anfragen der TREC Deep Learning Tracks und jeweils 100 Anfragen aus den MS MARCO Trainings-, Entwicklungs- und Evaluationsanfragen, um für jede Anfrage die Top-100 Dokumente zu erhalten und berechnen auf diesen Top-100 Dokumenten die IR Axiom-Präferenz für die dazugehörige Anfrage. Wir entscheiden uns für eine Bestimmung der IR Axiom-Präferenzen für die Top-100 Passagen und gegen die Bestimmung der IR Axiom-Präferenzen für zufällig gewählte Passagen, da die Top-100 Passagen ähnliche Zusammenhänge bezüglich der Relevanz und weiteren Eigenschaften wie der Dokumentlänge aufweisen, wie es dann auch bei den Dokumenten für die Ranking Aufgabe der Fall ist. Wir kennzeichnen die unterschiedlichen Testdaten durch die Art der Stichprobe und ob die Test-Daten aus den 700 Anfragen (Unique=Falsch, None-Unique) oder den extra 100 Anfragen (Unique=Wahr, Unique) erstellt wurden. Damit wir für die IR Axiom-Präferenz Klassen 1 und -1 ungefähr gleich viele Trainingsdaten erhalten und keinen systematischen Bias für die Trainingsdaten der IR Axiom-Präferenz Klassen 1 und -1 einführen, bestimmen wir die IR Axiom-Präferenz für ein Paar an Passagen mit der Bedingung, dass immer die Passage d_1 ist, welche eine kleinere Dokument-ID im Index hat. Wir berichten die absoluten Häufigkeiten jeder IR Axiom-Präferenz Klasse aus den je Axiom bestimmten 3.465.000 Vorbedingungen und Präferenzen in der Tabelle 4.1. Aus den absoluten Häufigkeiten für die IR Axiom-Präferenz Klassen geht hervor, dass eine beliebige Skalierung von IR Axiomen wie TFC3 mit einem größeren parallelisierbaren Rechenaufwand einhergeht, da die Vorbedingung der Axiome trotz den vorgenommenen Anpassungen immer noch zu restriktiv sind und eine starkes Ungleichgewicht im auftreten der Präferenzen vorliegt.

Die Trainingsdaten für die einzelnen IR Axiom-Präferenz Aufgaben bestimmen wir immer als stratifizierte Stichprobe aus der Teilmenge der 3.465.000 Vorbedingungen und Präferenzen, in der die Vorbedingung für das IR Axiom erfüllt ist. Für die Evaluierung der IR Axiom-Präferenz Aufgaben ziehen wir aus der Teilmenge der 3.465.000 Vorbedingungen und Präferenz, in der die

Vorbedingung für das IR Axiom erfüllt ist, je IR Axiom eine stratifizierte und eine zufällige Stichprobe mit jeweils 3.000 Einträgen. Da erste Experimente mit den IR Axiom-Präferenz Aufgaben ein auswendig Lernen der zu treffenden Entscheidungen für die Trainings Anfragen und Dokumentpaare vermuten ließen (Overfitting mit Data-Leakage, A.2.2), haben wir für weitere 100 Anfragen aus den MS MARCO Trainings-Anfragen, die sich nach einer Leerzeichen-Tokenisierung keine Token mit den bereits verwendeten 700 Anfragen teilen, nach dem gleichen Vorgehen wie für die 700 Anfragen eine stratifizierte und zufällige Stichprobe für jedes IR Axiom erstellt. Die stratifizierte und zufällige Stichprobe bestehen jeweils aus 1.000 Einträgen.

Für 43 der 200 Anfragen vom TREC Deep Learning Track aus dem Jahr 2019 wurden insgesamt 9.260 Relevanzbewertungen auf einer Skala von 0 bis 3 durch das NIST erstellt und frei zur Verfügung gestellt.[2] Ebenfalls für 54 der 200 Anfragen vom TREC Deep Learning Track aus dem Jahr 2020 wurden insgesamt 11.386 Relevanzbewertungen auf einer Skala von 0 bis 3 durch das NIST erstellt und im Web veröffentlicht.[3] Wir verwenden die 43 und 54 Anfragen mit den durch das NIST erstellten Relevanzbewertungen zur Evaluierung unseres Ansatzes, da jede einzelne Anfrage mehr Relevanzbewertungen als die Anfragen aus dem MS MARCO Datensatz hat und somit besser zur Evaluation der Effektiviät der erstellten Rankings bezüglich NDCG@10, P@1 und MRR@10 geeignet ist.

4.2 Implementations-Details

Für die Erstellung der Datensätze für die IR Axiom-Präferenz Aufgabe und der initialen Rankings verwenden wir das Pyserini-Toolkit (Release 0-13-0)[43]. Wir nutzen den Standard-Tokenizer, den Porter-Stemmer und die Standard-Stoppwortliste für die englische Sprache aus dem Pyserini-Toolkit zur Erstellung des Indexes und dem Ranken von Dokumenten. Die Top-100 Dokumente für die IR Axiom-Präferenz Aufgaben und die initialen Rankings werden mittels BM25 mit den im Pyserini-Toolkit voreingestellten Parametern $k_1 = 0.9$ und $b = 0.4$ bestimmt. Für die Berechnung der IR Axiom-Präferenzen verwenden wir die von Völske et al. [76] veröffentlichte Implementation der IR Axiome und nehmen Kompatibilitätsanpassungen für den Einsatz mit dem Pyserini-Toolkit vor.

Für das Multi-Task-Learning Setup verwenden wir das MT-DNN-Toolkit,[4] welches für Natural-Language-Understanding Aufgaben entworfen wurde, je-

[2] `https://trec.nist.gov/data/deep2019.html`
[3] `https://trec.nist.gov/data/deep2020.html`
[4] `https://github.com/microsoft/mt-dnn`

doch im Allgemeinen für beliebige Aufgaben verwendet werden kann [46]. Das MT-DNN-Toolkit ermöglicht es, die vordefinierten NLU Aufgaben um weitere Aufgaben zu erweitern, aus mehreren vortrainierten Transformer Modellen wie BERT-Base-Uncased und ALBERT auszuwählen und Machine-Learning Techniken wie Dropout, Knowledge-Distillation und SAN direkt anzuwenden [16]. Wir verwenden in unseren Experimenten das BERT-Base-Uncased Modell, welches wie BERT auf den Aufgaben Masked-Lanugage-Modeling und Next-Sentence-Prediction vortrainiert wurde, jedoch die Groß- und Kleinschreibung der Eingaben vereinheitlicht. Die allgemeine Multi-Task-Learning Architektur des MT-DNN-Toolkits basiert auf dem Fine-Tuning von vortrainierten Transformer Modellen für mehrere Aufgaben gleichzeitig, indem in jedem Trainingsschritt zufällig eine Aufgabe mit einem dazugehörigen noch nicht verwendeten Trainings-Batch zum Training ausgewählt wird. Die Aufgaben im Multi-Task-Learning teilen sich untereinander den Encoder zur Erstellung der Repräsentationen von den aufgabenspezifischen Eingaben und bestimmen mittels den aufgabenspezifischen Decodern eine Ausgabe für die jeweilige Aufgabe. Wir definieren die Ranking Aufgabe als eine Regressionsaufgabe mit einem Pointwise-Ansatz [41], so dass eine Trainings-Eingabe aus einer Anfrage, einer Passage und dem Relevanzlabel 0 oder 1 besteht. Die Ranking Aufgabe gibt durch die Definiton als eine Regressionsaufgabe für eine Test-Eingabe aus einer Anfrage und einem Dokument einen nummerischen Wert zurück, den wir als eine Relevanzbewertung auf einer Ordinalskala interpretieren. Die IR Axiom-Präferenz Aufgabe definieren wir als eine Klassifikationsaufgabe, die für eine Eingabe aus einer Anfrage und zwei Dokumenten Entscheiden muss, zu welche der Axiom-Präferenz Klassen 1, 0 und -1 die Eingabe gehört. Da BERT nur auf ein und zwei Texten als Eingabe trainiert wurde, testeten wir für die Eingabe der IR Axiom-Präferenz Aufgaben die Eingabe Variante A:

$$[CLS]Anfrage + Dokument_1[SEP]Anfrage + Dokument_2[SEP]$$

welche eine klare Unterscheidung der Dokumente erlaubt und dafür die Anfrage nicht von den Dokumenten unterscheiden kann und die Eingabe Variante B:

$$[CLS]Anfrage[SEP]Dokument_1 + Dokument_2[SEP]$$

welche die Anfrage klar von den Dokumenten unterscheidet und dafür die Dokumente untereinander nicht unterscheiden kann. Die Text-Eingaben werden in beiden Varianten durch ein Leerzeichen getrennt, so dass nicht das letzte und erste Token durch die Konkatenation verschmelzen. Da in den anfänglichen Experimente mit den Varianten A und B die Variante B ähnliche oder bessere Ergebnisse als die Variante A erzielte (A.2.2, A.2.3) und die Variante B durch die Trennung der Anfrage von den Dokumenten ähnlicher zur Eingabe

der Ranking Aufgabe ist, verwenden wir in unseren Experimenten stets die Eingabe Variante B und gehen im nächsten Kapitel nur auf die Experimente ein, die die Eingabevariante B nutzen.

Für die Wahl der restlichen Parameter für das Multi-Task-Learning Setup folgen wir den Entscheidungen von Liu et al. [46], die das MT-DNN-Toolkit für das Multi-Task-Learning von Natural-Language-Understanding Aufgaben verwendeten. Wir verwenden Adamax [40] als Optimierer für das stochastische Gradientenabstiegsverfahren mit einer Lernrate von 5e-5. Nach einer 0.1 Warm-Up Phase fällt die Lernrate linear ab. Wir setzen den Dropout für die Ranking und IR Axiom-Präferenz Aufgaben auf 0.1 und verwenden für jede Aufgabe ein SAN, um Overfitting zu verhindern und effektivere Ergebnisse zu erzielen. Wir verwenden für die Ranking Aufgabe den mittleren quadratischen Fehler und für die IR Axiom-Präferenz Aufgaben die Kreuzentropie als Verlustfunktionen [32]. Zur Beschränkung der notwendigen Rechen- und Speicherkapazitäten setzen wir die Batchgröße auf 16, die maximale Eingabelänge auf 512 Token und trainieren über 5 Epochen.

4.3 Auswahl der Experimente

Da die Anzahl an möglichen Experimenten für eine Ranking Aufgabe und 20 IR Axiom-Präferenzaufgaben mit möglichen unterschiedlichen Trainingsverhältnissen unbeschränkt ist, begründen wir von uns getroffene weitere Annahmen, die unsere Auswahl an Experimenten motivieren. Wir nehmen an, dass eine IR Axiom-Präferenz keine Aussage über eine Ranking Präferenz, bezüglich der Facette an Relevanz die das IR Axiom abdeckt, erlaubt, wenn die Vorbedingung des Axioms nicht erfüllt ist [34, 76]. Wir nehmen deshalb für die IR Axiom-Präferenz Aufgaben nur Trainings- und Test-Daten, bei denen die Vorbedingung erfüllt ist. Für die IR Axiom-Präferenz Aufgaben gehen wir davon aus, dass ein Training auf eine zufällige Strichprobe dem BERT Modell nicht die Bestimmung der IR Axiom-Präferenzen, sondern die relativen Häufigkeiten der IR Axiom-Präferenz Klassen antrainiert [29]. Damit BERT die IR Axiom-Präferenz und nicht die relativen Häufigkeiten erlernt, sind die Trainingsdaten für die IR Axiom-Präferenz Aufgaben stets stratifizierte Stichproben. Wir nehmen auch an, dass die Ergebnisse des Modells, wenn es nur auf wenige Trainingsdaten trainiert wurde, einer Varianz unterliegt, die durch das mehrmalige Training des Modells mit den gleichen Trainingsdaten und das Bilden von einem Ergebnis-Mittelwerten mit einem 95% Konfidenzintervall abgebildet werden kann [55]. Damit wir die Ergebnisse auch bei wenigen Trainingsdaten interpretieren können, wiederholen wir Experimente für wenige Trainingsdaten mehrfach und interpretieren die Ergebnisse anhand des Mittelwertes sowie

Tabelle 4.1: Die klassenweise Verteilung der IR Axiom-Präferenzen unter Berücksichtigung der Vorbedingung. Es wurden insgesamt je IR Axiom 3.465.000 Präferenzen unter Verwendung von 700 Anfragen und allen Dokumentpaaren aus dem Top-100 BM25 Retrieval berechnet.

	Vorbedingung=Wahr			**Vorbedingung=Falsch**		
IR Axiom	−1	0	1	−1	0	1
TFC1	216.744	165.080	211.852	1.205.526	427.602	1.193.196
TFC3	4.544	585.053	4.079	22.172	2.828.823	20.329
M-TDC	10.616	51.161	10.164	307.592	2.787.440	298.027
LB1	610.758	1.111.645	604.088	452.123	278.279	407.107
LNC1	104.010	14.647	99.261	1.596.916	61.800	1.588.366
TF-LNC	73.851	3.317.960	73.189	0	0	0
REG	951.658	1.562.557	950.785	0	0	0
ANTI-REG	1.089.194	1.300.155	1.075.651	0	0	0
DIV	1.695.911	116.491	1.652.598	0	0	0
STMC1	1.616.600	283.596	1.564.804	0	0	0
STMC2	59.472	3.342.367	63.161	0	0	0
PROX1	491.495	145.271	485.691	573.601	1.207.005	561.937
PROX2	538.603	52.329	521.525	1.018.009	309.503	1.015.031
PROX3	16.968	1.089.948	15.541	10.600	2.326.608	5.335
PROX4	1.606.588	261.922	1.594.190	0	0	0
PROX5	1.544.705	360.953	1.557.042	0	0	0
RS-TF	1.718.949	43.852	1.702.199	0	0	0
RS-TF-IDF	1.738.875	22.628	1.703.497	0	0	0
RS-BM25	1.774.654	19.477	1.670.869	0	0	0
RS-QL	1.772.231	19.473	1.673.296	0	0	0

des 95% Konfidenzintervall um den Mittelwert. Des weiteren gehen wir davon aus, dass die beste Kombination an IR Axiom-Präferenz Aufgaben zur Verbesserung der Ranking Aufgabe und das optimale Trainingsverhältnis der Aufgaben untereinander nicht analytisch bestimmt werden kann [16, 28]. Die Ungewissheit über ein optimales Setup zusammen mit der Annahme, dass jedes IR Axiom eine andere spezifische Facette an Relevanz abbildet, die alle gleich wichtig für ein effektives Ranking sind, führt uns zu der Entscheidung, das Verhältnis von 1:1 der Trainingsdaten der IR Axiom-Präferenz Aufgabe untereinander einzuhalten.

Kapitel 5

Evaluation

In dem folgenden Kapitel stellen wir die Metriken Normalized Discounted Cumulative Gain (NDCG), Mean Reciprocal Rank (MRR), Precision (P) und Accuracy (ACC) im Kontext der Ranking und Klassifikations Aufgabe vor. Mittels dieser Metriken interpretieren wir die Ergebnisse unseres Ansatzes in den Experimenten bezüglich der Effektivität. Wir stellen für einen besseren Lesefluss nur ausgewählte Ergebnisse von ausgewählten Epochen unserer Experimente in diesem Kapitel dar und verweisen für tabellarische und grafische Darstellungen aller Ergebnisse unserer Experimente auf den Anhang A.

5.1 Evaluationsmetriken

Das Hauptziel im Information-Retrieval ist es das Informationsbedürfnis eines Nutzers zu erfüllen. In der Ad-Hoc Suche wird das Informationsbedürfnis eines Nutzers durch eine Anfrage abgebildet und durch die Erstellung eines Rankings vermuteter relevanter Dokumente erfüllt. Wie effektiv ein Ranking das Informationsbedürfnis eines Nutzers erfüllt, wird mittels Relevanzbewertungen für Dokument-Anfrage Paare und einer Evaluationsmetrik bestimmt. Die Relevanzbewertungen werden von Menschen erstellt und je nach verwendeter Evaluationsmetrik bezüglich bestimmter Kriterien ausgewertet.

Betrachten wir Sprach-Assistenten wie Cortana, dann sollen diese auf natürlichsprachliche Fragen eines Nutzers mit der relevantesten Antwort reagieren. Eine Evaluationsmetrik, die die Relevanz nur bezüglich dem ersten Retrieval-Ergebnis bewertet, ist Precision@1 [69] die wie folgt für eine Anfrage definiert ist:

$$P@1 = relevance_1$$

Precision@1 betrachtet nur die Relevanz des ersten Dokumentes und gibt an, ob dieses relevant ist. Bezogen auf MS MARCO können wir mit Precision@1

evaluieren, wie gut unser Learning-to-Rank Ansatz darin ist, auf eine Anfrage genau eine Antwort-Passage zu finden. Da abseits von Sprach-Assistenten ein Nutzer nicht nur das erste Dokument auf der Suche nach einer Antwort in einem Ranking betrachten würde, evaluieren wir unseren Ansatz auch bezüglich MRR@10 [69].

$$MRR@10 = \frac{1}{\text{rank first relevant}}$$

MRR@10 betrachtet die ersten 10 Dokumente des Rankings und gibt das Reziproke des Ranges des ersten relevanten Dokumentes zurück. Mittels MRR@10 können wir untersuchen, wie gut unser Learning-to-Rank Ansatz im möglichst hohen Ranking einer Antwort-Passage ist und mit anderen auf MS MARCO evaluierten Ranking-Ansätzen vergleichen. Betrachten wir das von MRR implizierte Nutzerverhalten, dann würde ein Nutzer nach MRR sich das Ranking in absteigender Reihenfolge anschauen und beim Finden einer relevanten Antwort die nachfolgenden Dokumente des Rankings nicht weiter betrachten. Außerdem impliziert MRR, dass ein höheres Ranking eines relevanten Dokumentes förderlicher für die Erfüllung des Informationsbedürfnisses des Nutzers ist. Das ein Nutzer das Ranking von Rang 1 absteigend in genau der Reihenfolge des Rankings durchsucht und beim Finden einer relevanten Antwort die nachfolgenden Dokumente des Rankings nicht weiter betrachtet, sind Annahmen über Nutzer, die in der Realität nur in Ausnahmefällen zutriffen [57, 63]. Eine Evaluationsmetrik, die Berücksichtigt das Nutzer nach dem ersten relevanten Dokumente noch weitere Dokumente betrachten und hoch gerankte relevante Dokumente förderlicher zur Erfüllung des Informationsbedürfnisses eines Nutzers sind, ist DCG@10 [37].

$$DCG@10 = \sum_{rank=1}^{10} \frac{relevance_{rank}}{log_2(rank+1)}$$

Da sich die Menge an verfügbaren Relevanzbewertungen für mehrere Anfragen unterscheidet, kann sich der maximal erreichbaren DCG@10 Wert für zwei Anfragen erheblich unterscheiden. Damit die positiven Eigenschaften von DCG@10 nicht verloren gehen und die für zwei Anfragen erreichten Werte vergleichbarer sind, wurde DCG@10 zu NDCG@10 erweitert [37].

$$NDCG@10 = \frac{DCG@10}{iDCG@10}$$

Der NDCG@10 ergibt sich als der DCG@10 mit einer Normalisierung durch den optimalen iDCG@10 Wert. Der iDCG@10 Wert betrachtet alle Verfügbaren Relevanzbewertungen und bestimmt aus diesen den höchsten erreichbaren DCG@10 Wert. Wir betrachten für unseren Learning-to-Rank Ansatz NDCG@10 und MRR@10, da die erste Ergebnisseite in der Ad-Hoc Suche in der

Regel genau 10 Dokumente darstellt und Ansätze für den TREC Deep Learning Track aus dem Jahr 2019 und 2020 ebenfalls NDCG@10 und MRR@10 berichten. Wir können damit die Effektivität unseren Learning-to-Rank Ansatz im Kontext weiterer Ansätze evaluieren und das Potenzial unseres Ansatzes abschätzen.

Für die IR Axiom-Präferenz Klassifikationsaufgabe benötigen wir eine Evalutionsmetrik, die ein Verhältnis zwischen der Anzahl an richtigen Klassifikationen und der Anzahl an falschen Klassifikationen ausdrückt. Da wir für die IR Axiom-Präferenz Aufgabe jede Klasse als gleich bedeutsam annehmen, können wir die Evaluationsmetrik Accuracy verwenden [36]. Die Accuracy berechnet sich als die Anzahl richtiger Klassifikationen durch die Gesamtanzahl der getätigten Klassifikationen.

$$Accuracy = \frac{\text{Anzahl richtiger Klassifikationen}}{\text{Anzahl aller Klassifikationen}}$$

5.2 Experiment Ergebnisse

Tabelle 5.1: Übersicht der Ergebnisse unserer Baselines und ausgewählter Ansätze aus den TREC Deep Learning Overview-Papern [14, 15] auf den 43 und 54 Anfragen des TREC Deep Learning Tracks aus den Jahren 2019 und 2020. $BM25_{Buop}$ stellt die besten möglichen Ergebnisse dar, die durch ein Reranking des initialen Rankings erreicht werden können. Für $BM25_{uop}$ sind die Parameter $k_1 = 0.9$ und $b = 0.4$ und für $BM25_{op}$ sind die Parameter $k_1 = 0.82$ und $b = 0.68$ nach der Empfehlung des Pyserini-Toolkit für den MS MARCO Passage Datensatz.

	TREC 2019			**TREC 2020**		
Ansatz	NDCG@10	P@1	MRR@10	NDCG@10	P@1	MRR@10
$BM25_{op}$	0.4973	0.5581	0.6821	0.4876	0.5370	0.6554
$BM25_{uop}$	0.5058	0.0000	0.7024	0.0000	0.5370	0.6533
$BM25_{Buop}$	0.9640	0.9767	0.9767	0.9685	1.0000	1.0000
ICT-BERT2	0.6650	–	0.8743	–	–	–
nlm-prfun-bert	–	–	–	0.6648	–	0.8603

Um die Effektivität unseres Learning-to-Rank Ansatzes in den Kontext weiterer Ansätze zu stellen, definieren wir in Tabelle 5.1 mehrere Baselines. $BM25_{op}$ und $BM25_{uop}$ bieten als einfach zu implementierende und nicht stark optimierte klassische Ansätze eine untere Vergleichsgrenze. $BM25_{Buop}$ zeigt als obere Grenze unseres Learning-to-Rank Ansatzes die bestmöglichen Ergebnisse, die von der Ranking Aufgabe durch ein Reranking des initialen Ranking

erreicht werden können. Die Ergebnisse von $BM25_{Buop}$ zeigt, dass die Effektivität der Ranking Aufgabe nicht maßgeblich durch BM25 zur Erstellung der initialen Rankings beschränkt wird. Wir wählen ICT-BERT2 [15] und nlm-prfun-bert [14] als weitere Baselines, da diese sich bezüglich NDCG@10 im Mittelfeld unter allen für den TREC19 und TREC20 eingereichten Ansätzen befinden und auf den Einsatz von BERT basieren. ICT-BERT2 und nlm-prfun-bert sind für unseren Ansatz kompetitive Baselines, die wir mit unseren Untersuchungen versuchen zu erreichen, um das Potenzial des Multi-Task-Learnings von einer Ranking Aufgabe und IR Axiom-Präferenz Aufgaben aufzuzeigen. Zusätzlich zu diesen Baselines betrachten wir die Ergebnisse der IR Axiom-Präferenz Aufgaben und der Ranking Aufgabe (Tabelle 5.3) ohne Multi-Task-Learning (Single-Task), um Vergleichswerte zur Untersuchung des Einflusses des Multi-Task-Learnings auf die Effektivität der einzelnen Aufgaben zu erhalten. Die Single-Task Setups für die IR Axiom-Präferenz Aufgaben erlauben es uns einzuschätzen, inwiefern die einzelnen IR Axiom-Präferenz Aufgaben durch BERT erlernbar sind. Abbildung 5.1 gibt eine Übersicht über die Accuracy Ergebnisse des Single-Task Setups der Termhäufigkeits-Axiome. Der Wachstum des Scores für die None-Unique-Testdaten und die Stagnation oder das Sinken der Scores für die Unique-Testdaten zeigen das zunehmende Overfitting des Modells auf die Trainingsdaten über die Trainingsepochen. Außerdem zeigen die Score Unterschiede zwischen den Stratified- und Random-Datensätzen für TFC3, dass BERT für unbalancierte Klassenverteilung zur Präferenz Klasse 0 (Tabelle 4.1) die Klassifikationsentscheidung Präferenz gleich 0 oder Präferenz ungleich 0 besser erlernt. Die Unterschiede zwischen LB1 und M-TDC in dem Random- und Stratified-Datensätzen für Unique und None-Unique bedeuten, dass die Axiome unterschiedlich gut von BERT erlernt werden. Für die anderen Axiome-Gruppen im Single-Task Setup zeigten sich die gleichen Beobachtungen wie für die Termhäufigkeits-Axiome. Aus den Single-Task Setup ist kein Axiom zu erkennen, dass gar nicht von BERT erlernt werden kann.

Da wir durch die Single-Task Setups kein nicht erlernbares Axiom ausschließen können, betrachten wir nach den in Abschnitt 4.3 getroffenen Annahmen das Multi-Task-Learning von der Ranking Aufgabe und allen 20 IR Axiom-Präferenzaufgaben für jeweils 10.000 Trainingsdaten als Experiment 1. Der Vergleich der Ergebnisse der Ranking Aufgabe im Single-Task Setup in Abbildung 5.2 mit denen der Ranking Aufgabe im Experiment 1 in Abbildung 5.3 zeigt, dass das Multi-Task-Learning mit 20 IR Axiom-Präferenzaufgaben in einem $1 : \cdots : 1$ Trainingsverhältnis die Effektivität der Ranking Aufgabe verschlechtert. Die Verschlechterung der Ranking Ergebnisse erklären wir uns durch das stark ungleiche Verhältnis der Trainingsdaten zwischen der Ranking Aufgabe und allen IR Axiom Aufgaben. Für jede Eingabe für die Ranking Aufgabe sieht BERT 20 weitere Eingabe für 20 verschiedene IR Axiome, so

dass wenn wir das Ranken und die Axiome jeweils als einen übergeordneten Aufgabenkomplex betrachten, dass Verhältnis der Trainingsdaten 1:20 ist. Durch das ungleiche Trainingsverhältnis von 1:20 wird es BERT dann auch erschwert geeignete Repräsentationen für die Ranking Aufgabe zu bestimmen, da BERT mit dem Trainingsverhältnis von 1:20 vor allem auf das Erstellen von Repräsentationen für die IR Axiom Aufgaben trainiert wird. Die Accuracy Ergebnisse der Termhäufigkeits-Axiome in Abbildung 5.4 zeigen, dass sich das Multi-Task-Learning auf die einzelnen Axiome unterschiedlich ausgewirkt hat. Alle Axiome zeigen nach wie vor ein Overfitting auf den Trainingsdaten. Die Wirkung gegen Overfitting, die normalerweise beim Multi-Task-Learning beobachtet werden kann, ist in unserem Experiment 1 für die Axiome kaum zu beobachten, da sich Trainingsdaten für die einzelnen IR Axiom-Präferenz Aufgaben in den Dokumenten, Anfragen und Präferenz-Labeln teils überschneiden. Die teilweisen Überschneidungen der Trainingsdaten führen dazu, dass die IR Axiome STMC2, DIV, Retrieval-Score-Axiome, PROX3 und TF-LNC1 um bis zu 20% stärker als im Single-Task Setup auf die Trainingsdaten Overfitten und für die Random-Unique-Testdaten im Mittel um 15% schlechtere Ergebnisse über die Epochen hinweg erzielen. Die Ergebnisse zeigen, dass die Axiome LB1, M-TDC, ANTI-REG, REG und PROX5 durch das Multi-Task-Learning Setup effektivere Ergebnisse auf den Unique-Testdaten erzielten. Da die Ranking Aufgabe im Experiment 1 schlechtere Ergebnisse erzielte und nur die Axiome LB1, M-TDC, ANTI-REG, REG und PROX5 vom Multi-Task-Learning trotz Overfitting profitierten, betrachten wir für die Experimente 2, 3 und 4 das Trainingsverhältnis von der Ranking Aufgabe zu den IR Axiom-Präferenz Aufgaben insgesamt, Ändern das Trainingsverhältnis zu 2:1 für die Ranking Aufgabe und nehmen nur zwei IR Axiom-Präferenz Aufgaben für das Multi-Task-Learning mit der Ranking Aufgabe.

Für das Experiment 2 trainieren wir die Ranking Aufgabe mit 100.000 Trainingsdaten gemeinsam mit M-TDC und LB1 mit jeweils 25.000 Trainingsdaten, da M-TDC und LB1 bereits Verbesserungen im Experiment 1 zeigten. Da M-TDC und LB1 beides Termhäufigkeits-Axiome sind, erwarten wir keine negative Interferenzen zwischen den IR Axiom-Präferenz Aufgaben, so dass ein positiver Wissenstransfer zwischen IR Axiom-Präferenz Aufgaben und der Ranking Aufgabe erleichtert wird. Vergleichen wir die Ergebnisse der Ranking Aufgabe im Single-Task Setup in Abbildung 5.5 mit den Ergebnissen der Ranking Aufgabe im Experiment 2 in Abbildung 5.6 zeigt sich, dass Multi-Task-Learning mit M-TDC und LB1 bei dem Vergleich der besten Ergebnisse über alle Epochen den NDCG@10 von 0.6138 auf 0.6257 und den MRR@10 von 0.7494 auf 0.7605 für TREC20 verbessert. Obwohl sich die Ergebnisse für die anderen Metriken auf TREC19 und TREC20 im Multi-Task-Learning verschlechtern, zeigt die Verbesserung des NDCG@10 und MRR@10 für TREC20,

das Multi-Task-Learning die Effektivität der Ranking Aufgabe steigern kann. Die effektiveren Ergebnisse der Ranking Aufgabe im Experiment 2 gegenüber dem Single-Task Setup für die Epochen 3, 4 und 5 deutet auf die Wirkung des Multi-Task-Learnings gegen Overfitting hin. Die Accuracy Ergebnisse von M-TDC und LB1 in Abbildung 5.7 zeigen, dass im Setup von Experiment 2 weniger Overfitting auf die Trainingsdaten der IR Axiom-Präferenz Aufgaben statt findet. Das Overfitting wird durch das Trainingsverhältnis von 4:1:1 von der Ranking Aufgabe zu M-TDC und LB1 abgeschwächt. LB1 und M-TDC erzielen auf den Stratified-Unique-Testdaten vergleichbare Ergebnisse zum Single-Task Setup und erzielen bessere Ergebnisse auf den Random-Unique-Testdaten. Aus unseren Experimenten ist nicht direkt ersichtlich, ob die Verbesserung der Ergebnisse der IR Axiom-Präferenz Aufgaben ein Resultat der größeren Trainingsdatenmenge für die IR Axiome, des positiven Wissentransfers von der Ranking Aufgabe zu den Axiom Aufgaben oder ein Resultat des geringeren Overfittings auf die Trainingsdaten ist.

Für das Experiment 3 verwenden wir das Setup von Experiment 2 und untersuchen, ob sich die Ergebnisse für das Multi-Task-Learning mit den IR Axiomen M-TDC und LB1 auch für das Multi-Task-Learning mit IR Axiomen REG und ANTI-REG reproduzieren lassen. Die Wahl der IR Axiome REG und ANTI-REG, entspringt der Beobachtung der Verbesserung der Ergebnisse für REG und ANTI-REG im Experiment 1 und das die Präferenzen für REG und ANTI-REG sehr ähnlich definiert sind, so dass negative Interferenzen zwischen den Axiom Aufgaben nicht zu erwarten sind. Im Experiment 3 trainieren wir die Ranking Aufgabe mit 100.000 Trainingsdaten gemeinsam mit REG und ANTI-REG mit jeweils 25.000 Trainingsdaten. Ein Vergleich der Ergebnisse der Ranking Aufgabe im Single-Task Setup in Abbildung 5.5 mit den Ergebnissen der Ranking Aufgabe im Experiment 3 in Abbildung 5.8 zeigt, dass Multi-Task-Learning mit REG und ANTI-REG bei dem Vergleich der besten Ergebnisse über alle Epochen die P@1 um 3%, den NDCG@10 um 5% und den MRR@10 um 7% auf TREC20 verbessert. Gleichzeitig liegt die Verschlechterung der Ergebnisse für NDCG@10 und MRR@10 auf TREC19 bei unter 1.5%, so dass REG und ANTI-REG die Ergebnisse von M-TDC und LB1 im Experiment 2 bezüglich der Verbesserung der Ranking Aufgabe übertreffen. Ebenfalls deutet sich erneut die erwartete Wirkung des Multi-Task-Learnings gegen Overfitting an, indem das Multi-Task-Learning Setup in Experiment 3 für die Ranking Aufgabe effektivere Ergebnisse als das Single-Task Setup in den Epochen 3, 4 und 5 erzielt. Die Accuracy Ergebnisse von REG und ANTI-REG in Abbildung 5.9 zeigen, dass im Setup von Experiment 3 weniger Overfitting auf die Trainingsdaten der IR Axiom-Präferenz Aufgaben statt findet. Das Overfitting wird wie in Experiment 2 durch das Trainingsverhältnis von 4:1:1 von der Ranking Aufgabe zu REG und ANTI-REG abgeschwächt. REG und

ANTI-REG erzielen auf den Stratified- und Random-Unique-Testdaten rund 10% bessere Ergebnisse als im Single-Task Setup. Abseits der in Experiment 2 genannten möglichen Erklärungen für diese Verbesserung der Accuracy auf den Unique-Testdaten, ist im Fall des Experimentes 3 die Verbesserung durch den positiven Wissenstransfer von REG und ANTI-REG untereinander plausibler, da REG und ANTI-REG in der Defintion und Berechnung der Axiom-Präferenzen sich ähnlicher als M-TDC und LB1 sind.

Für das Experiment 4 verwenden wir das Setup von Experiment 3 und untersuchen, ob sich die Ergebnisse für das Multi-Task-Learning mit den Anfrage-Facetten-Axiomen REG und ANTI-REG auch für das Multi-Task-Learning mit den Term-Nachbarschafts-Axiomen PROX4 und PROX5 reproduzieren lassen. Wir wählen das Term-Nachbarschafts-Axiom PROX5, da dieses sowohl im Experiment 1 als auch im Single-Task Setup bessere Ergebnisse als die anderen Term-Nachbarschafts-Axiome erzielt. Damit die Ergebnisse von Experiment 4 genauer mit denen von Experiment 2 und 3 verglichen werden können, brauchten wir noch eine weitere Term-Nachbarschafts-Axiom Aufgabe. Für die Wahl von PROX4 als zweite Axiom-Aufgabe orientieren wir uns an der Vermutung aus Experiment 3, dass eine ähnliche Definition und Berechnung der Präferenzen den positiven Wissenstransfer zwischen den Axiom Aufgaben fördert. Für das Experiment 4 trainieren wir die Ranking Aufgabe mit 100.000 Trainingsdaten gemeinsam mit PROX4 und PROX5 mit jeweils 25.000 Trainingsdaten. Vergleichen wir die Ergebnisse der Ranking Aufgabe im Single-Task Setup in Abbildung 5.5 mit den Ergebnissen der Ranking Aufgabe im Experiment 4 in Abbildung 5.6 zeigt sich, dass Multi-Task-Learning mit PROX4 und PROX5 bei dem Vergleich der besten Ergebnisse über alle Epochen die Ergebnisse auf TREC20 in einem kleineren Ausmaß als REG und ANTI-REG im Experiment 3 verbessert. Im Gegensatz zu Experiment 3 erzielt die Ranking Aufgabe in Experiment 4 um im Mittel 8% schlechtere Ergebnisse auf TREC19 als das Single-Task Setup. Die Minderung des Overfitting auf die Ranking Aufgabe durch Multi-Task-Learning ergibt sich auch im Experiment 4, indem die Effektivität der Ergebnisse im Experiment 4 für die Epochen 3, 4 und 5 höher ist als im Single-Task Setup. Die Accuracy Ergebnisse von PROX4 und PROX5 in Abbildung 5.11 zeigen, dass durch das Trainingsverhältnis von 4:1:1 im Setup von Experiment 4 weniger Overfitting auf die Trainingsdaten der IR Axiom-Präferenz Aufgaben statt findet. PROX4 und PROX5 erzielen auf den Stratified- und Random-Unique-Testdaten rund 7% bessere Ergebnisse als im Single-Task Setup. Im Gegensatz zum Single-Task Setup von PROX5 sinkt im Multi-Task-Learning von Experiment 4 die Accuracy über den Verlauf der Epochen. Wir schätzen, dass die Abnahme der Accuracy über den Verlauf der Epochen durch das zunehmende Overfitting der Ranking Aufgabe zustande kommt. Wir vermuten, dass das Overfitting der Ranking Aufgabe

einen negativen Wissenstransfer auf PROX5 ausübt.

Zuletzt betrachten wir dann im Experiment 5, ob sich die in Experiment 2, 3 und 4 erzielten Verbesserungen mittels Multi-Task-Learning zu einer größeren Verbesserung der Effektivität der Ranking Aufgabe kombinieren lassen. Im Experiment 5 trainieren wir die Ranking-Aufgabe auf 100.000 Trainingsdaten gemeinsam mit M-TDC, LB1, REG, ANTI-REG, PROX4 und PROX5 auf jeweils 25.000 Trainingsdaten. Ein Vergleich der Ergebnisse der Ranking Aufgabe im Single-Task Setup in Abbildung 5.5 mit den Ergebnissen der Ranking Aufgabe im Experiment 5 in Abbildung 5.12 zeigt, dass Multi-Task-Learning M-TDC, LB1, REG, ANTI-REG, PROX4 und PROX5 bei dem Vergleich der besten Ergebnisse über alle Epochen vergleichbare Ergebnisse auf TREC19 erreicht und die Ergebnisse für TREC20 verbessert. Die Ranking Aufgabe im Experiment 5 erzielt auf P@1 und MRR@10 rund 3% schlechtere Ergebnisse als die Ranking Aufgabe aus Experiment 3. Im Experiment 5 zeigt sich erneut die Wirkung des Multi-Task-Learnings mit den Axiom Aufgaben gegen das Overfitting der Ranking Aufgabe. Die Accuracy Ergebnisse von M-TDC, LB1, REG, ANTI-REG, PROX4 und PROX5 in der Tabelle 5.2 zeigen, dass im Setup von Experiment 5 trotz dem Trainingsverhältnis von 3:2 für die Axiom-Aufgaben weniger Overfitting auf die Trainingsdaten der IR Axiom-Präferenz Aufgaben statt findet. PROX4 und PROX5 erzielen im Experiment 5 im Schnitt 7% schlechtere Ergebnisse auf den Random- und Stratified-Unique Testdaten, so dass wir davon ausgehen, dass eine Interferenz im Training zwischen den Term-Nachbarschaft-Axiomen PROX4, PROX5 und einem oder mehreren Axiomen von REG, ANTI-REG, M-TDC und LB1 vorliegt. Für die Axiome REG, ANTI-REG, M-TDC und LB1 liegen die Accuracy Ergebnisse für die Random- und Stratified-Unique Testdaten auf einem vergleichbaren Niveau mit Experiment 3 beziehungsweise Experiment 2.

Vergleichswerte aus dem Single-Task Setup und die besten Ergebnisse über alle Epochen für die Experimente 2, 3, 4 und 5 wird für die Ranking Aufgabe in Tabelle 5.3 und für die Axiom Aufgaben in Tabelle 5.2 dargestellt. Die Accuracy Ergebnisse für die Axiom Aufgaben deuten darauf hin, dass zwischen den Axiomen im Multi-Task-Learning sowohl ein positiver Wissenstransfer als auch hinderliche Interferenzen auftreten. Die Ranking Aufgabe konnte durch das Multi-Task-Learning mit Axiomen effektivere Ergebnisse erzielen. Die Ranking Aufgabe wirkt außerdem gegen das Overfitting auf die Axiom Trainingsdaten, sowie die Axiom Aufgaben gegen das Overfitting auf die Ranking Trainingsdaten wirken. Die besten Ergebnisse der Ranking Aufgabe im Multi-Task-Learning Setup erreichen vergleichbare NDCG@10 Wert zu den kompetetiven Baselines ICT-BERT2 und nlm-prfun-bert und validieren damit das Potenzial des Multi-Task-Learning von einer Ranking Aufgabe mit Axiom Hilfsaufgaben. Die Ergebnisse für die Axiom Aufgaben und dabei ins-

besondere die Ergebnisse von REG und ANTI-REG lassen vermuten, dass bei unserem verwendeten Eingabeformat der Axiom Trainingsdaten als [CLS] q [SEP] $d_1 + d_2$ [SEP] nicht die zu erlernenden Eigenschaften wie Termhäufigkeit und Dokumentlänge (Tabelle 3.1) die Erlernbarkeit der Axiome bestimmen, sondern wie wichtig die klare Trennung der Anfrage und Dokumente untereinander zur Berechnung der Präferenz ist.

Tabelle 5.2: Übersicht der besten Accuracy Ergebnisse der Axiom Aufgabe aus allen 5 Epochen des Experimentes 2 $Multi_{Term}$, des Experimentes 3 $Multi_{Anfrage}$, des Experimentes 4 $Multi_{Prox}$ und des Experimentes 5 $Multi_{Multi}$ mit 25.000 Trainingsdaten für jede Axiom Aufgabe. Zum Vergleichen geben wir die besten Accuracy Ergebnisse aus allen 5 Epochen der jeweiligen Axiom Aufgabe ohne Multi-Task-Learning für 10.000 Trainingsdaten mit 5 Wiederholungen und einem 95% Konfidenzintervall als *Single* an.

		Stratified		**Random**	
Ansatz	**Axiom**	None-Unique	Unique	None-Unique	Unique
Single	REG	0.6097±0.0393	0.3403±0.0094	0.6352±0.0431	0.3293±0.0038
$Multi_{Anfrage}$	REG	0.4637	0.3700	0.4850	0.3670
$Multi_{Multi}$	REG	0.4720	0.3730	0.4850	0.3690
Single	ANTI-REG	0.5928±0.0983	0.3303±0.0123	0.5968±0.0987	0.3197±0.0152
$Multi_{Anfrage}$	ANTI-REG	0.4460	0.3620	0.4540	0.3570
$Multi_{Multi}$	ANTI-REG	0.4587	0.3350	0.4633	0.3440
Single	M-TDC	0.6681±0.0273	0.3580±0.0310	0.7706±0.0146	0.5287±0.0971
$Multi_{Term}$	M-TDC	0.5240	0.3420	0.5733	0.6300
$Multi_{Multi}$	M-TDC	0.5407	0.3530	0.5690	0.5510
Single	LB1	0.6382±0.0525	0.3860±0.0197	0.6709±042846	0.4990±0.0474
$Multi_{Term}$	LB1	0.5050	0.4000	0.5447	0.6360
$Multi_{Multi}$	LB1	0.4990	0.3910	0.5323	0.6010
Single	PROX4	0.6217±0.0265	0.3730±0.0203	0.5673±0.0251	0.2960±0.0584
$Multi_{Prox}$	PROX4	0.4777	0.3840	0.4210	0.3170
$Multi_{Multi}$	PROX4	0.4737	0.3690	0.4227	0.2830
Single	PROX5	0.7000±0.0384	0.4797±0.0255	0.6073±0.0548	0.4683±0.0165
$Multi_{Prox}$	PROX5	0.6193	0.5220	0.5277	0.5160
$Multi_{Multi}$	PROX5	0.6167	0.4740	0.5267	0.4670

Tabelle 5.3: Übersicht der besten Ergebnisse der Ranking Aufgabe aus allen 5 Epochen des Experimentes 2 $Multi_{Term}$, des Experimentes 3 $Multi_{Anfrage}$, des Experimentes 4 $Multi_{Prox}$ und des Experimentes 5 $Multi_{Multi}$, die jeweils dieselben 100.000 Trainingsdaten für die Ranking Aufgabe erhalten. Die Experimente unterscheiden sich in den weiteren Aufgaben, die im Multi-Task-Learning eingesetzt werden. Zum Vergleichen geben wir die besten Ergebnisse aus allen 5 Epochen der Ranking Aufgabe ohne Multi-Task-Learning für 100.000 Trainingsdaten $Rank_{100t}$ und 1.000.000 Trainingsdaten $Rank_{1m}$ an. Die besten Ergebnisse unserer Ansätze sind fett markiert.

	TREC 2019			**TREC 2020**		
Ansatz	NDCG@10	P@1	MRR@10	NDCG@10	P@1	MRR@10
$Rank_{100t}$	0.6855	0.7674	0.8547	0.6138	0.6481	0.7494
$Rank_{1m}$	0.6963	0.7442	0.8341	0.6747	0.6852	0.7968
$Multi_{Multi}$	**0.6843**	0.7674	**0.8547**	0.6551	0.6481	0.7687
$Multi_{Term}$	0.6741	0.6977	0.7940	0.6257	0.6296	0.7605
$Multi_{Anfrage}$	0.6830	**0.7674**	0.8434	**0.6558**	**0.6667**	**0.7901**
$Multi_{Prox}$	0.6584	0.6512	0.7469	0.6428	0.6667	0.7670

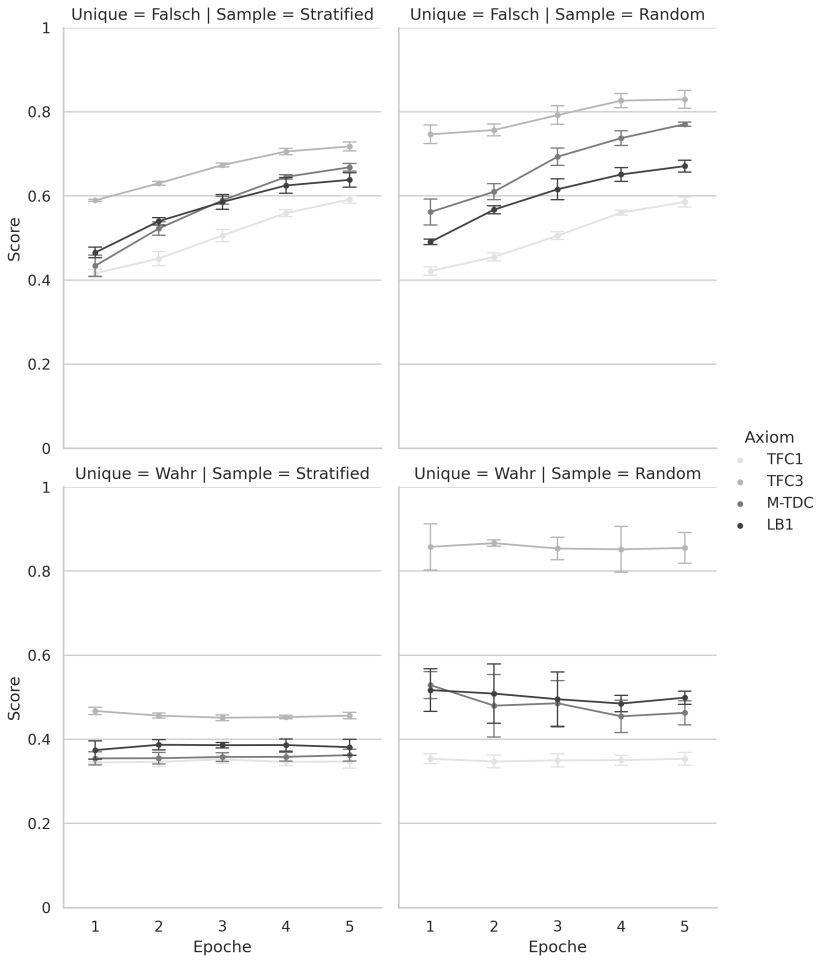

Abbildung 5.1: Accuracy Ergebnisse der Termhäufigkeits-Axiome für 10.000 Trainingsdaten mit einem 95% Konfidenzintervall um den Mittelwert der Accuracy Ergebnisse für 5 Wiederholungen. Jedes Axiom wurde ohne Multi-Task-Learning trainiert.

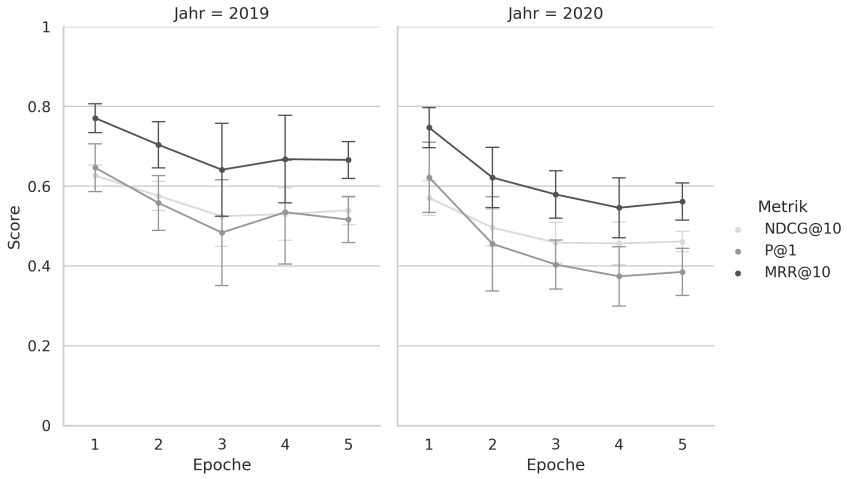

Abbildung 5.2: Ergebnisse der Ranking Aufgabe für 10.000 Trainingsdaten mit einem 95% Konfidenzintervall um den Mittelwert der Ergebnisse für 5 Wiederholungen.

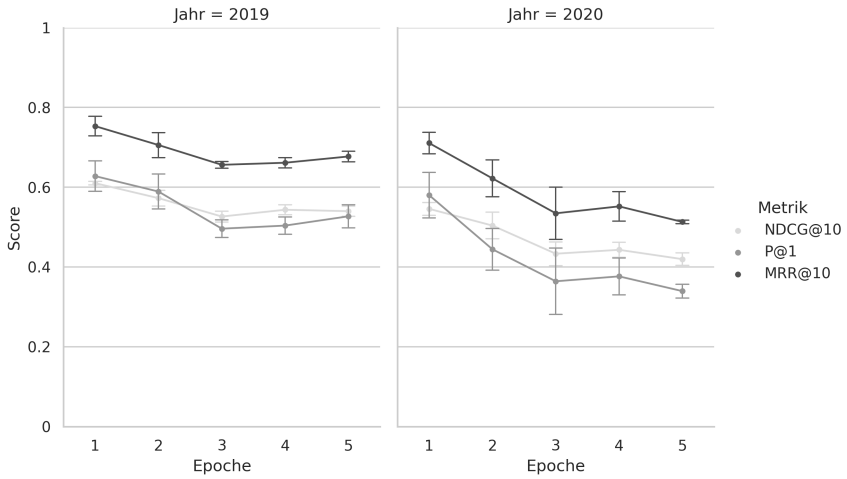

Abbildung 5.3: Ergebnisse der Ranking Aufgabe für 10.000 Trainingsdaten mit einem 95% Konfidenzintervall um den Mittelwert der Accuracy Ergebnisse für 3 Wiederholungen im Experiment 1. Es wurde Multi-Task-Learning mit 20 Axiom Aufgaben und einer Ranking Aufgabe zu jeweils 10.000 Trainingsdaten durchgeführt.

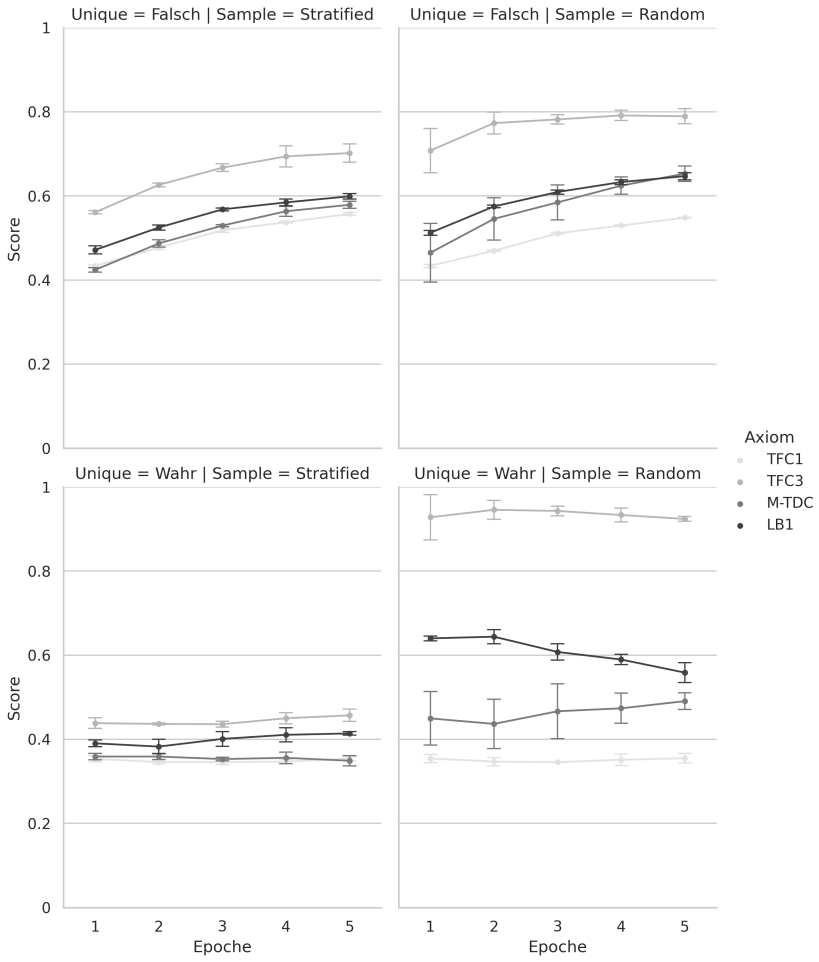

Abbildung 5.4: Accuracy Ergebnisse der Termhäufigkeits-Axiome für 10.000 Trainingsdaten mit einem 95% Konfidenzintervall um den Mittelwert der Accuracy Ergebnisse für 3 Wiederholungen im Experiment 1. Es wurde Multi-Task-Learning mit 20 Axiom Aufgaben und einer Ranking Aufgabe zu jeweils 10.000 Trainingsdaten durchgeführt.

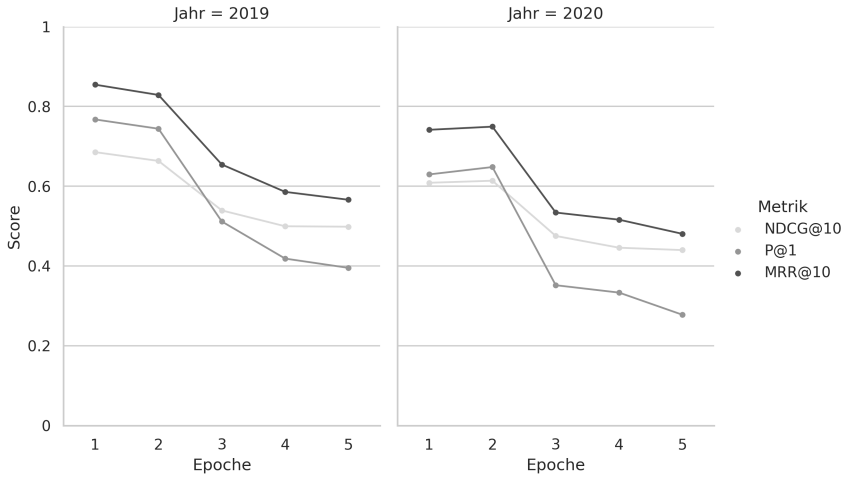

Abbildung 5.5: Ergebnisse der Ranking Aufgabe für 100.000 Trainingsdaten im Single-Task Setup.

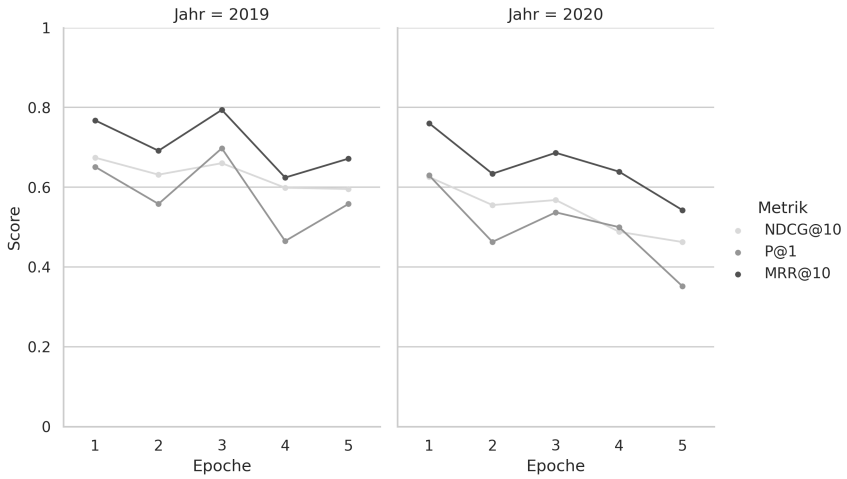

Abbildung 5.6: Ergebnisse der Ranking Aufgabe für 100.000 im Experiment 2. Es wurde Multi-Task-Learning mit M-TDC und LB1 zu jeweils 25.000 Trainingsdaten durchgeführt.

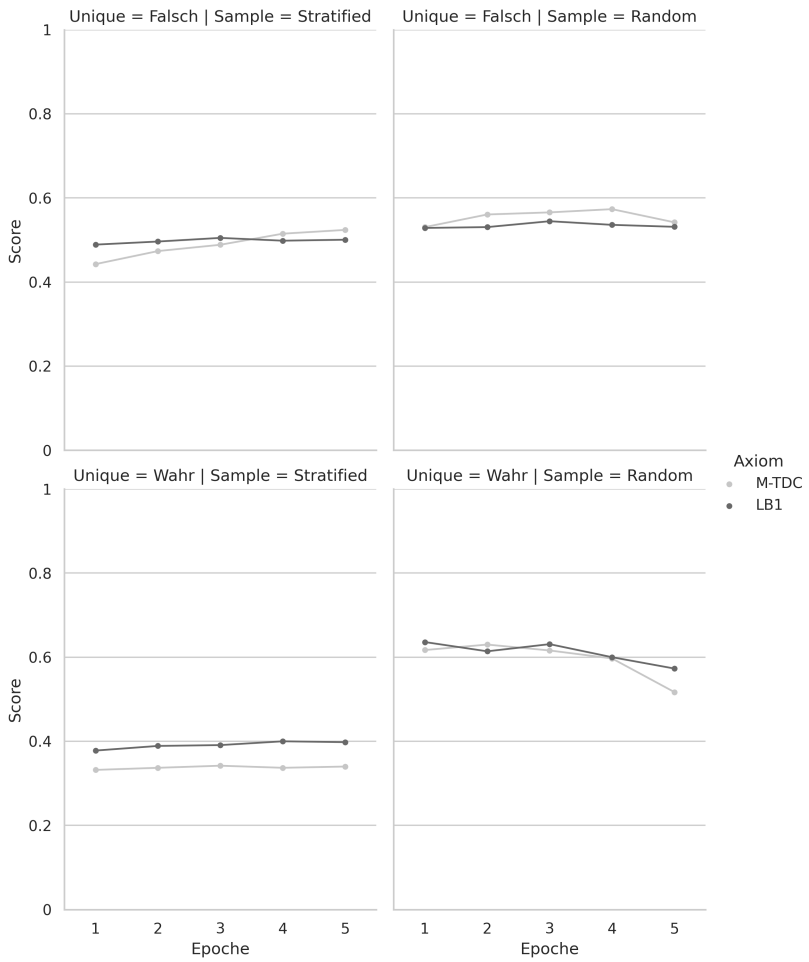

Abbildung 5.7: Accuracy Ergebnisse von M-TDC und LB1 für 25.000 Trainingsdaten im Experiment 2. Es wurde Multi-Task-Learning mit M-TDC und LB1 und einer Ranking Aufgabe mit 100.000 Trainingsdaten durchgeführt.

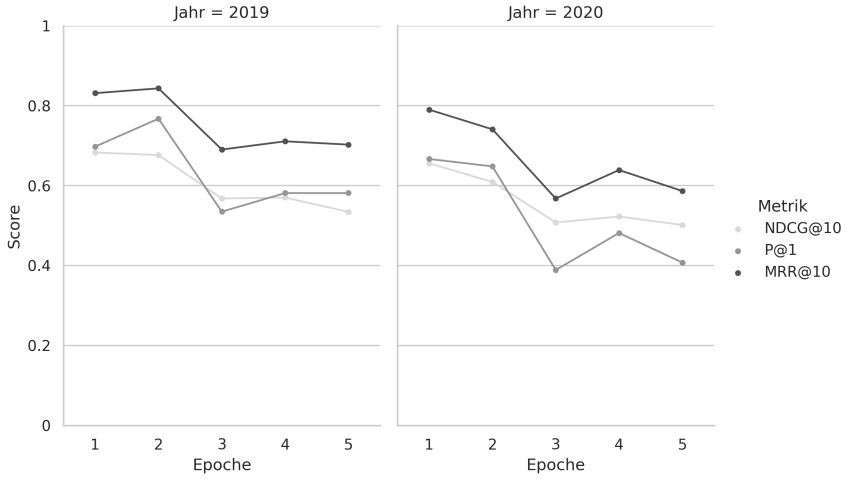

Abbildung 5.8: Ergebnisse der Ranking Aufgabe für 100.000 im Experiment 3. Es wurde Multi-Task-Learning mit REG und ANTI-REG zu jeweils 25.000 Trainingsdaten durchgeführt.

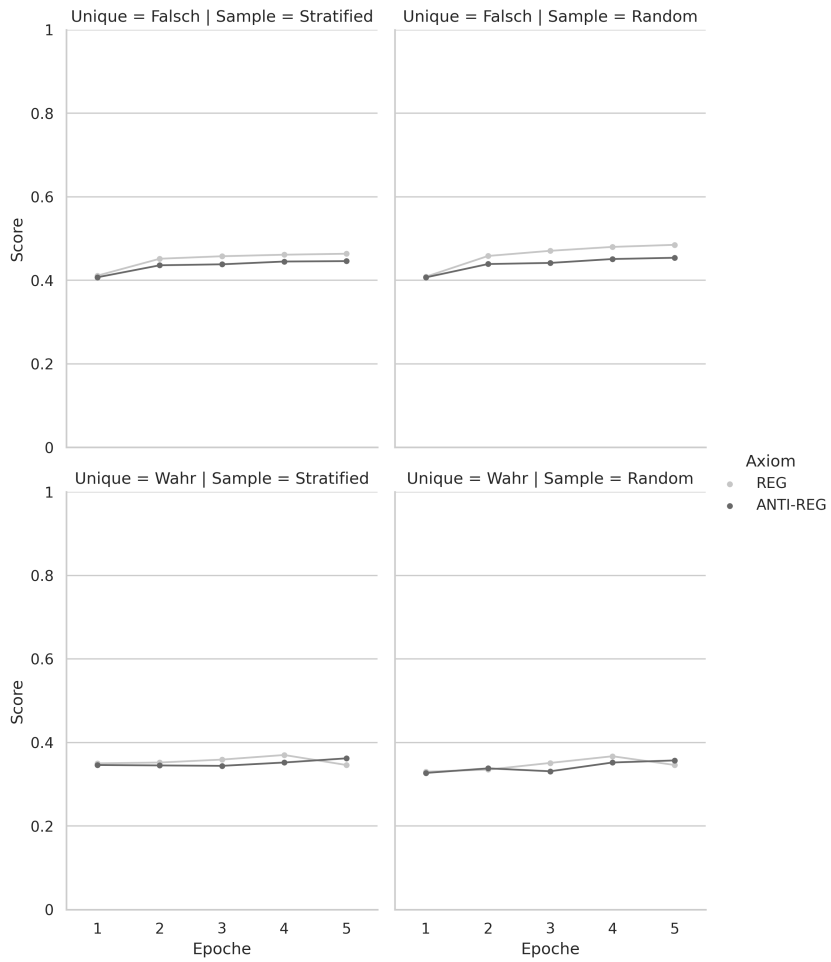

Abbildung 5.9: Accuracy Ergebnisse von REG und ANTI-REG für 25.000 Trainingsdaten im Experiment 3. Es wurde Multi-Task-Learning mit REG und ANTI-REG und einer Ranking Aufgabe mit 100.000 Trainingsdaten durchgeführt.

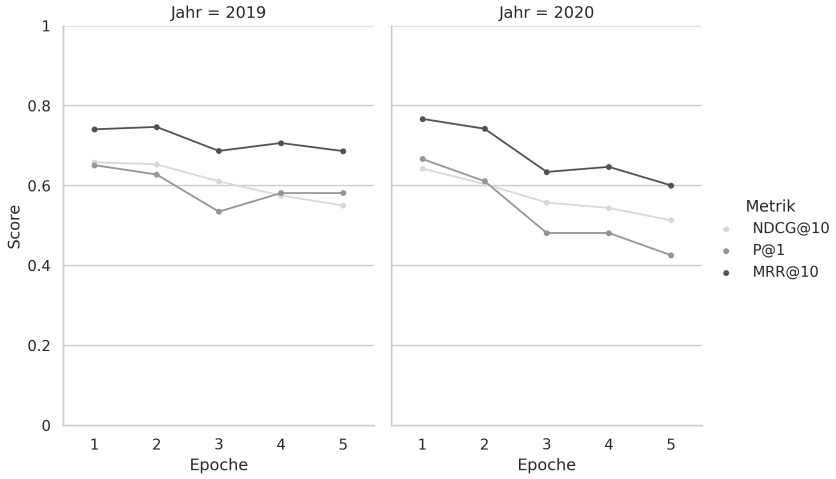

Abbildung 5.10: Ergebnisse der Ranking Aufgabe für 100.000 im Experiment 4. Es wurde Multi-Task-Learning mit PROX4 und PROX5 zu jeweils 25.000 Trainingsdaten durchgeführt.

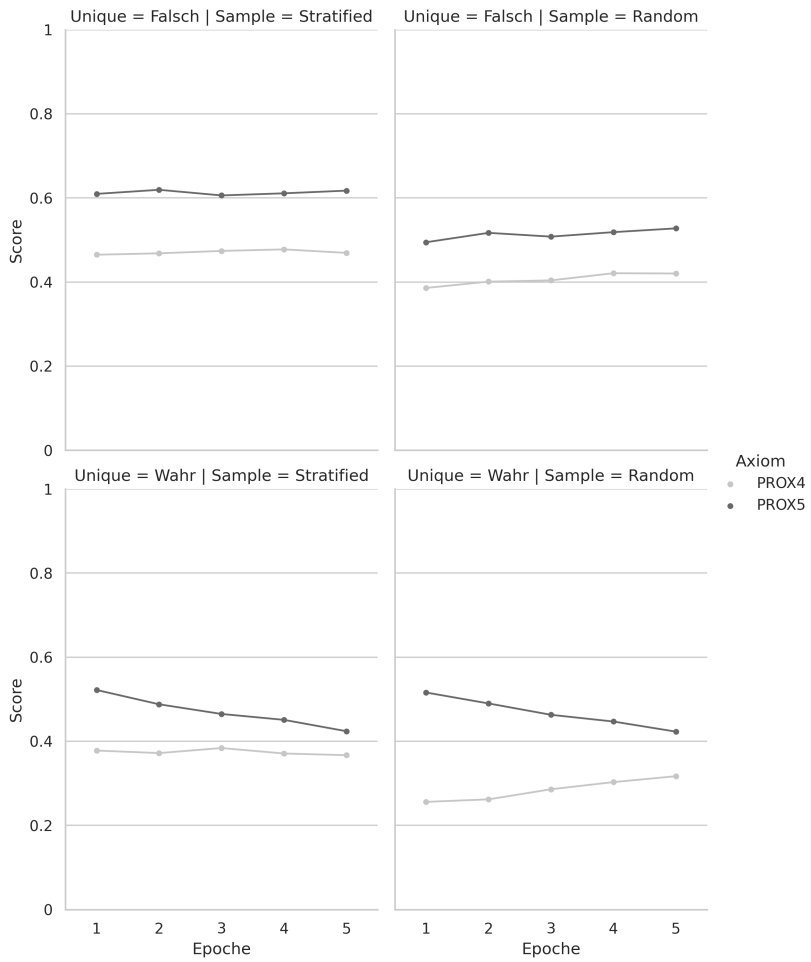

Abbildung 5.11: Accuracy Ergebnisse von PROX4 und PROX5 für 25.000 Trainingsdaten im Experiment 4. Es wurde Multi-Task-Learning mit PROX4 und PROX5 und einer Ranking Aufgabe mit 100.000 Trainingsdaten durchgeführt.

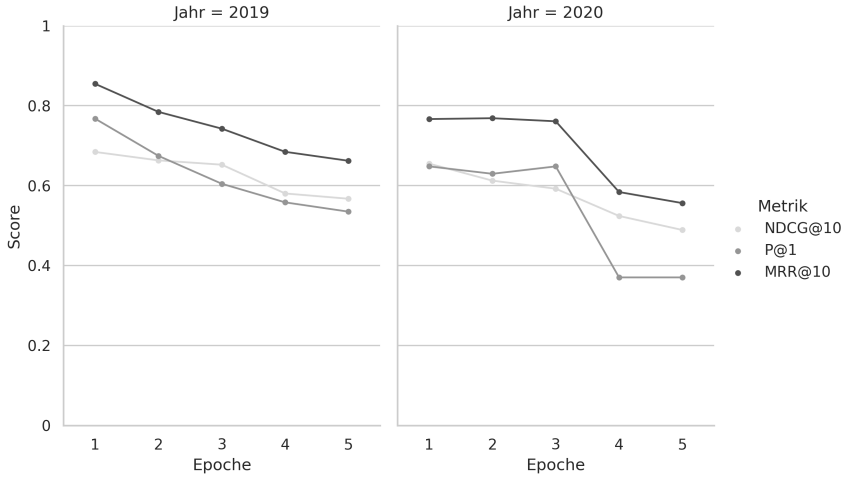

Abbildung 5.12: Ergebnisse der Ranking Aufgabe für 100.000 im Experiment 5. Es wurde Multi-Task-Learning mit M-TDC, LB1, REG, ANTI-REG PROX4 und PROX5 zu jeweils 25.000 Trainingsdaten durchgeführt.

Kapitel 6

Fazit

In unserer Forschungsarbeit wollten wir untersuchen, ob und wie ein optimaler positiver Wissenstransfer zwischen einem Ranker und IR Axiomen im Multi-Task-Learning gewährleistet werden kann. Wir haben dazu einen Learning-to-Rank Ansatz umgesetzt, der das vortrainierte Transfomer Modell BERT zum Multi-Task-Learning von einem Ranker und 20 IR Axiom Aufgaben einsetzt. Unsere Experimente zeigen, dass BERT aus automatisch generierten Trainingsdaten für die IR Axiom als eine Klassifikationsaufgabe die IR Axiome ohne Multi-Task-Learning erlernen kann. Unsere Experimente zeigen außerdem, dass BERT die verschiedenen Axiome unterschiedlich gut erlernen kann. Dabei erlernt BERT die IR Axiom TFC3 und PROX5 mit einer Accuracy von rund 0.5 am besten und die IR Axiome DIV und STMC2 mit einer Accuracy von rund 0.3 am schlechtesten

Unsere Evaluation auf den TREC Deep Learning Tracks aus den Jahren 2019 und 2020 zeigte, dass Multi-Task-Learning mit IR Axiomen die Effektivität von Rankern erhöhen kann. Unsere Evaluation verschiedener Trainingsdaten Verhältnisse und IR Axiom-Kombinationen zeigte, dass eine Erhöhung der Effektivität des Rankers maßgeblich von dem hinzugefügten induktiven Bias in Form der IR Axiome und der Menge an IR Axiom Trainingsdaten abhängt. Für unsere beste Kombination der Aufgaben und Wahl der Trainingsverhältnisse im Multi-Task-Learning konnten wir im Vergleich zum Training des Rankers auf denselben Test-Daten ohne Multi-Task-Learning zeigen, dass sich die Effektivität der Ranking Aufgabe bezogen auf TREC20 im Mittel um 5% bezüglich NDCG@10, P@1 und MRR@10 erhöht und auf TREC 19 vergleichbare Ergebnisse erreicht. Wir konnten außerdem feststellen, dass Multi-Task-Learning im Allgemeinen für den Ranker und die IR Axiom Aufgaben eine generalisierende Wirkung hat und so gegen das auftretende Overfitting der einzelnen Aufgaben wirkt. Diesen Effekt konnten wir durch die Evaluation der Effektivität der einzelnen Aufgaben über mehrere Epochen feststellen. Für das Multi-

Task-Learning des Rankers und der IR Axiome konnten wir keinen Einfluss des Rankers abseits des Minderns des Overfittings auf die Effektivität der IR Axiome feststellen. Außerdem lassen unsere Ergebnisse auf Interferenzen im Training zwischen den IR Axiomen PROX4 und PROX5 bezüglich REG und ANTI-REG schließen, die in zukünftigen Arbeiten noch genauer zu betrachten sind. Unsere Untersuchungen und Ergebnisse zeigen das Potenzial von Multi-Task-Learning von einem Ranker und IR Axiom Aufgaben, wobei unklar ist, in welchem Rahmen sich das Potenzial für große Mengen an Trainingsdaten bewegt.

Für zukünftige Arbeiten rund um unseren Ansatz sehen wir drei mögliche Richtungen. Die erste Richtung ist es die IR Axiome und IR Axiom Aufgaben geeigneter für einen Machine Learning Ansatz zu formulieren. Insbesondere eine Entfernung der Vorbedingung von IR Axiomen, ohne dabei die Aussagekraft über eine Ranking-Präferenz zu verlieren, erwarten wir insofern als eine Verbesserung, dass das im Multi-Task-Learning gelernte Modell dann nicht die Vorbedingungen mit erlernen muss und sich auf das erlernen von Zusammenhänge zwischen dem Ranking und der IR Axiome konzentrieren kann. Ohne Vorbedingungen vermuten wir, dass BERT die einzelnen IR Axiome besser erlernen kann. Als eine zweite Richtung betrachten wir Untersuchunge zur Übertragbarkeit unseres Ansatzes. Es ist wichtig zu untersuchen, ob sich die Ergebnisse unseres Ansatzes auf dem MS MARCO Passagen Datensatz auch auf andere Korpora und das Ranking von längeren Dokumenten übertragen lassen. Interessant ist auch zu betrachten, ob sich unsere Ergebnisse für das Multi-Task-Learning auf mehr Trainingsdaten skalieren lassen oder ob dann bisher noch nicht beobachtete Veränderungen auftreten. Die vielältigsten Möglichkeiten für zukünftige Arbeiten rund um unseren Ansatz sehen wir in der Anpassung der Multi-Task-Learning Architektur. Für Forschungsarbeiten bezüglich der Architektur sind Änderungen an dem Tranformer Modell von BERT zu T5 [64], Untersuchungen zur Task-Affinity [28], mehr Setup-Kombinationen bezüglich Parameterwahl und den Aufgaben, Pairwise und Listwise Ranking Aufgaben und das Einführen weiterer Aufgaben wie Reading Comprehension mögliche spezifischere Themen. In Anbetracht des Erfolges von BERTSUM [47] und duoBERT [54] erwarten wir durch eine Anpassung des Eingabeformates der IR Axiom Aufgaben eine deutlich Verbesserung unseres Multi-Task-Learning Ansatzes für die IR Axiom Aufgaben und die Ranking Aufgabe.

Anhang A

Ergebnisse aller Experimente

A.1 Ranking Baselines

Tabelle A.1: Übersicht der Ergebnisse unserer Baselines und ausgewählter Ansätze aus den TREC Deep Learning Overview-Papern [14, 15] auf den 43 und 54 Anfragen des TREC Deep Learning Tracks aus den Jahren 2019 und 2020. $BM25_{Buop}$ stellt die besten möglichen Ergebnisse dar, die durch ein Reranking des initialen Rankings erreicht werden können. Für $BM25_{uop}$ sind die Parameter $k_1 = 0.9$ und $b = 0.4$ und für $BM25_{op}$ sind die Parameter $k_1 = 0.82$ und $b = 0.68$ nach der Empfehlung des Pyserini-Toolkit für den MS MARCO Passage Datensatz.

	TREC 2019			**TREC 2020**		
Ansatz	NDCG@10	P@1	MRR@10	NDCG@10	P@1	MRR@10
$BM25_{op}$	0.4973	0.5581	0.6821	0.4876	0.5370	0.6554
$BM25_{uop}$	0.5058	0.0000	0.7024	0.0000	0.5370	0.6533
$BM25_{Buop}$	0.9640	0.9767	0.9767	0.9685	1.0000	1.0000
ICT-BERT2	0.6650	–	0.8743	–	–	–
nlm-prfun-bert	–	–	–	0.6648	–	0.8603

A.2 Ergebnisse Single-Task-Setup

A.2.1 Ranking Aufgabe

Tabelle A.2: Übersicht der Ergebnisse unseres Learning-to-Rank Ansatzes für 10.000 Trainingsdaten und kein Multi-Task-Learning auf den 43 und 54 Anfragen des TREC Deep Learning Tracks aus den Jahren 2019 und 2020 für 5 Wiederholungen mit 95% Konfidenzintervallen.

	TREC 2019			**TREC 2020**		
Epoche	NDCG@10	P@1	MRR@10	NDCG@10	P@1	MRR@10
1	0.6269±0.0371	0.6465±0.0827	0.7709±0.0502	0.5707±0.0604	0.6222±0.1223	0.7470±0.0701
2	0.5757±0.0508	0.5581±0.0958	0.7040±0.0807	0.4962±0.0640	0.4556±0.1639	0.6218±0.1051
3	0.5247±0.1052	0.4837±0.1842	0.6414±0.1619	0.4587±0.0708	0.4037±0.0851	0.5795±0.0829
4	0.5303±0.0917	0.5349±0.1803	0.6679±0.1521	0.4567±0.0739	0.3741±0.1033	0.5461±0.1044
5	0.5395±0.0494	0.5163±0.0801	0.6660±0.0643	0.4614±0.0348	0.3852±0.0819	0.5616±0.0643

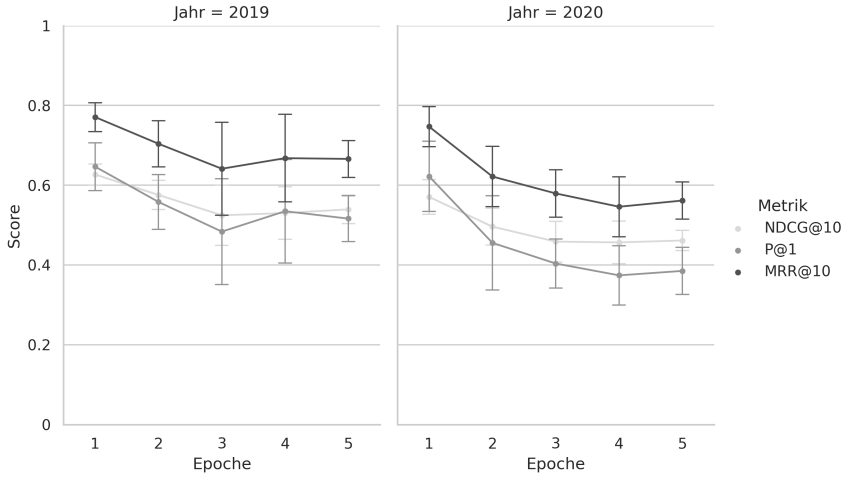

Abbildung A.1: Grafische Übersicht der Ergebnisse unseres Learning-to-Rank Ansatzes für 10.000 Trainingsdaten und kein Multi-Task-Learning auf den 43 und 54 Anfragen des TREC Deep Learning Tracks aus den Jahren 2019 und 2020 für 5 Wiederholungen mit 95% Konfidenzintervallen.

Tabelle A.3: Übersicht der Ergebnisse unseres Learning-to-Rank Ansatzes für 100.000 Trainingsdaten und kein Multi-Task-Learning auf den 43 und 54 Anfragen des TREC Deep Learning Tracks aus den Jahren 2019 und 2020. Die besten Ergebnisse sind fett markiert.

	TREC 2019			**TREC 2020**		
Epoche	NDCG@10	P@1	MRR@10	NDCG@10	P@1	MRR@10
1	**0.6855**	**0.7674**	**0.8547**	0.6087	0.6296	0.7415
2	0.6635	0.7442	0.8289	**0.6138**	**0.6481**	**0.7494**
3	0.5394	0.5116	0.6545	0.4753	0.3519	0.5340
4	0.4997	0.4186	0.5859	0.4459	0.3333	0.5161
5	0.4986	0.3953	0.5661	0.4399	0.2778	0.4807

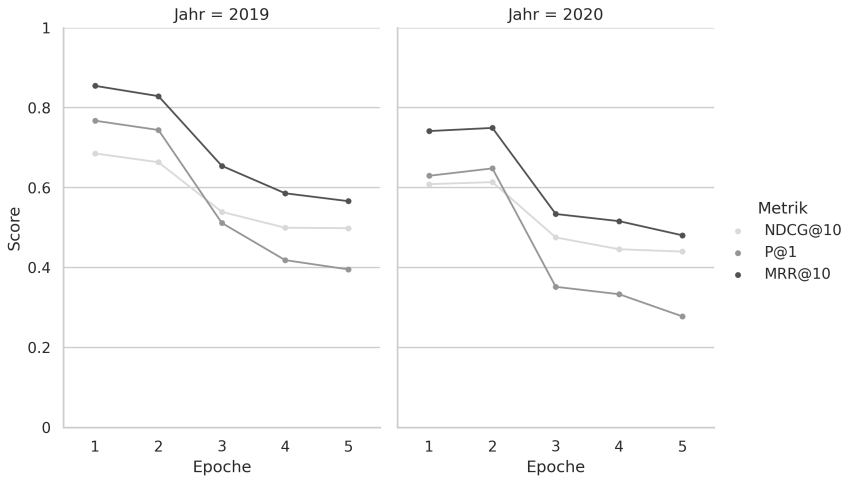

Abbildung A.2: Grafische Übersicht der Ergebnisse unseres Learning-to-Rank Ansatzes für 100.000 Trainingsdaten und kein Multi-Task-Learning auf den 43 und 54 Anfragen des TREC Deep Learning Tracks aus den Jahren 2019 und 2020.

Tabelle A.4: Übersicht der Ergebnisse unseres Learning-to-Rank Ansatzes für 1.000.000 Trainingsdaten und kein Multi-Task-Learning auf den 43 und 54 Anfragen des TREC Deep Learning Tracks aus den Jahren 2019 und 2020. Die besten Ergebnisse sind fett markiert.

	TREC 2019			**TREC 2020**		
Epoche	NDCG@10	P@1	MRR@10	NDCG@10	P@1	MRR@10
1	0.6871	0.7209	0.7990	**0.6747**	**0.6852**	0.7879
2	0.6902	0.7209	0.8076	0.6606	0.6667	**0.7968**
3	**0.6963**	0.7209	0.8283	0.6666	0.6481	0.7908
4	0.6807	**0.7442**	**0.8341**	0.6622	0.6852	0.7909
5	0.6442	0.6744	0.7814	0.6424	0.5926	0.7341

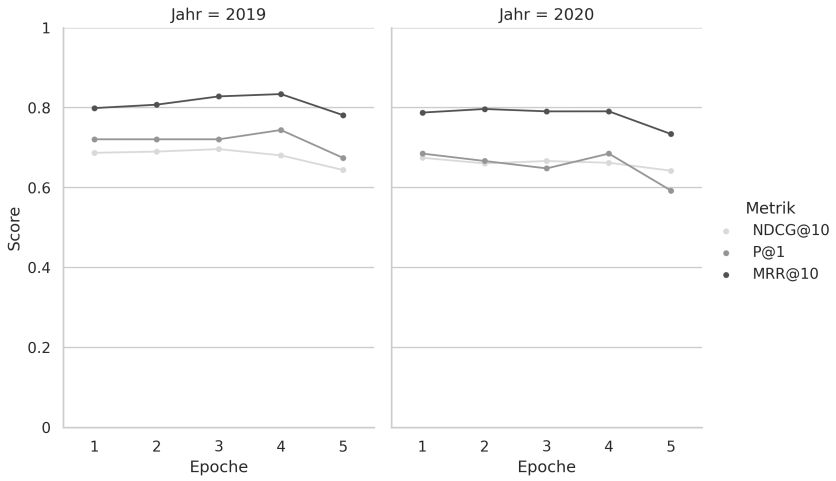

Abbildung A.3: Grafische Übersicht der Ergebnisse unseres Learning-to-Rank Ansatzes für 1.000.000 Trainingsdaten und kein Multi-Task-Learning auf den 43 und 54 Anfragen des TREC Deep Learning Tracks aus den Jahren 2019 und 2020.

A.2.2 Axiomen Aufgaben Eingabe Variante A

Nachfolgend eine tabellarische und grafische Darstellung der Accuracy Ergebnisse jedes Axiomes für 10.000 Trainingsdaten ohne Multi-Task-Learning bezogen auf die Accuracy auf den Test-Datensätzen für 5 Wiederholungen mit 95% Konfidenzintervallen. Die Unique Test-Datensätze bestehen aus jeweils 1.000 Testdaten, deren Anfragen nicht im Training gesehen wurden und die None-Unique Test-Datensätze bestehen aus 3.000 Testdaten. Wir gruppieren die Ergebnisse nach den Axiom-Gruppen. Die Axiom Aufgaben nutzen die Eingabe Variante A $[CLS]Anfrage + Dokument_1[SEP]Anfrage + Dokument_2[SEP]$.

Tabelle A.5: Termhäufigkeits-Axiome

		Stratified		**Random**	
Axiom	**Epoche**	None-Unique	Unique	None-Unique	Unique
TFC1	1	0.4107±0.0282	0.3527±0.0103	0.4284±0.0246	0.3530±0.0174
	2	0.4789±0.0756	0.3600±0.0273	0.4897±0.0665	0.3640±0.0197
	3	0.5572±0.0451	0.3577±0.0076	0.5551±0.0322	0.3710±0.0194
	4	0.6032±0.0292	0.3507±0.0229	0.5982±0.0273	0.3630±0.0086
	5	0.6211±0.0166	0.3510±0.0199	0.6130±0.0158	0.3620±0.0090
TFC3	1	0.5741±0.0697	0.4530±0.0163	0.6709±0.3018	0.7130±0.4018
	2	0.6440±0.0231	0.4460±0.0348	0.7788±0.1296	0.8777±0.0697
	3	0.6856±0.0338	0.4403±0.0286	0.8268±0.1375	0.8700±0.1150
	4	0.7351±0.0543	0.4383±0.0466	0.8722±0.1240	0.8823±0.1006
	5	0.7540±0.0731	0.4387±0.0402	0.8829±0.1040	0.8890±0.0252
M-TDC	1	0.4101±0.1011	0.3403±0.0589	0.5159±0.0335	0.4330±0.1496
	2	0.4683±0.2097	0.3517±0.0100	0.4826±0.3282	0.3473±0.1137
	3	0.5397±0.2718	0.3520±0.0410	0.5834±0.3040	0.3907±0.1385
	4	0.6017±0.2809	0.3460±0.0245	0.6468±0.3498	0.3813±0.1047
	5	0.6037±0.2585	0.3357±0.0158	0.6660±0.2512	0.3657±0.0460
LB1	1	0.4861±0.0557	0.3800±0.0323	0.5207±0.0890	0.5770±0.1843
	2	0.5534±0.0308	0.4023±0.0747	0.5776±0.0378	0.4837±0.0442
	3	0.6104±0.0297	0.3907±0.0261	0.6416±0.0464	0.4900±0.0212
	4	0.6434±0.0139	0.3817±0.0162	0.6781±0.0296	0.4883±0.1449
	5	0.6552±0.0285	0.3863±0.0362	0.6944±0.0083	0.4867±0.1502

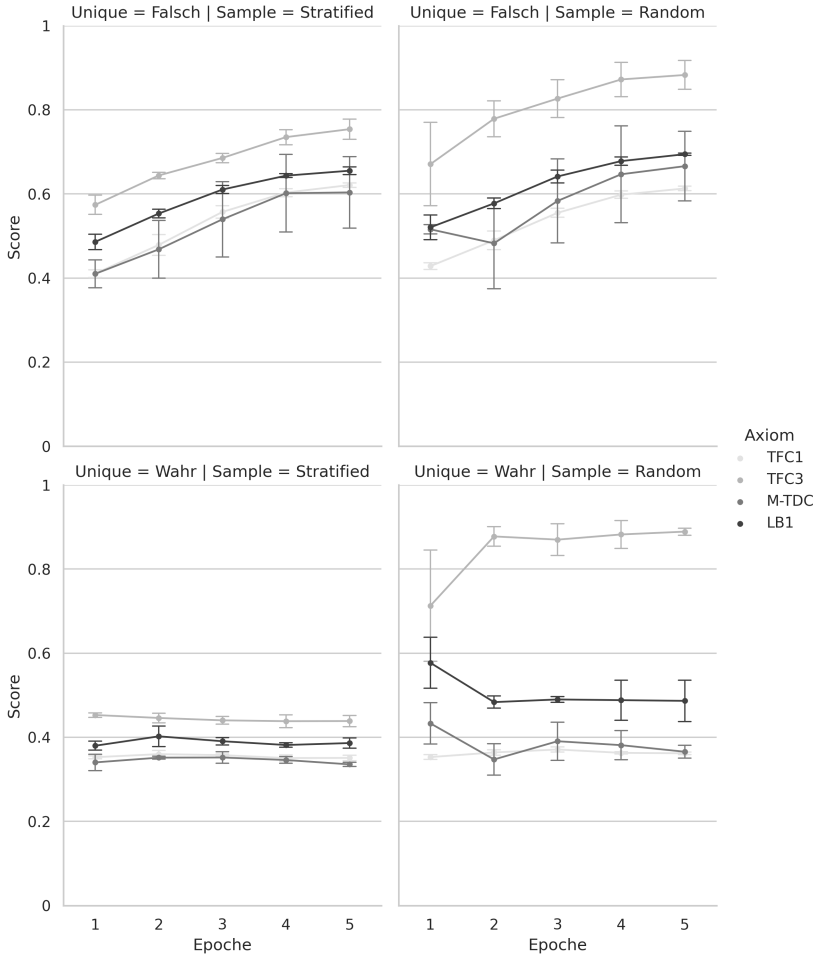

Abbildung A.4: Termhäufigkeits-Axiome

Tabelle A.6: Längen-Normierungs-Axiome

		Stratified		Random	
Axiom	**Epoche**	None-Unique	Unique	None-Unique	Unique
LNC1	1	0.4508±0.0230	0.3427±0.0217	0.4236±0.0842	0.4227±0.0987
	2	0.5223±0.0364	0.3513±0.0038	0.4786±0.0077	0.3917±0.1071
	3	0.5947±0.0621	0.3513±0.0338	0.5328±0.1064	0.3777±0.1277
	4	0.6559±0.0706	0.3560±0.0025	0.5908±0.0953	0.3790±0.0748
	5	0.6866±0.0631	0.3520±0.0388	0.6248±0.0622	0.3807±0.0569
TF-LNC	1	0.3749±0.0506	0.3473±0.0123	0.2642±0.4274	0.3907±0.8087
	2	0.4551±0.1089	0.3357±0.0094	0.3786±0.3230	0.3563±0.5763
	3	0.5529±0.0510	0.3377±0.0254	0.5624±0.2073	0.3877±0.4574
	4	0.5996±0.0332	0.3383±0.0100	0.6239±0.0974	0.3250±0.2361
	5	0.6220±0.0554	0.3380±0.0151	0.6522±0.0571	0.2877±0.0572

Tabelle A.7: Semantische-Ähnlichkeits-Axiome

		Stratified		Random	
Axiom	**Epoche**	None-Unique	Unique	None-Unique	Unique
STMC1	1	0.6617±0.0133	0.4200±0.1594	0.5401±0.0127	0.4630±0.0397
	2	0.6729±0.0368	0.4397±0.0423	0.5529±0.0482	0.4787±0.0183
	3	0.7018±0.1054	0.4530±0.0066	0.5917±0.1410	0.4777±0.0225
	4	0.7430±0.1197	0.4403±0.0460	0.6494±0.1529	0.4753±0.0288
	5	0.7840±0.0578	0.4490±0.0407	0.7084±0.0829	0.4800±0.0336
STMC2	1	0.3492±0.0343	0.3393±0.0158	0.0511±0.1418	0.0357±0.0933
	2	0.4498±0.1487	0.3317±0.0087	0.3787±0.4265	0.1730±0.1850
	3	0.5314±0.1504	0.3253±0.0423	0.4827±0.4463	0.1827±0.2433
	4	0.5833±0.1445	0.3217±0.0345	0.5753±0.3226	0.2370±0.2145
	5	0.6256±0.1443	0.3190±0.0286	0.6962±0.1611	0.3137±0.1402

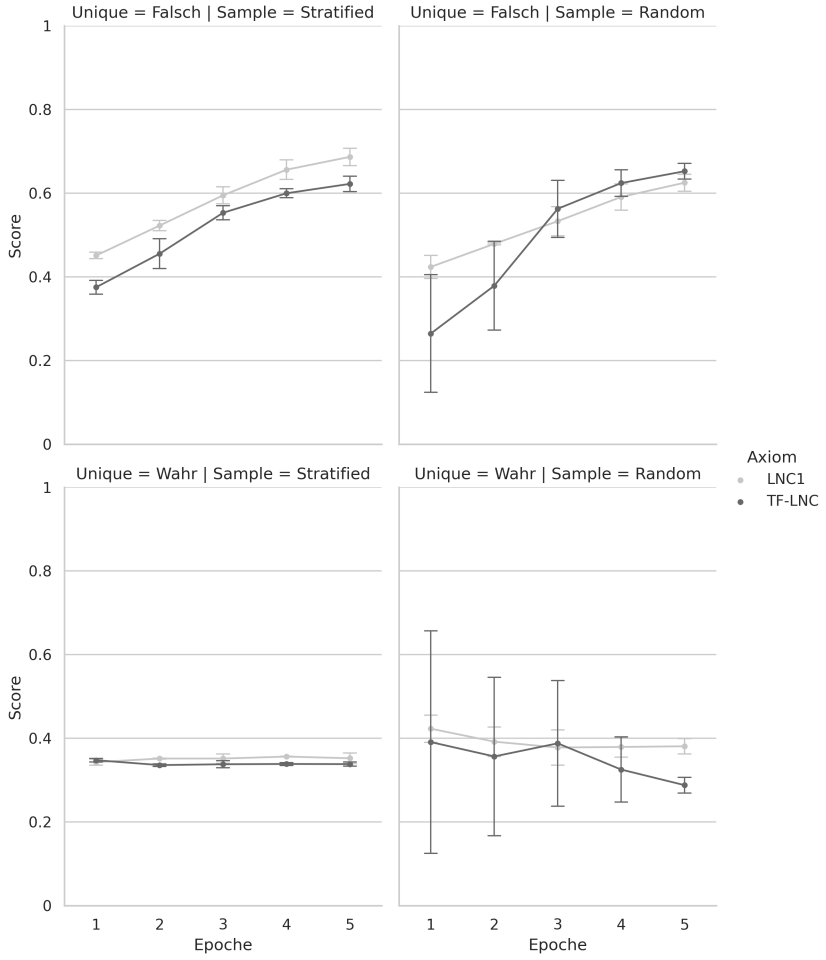

Abbildung A.5: Längen-Normierungs-Axiome

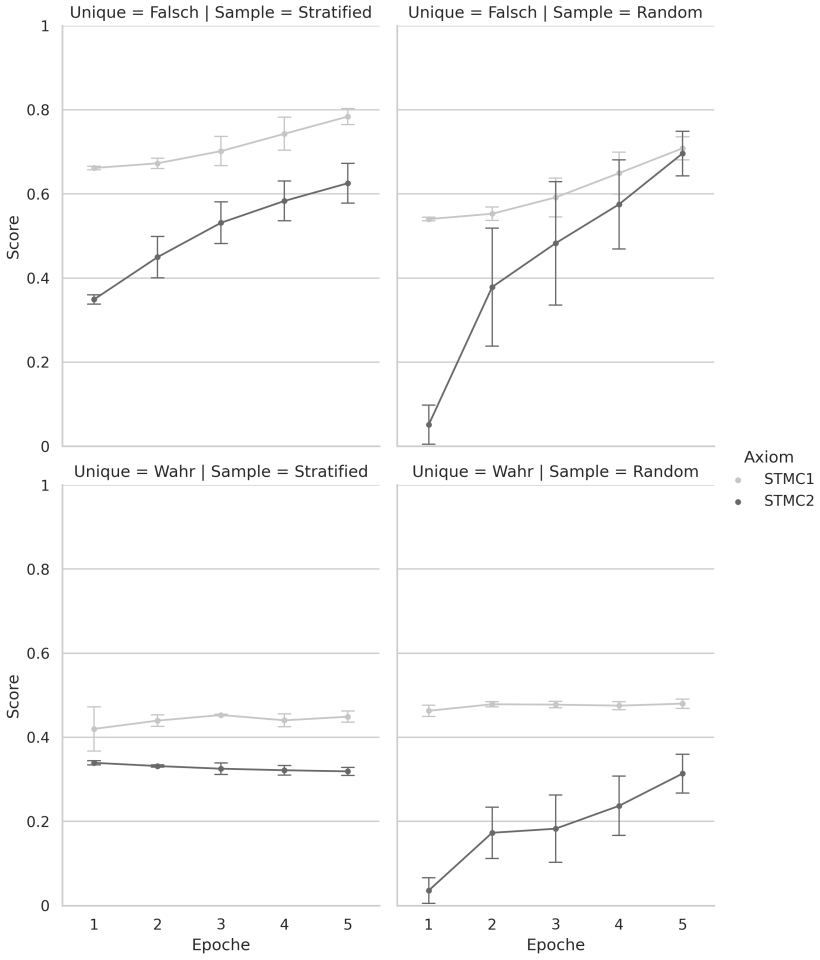

Abbildung A.6: Semantische-Ähnlichkeits-Axiome

Tabelle A.8: Anfrage-Facetten-Axiome

		Stratified		**Random**	
Axiom	**Epoche**	None-Unique	Unique	None-Unique	Unique
REG	1	0.4051±0.0792	0.3357±0.0103	0.4426±0.1636	0.3247±0.0393
	2	0.4793±0.1259	0.3480±0.0258	0.5227±0.1562	0.3457±0.0658
	3	0.5461±0.1030	0.3493±0.0186	0.5847±0.1339	0.3470±0.0449
	4	0.5990±0.0990	0.3580±0.0221	0.6413±0.1084	0.3553±0.0380
	5	0.6364±0.0614	0.3593±0.0259	0.6741±0.0856	0.3550±0.0366
ANTI-REG	1	0.3576±0.0619	0.3343±0.0374	0.3486±0.0830	0.3063±0.0696
	2	0.4610±0.1462	0.3553±0.0396	0.4680±0.1757	0.3413±0.1027
	3	0.5336±0.1646	0.3470±0.0221	0.5416±0.1734	0.3373±0.0500
	4	0.5859±0.1304	0.3453±0.0193	0.5900±0.1358	0.3267±0.0225
	5	0.6153±0.1415	0.3483±0.0217	0.6214±0.1411	0.3377±0.0211
DIV	1	0.3399±0.0077	0.3347±0.0100	0.3296±0.5980	0.3283±0.6168
	2	0.4073±0.1068	0.3370±0.0485	0.3718±0.1603	0.2577±0.3260
	3	0.5083±0.1764	0.3317±0.0646	0.4759±0.1186	0.2810±0.1195
	4	0.5871±0.1709	0.3183±0.0315	0.5584±0.0973	0.2633±0.0291
	5	0.6196±0.1289	0.3230±0.0366	0.6069±0.1853	0.2653±0.1810

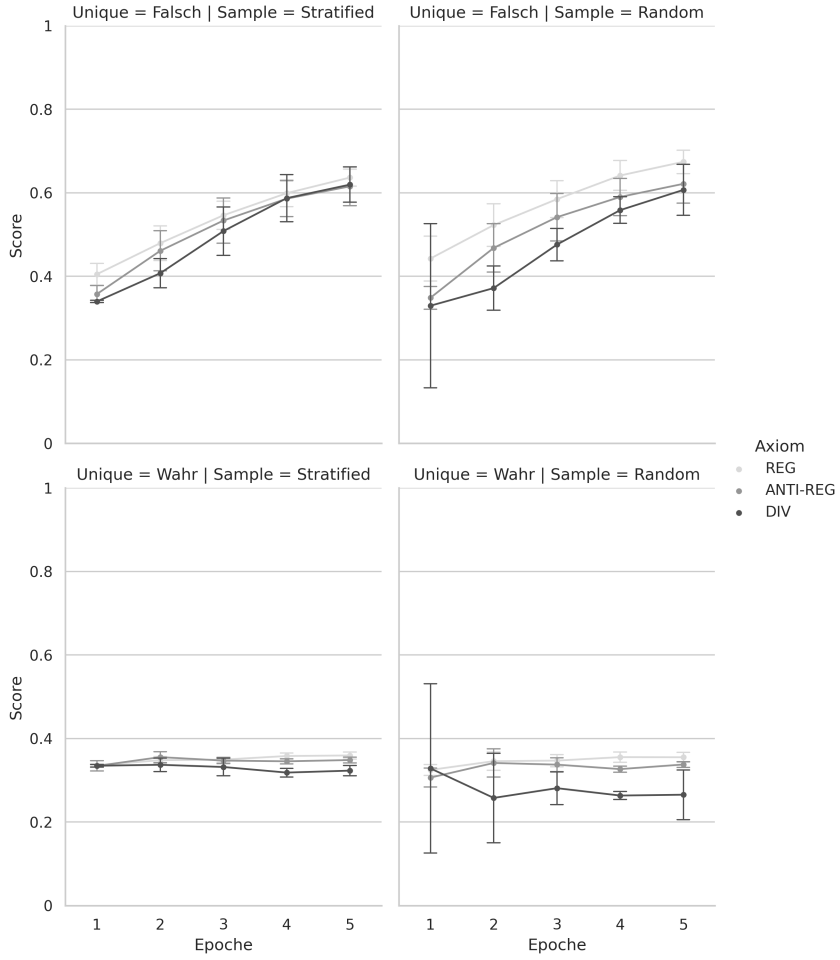

Abbildung A.7: Anfrage-Facetten-Axiome

Tabelle A.9: Term-Nachbarschafts-Axiome

		Stratified		**Random**	
Axiom	**Epoche**	None-Unique	Unique	None-Unique	Unique
PROX1	1	0.5049±0.0371	0.3993±0.0518	0.4836±0.0292	0.3760±0.0452
	2	0.5688±0.0208	0.3763±0.0648	0.5284±0.0205	0.3397±0.0535
	3	0.6171±0.0251	0.3757±0.0362	0.5640±0.0369	0.3390±0.0519
	4	0.6524±0.0325	0.3770±0.0497	0.5997±0.0296	0.3523±0.0682
	5	0.6691±0.0592	0.3800±0.0456	0.6096±0.0302	0.3527±0.0535
PROX2	1	0.3933±0.0261	0.3427±0.0439	0.4692±0.0154	0.4487±0.0856
	2	0.4834±0.0433	0.3553±0.0434	0.4956±0.0672	0.4017±0.0601
	3	0.5679±0.0546	0.3623±0.0470	0.5422±0.0485	0.3970±0.0489
	4	0.6297±0.0774	0.3550±0.0499	0.5883±0.0574	0.3843±0.0174
	5	0.6694±0.0557	0.3650±0.0409	0.6059±0.0116	0.3517±0.0311
PROX3	1	0.6990±0.0605	0.3297±0.0063	0.9228±0.0180	0.9200±0.0410
	2	0.7848±0.0404	0.3307±0.0255	0.9453±0.0254	0.8970±0.0453
	3	0.8400±0.0495	0.3323±0.0241	0.9633±0.0093	0.9100±0.0203
	4	0.8712±0.0528	0.3287±0.0112	0.9716±0.0038	0.9230±0.0277
	5	0.8901±0.0480	0.3307±0.0038	0.9741±0.0027	0.9213±0.0320
PROX4	1	0.4586±0.0355	0.3703±0.0146	0.4140±0.0675	0.2593±0.0551
	2	0.5223±0.0447	0.3690±0.0203	0.4593±0.0373	0.2930±0.0522
	3	0.5834±0.0440	0.3653±0.0352	0.5126±0.0557	0.3080±0.0772
	4	0.6228±0.0383	0.3660±0.0258	0.5498±0.0578	0.3250±0.0797
	5	0.6503±0.0166	0.3593±0.0251	0.5880±0.0323	0.3173±0.0160
PROX5	1	0.6101±0.0254	0.4740±0.0630	0.5321±0.0133	0.4623±0.0550
	2	0.6450±0.0258	0.4750±0.0652	0.5586±0.0307	0.4623±0.0561
	3	0.6747±0.0417	0.4487±0.0590	0.5874±0.0476	0.4417±0.0330
	4	0.6937±0.0353	0.4410±0.0431	0.6098±0.0538	0.4393±0.0382
	5	0.7086±0.0446	0.4513±0.0296	0.6193±0.0524	0.4443±0.0146

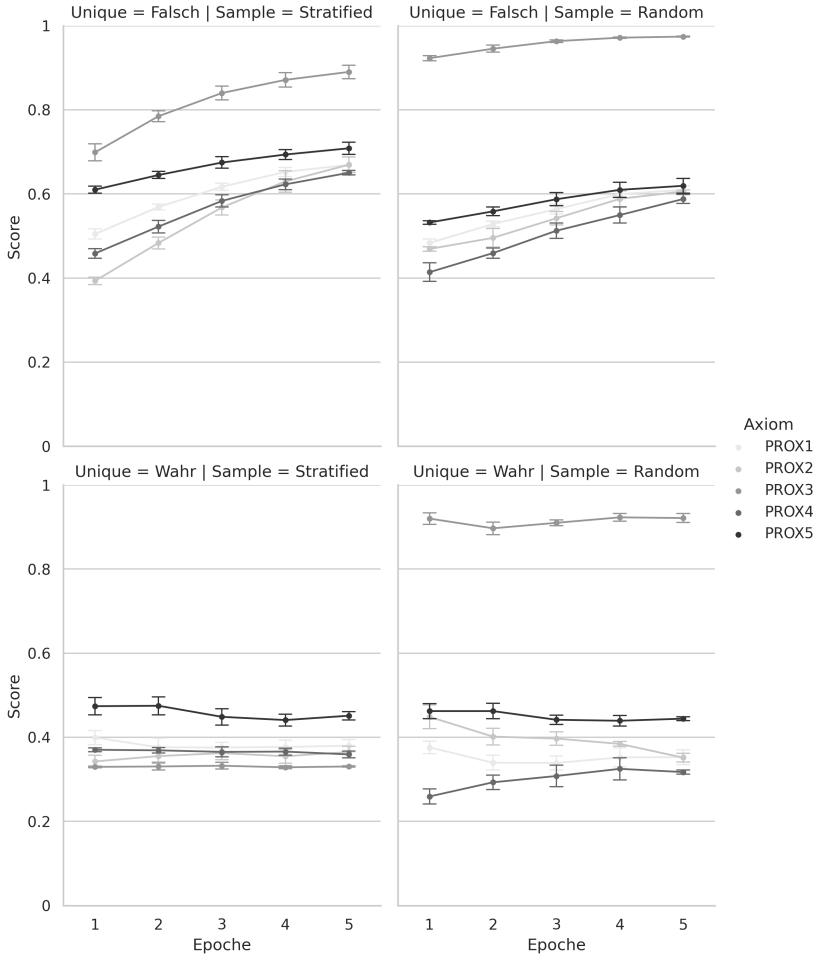

Abbildung A.8: Term-Nachbarschafts-Axiome

Tabelle A.10: Retrieval-Score-Axiome

		Stratified		Random	
Axiom	**Epoche**	None-Unique	Unique	None-Unique	Unique
RS-TF	1	0.4410±0.0116	0.3623±0.0426	0.4964±0.0229	0.4860±0.0541
	2	0.5036±0.0442	0.3553±0.0183	0.5219±0.0272	0.4160±0.2005
	3	0.5728±0.0164	0.3697±0.0103	0.5623±0.0182	0.4160±0.1020
	4	0.6226±0.0367	0.3457±0.0190	0.5946±0.0137	0.3807±0.0856
	5	0.6372±0.0383	0.3457±0.0331	0.6040±0.0186	0.3763±0.0396
RS-TF-IDF	1	0.5481±0.0261	0.3770±0.0114	0.4961±0.0376	0.4250±0.0805
	2	0.5930±0.0398	0.3780±0.0138	0.5123±0.0617	0.4117±0.1026
	3	0.6490±0.0337	0.3863±0.0304	0.5617±0.0542	0.4147±0.0870
	4	0.6934±0.0309	0.3830±0.0263	0.6044±0.0739	0.4037±0.0648
	5	0.7069±0.0349	0.3947±0.0100	0.6244±0.0700	0.4173±0.0324
RS-BM25	1	0.5080±0.0171	0.3607±0.0231	0.4651±0.0814	0.4377±0.1697
	2	0.5613±0.0155	0.3667±0.0296	0.4967±0.0291	0.4283±0.0982
	3	0.6110±0.0296	0.3707±0.0503	0.5470±0.0266	0.4380±0.0545
	4	0.6328±0.0146	0.3573±0.0183	0.5598±0.0232	0.4330±0.0437
	5	0.6512±0.0286	0.3727±0.0143	0.5662±0.0176	0.4157±0.0188
RS-QL	1	0.4957±0.0217	0.3520±0.0240	0.4991±0.0243	0.4510±0.0460
	2	0.5512±0.0583	0.3557±0.0169	0.5224±0.0683	0.4360±0.0697
	3	0.6076±0.0587	0.3530±0.0277	0.5673±0.0835	0.4253±0.0362
	4	0.6579±0.0355	0.3527±0.0419	0.6033±0.0423	0.4077±0.0810
	5	0.6809±0.0655	0.3507±0.0357	0.6041±0.0830	0.3803±0.0423

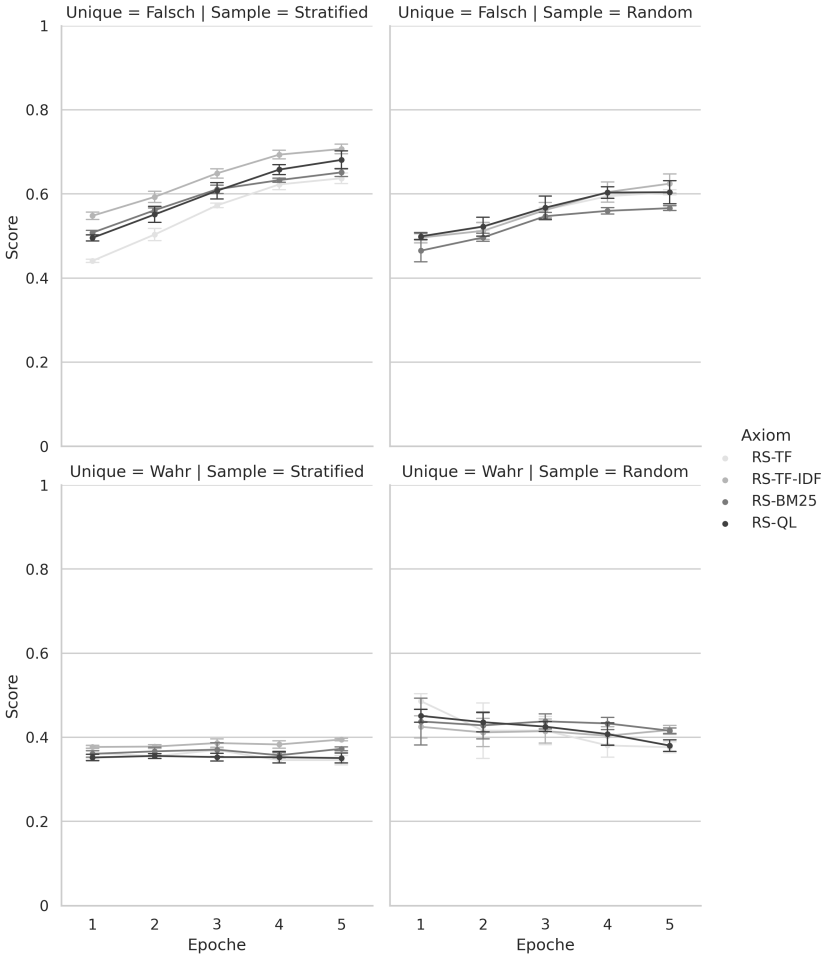

Abbildung A.9: Retrieval-Score-Axiome

A.2.3 Axiomen Aufgaben Eingabe Variante B

Nachfolgend eine tabellarische und grafische Darstellung der Accuracy Ergebnisse jedes Axiomes für 10.000 Trainingsdaten ohne Multi-Task-Learning bezogen auf die Accuracy auf den Test-Datensätzen für 5 Wiederholungen mit 95% Konfidenzintervallen. Die Unique Test-Datensätze bestehen aus jeweils 1.000 Testdaten, deren Anfragen nicht im Training gesehen wurden und die None-Unique Test-Datensätze bestehen aus 3.000 Testdaten. Wir gruppieren die Ergebnisse nach den Axiom-Gruppen. Die Axiom Aufgaben nutzen die Eingabe Variante B $[CLS]Anfrage[SEP]Dokument_1 + Dokument_2[SEP]$.

Tabelle A.11: Termhäufigkeits-Axiome

		Stratified		**Random**	
Axiom	**Epoche**	None-Unique	Unique	None-Unique	Unique
TFC1	1	0.4161±0.0299	0.3450±0.0075	0.4214±0.0320	0.3537±0.0362
	2	0.4511±0.0507	0.3463±0.0334	0.4549±0.0284	0.3473±0.0458
	3	0.5060±0.0439	0.3523±0.0277	0.5056±0.0278	0.3500±0.0476
	4	0.5593±0.0252	0.3463±0.0273	0.5603±0.0167	0.3503±0.0357
	5	0.5916±0.0273	0.3470±0.0472	0.5854±0.0358	0.3537±0.0466
TFC3	1	0.5896±0.0064	0.4673±0.0273	0.7464±0.0674	0.8577±0.1673
	2	0.6301±0.0145	0.4563±0.0180	0.7568±0.0426	0.8663±0.0235
	3	0.6734±0.0143	0.4513±0.0211	0.7922±0.0674	0.8537±0.0811
	4	0.7057±0.0231	0.4527±0.0127	0.8266±0.0509	0.8517±0.1657
	5	0.7177±0.0331	0.4563±0.0229	0.8297±0.0659	0.8550±0.1119
M-TDC	1	0.4341±0.0761	0.3547±0.0465	0.5618±0.0931	0.5287±0.0971
	2	0.5227±0.0483	0.3550±0.0415	0.6100±0.0575	0.4800±0.2261
	3	0.5896±0.0279	0.3580±0.0310	0.6933±0.0635	0.4857±0.1648
	4	0.6453±0.0160	0.3583±0.0315	0.7374±0.0535	0.4547±0.1178
	5	0.6681±0.0273	0.3623±0.0425	0.7706±0.0146	0.4630±0.0864
LB1	1	0.4654±0.0382	0.3743±0.0662	0.4907±0.0197	0.5170±0.1537
	2	0.5399±0.0264	0.3870±0.0366	0.5674±0.0290	0.5087±0.2138
	3	0.5857±0.0538	0.3860±0.0197	0.6159±0.0759	0.4953±0.1983
	4	0.6249±0.0563	0.3863±0.0447	0.6510±0.0509	0.4850±0.0594
	5	0.6382±0.0525	0.3813±0.0580	0.6709±0.0428	0.4990±0.0474

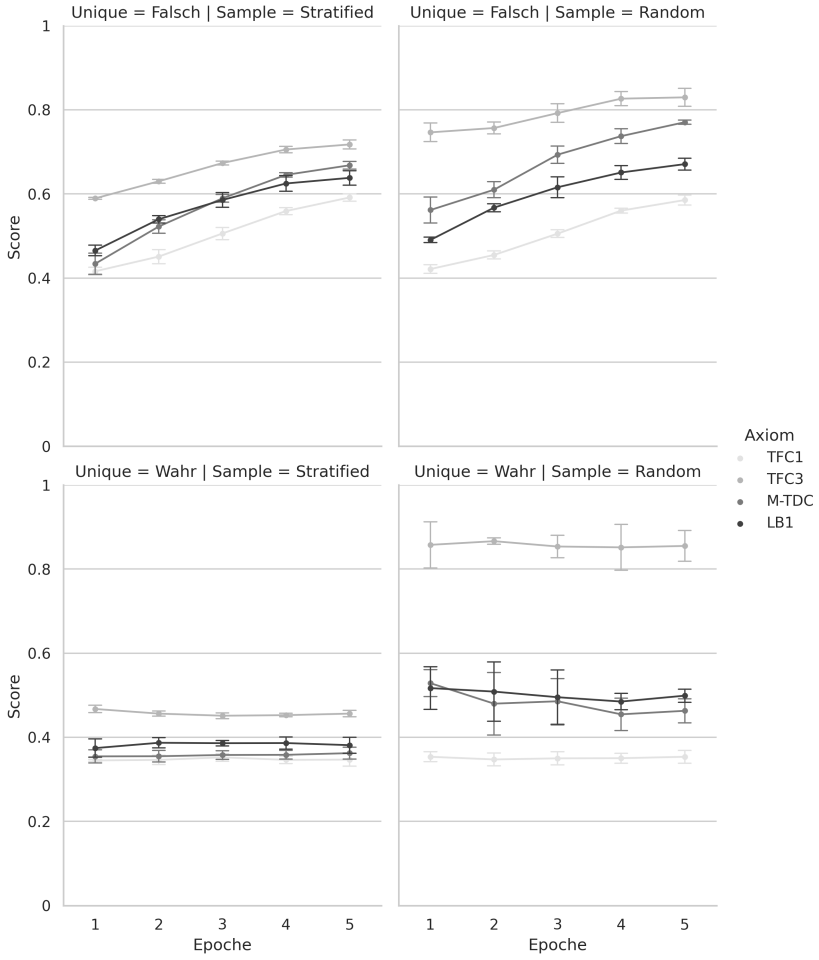

Abbildung A.10: Termhäufigkeits-Axiome

Tabelle A.12: Längen-Normierungs-Axiome

		Stratified		Random	
Axiom	**Epoche**	None-Unique	Unique	None-Unique	Unique
LNC1	1	0.4353±0.0241	0.3487±0.0338	0.3947±0.1970	0.4107±0.1875
	2	0.5186±0.0381	0.3597±0.0423	0.5063±0.0526	0.4193±0.1103
	3	0.5770±0.0909	0.3530±0.0415	0.5469±0.1590	0.3807±0.1473
	4	0.6384±0.1183	0.3467±0.0152	0.6324±0.2013	0.4007±0.1219
	5	0.6646±0.1333	0.3530±0.0172	0.6507±0.2011	0.3983±0.0574
TF-LNC	1	0.3624±0.0253	0.3430±0.0437	0.4578±0.8610	0.6070±0.7859
	2	0.4229±0.0635	0.3440±0.0352	0.5772±0.4591	0.7140±0.3910
	3	0.4798±0.1110	0.3483±0.0479	0.5857±0.3896	0.6903±0.0254
	4	0.5376±0.1013	0.3410±0.0108	0.6186±0.3876	0.5800±0.1368
	5	0.5603±0.0791	0.3507±0.0500	0.6431±0.2177	0.5190±0.1643

Tabelle A.13: Semantische-Ähnlichkeits-Axiome

		Stratified		Random	
Axiom	**Epoche**	None-Unique	Unique	None-Unique	Unique
STMC1	1	0.6620±0.0251	0.3587±0.0625	0.5428±0.0191	0.4460±0.0179
	2	0.6807±0.0075	0.3520±0.0643	0.5613±0.0176	0.4453±0.0287
	3	0.7386±0.0396	0.3707±0.0882	0.6343±0.0440	0.4460±0.0373
	4	0.7584±0.0422	0.3613±0.0487	0.6642±0.0463	0.4380±0.0348
	5	0.7797±0.0711	0.3707±0.0241	0.6941±0.0909	0.4417±0.0254
STMC2	1	0.3602±0.0505	0.3363±0.0174	0.1771±0.2645	0.1570±0.2656
	2	0.4693±0.0518	0.3437±0.0274	0.5508±0.3670	0.3800±0.5167
	3	0.5436±0.0221	0.3383±0.0225	0.6353±0.1686	0.4723±0.1616
	4	0.6006±0.0279	0.3343±0.0349	0.6781±0.1282	0.4763±0.1494
	5	0.6232±0.0204	0.3407±0.0357	0.7022±0.1966	0.4863±0.2448

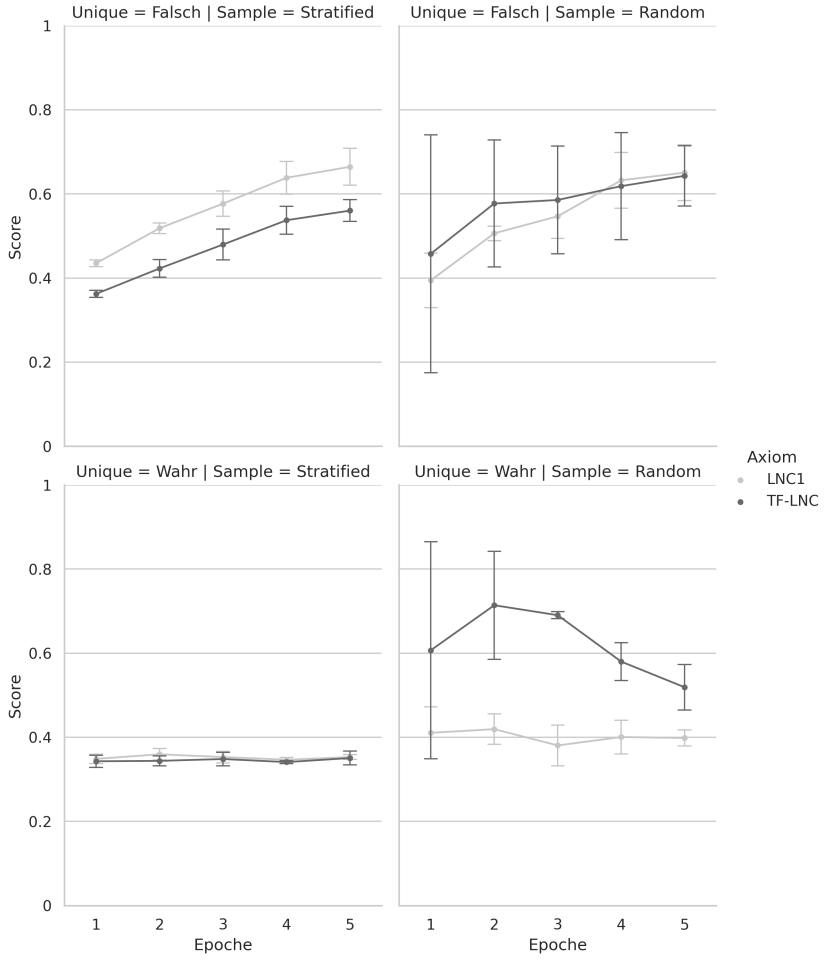

Abbildung A.11: Längen-Normierungs-Axiome

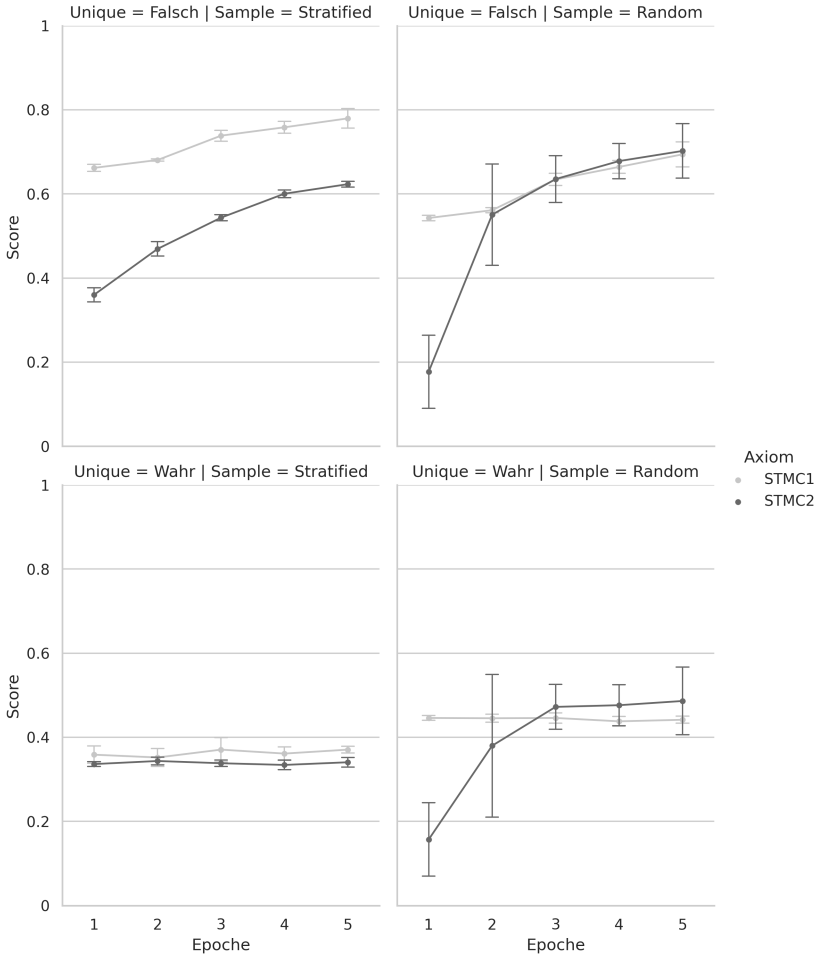

Abbildung A.12: Semantische-Ähnlichkeits-Axiome

Tabelle A.14: Anfrage-Facetten-Axiome

		Stratified		**Random**	
Axiom	**Epoche**	None-Unique	Unique	None-Unique	Unique
REG	1	0.4353±0.0278	0.3317±0.0165	0.4646±0.0357	0.3123±0.0180
	2	0.5183±0.0313	0.3380±0.0174	0.5412±0.0199	0.3250±0.0108
	3	0.5578±0.0328	0.3403±0.0094	0.5854±0.0238	0.3293±0.0038
	4	0.6006±0.0131	0.3417±0.0274	0.6241±0.0322	0.3310±0.0323
	5	0.6097±0.0393	0.3370±0.0149	0.6352±0.0431	0.3290±0.0149
ANTI-REG	1	0.3854±0.0655	0.3177±0.0355	0.3860±0.0060	0.3130±0.1484
	2	0.4580±0.0739	0.3307±0.0231	0.4671±0.0620	0.3263±0.0511
	3	0.5184±0.1038	0.3303±0.0123	0.5259±0.1036	0.3197±0.0152
	4	0.5764±0.0870	0.3290±0.0269	0.5818±0.0862	0.3207±0.0244
	5	0.5928±0.0983	0.3297±0.0531	0.5968±0.0987	0.3207±0.0486
DIV	1	0.3610±0.0372	0.3233±0.0038	0.4487±0.1174	0.3857±0.1006
	2	0.4311±0.0970	0.3453±0.0038	0.3940±0.0437	0.2550±0.1624
	3	0.5067±0.0814	0.3380±0.0302	0.4581±0.0394	0.2807±0.0448
	4	0.5691±0.0688	0.3293±0.0239	0.5083±0.0402	0.2790±0.0325
	5	0.6034±0.0425	0.3343±0.0249	0.5420±0.0279	0.2737±0.0014

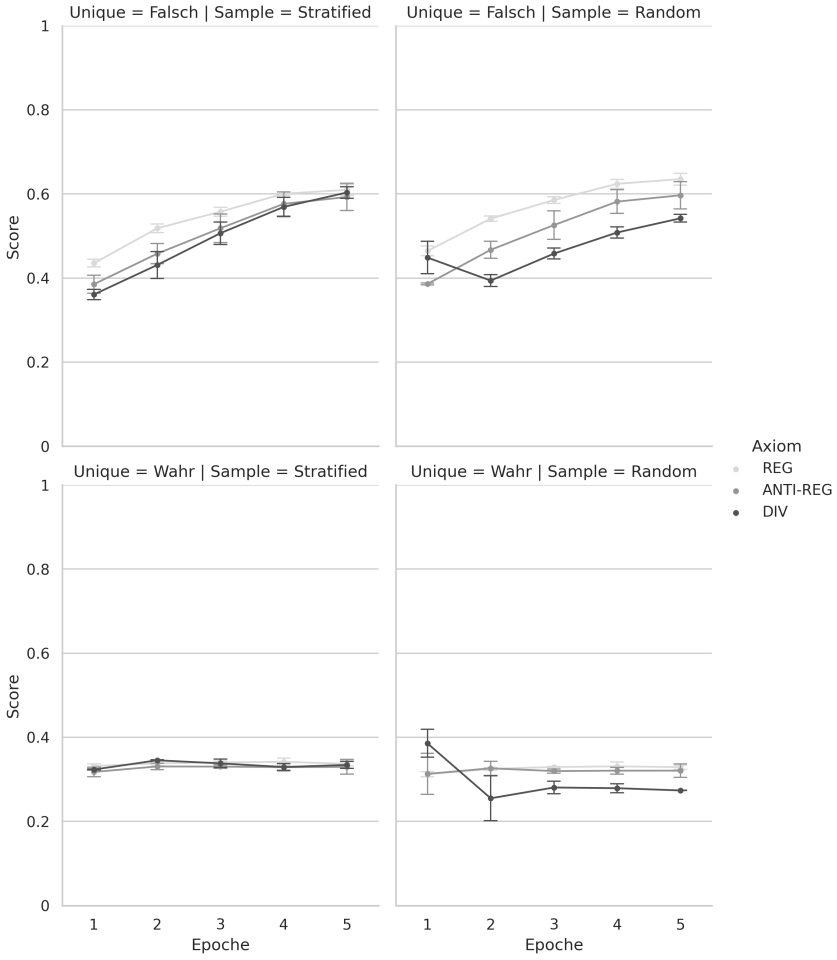

Abbildung A.13: Anfrage-Facetten-Axiome

Tabelle A.15: Term-Nachbarschafts-Axiome

		Stratified		**Random**	
Axiom	**Epoche**	None-Unique	Unique	None-Unique	Unique
PROX1	1	0.4918±0.0421	0.4103±0.0117	0.4777±0.0537	0.3807±0.0334
	2	0.5393±0.0664	0.3760±0.0366	0.5076±0.0981	0.3663±0.0632
	3	0.5811±0.0694	0.3813±0.0597	0.5361±0.0857	0.3590±0.0522
	4	0.6198±0.0595	0.3723±0.0714	0.5730±0.0695	0.3537±0.0834
	5	0.6312±0.0615	0.3573±0.0190	0.5701±0.0734	0.3287±0.0229
PROX2	1	0.4140±0.0347	0.3440±0.0043	0.3724±0.2925	0.3407±0.3157
	2	0.4789±0.0572	0.3603±0.0160	0.4658±0.1330	0.4160±0.1940
	3	0.5398±0.0389	0.3473±0.0304	0.5062±0.1227	0.4020±0.1207
	4	0.5833±0.0133	0.3597±0.0162	0.5418±0.1282	0.3943±0.1227
	5	0.6166±0.0654	0.3580±0.0215	0.5634±0.1668	0.3800±0.0854
PROX3	1	0.6592±0.0312	0.4000±0.2065	0.9066±0.0106	0.9057±0.0503
	2	0.7348±0.0761	0.4133±0.1934	0.9333±0.0371	0.9237±0.0190
	3	0.7871±0.0686	0.3933±0.2381	0.9537±0.0259	0.9057±0.0700
	4	0.8270±0.0792	0.4007±0.2257	0.9630±0.0251	0.9033±0.0743
	5	0.8419±0.0758	0.4053±0.2102	0.9659±0.0219	0.9013±0.0624
PROX4	1	0.4617±0.0304	0.3730±0.0203	0.4147±0.0545	0.2443±0.1280
	2	0.5223±0.0022	0.3723±0.0241	0.4687±0.0515	0.2637±0.1334
	3	0.5682±0.0141	0.3770±0.0263	0.5139±0.0221	0.2820±0.0838
	4	0.6084±0.0410	0.3790±0.0452	0.5496±0.0220	0.2917±0.0963
	5	0.6217±0.0265	0.3663±0.0338	0.5673±0.0251	0.2960±0.0584
PROX5	1	0.6036±0.0247	0.4810±0.0410	0.5012±0.0344	0.4687±0.0223
	2	0.6398±0.0275	0.4797±0.0255	0.5386±0.0680	0.4683±0.0165
	3	0.6729±0.0291	0.4657±0.0648	0.5767±0.0430	0.4607±0.0534
	4	0.7000±0.0384	0.4490±0.0489	0.6073±0.0548	0.4460±0.0474
	5	0.7150±0.0660	0.4483±0.0160	0.6298±0.0921	0.4440±0.0194

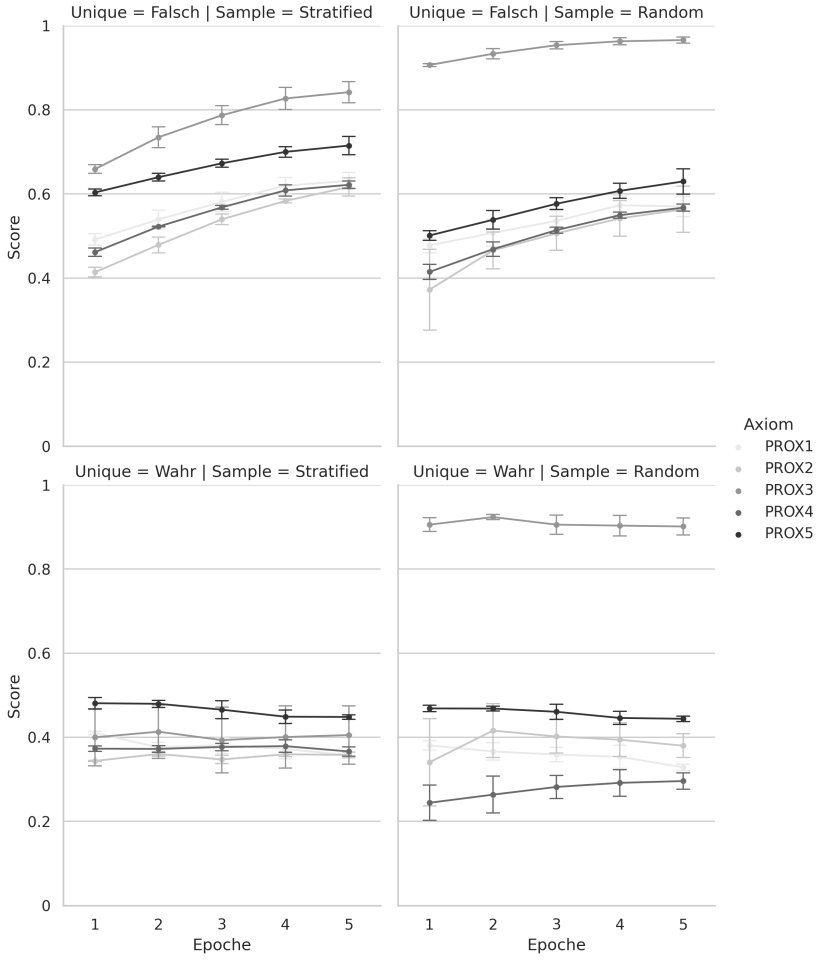

Abbildung A.14: Term-Nachbarschafts-Axiome

Tabelle A.16: Retrieval-Score-Axiome

		Stratified		Random	
Axiom	**Epoche**	None-Unique	Unique	None-Unique	Unique
RS-TF	1	0.4441±0.0347	0.3470±0.0155	0.4851±0.0848	0.4517±0.0682
	2	0.4882±0.0604	0.3420±0.0174	0.4869±0.1086	0.3750±0.1431
	3	0.5568±0.0634	0.3507±0.0402	0.5090±0.0331	0.3587±0.0593
	4	0.6006±0.0479	0.3453±0.0317	0.5371±0.0438	0.3480±0.0609
	5	0.6180±0.0565	0.3503±0.0251	0.5458±0.0612	0.3460±0.1269
RS-TF-IDF	1	0.5426±0.0285	0.3780±0.0245	0.4458±0.0789	0.3757±0.1952
	2	0.5836±0.0164	0.3890±0.0258	0.4692±0.0655	0.3447±0.1865
	3	0.6372±0.0113	0.3907±0.0125	0.5312±0.0462	0.3627±0.2023
	4	0.6757±0.0149	0.3980±0.0075	0.5714±0.0550	0.3687±0.1299
	5	0.6983±0.0195	0.3980±0.0174	0.6089±0.0476	0.3790±0.1212
RS-BM25	1	0.5050±0.0130	0.3753±0.0428	0.4342±0.1085	0.3510±0.2199
	2	0.5421±0.0084	0.3627±0.0207	0.4771±0.0351	0.3917±0.0435
	3	0.5970±0.0191	0.3643±0.0456	0.5303±0.0424	0.4037±0.0426
	4	0.6129±0.0186	0.3740±0.0301	0.5312±0.0715	0.3940±0.0449
	5	0.6340±0.0119	0.3740±0.0237	0.5419±0.0485	0.3790±0.0203
RS-QL	1	0.5164±0.0099	0.3627±0.0331	0.4500±0.0899	0.4137±0.0052
	2	0.5631±0.0261	0.3690±0.1103	0.4858±0.0650	0.3810±0.1122
	3	0.6104±0.0402	0.3820±0.0997	0.5258±0.0385	0.3743±0.0872
	4	0.6539±0.0424	0.3803±0.0749	0.5640±0.0431	0.3767±0.0211
	5	0.6633±0.0329	0.3780±0.0701	0.5712±0.0125	0.3653±0.0837

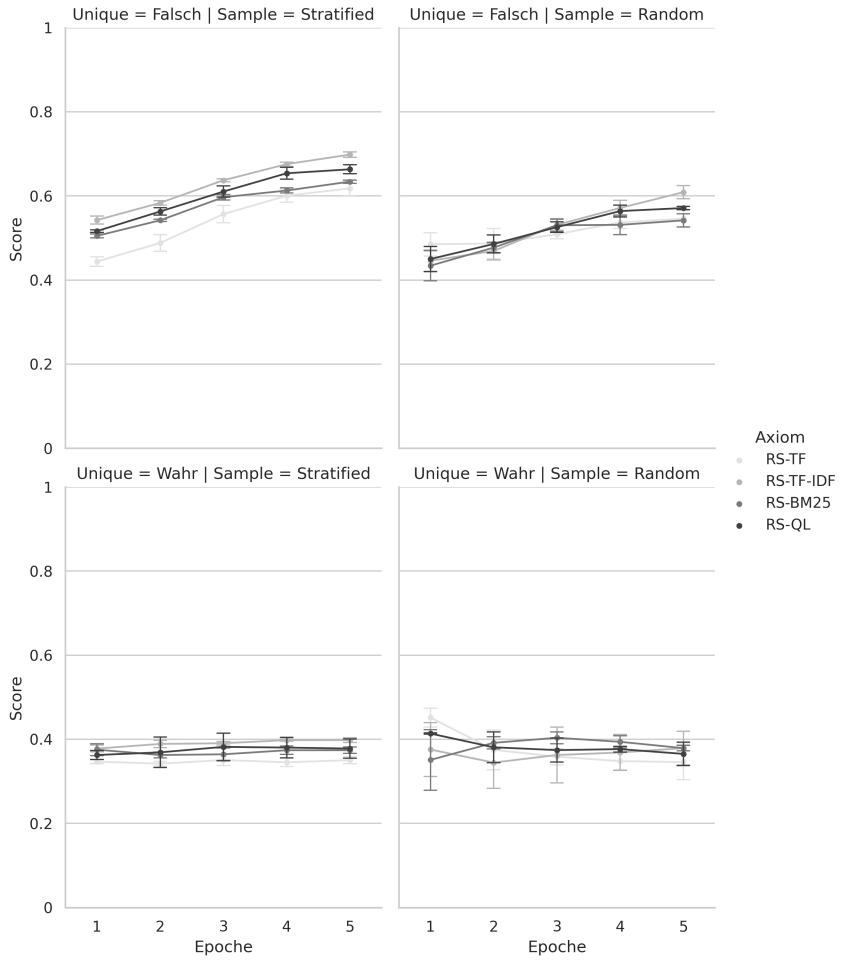

Abbildung A.15: Retrieval-Score-Axiome

A.3 Ergebnisse Experiment 1-A

Nachfolgend eine tabellarische und grafische Darstellung der Accuracy Ergebnisse jedes Axiomes für 10.000 Trainingsdaten und die Ergebnisse der Ranking Aufgabe für 10.000 Trainingsdaten im Multi-Task-Learning für die jeweiligen Test-DatensÃ¤tze für 3 Wiederholungen mit 95% Konfidenzintervallen. Die Unique Test-Datensätze bestehen aus jeweils 1.000 Testdaten, deren Anfragen nicht im Training gesehen wurden und die None-Unique Test-Datensätze bestehen aus 3.000 Testdaten. Wir gruppieren die Ergebnisse für die Axiome nach den Axiom-Gruppen. Die Axiom Aufgaben nutzen die Eingabe Variante A $[CLS]Anfrage + Dokument_1[SEP]Anfrage + Dokument_2[SEP]$.

Tabelle A.17: Übersicht der Ergebnisse unseres Learning-to-Rank Ansatzes für 10.000 Trainingsdaten der Ranking Aufgabe und Multi-Task-Learning mit allen Axiomen in der Eingabe Variante A mit jeweils 10.000 Trainingsdaten auf den 43 und 54 Anfragen des TREC Deep Learning Tracks aus den Jahren 2019 und 2020 für 3 Wiederholungen mit 95% Konfidenzintervallen.

	TREC 2019			**TREC 2020**		
Epoche	NDCG@10	P@1	MRR@10	NDCG@10	P@1	MRR@10
1	0.5712±0.0475	0.5814±0.0000	0.7176±0.0466	0.4974±0.0979	0.5370±0.1380	0.6625±0.0993
2	0.5056±0.0424	0.5194±0.2605	0.6430±0.1710	0.4438±0.0304	0.3704±0.2434	0.5364±0.1890
3	0.4820±0.1045	0.3953±0.2518	0.5729±0.1528	0.4066±0.1243	0.3580±0.3391	0.5029±0.2920
4	0.4694±0.0684	0.4186±0.1155	0.5695±0.0774	0.3887±0.0588	0.3148±0.2005	0.4884±0.1459
5	0.4872±0.0071	0.4341±0.0667	0.6008±0.0145	0.4032±0.0294	0.2778±0.0797	0.4735±0.0294

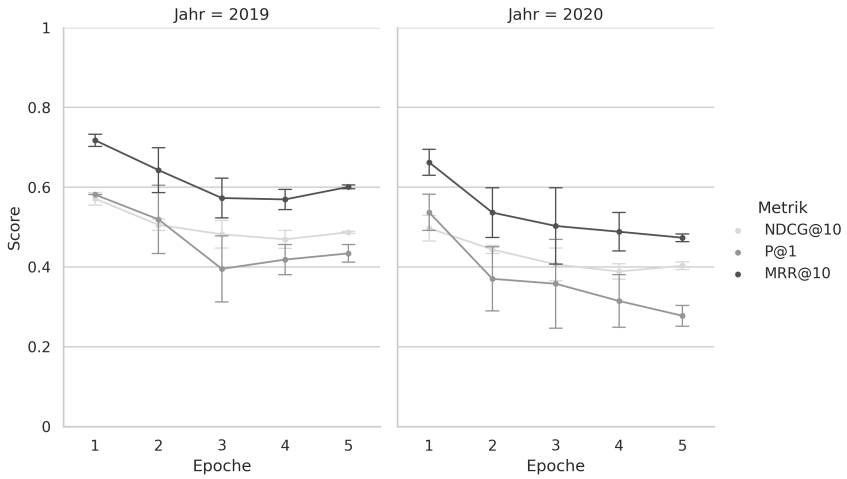

Abbildung A.16: Grafische Übersicht der Ergebnisse unseres Learning-to-Rank Ansatzes für 10.000 Trainingsdaten im Multi-Task-Learning mit allen Axiom Aufgaben in der Eingabe Variante A auf den 43 und 54 Anfragen des TREC Deep Learning Tracks aus den Jahren 2019 und 2020 für 3 Wiederholungen mit 95% Konfidenzintervallen.

Tabelle A.18: Termhäufigkeits-Axiome

		Stratified		Random	
Axiom	**Epoche**	None-Unique	Unique	None-Unique	Unique
TFC1	1	0.4272±0.0252	0.3693±0.0419	0.4290±0.0364	0.3667±0.0453
	2	0.4659±0.0557	0.3560±0.0172	0.4674±0.0720	0.3643±0.0374
	3	0.5091±0.0139	0.3677±0.0235	0.5056±0.0233	0.3710±0.0273
	4	0.5333±0.0158	0.3643±0.0072	0.5303±0.0116	0.3647±0.0103
	5	0.5541±0.0082	0.3707±0.0382	0.5498±0.0177	0.3710±0.0410
TFC3	1	0.5473±0.0167	0.4307±0.0255	0.6621±0.0476	0.9033±0.1066
	2	0.6029±0.0428	0.4313±0.0277	0.7394±0.1915	0.9023±0.1345
	3	0.6489±0.0037	0.4300±0.0418	0.7730±0.0835	0.9163±0.0586
	4	0.6837±0.0251	0.4290±0.0352	0.8060±0.0556	0.8993±0.0703
	5	0.6923±0.0162	0.4317±0.0236	0.8043±0.0223	0.8987±0.0411
M-TDC	1	0.3942±0.0751	0.3427±0.0402	0.3453±0.1951	0.2980±0.2805
	2	0.4601±0.0557	0.3450±0.0277	0.4742±0.1357	0.3587±0.1967
	3	0.5303±0.0539	0.3427±0.0274	0.5652±0.0427	0.4080±0.1043
	4	0.5791±0.0395	0.3427±0.0165	0.6171±0.0153	0.4013±0.0739
	5	0.6010±0.0299	0.3457±0.0207	0.6611±0.0328	0.4350±0.0194
LB1	1	0.4693±0.0189	0.4137±0.0217	0.4848±0.0170	0.5607±0.1382
	2	0.5183±0.0082	0.4067±0.0208	0.5439±0.0554	0.5630±0.1582
	3	0.5687±0.0316	0.3953±0.0460	0.6071±0.0195	0.5467±0.1123
	4	0.6028±0.0605	0.3853±0.0466	0.6476±0.0378	0.4843±0.1742
	5	0.6240±0.0288	0.3823±0.0475	0.6659±0.0224	0.4093±0.1128

Tabelle A.19: Längen-Normierungs-Axiome

		Stratified		Random	
Axiom	**Epoche**	None-Unique	Unique	None-Unique	Unique
LNC1	1	0.4577±0.0448	0.3450±0.0476	0.4311±0.0332	0.4460±0.0548
	2	0.5568±0.0411	0.3300±0.0431	0.4571±0.1791	0.4237±0.1428
	3	0.6100±0.0453	0.3420±0.0066	0.4528±0.0935	0.3860±0.0814
	4	0.6491±0.0088	0.3303±0.0162	0.5009±0.0240	0.3893±0.0343
	5	0.6532±0.0403	0.3297±0.0125	0.5033±0.0460	0.3707±0.0376
TF-LNC	1	0.3626±0.0199	0.3610±0.0203	0.4148±0.3483	0.6737±0.5121
	2	0.4360±0.0119	0.3440±0.0368	0.6584±0.5491	0.4470±0.6863
	3	0.5593±0.0513	0.3383±0.0277	0.8197±0.1464	0.2510±0.3148
	4	0.6008±0.0457	0.3383±0.0309	0.9182±0.0388	0.2170±0.1777
	5	0.6294±0.0353	0.3377±0.0362	0.9337±0.0268	0.1597±0.1141

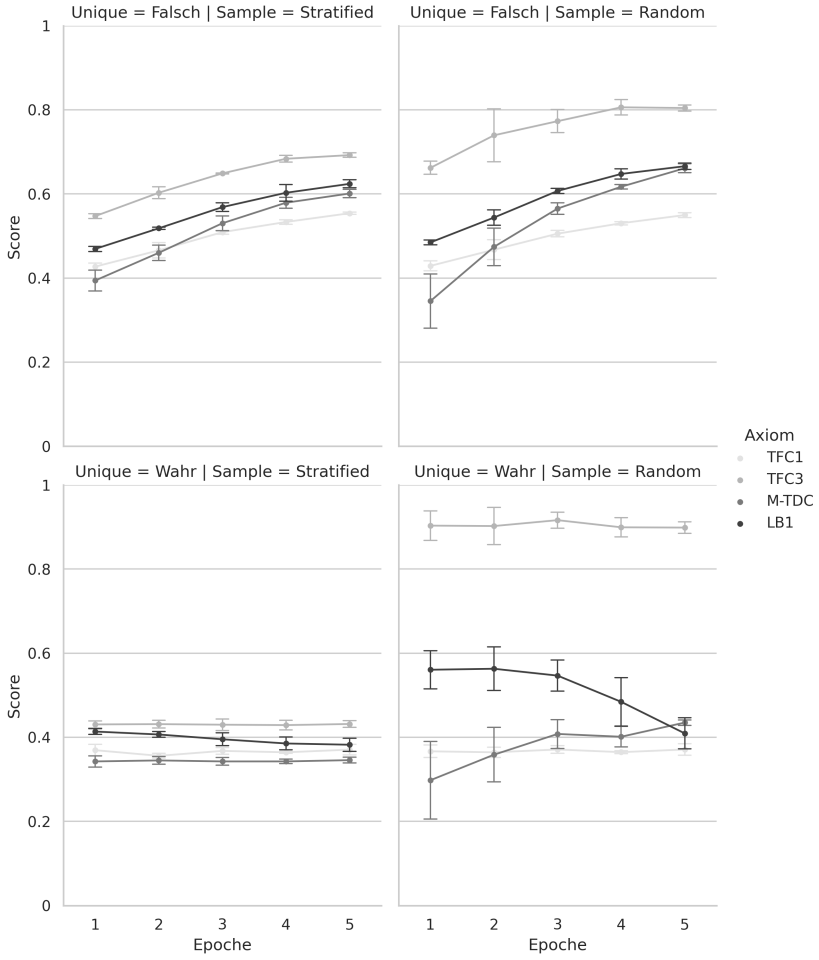

Abbildung A.17: Termhäufigkeits-Axiome

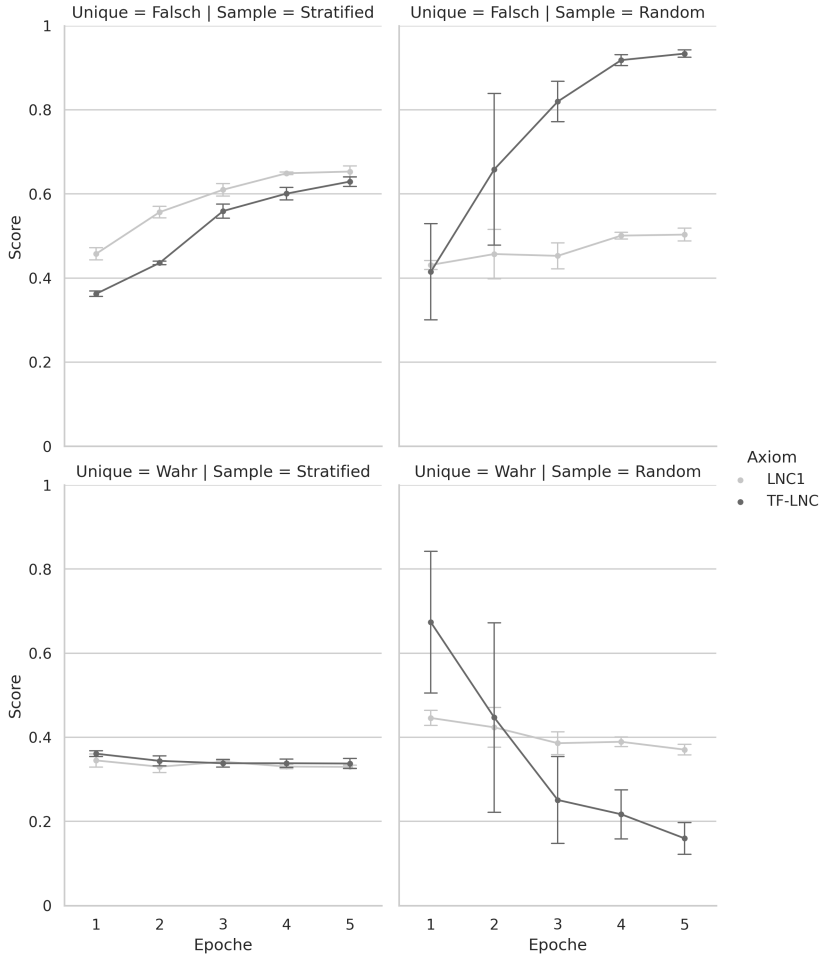

Abbildung A.18: Längen-Normierungs-Axiome

Tabelle A.20: Semantische-Ähnlichkeits-Axiome

		Stratified		Random	
Axiom	**Epoche**	None-Unique	Unique	None-Unique	Unique
STMC1	1	0.6449±0.0469	0.4593±0.0423	0.5270±0.0173	0.4377±0.0432
	2	0.6676±0.0852	0.4613±0.0349	0.5571±0.0633	0.4177±0.0225
	3	0.7154±0.0164	0.4570±0.0704	0.6131±0.0133	0.3703±0.1290
	4	0.7410±0.0334	0.4553±0.0362	0.6469±0.0340	0.3513±0.0932
	5	0.7620±0.0137	0.4407±0.0447	0.6736±0.0140	0.3293±0.1065
STMC2	1	0.3694±0.0433	0.3467±0.0208	0.4529±0.6757	0.3227±0.1351
	2	0.4966±0.0354	0.3317±0.0214	0.8084±0.0894	0.4523±0.5066
	3	0.5940±0.0484	0.3380±0.0237	0.8914±0.1128	0.2783±0.3168
	4	0.6439±0.0228	0.3410±0.0075	0.9459±0.0247	0.2290±0.1864
	5	0.6654±0.0342	0.3410±0.0495	0.9550±0.0138	0.1687±0.1385

Tabelle A.21: Anfrage-Facetten-Axiome

		Stratified		Random	
Axiom	**Epoche**	None-Unique	Unique	None-Unique	Unique
REG	1	0.4049±0.0029	0.3627±0.0428	0.4313±0.0582	0.3677±0.0966
	2	0.4837±0.0238	0.3647±0.0299	0.5281±0.0221	0.3693±0.0540
	3	0.5278±0.0244	0.3550±0.0361	0.5738±0.0243	0.3473±0.0413
	4	0.5536±0.0361	0.3500±0.0317	0.6077±0.0373	0.3380±0.0348
	5	0.5703±0.0428	0.3390±0.0344	0.6251±0.0328	0.3243±0.0366
ANTI-REG	1	0.3907±0.0436	0.3663±0.0268	0.3893±0.0416	0.3863±0.0933
	2	0.4481±0.0342	0.3737±0.0223	0.4537±0.0274	0.4067±0.0559
	3	0.4976±0.0326	0.3737±0.0352	0.5071±0.0233	0.3940±0.0701
	4	0.5336±0.0147	0.3740±0.0124	0.5449±0.0075	0.3793±0.0241
	5	0.5538±0.0322	0.3620±0.0269	0.5654±0.0348	0.3577±0.0402
DIV	1	0.3959±0.0222	0.3407±0.0393	0.4564±0.0399	0.3370±0.1344
	2	0.4807±0.0144	0.3363±0.0029	0.5161±0.1176	0.2827±0.2746
	3	0.5733±0.0588	0.3283±0.0087	0.5570±0.0225	0.1733±0.1293
	4	0.6187±0.0410	0.3247±0.0127	0.6004±0.0311	0.1490±0.0810
	5	0.6440±0.0352	0.3303±0.0208	0.6122±0.0379	0.1247±0.0688

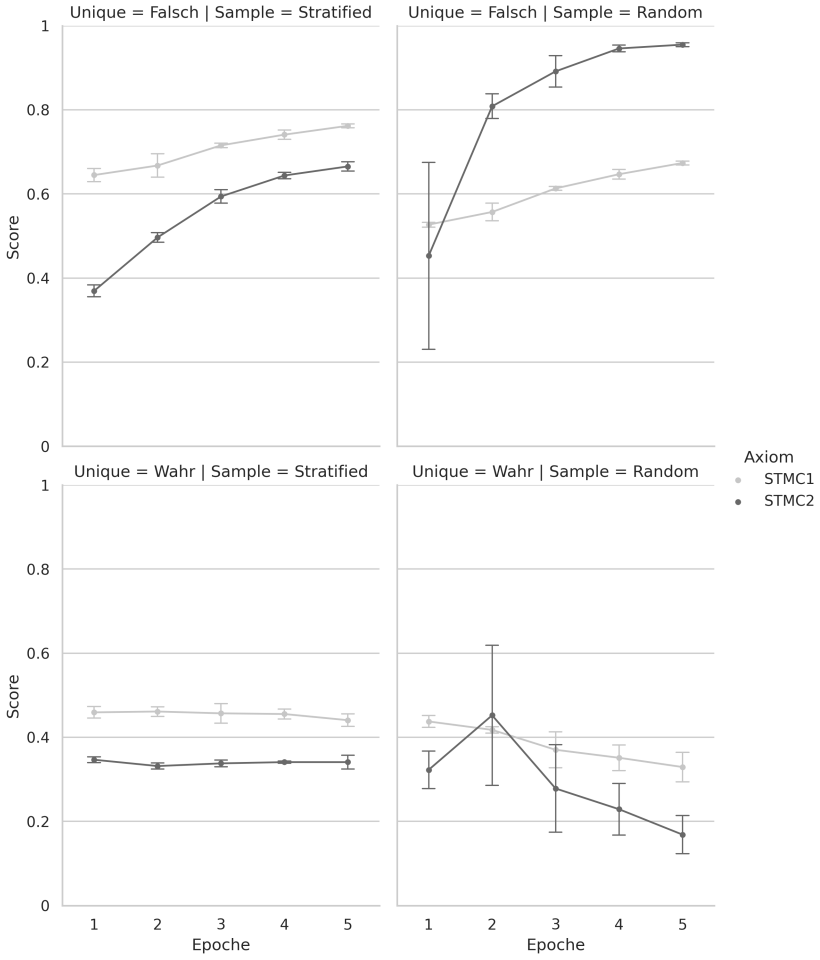

Abbildung A.19: Semantische-Ähnlichkeits-Axiome

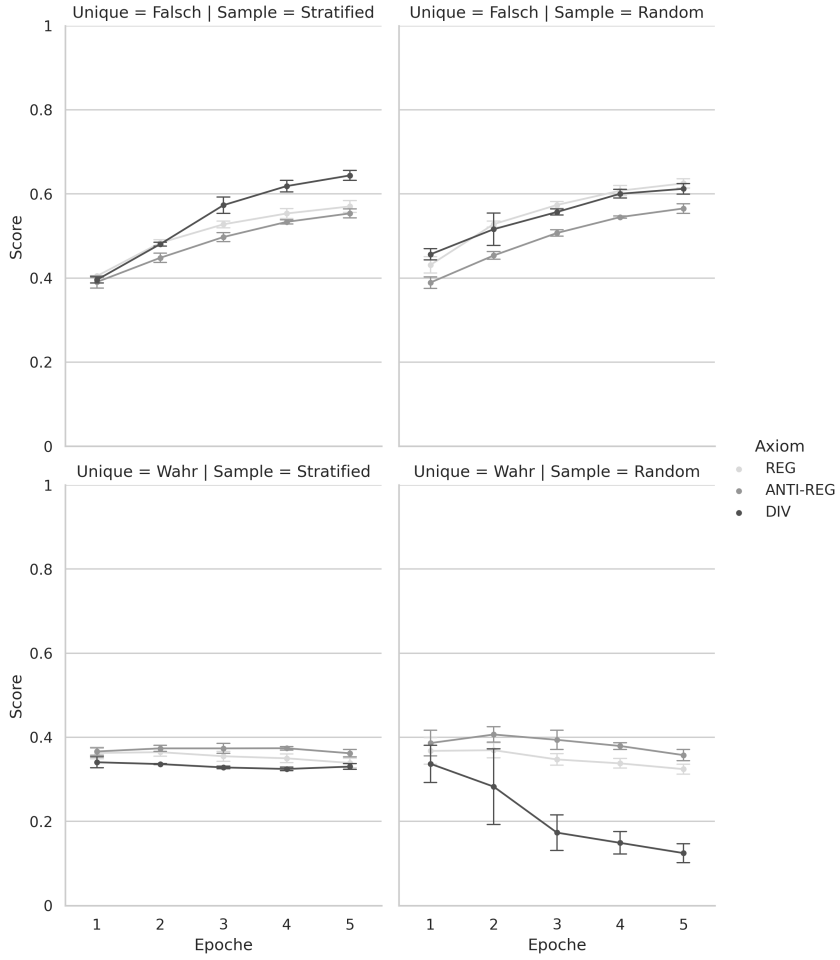

Abbildung A.20: Anfrage-Facetten-Axiome

Tabelle A.22: Term-Nachbarschafts-Axiome

		Stratified		**Random**	
Axiom	**Epoche**	None-Unique	Unique	None-Unique	Unique
PROX1	1	0.4860±0.0104	0.4150±0.0944	0.4343±0.0502	0.3513±0.1355
	2	0.5612±0.0681	0.4050±0.0846	0.5049±0.0201	0.3480±0.1280
	3	0.6207±0.0453	0.4010±0.1130	0.5542±0.0390	0.3270±0.1061
	4	0.6384±0.0393	0.4053±0.0897	0.5766±0.0458	0.3323±0.0783
	5	0.6588±0.0346	0.3990±0.0736	0.5947±0.0345	0.3260±0.0602
PROX2	1	0.4336±0.0186	0.3850±0.0413	0.4184±0.0496	0.3183±0.1115
	2	0.5434±0.0708	0.3803±0.0488	0.4859±0.0509	0.3413±0.2432
	3	0.6414±0.0424	0.3790±0.0400	0.5439±0.0335	0.2660±0.1212
	4	0.6744±0.0105	0.3730±0.0382	0.5818±0.0161	0.2660±0.0869
	5	0.6883±0.0079	0.3783±0.0445	0.5926±0.0096	0.2500±0.0845
PROX3	1	0.6680±0.0187	0.3417±0.0112	0.9129±0.0241	0.8793±0.0607
	2	0.7476±0.0162	0.3537±0.0255	0.9501±0.0218	0.8343±0.0625
	3	0.7982±0.0218	0.3363±0.0470	0.9613±0.0068	0.7267±0.0681
	4	0.8376±0.0277	0.3380±0.0947	0.9659±0.0076	0.6673±0.0616
	5	0.8567±0.0044	0.3327±0.1072	0.9673±0.0022	0.6047±0.1329
PROX4	1	0.4843±0.0136	0.3737±0.0324	0.4249±0.0336	0.2677±0.0231
	2	0.5238±0.0626	0.3677±0.0266	0.5131±0.0154	0.2740±0.1012
	3	0.5630±0.0565	0.3607±0.0288	0.5577±0.0280	0.2277±0.0864
	4	0.5828±0.0570	0.3557±0.0337	0.5871±0.0559	0.2207±0.0598
	5	0.5919±0.0620	0.3543±0.0402	0.5963±0.0643	0.2040±0.0548
PROX5	1	0.5780±0.0299	0.4910±0.0597	0.4991±0.0221	0.4833±0.0567
	2	0.5834±0.0222	0.4650±0.0221	0.5369±0.0080	0.4513±0.0214
	3	0.6311±0.0108	0.4553±0.0497	0.5844±0.0075	0.4313±0.0525
	4	0.6648±0.0342	0.4300±0.0352	0.6217±0.0459	0.4043±0.0352
	5	0.6786±0.0344	0.4183±0.0279	0.6360±0.0612	0.3917±0.0236

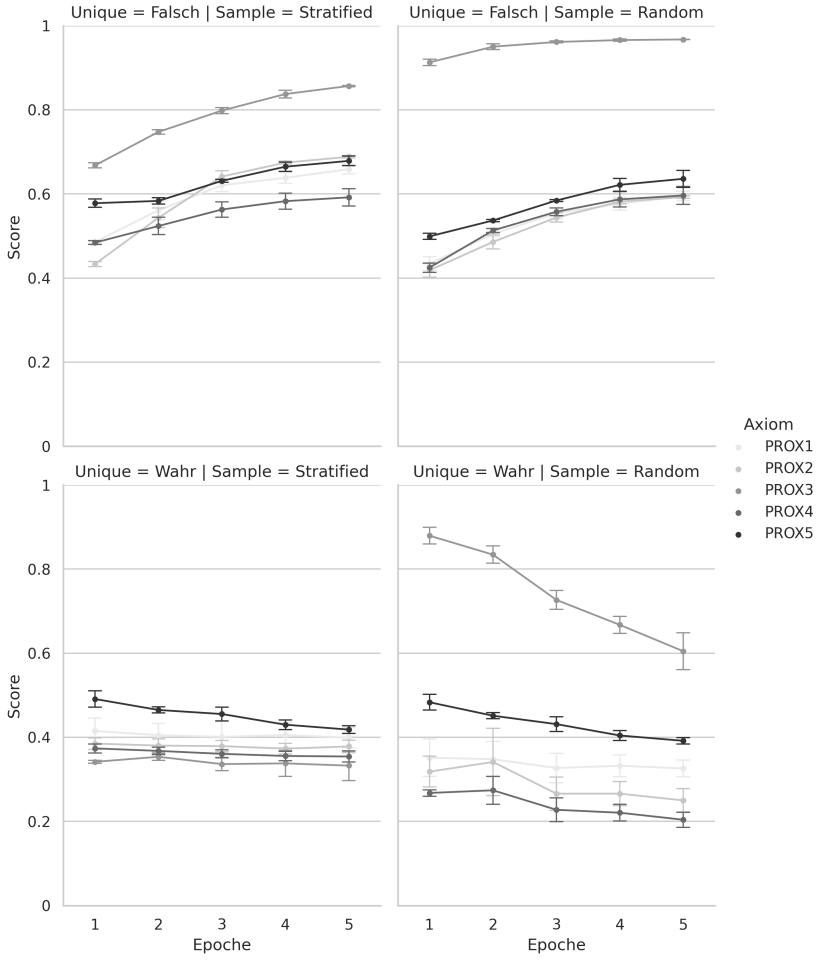

Abbildung A.21: Term-Nachbarschafts-Axiome

Tabelle A.23: Retrieval-Score-Axiome

		Stratified		Random	
Axiom	**Epoche**	None-Unique	Unique	None-Unique	Unique
RS-TF	1	0.4701±0.0277	0.3553±0.0137	0.4718±0.0561	0.4327±0.1191
	2	0.5659±0.0623	0.3577±0.0076	0.5394±0.0376	0.3543±0.1563
	3	0.6593±0.0504	0.3513±0.0112	0.5921±0.0370	0.2397±0.1231
	4	0.7158±0.0712	0.3530±0.0407	0.6477±0.0942	0.2100±0.0263
	5	0.7364±0.0791	0.3550±0.0286	0.6694±0.0987	0.1793±0.0390
RS-TF-IDF	1	0.5820±0.0573	0.3843±0.0014	0.4920±0.0668	0.4367±0.0737
	2	0.6764±0.0383	0.3873±0.0410	0.5560±0.0277	0.3760±0.1157
	3	0.7404±0.0677	0.3903±0.0266	0.6184±0.0865	0.2853±0.0910
	4	0.7763±0.0818	0.3770±0.0277	0.6716±0.1244	0.2467±0.0434
	5	0.7953±0.0837	0.3750±0.0262	0.6987±0.1322	0.2170±0.0348
RS-BM25	1	0.5550±0.0345	0.3507±0.0671	0.4834±0.0381	0.4160±0.0804
	2	0.6524±0.0406	0.3617±0.0288	0.5474±0.0264	0.3643±0.1377
	3	0.7186±0.0302	0.3670±0.0538	0.5988±0.0290	0.2607±0.1209
	4	0.7506±0.0261	0.3557±0.0583	0.6438±0.0268	0.2240±0.0742
	5	0.7714±0.0260	0.3613±0.0684	0.6722±0.0278	0.1990±0.0800
RS-QL	1	0.5476±0.0353	0.3723±0.0125	0.4577±0.0323	0.4033±0.0359
	2	0.6650±0.0242	0.3657±0.0072	0.5600±0.0217	0.3567±0.1177
	3	0.7128±0.0308	0.3713±0.0411	0.5903±0.0245	0.2667±0.1376
	4	0.7396±0.0409	0.3597±0.0311	0.6323±0.0441	0.2303±0.0551
	5	0.7613±0.0530	0.3640±0.0358	0.6596±0.0578	0.2070±0.0277

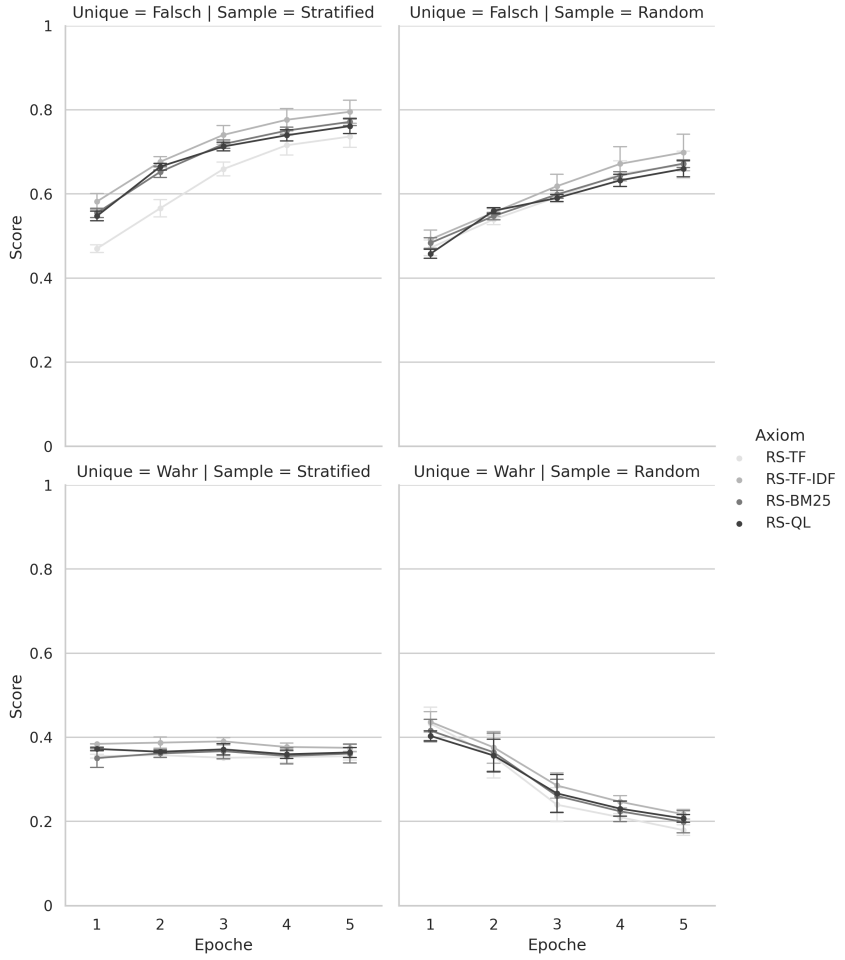

Abbildung A.22: Retrieval-Score-Axiome

A.4 Ergebnisse Experiment 1-B

Nachfolgend eine tabellarische und grafische Darstellung der Accuracy Ergebnisse jedes Axiomes für 10.000 Trainingsdaten und die Ergebnisse der Ranking Aufgabe für 10.000 Trainingsdaten im Multi-Task-Learning für die jeweiligen Test-DatensÃ¤tze für 3 Wiederholungen mit 95% Konfidenzintervallen. Die Unique Test-Datensätze bestehen aus jeweils 1.000 Testdaten, deren Anfragen nicht im Training gesehen wurden und die None-Unique Test-Datensätze bestehen aus 3.000 Testdaten. Wir gruppieren die Ergebnisse fÃ¼r die Axiome nach den Axiom-Gruppen. Die Axiom Aufgaben nutzen die Eingabe Variante B $[CLS]Anfrage[SEP]Dokument_1 + Dokument_2[SEP]$.

Tabelle A.24: Übersicht der Ergebnisse unseres Learning-to-Rank Ansatzes für 10.000 Trainingsdaten der Ranking Aufgabe und Multi-Task-Learning mit allen Axiomen in der Eingabe Variante B mit jeweils 10.000 Trainingsdaten auf den 43 und 54 Anfragen des TREC Deep Learning Tracks aus den Jahren 2019 und 2020 für 3 Wiederholungen mit 95% Konfidenzintervallen.

	TREC 2019			**TREC 2020**		
Epoche	NDCG@10	P@1	MRR@10	NDCG@10	P@1	MRR@10
1	0.6102±0.0143	0.6279±0.1155	0.7532±0.0747	0.5458±0.0493	0.5802±0.1742	0.7108±0.0824
2	0.5727±0.0603	0.5891±0.1334	0.7058±0.0951	0.5039±0.1014	0.4444±0.1594	0.6222±0.1404
3	0.5265±0.0412	0.4961±0.0667	0.6562±0.0248	0.4333±0.0904	0.3642±0.2534	0.5346±0.1990
4	0.5436±0.0375	0.5039±0.0667	0.6614±0.0384	0.4431±0.0584	0.3765±0.1405	0.5520±0.1123
5	0.5399±0.0385	0.5271±0.0882	0.6772±0.0408	0.4197±0.0469	0.3395±0.0531	0.5130±0.0128

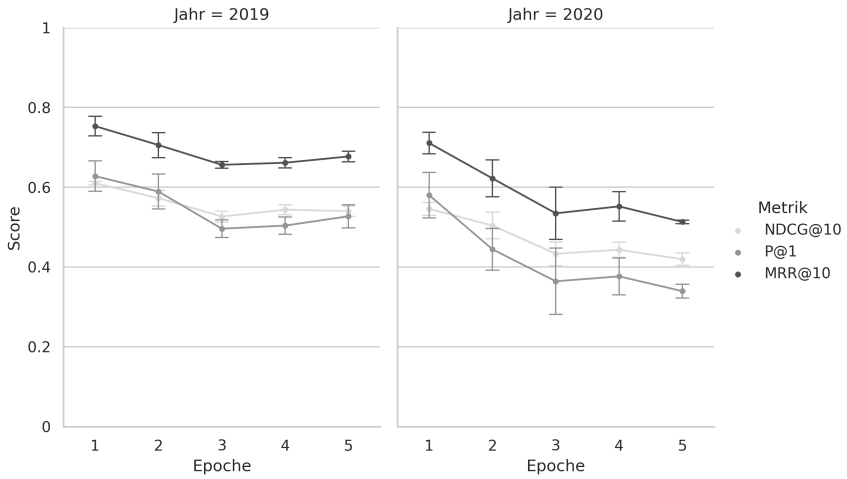

Abbildung A.23: Grafische Übersicht der Ergebnisse unseres Learning-to-Rank Ansatzes für 10.000 Trainingsdaten im Multi-Task-Learning mit allen Axiom Aufgaben in der Eingabe Variante B auf den 43 und 54 Anfragen des TREC Deep Learning Tracks aus den Jahren 2019 und 2020 für 3 Wiederholungen mit 95% Konfidenzintervallen.

Tabelle A.25: Termhäufigkeits-Axiome

		Stratified		Random	
Axiom	**Epoche**	None-Unique	Unique	None-Unique	Unique
TFC1	1	0.4338±0.0085	0.3533±0.0231	0.4337±0.0116	0.3543±0.0311
	2	0.4772±0.0121	0.3460±0.0138	0.4698±0.0051	0.3470±0.0301
	3	0.5183±0.0163	0.3460±0.0199	0.5111±0.0071	0.3457±0.0014
	4	0.5373±0.0063	0.3470±0.0155	0.5298±0.0037	0.3513±0.0419
	5	0.5572±0.0100	0.3540±0.0155	0.5481±0.0025	0.3550±0.0344
TFC3	1	0.5613±0.0108	0.4383±0.0387	0.7080±0.1600	0.9280±0.1637
	2	0.6262±0.0133	0.4367±0.0080	0.7732±0.0783	0.9457±0.0689
	3	0.6672±0.0272	0.4360±0.0217	0.7820±0.0341	0.9430±0.0342
	4	0.6940±0.0775	0.4500±0.0407	0.7914±0.0371	0.9333±0.0488
	5	0.7019±0.0665	0.4570±0.0437	0.7894±0.0543	0.9240±0.0174
M-TDC	1	0.4243±0.0168	0.3590±0.0237	0.4651±0.2122	0.4497±0.1937
	2	0.4871±0.0264	0.3590±0.0221	0.5456±0.1535	0.4367±0.1777
	3	0.5296±0.0072	0.3530±0.0129	0.5847±0.1269	0.4667±0.1986
	4	0.5636±0.0361	0.3560±0.0422	0.6244±0.0627	0.4737±0.1089
	5	0.5789±0.0248	0.3490±0.0366	0.6530±0.0558	0.4907±0.0598
LB1	1	0.4717±0.0287	0.3903±0.0249	0.5124±0.0178	0.6400±0.0174
	2	0.5249±0.0177	0.3827±0.0522	0.5751±0.0082	0.6440±0.0519
	3	0.5680±0.0108	0.4010±0.0539	0.6093±0.0152	0.6077±0.0586
	4	0.5847±0.0251	0.4107±0.0509	0.6330±0.0174	0.5897±0.0373
	5	0.5991±0.0199	0.4140±0.0124	0.6470±0.0253	0.5587±0.0727

Tabelle A.26: Längen-Normierungs-Axiome

		Stratified		Random	
Axiom	**Epoche**	None-Unique	Unique	None-Unique	Unique
LNC1	1	0.4606±0.0031	0.3410±0.0066	0.4289±0.0121	0.4653±0.1184
	2	0.5418±0.0194	0.3507±0.0304	0.4391±0.0332	0.4327±0.0901
	3	0.5981±0.0335	0.3530±0.0174	0.4533±0.0328	0.3943±0.0593
	4	0.6271±0.0768	0.3557±0.0255	0.4779±0.1053	0.3910±0.0778
	5	0.6328±0.0642	0.3517±0.0572	0.4859±0.0836	0.3810±0.1056
TF-LNC	1	0.3670±0.0086	0.3557±0.0366	0.3070±0.1740	0.4550±0.3386
	2	0.4280±0.0575	0.3427±0.0560	0.5057±0.2039	0.3700±0.2608
	3	0.5312±0.0281	0.3320±0.0345	0.7574±0.0971	0.2363±0.0466
	4	0.5879±0.0139	0.3340±0.0410	0.8907±0.0609	0.2260±0.0579
	5	0.6123±0.0244	0.3400±0.0262	0.9202±0.0070	0.2107±0.0165

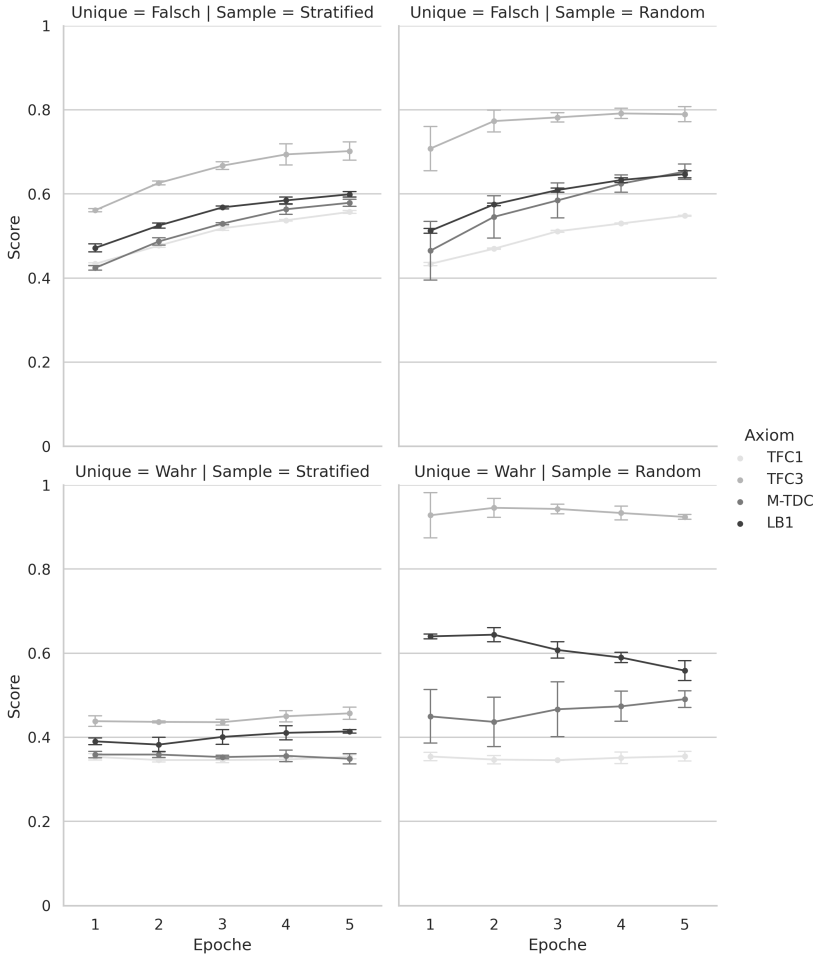

Abbildung A.24: Termhäufigkeits-Axiome

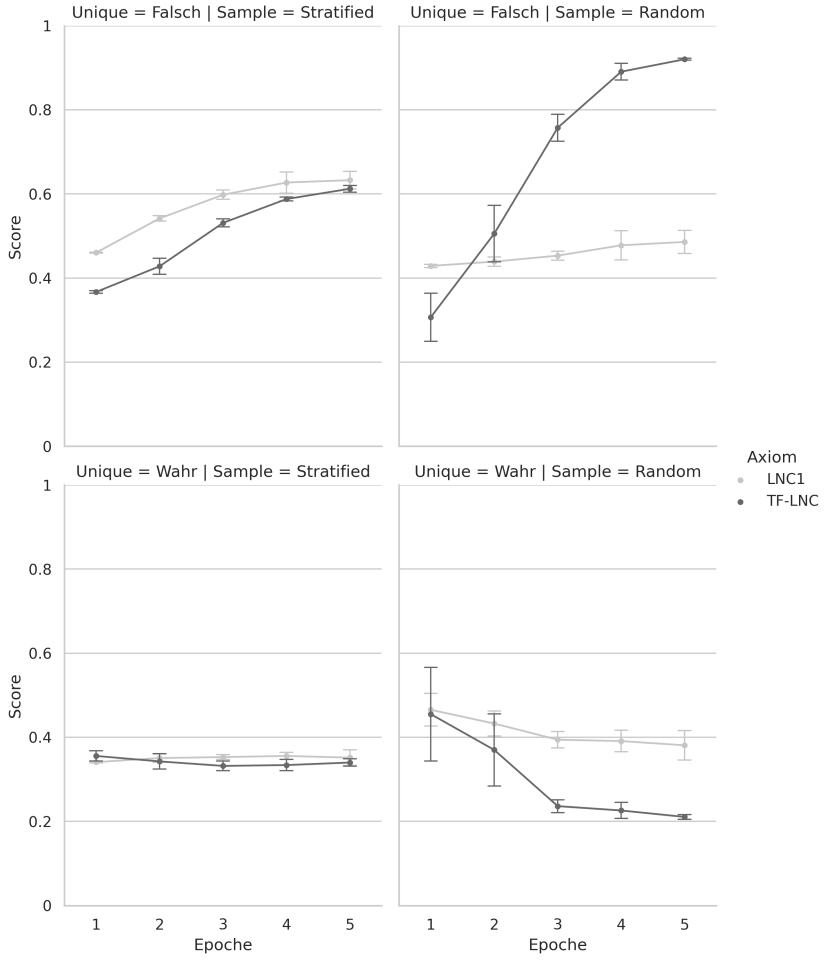

Abbildung A.25: Längen-Normierungs-Axiome

Tabelle A.27: Semantische-Ähnlichkeits-Axiome

		Stratified		Random	
Axiom	**Epoche**	None-Unique	Unique	None-Unique	Unique
STMC1	1	0.6642±0.0139	0.4013±0.0466	0.5416±0.0207	0.4437±0.0377
	2	0.7110±0.0186	0.4403±0.0556	0.6011±0.0323	0.4527±0.0445
	3	0.7377±0.0543	0.4447±0.0673	0.6347±0.0700	0.4407±0.0758
	4	0.7574±0.0655	0.4327±0.0635	0.6629±0.0920	0.4127±0.0676
	5	0.7669±0.0598	0.4337±0.0428	0.6739±0.0800	0.4010±0.0621
STMC2	1	0.3860±0.0252	0.3280±0.0361	0.4900±0.5054	0.5077±0.4622
	2	0.4847±0.0637	0.3253±0.0137	0.6269±0.2415	0.3233±0.1416
	3	0.5812±0.0332	0.3313±0.0251	0.8319±0.0489	0.2757±0.0487
	4	0.6310±0.0152	0.3340±0.0174	0.9242±0.0249	0.2483±0.0362
	5	0.6459±0.0170	0.3287±0.0103	0.9438±0.0029	0.2307±0.0372

Tabelle A.28: Anfrage-Facetten-Axiome

		Stratified		Random	
Axiom	**Epoche**	None-Unique	Unique	None-Unique	Unique
REG	1	0.4207±0.0171	0.3463±0.0117	0.4361±0.0730	0.3397±0.0263
	2	0.4890±0.0167	0.3523±0.0127	0.5081±0.0449	0.3443±0.0239
	3	0.5262±0.0371	0.3337±0.0417	0.5524±0.0462	0.3197±0.0547
	4	0.5472±0.0502	0.3453±0.0575	0.5889±0.0619	0.3303±0.0559
	5	0.5497±0.0392	0.3397±0.0331	0.6007±0.0475	0.3243±0.0317
ANTI-REG	1	0.4062±0.0112	0.3637±0.0038	0.4080±0.0204	0.3987±0.0355
	2	0.4688±0.0113	0.3597±0.0152	0.4747±0.0177	0.3857±0.0714
	3	0.5090±0.0320	0.3617±0.0328	0.5179±0.0383	0.3810±0.0435
	4	0.5396±0.0580	0.3663±0.0169	0.5500±0.0623	0.3780±0.0163
	5	0.5512±0.0619	0.3590±0.0240	0.5626±0.0662	0.3710±0.0523
DIV	1	0.3789±0.0089	0.3403±0.0275	0.4093±0.1226	0.3007±0.3580
	2	0.4672±0.0368	0.3257±0.0531	0.4549±0.0569	0.2010±0.1171
	3	0.5631±0.0205	0.3383±0.0319	0.5363±0.0466	0.1670±0.0114
	4	0.6114±0.0157	0.3300±0.0529	0.5838±0.0371	0.1553±0.0534
	5	0.6293±0.0448	0.3353±0.0408	0.5952±0.0414	0.1563±0.0320

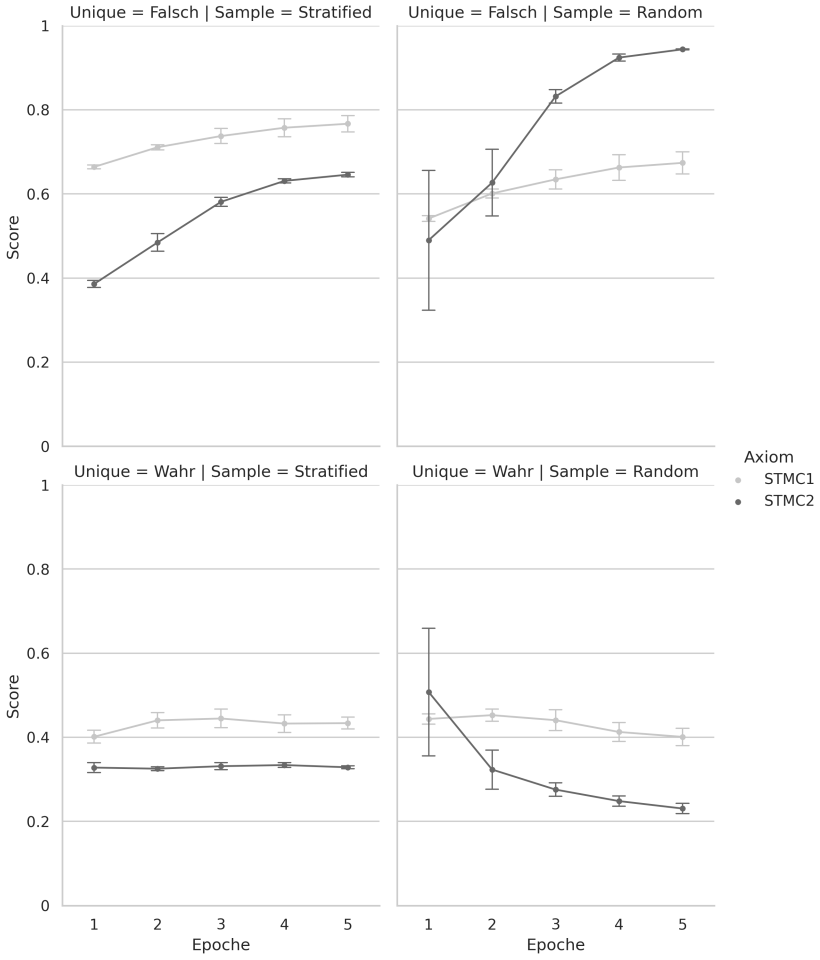

Abbildung A.26: Semantische-Ähnlichkeits-Axiome

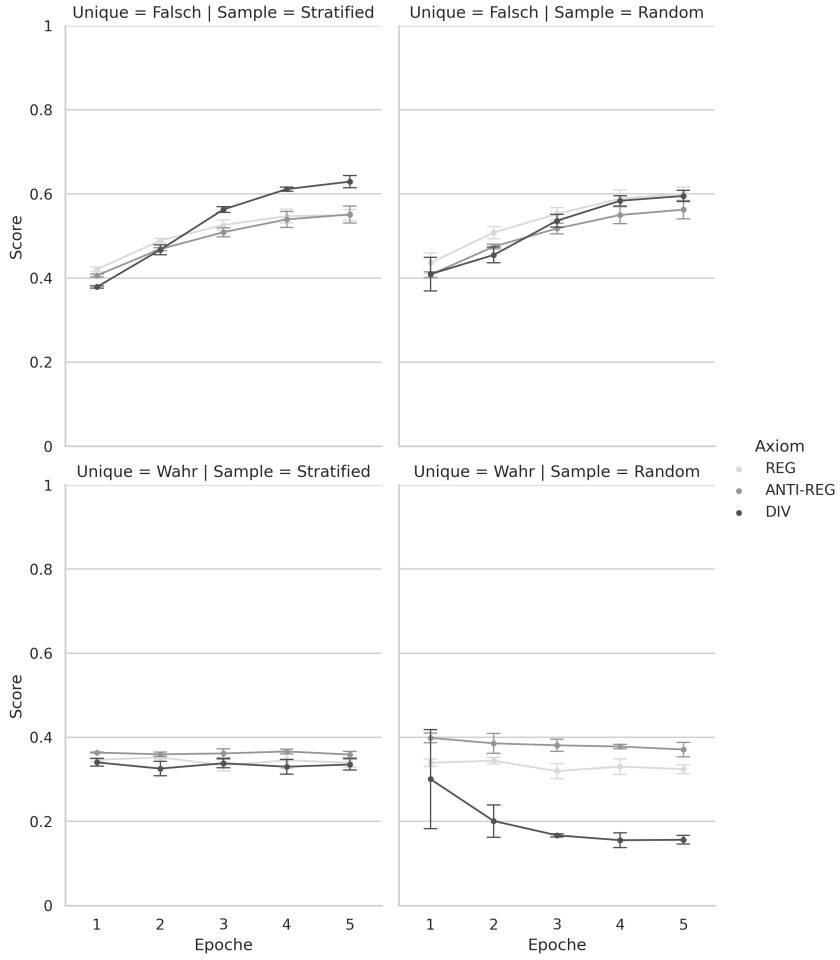

Abbildung A.27: Anfrage-Facetten-Axiome

Tabelle A.29: Term-Nachbarschafts-Axiome

		Stratified		**Random**	
Axiom	**Epoche**	None-Unique	Unique	None-Unique	Unique
PROX1	1	0.4850±0.0129	0.4177±0.0696	0.4670±0.0389	0.3870±0.0672
	2	0.5407±0.0316	0.3967±0.0029	0.5058±0.0284	0.3543±0.0417
	3	0.5990±0.0237	0.3803±0.0211	0.5516±0.0342	0.3317±0.0531
	4	0.6341±0.0077	0.3857±0.0294	0.5853±0.0169	0.3363±0.0632
	5	0.6562±0.0027	0.3827±0.0162	0.6037±0.0215	0.3280±0.0413
PROX2	1	0.4319±0.0109	0.3700±0.0293	0.4209±0.0400	0.3350±0.1457
	2	0.5080±0.0535	0.3867±0.0307	0.4453±0.0679	0.3150±0.0898
	3	0.5917±0.0826	0.3643±0.0475	0.5166±0.0520	0.3003±0.0475
	4	0.6452±0.0580	0.3660±0.0713	0.5670±0.0579	0.3020±0.0553
	5	0.6673±0.0504	0.3650±0.0652	0.5797±0.0598	0.2797±0.0225
PROX3	1	0.6687±0.0153	0.3627±0.1630	0.8882±0.0523	0.9220±0.0652
	2	0.7139±0.0338	0.3383±0.0432	0.9098±0.0402	0.9123±0.0617
	3	0.7526±0.0461	0.3430±0.0301	0.9263±0.0402	0.8807±0.0536
	4	0.7904±0.0650	0.3397±0.0255	0.9420±0.0461	0.8507±0.1184
	5	0.8060±0.0512	0.3347±0.0324	0.9471±0.0436	0.8113±0.1671
PROX4	1	0.4748±0.0037	0.3797±0.0385	0.4166±0.0254	0.2700±0.0407
	2	0.5368±0.0524	0.3613±0.0038	0.4733±0.0574	0.2360±0.0510
	3	0.5909±0.0072	0.3617±0.0057	0.5581±0.0222	0.2297±0.0249
	4	0.6103±0.0169	0.3517±0.0180	0.5910±0.0522	0.2257±0.0291
	5	0.6114±0.0237	0.3490±0.0174	0.5943±0.0331	0.2167±0.0231
PROX5	1	0.5957±0.0363	0.4907±0.0165	0.4987±0.0335	0.4813±0.0311
	2	0.6477±0.0412	0.4910±0.0348	0.5548±0.0497	0.4743±0.0524
	3	0.6828±0.0342	0.4693±0.0563	0.6046±0.0379	0.4450±0.0564
	4	0.7023±0.0217	0.4537±0.0387	0.6340±0.0318	0.4297±0.0328
	5	0.7160±0.0155	0.4423±0.0548	0.6466±0.0272	0.4157±0.0510

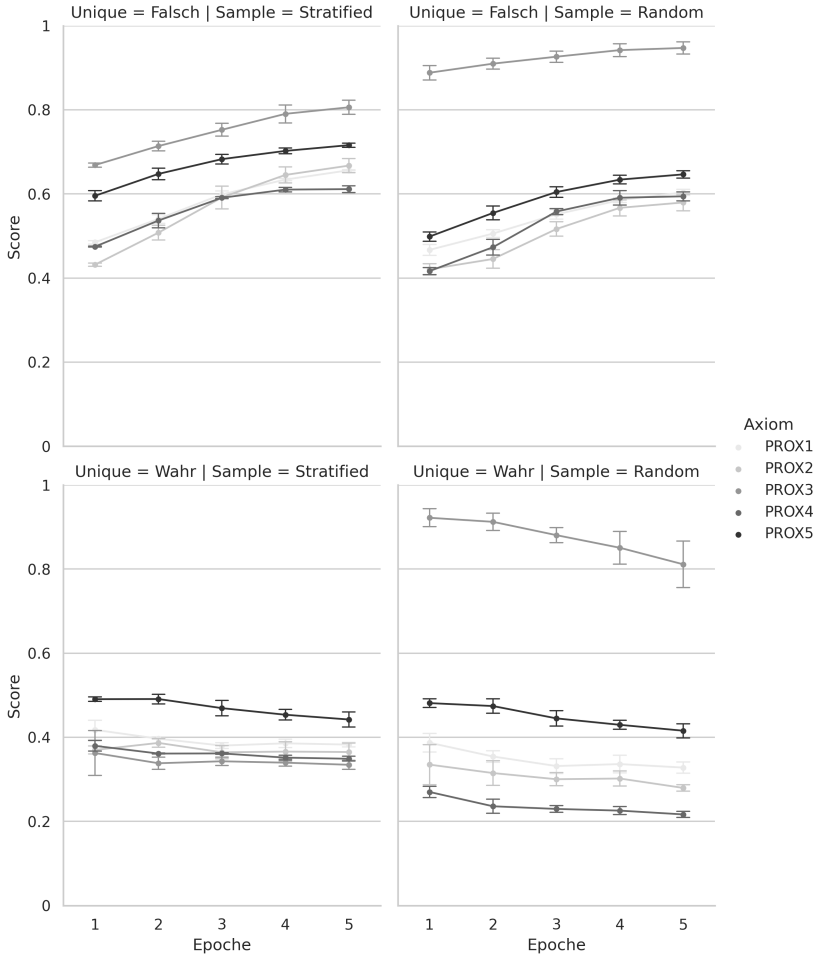

Abbildung A.28: Term-Nachbarschafts-Axiome

Tabelle A.30: Retrieval-Score-Axiome

		Stratified		Random	
Axiom	**Epoche**	None-Unique	Unique	None-Unique	Unique
RS-TF	1	0.4414±0.0238	0.3507±0.0313	0.4567±0.0236	0.4610±0.0986
	2	0.5531±0.0765	0.3490±0.0221	0.5130±0.0613	0.3077±0.1251
	3	0.6460±0.0892	0.3443±0.0330	0.5781±0.0970	0.2040±0.0923
	4	0.6973±0.0672	0.3380±0.0375	0.6299±0.0885	0.1967±0.0625
	5	0.7120±0.0545	0.3440±0.0163	0.6399±0.0652	0.1887±0.0662
RS-TF-IDF	1	0.5550±0.0712	0.3720±0.0155	0.4764±0.0746	0.4500±0.1081
	2	0.6499±0.0944	0.3910±0.0336	0.5389±0.1085	0.3473±0.0891
	3	0.7202±0.0850	0.3870±0.0050	0.5987±0.1290	0.2630±0.0485
	4	0.7514±0.0804	0.3753±0.0268	0.6358±0.1130	0.2470±0.0857
	5	0.7626±0.0447	0.3720±0.0224	0.6512±0.0553	0.2293±0.0714
RS-BM25	1	0.5186±0.0292	0.3723±0.0244	0.4464±0.0596	0.3510±0.0808
	2	0.6287±0.0424	0.3763±0.0100	0.5192±0.0157	0.2873±0.0590
	3	0.7113±0.0209	0.3670±0.0151	0.5908±0.0179	0.2417±0.0511
	4	0.7417±0.0244	0.3527±0.0052	0.6257±0.0335	0.2240±0.0673
	5	0.7523±0.0421	0.3570±0.0149	0.6407±0.0399	0.2173±0.0625
RS-QL	1	0.5327±0.0374	0.3673±0.0223	0.4570±0.0514	0.3593±0.0288
	2	0.6259±0.0601	0.3707±0.0634	0.5181±0.0730	0.2810±0.0821
	3	0.7017±0.0245	0.3617±0.0379	0.5892±0.0270	0.2250±0.0424
	4	0.7220±0.0216	0.3563±0.0165	0.6084±0.0348	0.2220±0.0562
	5	0.7338±0.0358	0.3567±0.0183	0.6240±0.0393	0.2147±0.0659

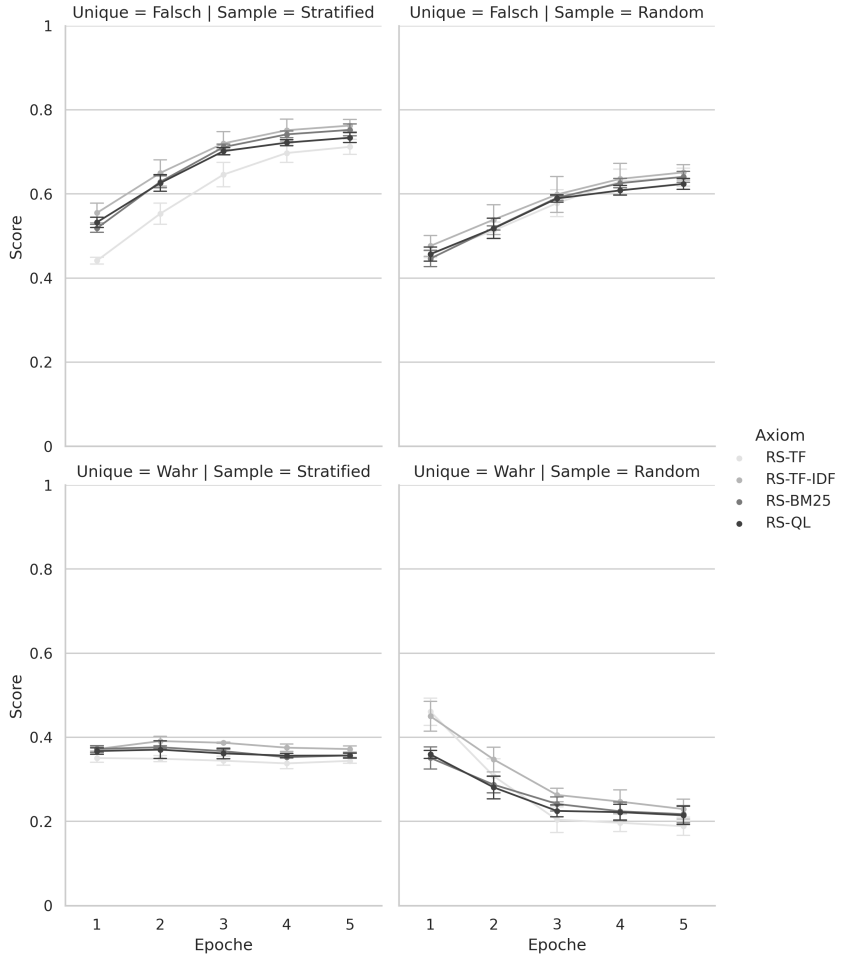

Abbildung A.29: Retrieval-Score-Axiome

A.5 Ergebnisse Experiment 2

Nachfolgend eine tabellarische und grafische Darstellung der Ergebnisse des Multi-Task-Learnings der Ranking Aufgabe mit 100.000 Trainingsdaten und den Axiom Aufgaben M-TDC und LB1 mit jeweils 25.000. Die Axiome werden bezüglich Accurcy evaluiert.

Tabelle A.31: Die Ergebnisse der Ranking Aufgabe mit 100.000 Trainingsdaten im Experiment 2. Die besten Ergebnisse sind fett markiert.

	TREC 2019			**TREC 2020**		
Epoche	NDCG@10	P@1	MRR@10	NDCG@10	P@1	MRR@10
1	**0.6741**	0.6512	0.7677	**0.6257**	**0.6296**	**0.7605**
2	0.6315	0.5581	0.6915	0.5553	0.4630	0.6339
3	0.6603	**0.6977**	**0.7940**	0.5680	0.5370	0.6864
4	0.5988	0.4651	0.6243	0.4883	0.5000	0.6391
5	0.5956	0.5581	0.6718	0.4626	0.3519	0.5428

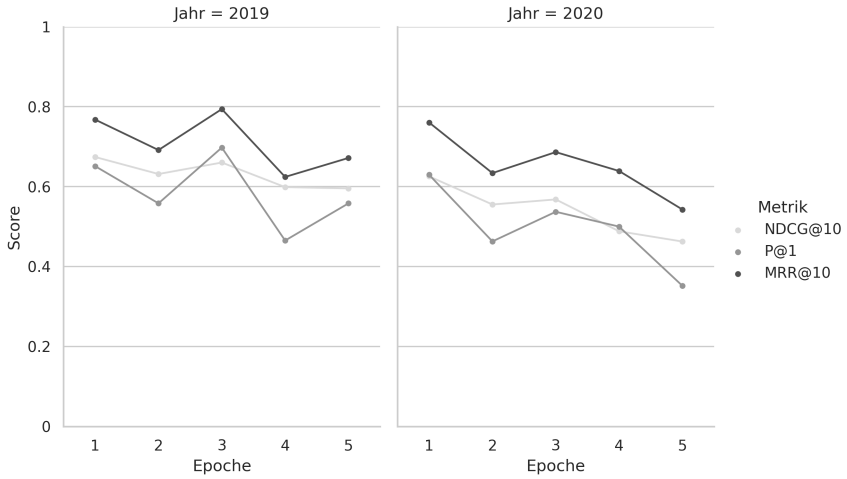

Abbildung A.30: Die Ergebnisse der Ranking Aufgabe mit 100.000 Trainingsdaten im Experiment 2.

Tabelle A.32: Die Accuracy Ergebnisse der M-TDC und LB1 Axiom Aufgabe mit 25.000 Trainingsdaten im Experiment 2.

		Stratified		**Random**	
Axiom	**Epoche**	None-Unique	Unique	None-Unique	Unique
M-TDC	1	0.4427	0.3320	0.5307	0.6170
	2	0.4737	0.3370	0.5607	0.6300
	3	0.4887	0.3420	0.5657	0.6160
	4	0.5150	0.3370	0.5733	0.5970
	5	0.5240	0.3400	0.5420	0.5170
LB1	1	0.4890	0.3780	0.5287	0.6360
	2	0.4963	0.3890	0.5307	0.6140
	3	0.5050	0.3910	0.5447	0.6310
	4	0.4983	0.4000	0.5360	0.6000
	5	0.5007	0.3980	0.5313	0.5730

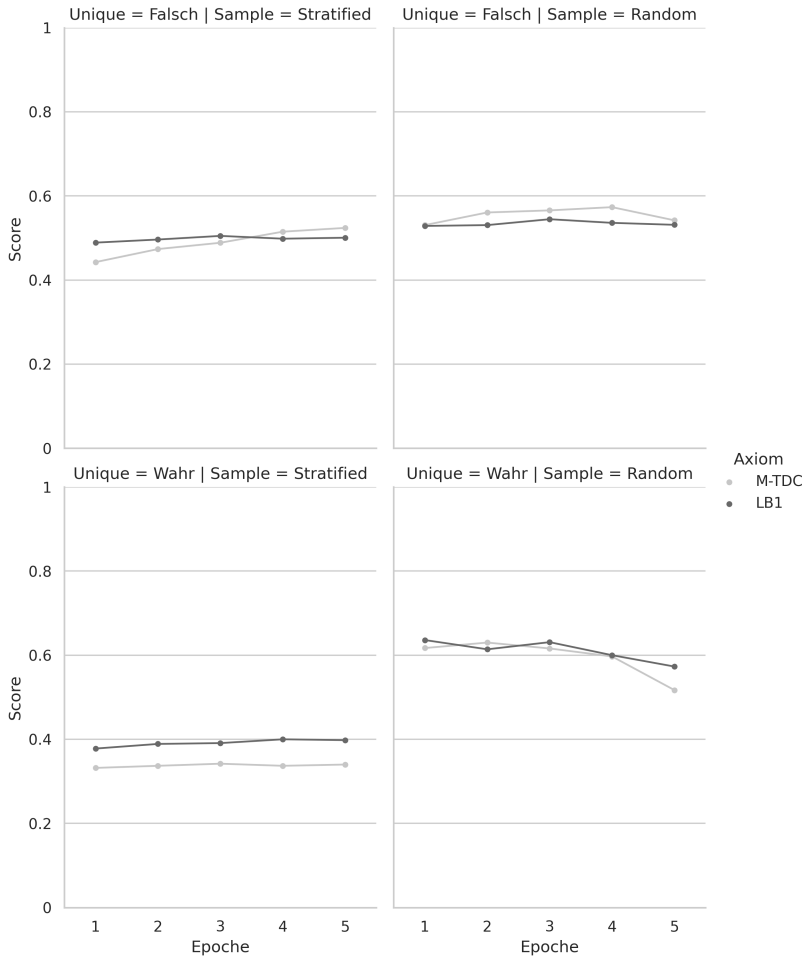

Abbildung A.31: Die Accuracy Ergebnisse von M-TDC und LB1 mit jeweils 25.000 stratifizierten Trainingsdaten im Experiment 2.

A.6 Ergebnisse Experiment 3

Nachfolgend eine tabellarische und grafische Darstellung der Ergebnisse des Multi-Task-Learnings der Ranking Aufgabe mit 100.000 Trainingsdaten und den Axiom Aufgaben REG und ANTI-REG mit jeweils 25.000. Die Axiome werden bezüglich Accurcy evaluiert.

Tabelle A.33: Die Ergebnisse der Ranking Aufgabe mit 100.000 Trainingsdaten im Experiment 3. Die besten Ergebnisse sind fett markiert.

	TREC 2019			**TREC 2020**		
Epoche	NDCG@10	P@1	MRR@10	NDCG@10	P@1	MRR@10
1	**0.6830**	0.6977	0.8314	**0.6558**	**0.6667**	**0.7901**
2	0.6764	**0.7674**	**0.8434**	0.6094	0.6481	0.7409
3	0.5678	0.5349	0.6903	0.5078	0.3889	0.5677
4	0.5702	0.5814	0.7109	0.5227	0.4815	0.6391
5	0.5343	0.5814	0.7026	0.5014	0.4074	0.5864

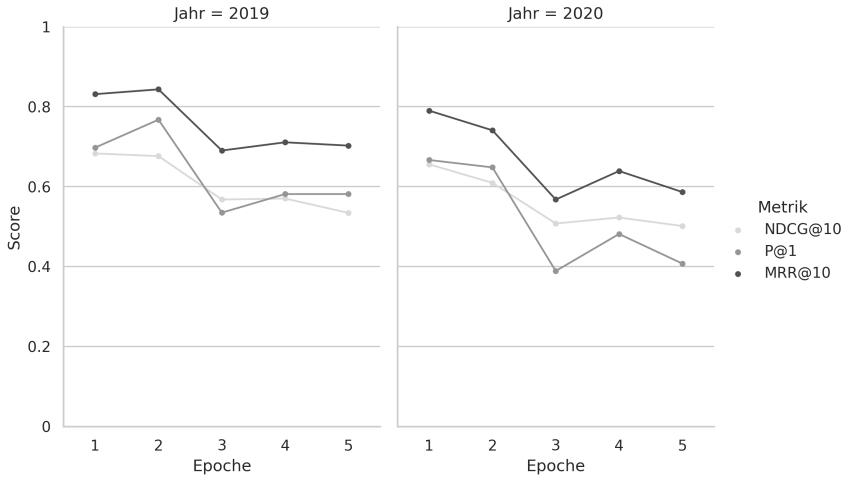

Abbildung A.32: Die Ergebnisse der Ranking Aufgabe mit 100.000 Trainingsdaten im Experiment 3.

Tabelle A.34: Die Accuracy Ergebnisse der REG und ANTI-REG Axiom Aufgabe mit 25.000 Trainingsdaten im Experiment 3.

		Stratified		**Random**	
Axiom	**Epoche**	None-Unique	Unique	None-Unique	Unique
REG	1	0.4113	0.3500	0.4087	0.3300
	2	0.4517	0.3520	0.4583	0.3350
	3	0.4577	0.3590	0.4707	0.3510
	4	0.4613	0.3700	0.4800	0.3670
	5	0.4637	0.3460	0.4850	0.3460
ANTI-REG	1	0.4073	0.3460	0.4070	0.3270
	2	0.4360	0.3450	0.4390	0.3380
	3	0.4383	0.3440	0.4417	0.3310
	4	0.4450	0.3520	0.4510	0.3520
	5	0.4460	0.3620	0.4540	0.3570

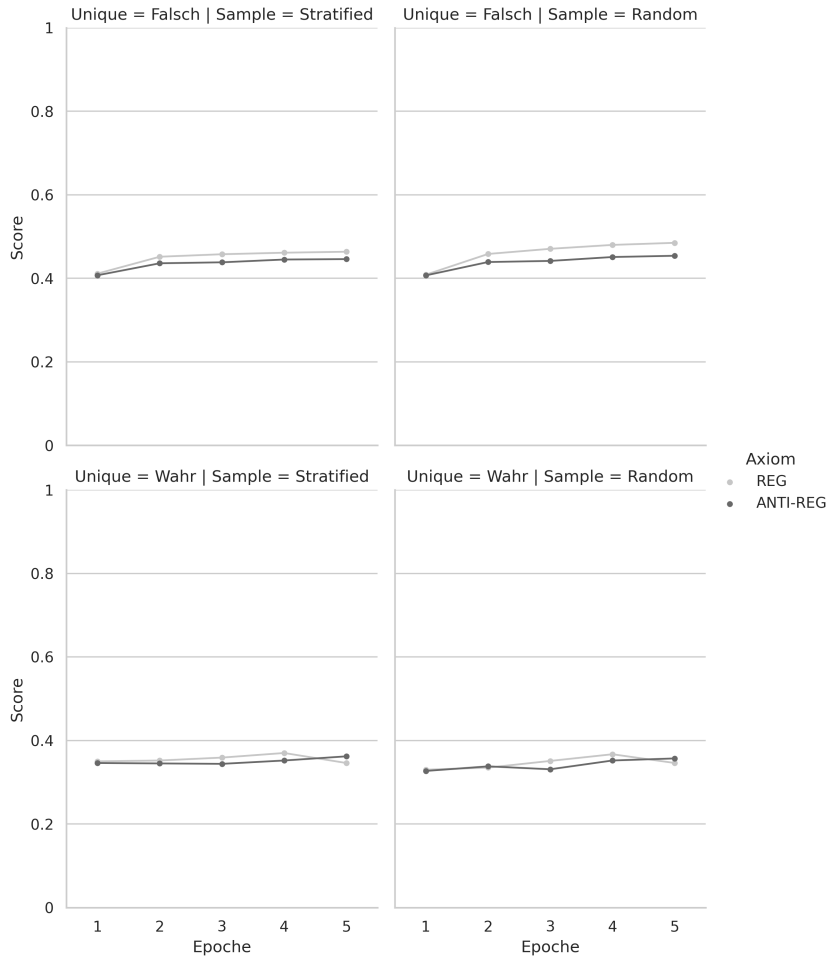

Abbildung A.33: Die Accuracy Ergebnisse von REG und ANTI-REG mit jeweils 25.000 stratifizierten Trainingsdaten im Experiment 3.

A.7 Ergebnisse Experiment 4

Nachfolgend eine tabellarische und grafische Darstellung der Ergebnisse des Multi-Task-Learnings der Ranking Aufgabe mit 100.000 Trainingsdaten und den Axiom Aufgaben PROX4 und PROX5 mit jeweils 25.000. Die Axiome werden bezüglich Accurcy evaluiert.

Tabelle A.35: Die Ergebnisse der Ranking Aufgabe mit 100.000 Trainingsdaten im Experiment 4. Die besten Ergebnisse sind fett markiert.

	TREC 2019			**TREC 2020**		
Epoche	NDCG@10	P@1	MRR@10	NDCG@10	P@1	MRR@10
1	**0.6584**	**0.6512**	0.7409	**0.6428**	**0.6667**	**0.7670**
2	0.6534	0.6279	**0.7469**	0.6040	0.6111	0.7424
3	0.6108	0.5349	0.6868	0.5575	0.4815	0.6343
4	0.5752	0.5814	0.7067	0.5442	0.4815	0.6470
5	0.5504	0.5814	0.6864	0.5132	0.4259	0.6006

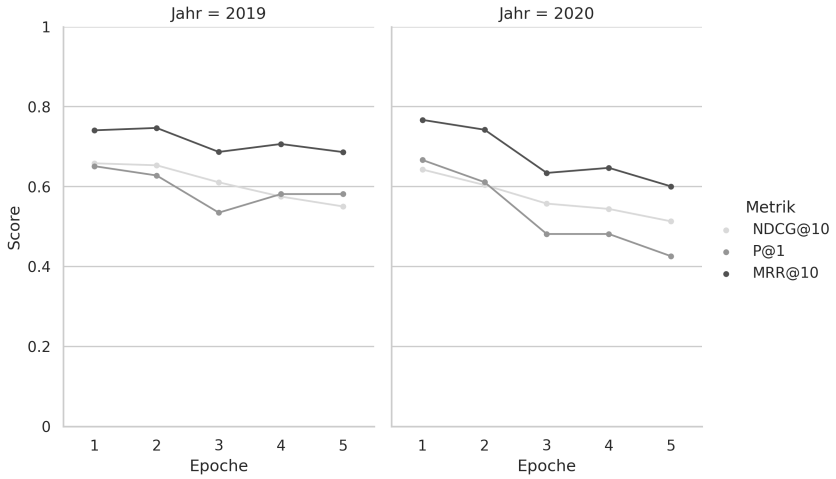

Abbildung A.34: Die Ergebnisse der Ranking Aufgabe mit 100.000 Trainingsdaten im Experiment 4.

Tabelle A.36: Die Accuracy Ergebnisse der PROX4 und PROX5 Axiom Aufgabe mit 25.000 Trainingsdaten im Experiment 4.

		Stratified		**Random**	
Axiom	**Epoche**	None-Unique	Unique	None-Unique	Unique
PROX4	1	0.4650	0.3780	0.3860	0.2560
	2	0.4683	0.3720	0.4010	0.2620
	3	0.4740	0.3840	0.4040	0.2860
	4	0.4777	0.3710	0.4210	0.3030
	5	0.4693	0.3670	0.4203	0.3170
PROX5	1	0.6097	0.5220	0.4947	0.5160
	2	0.6193	0.4880	0.5170	0.4900
	3	0.6060	0.4650	0.5080	0.4630
	4	0.6110	0.4510	0.5187	0.4470
	5	0.6173	0.4240	0.5277	0.4230

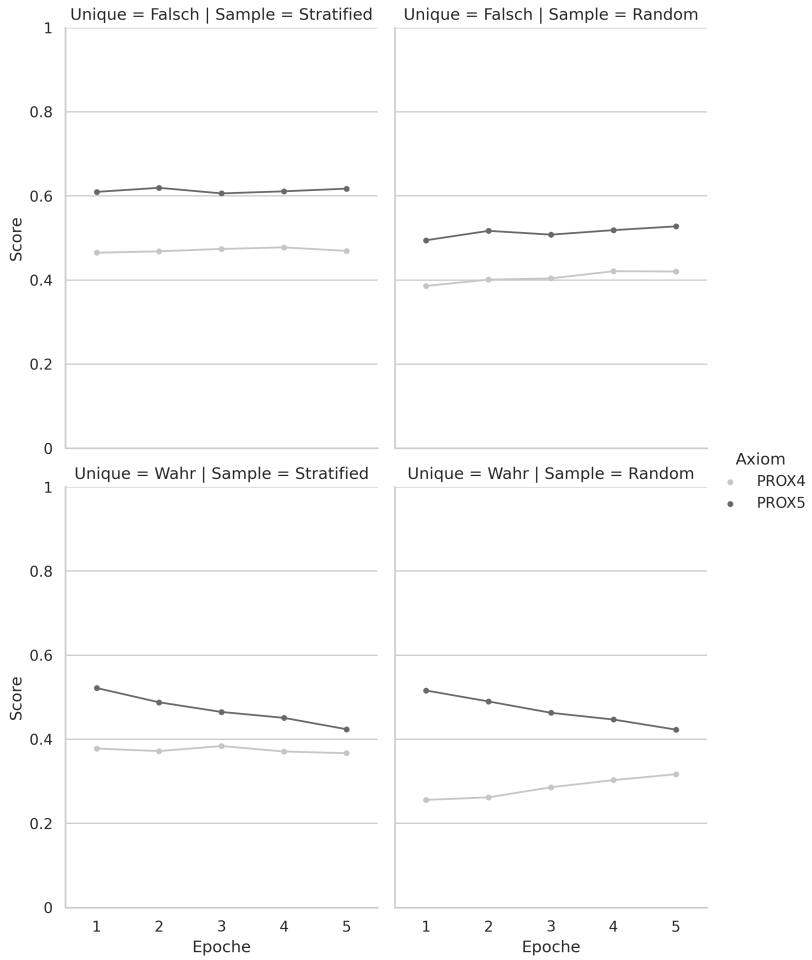

Abbildung A.35: Die Accuracy Ergebnisse von PROX4 und PROX5 mit jeweils 25.000 stratifizierten Trainingsdaten im Experiment 4.

A.8 Ergebnisse Experiment 5

Nachfolgend eine tabellarische und grafische Darstellung der Ergebnisse des Multi-Task-Learnings der Ranking Aufgabe mit 100.000 Trainingsdaten und den Axiom Aufgaben M-TDC, LB1, REG, ANTI-REG, PROX4 und PROX5 mit jeweils 25.000 Trainingsdaten. Die Axiome werden bezüglich Accurcy evaluiert.

Tabelle A.37: Die Ergebnisse der Ranking Aufgabe mit 100.000 Trainingsdaten im Experiment 5. Die besten Ergebnisse sind fett markiert.

	TREC 2019			**TREC 2020**		
Epoche	NDCG@10	P@1	MRR@10	NDCG@10	P@1	MRR@10
1	**0.6843**	**0.7674**	**0.8547**	**0.6551**	**0.6481**	0.7666
2	0.6632	0.6744	0.7845	0.6125	0.6296	**0.7687**
3	0.6524	0.6047	0.7425	0.5923	0.6481	0.7610
4	0.5804	0.5581	0.6846	0.5241	0.3704	0.5840
5	0.5672	0.5349	0.6622	0.4890	0.3704	0.5563

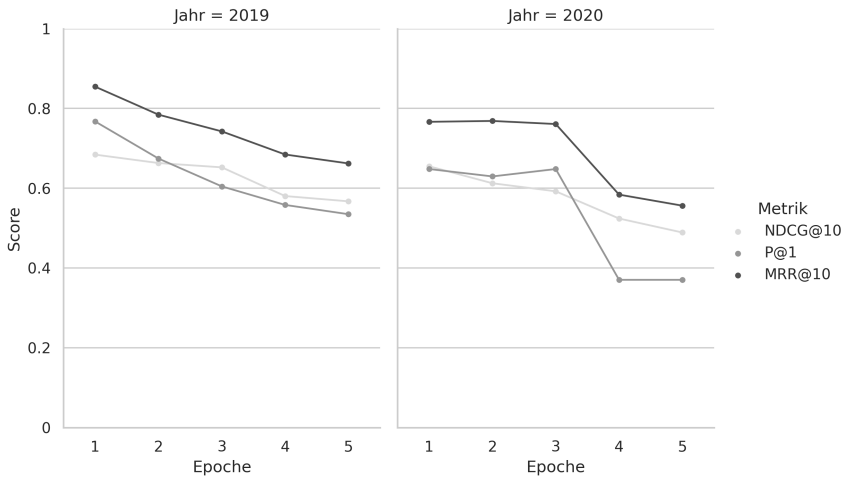

Abbildung A.36: Die Ergebnisse der Ranking Aufgabe mit 100.000 Trainingsdaten im Experiment 5.

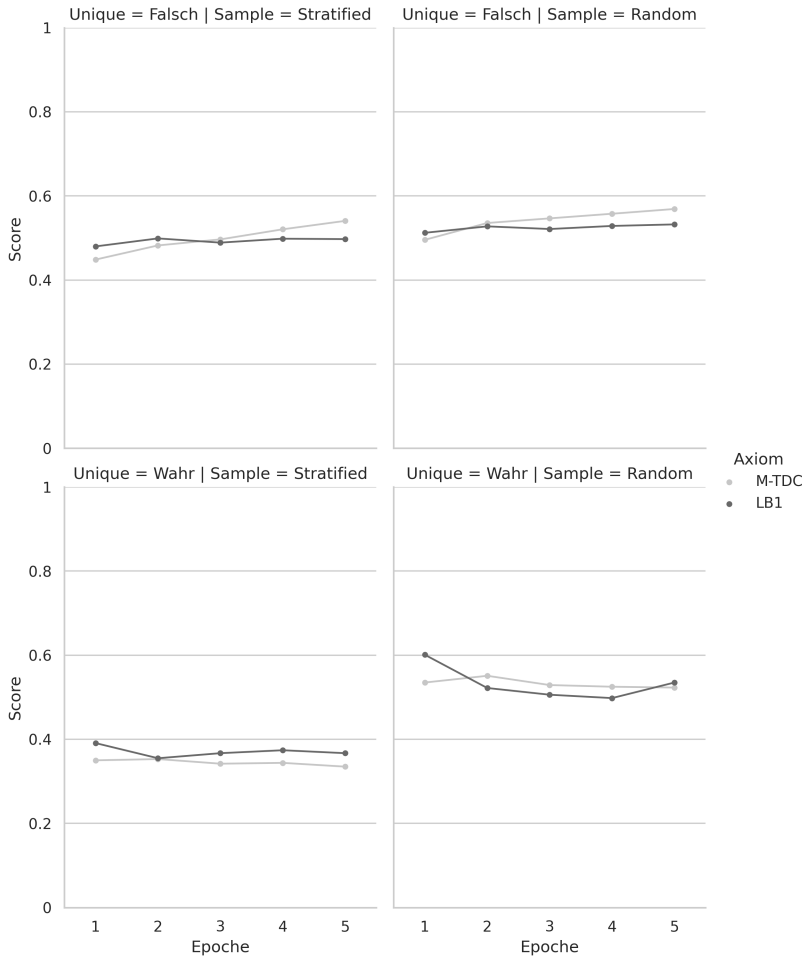

Abbildung A.37: Die Accuracy Ergebnisse von M-TDC und LB1 mit jeweils 25.000 stratifizierten Trainingsdaten im Experiment 5.

Tabelle A.38: Die Accuracy Ergebnisse der M-TDC, LB1, REG, ANTI-REG, PROX4 und PROX5 Axiom Aufgabe mit 25.000 Trainingsdaten im Experiment 5.

		Stratified		**Random**	
Axiom	**Epoche**	None-Unique	Unique	None-Unique	Unique
M-TDC	1	0.4487	0.3500	0.4957	0.5350
	2	0.4823	0.3530	0.5357	0.5510
	3	0.4967	0.3420	0.5467	0.5290
	4	0.5207	0.3440	0.5577	0.5250
	5	0.5407	0.3350	0.5690	0.5230
LB1	1	0.4800	0.3910	0.5123	0.6010
	2	0.4990	0.3550	0.5277	0.5220
	3	0.4890	0.3670	0.5213	0.5060
	4	0.4983	0.3740	0.5287	0.4980
	5	0.4973	0.3670	0.5323	0.5350
REG	1	0.4307	0.3710	0.4553	0.3660
	2	0.4470	0.3700	0.4753	0.3680
	3	0.4650	0.3700	0.4810	0.3680
	4	0.4717	0.3730	0.4833	0.3690
	5	0.4720	0.3730	0.4850	0.3680
ANTI-REG	1	0.4040	0.3330	0.4063	0.3420
	2	0.4360	0.3300	0.4377	0.3290
	3	0.4477	0.3330	0.4490	0.3360
	4	0.4507	0.3350	0.4530	0.3380
	5	0.4587	0.3320	0.4633	0.3440
PROX4	1	0.4600	0.3640	0.4193	0.2830
	2	0.4640	0.3600	0.4100	0.2380
	3	0.4677	0.3610	0.4127	0.2510
	4	0.4717	0.3690	0.4223	0.2880
	5	0.4737	0.3650	0.4227	0.2800
PROX5	1	0.5923	0.4710	0.5073	0.4640
	2	0.6010	0.4740	0.4997	0.4670
	3	0.6123	0.4510	0.5183	0.4480
	4	0.6020	0.4490	0.5127	0.4430
	5	0.6167	0.4650	0.5267	0.4600

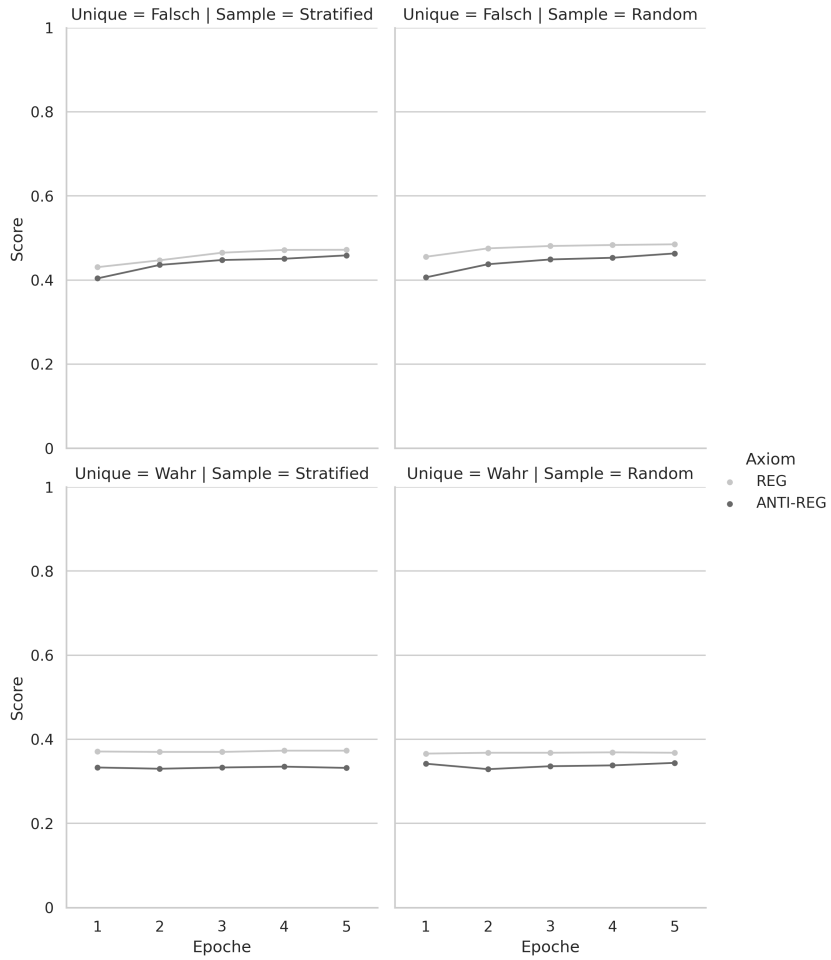

Abbildung A.38: Die Accuracy Ergebnisse von REG und ANTI-REG mit jeweils 25.000 stratifizierten Trainingsdaten im Experiment 5.

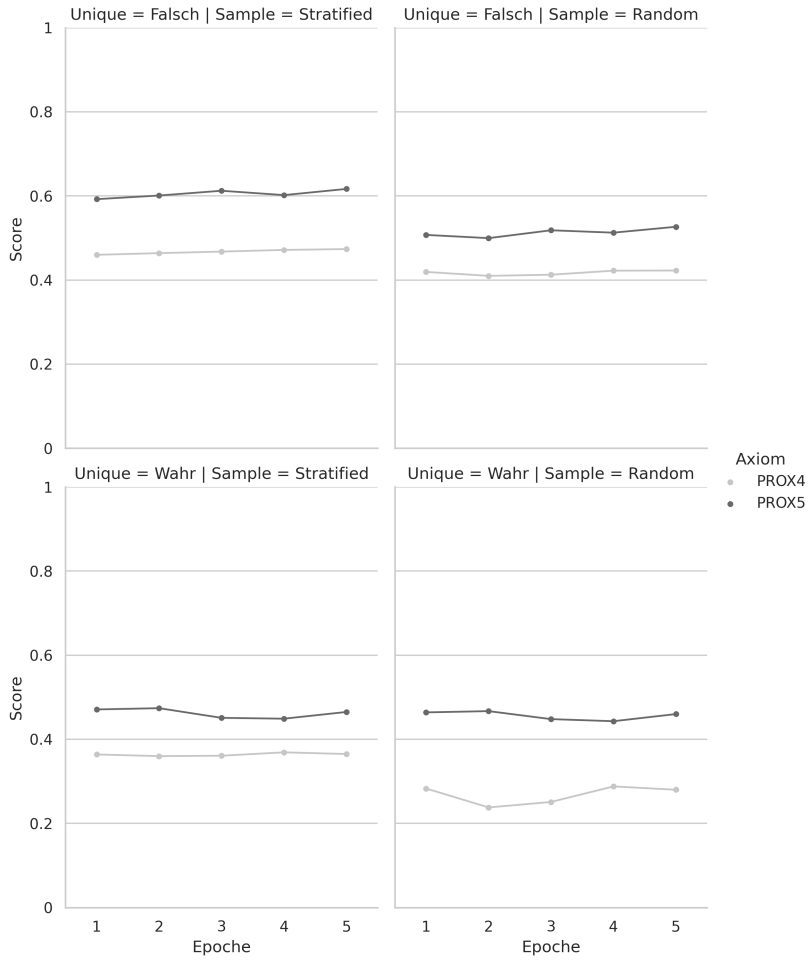

Abbildung A.39: Die Accuracy Ergebnisse von PROX4 und PROX5 mit jeweils 25.000 stratifizierten Trainingsdaten im Experiment 5.

Literaturverzeichnis

[1] Wasi Uddin Ahmad, Kai-Wei Chang, and Hongning Wang. Multi-task learning for document ranking and query suggestion. In *International Conference on Learning Representations*, 2018.

[2] Alon Altman and Moshe Tennenholtz. Ranking systems: the pagerank axioms. In *Proceedings of the 6th ACM conference on Electronic commerce*, pages 1–8, 2005.

[3] Enrique Amigó, Julio Gonzalo, and Felisa Verdejo. A general evaluation measure for document organization tasks. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 643–652, 2013.

[4] Mozhdeh Ariannezhad, Ali Montazeralghaem, Hamed Zamani, and Azadeh Shakery. Improving retrieval performance for verbose queries via axiomatic analysis of term discrimination heuristic. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1201–1204, 2017.

[5] Siddhant Arora and Andrew Yates. Investigating retrieval method selection with axiomatic features. *arXiv preprint arXiv:1904.05737*, 2019.

[6] Anne Aula. Query formulation in web information search. In *ICWI*, pages 403–410, 2003.

[7] Richard K Belew and Richard K Belew. *Finding out about: a cognitive perspective on search engine technology and the WWW*. Cambridge University Press, 2000.

[8] Ann L Brown and Mary Jo Kane. Preschool children can learn to transfer: Learning to learn and learning from example. *Cognitive psychology*, 20(4): 493–523, 1988.

[9] Peter D Bruza and Theo WC Huibers. Investigating aboutness axioms using information fields. In *SIGIR94*, pages 112–121. Springer, 1994.

[10] Luca Busin and Stefano Mizzaro. Axiometrics: An axiomatic approach to information retrieval effectiveness metrics. In *Proceedings of the 2013 Conference on the Theory of Information Retrieval*, pages 22–29, 2013.

[11] Arthur Câmara and Claudia Hauff. Diagnosing bert with retrieval heuristics. *Advances in Information Retrieval*, 12035:605, 2020.

[12] UP Cambridge. Online edition (c) 2009 cambridge up an introduction to information retrieval christopher d, 2009.

[13] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[14] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, and Daniel Campos. Overview of the trec 2020 deep learning track.

[15] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M Voorhees. Overview of the trec 2019 deep learning track. *arXiv preprint arXiv:2003.07820*, 2020.

[16] Michael Crawshaw. Multi-task learning with deep neural networks: A survey. *arXiv preprint arXiv:2009.09796*, 2020.

[17] Ronan Cummins and Colm O'Riordan. An axiomatic comparison of learned term-weighting schemes in information retrieval: clarifications and extensions. *Artificial Intelligence Review*, 28(1):51–68, 2007.

[18] Ronan Cummins and Colm O'Riordan. Analysing ranking functions in information retrieval using constraints. *Information extraction from the Internet*, 2009.

[19] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[20] Laura Dietz, Manisha Verma, Filip Radlinski, and Nick Craswell. Trec complex answer retrieval overview. In *TREC*, 2017.

[21] Fan Ding and Bin Wang. An axiomatic approach to exploit term dependencies in language model. In *Asia Information Retrieval Symposium*, pages 586–591. Springer, 2008.

[22] Dudenredaktion (o. J.). äxiomäuf duden online., 2021. URL `https://www.duden.de/node/11843/revision/11870`.

[23] Hui Fang and ChengXiang Zhai. An exploration of axiomatic approaches to information retrieval. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 480–487, 2005.

[24] Hui Fang and ChengXiang Zhai. Semantic term matching in axiomatic approaches to information retrieval. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 115–122, 2006.

[25] Hui Fang, Tao Tao, and ChengXiang Zhai. A formal study of information retrieval heuristics. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 49–56, 2004.

[26] Hui Fang, Tao Tao, and Chengxiang Zhai. Diagnostic evaluation of information retrieval models. *ACM Transactions on Information Systems (TOIS)*, 29(2):1–42, 2011.

[27] Julia Festman. Learning and processing multiple languages: The more the easier? *Language Learning*, 71(S1):121–162, 2021.

[28] Christopher Fifty, Ehsan Amid, Zhe Zhao, Tianhe Yu, Rohan Anil, and Chelsea Finn. Efficiently identifying task groupings for multi-task learning. *arXiv preprint arXiv:2109.04617*, 2021.

[29] George Forman and Martin Scholz. Apples-to-apples in cross-validation studies: pitfalls in classifier performance measurement. *Acm Sigkdd Explorations Newsletter*, 12(1):49–57, 2010.

[30] Shima Gerani, ChengXiang Zhai, and Fabio Crestani. Score transformation in linear combination for multi-criteria relevance ranking. In *European Conference on Information Retrieval*, pages 256–267. Springer, 2012.

[31] Rayid Ghani, Rosie Jones, and Dunja Mladenic. Building minority language corpora by learning to generate web search queries. *Knowledge and information systems*, 7(1):56–83, 2005.

[32] Pavel Golik, Patrick Doetsch, and Hermann Ney. Cross-entropy vs. squared error training: a theoretical and experimental comparison. In *Interspeech*, volume 13, pages 1756–1760, 2013.

[33] Sreenivas Gollapudi and Aneesh Sharma. An axiomatic approach for result diversification. In *Proceedings of the 18th international conference on World wide web*, pages 381–390, 2009.

[34] Matthias Hagen, Michael Völske, Steve Göring, and Benno Stein. Axiomatic result re-ranking. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 721–730, 2016.

[35] Sepp Hochreiter, Yoshua Bengio, Paolo Frasconi, Jürgen Schmidhuber, et al. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies, 2001.

[36] Mohammad Hossin and Md Nasir Sulaiman. A review on evaluation metrics for data classification evaluations. *International journal of data mining & knowledge management process*, 5(2):1, 2015.

[37] Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems (TOIS)*, 20 (4):422–446, 2002.

[38] Anjali Ganesh Jivani et al. A comparative study of stemming algorithms. *Int. J. Comp. Tech. Appl*, 2(6):1930–1938, 2011.

[39] Maryam Karimzadehgan and ChengXiang Zhai. Axiomatic analysis of translation language model for information retrieval. In *European Conference on Information Retrieval*, pages 268–280. Springer, 2012.

[40] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[41] Hang Li. A short introduction to learning to rank. *IEICE TRANSACTIONS on Information and Systems*, 94(10):1854–1862, 2011.

[42] Jimmy Lin. The neural hype and comparisons against weak baselines. In *ACM SIGIR Forum*, volume 52, pages 40–51. ACM New York, NY, USA, 2019.

[43] Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. Pyserini: A python toolkit for reproducible information retrieval research with sparse and dense representations. In *Proceedings of the 44th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2021)*, 2021.

[44] Tie-Yan Liu. Learning to rank for information retrieval. 2011.

[45] Xiaodong Liu, Kevin Duh, and Jianfeng Gao. Stochastic answer networks for natural language inference. *arXiv preprint arXiv:1804.07888*, 2018.

[46] Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. Multi-task deep neural networks for natural language understanding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4487–4496, Florence, Italy, July 2019. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/P19-1441`.

[47] Yang Liu and Mirella Lapata. Text summarization with pretrained encoders. *arXiv preprint arXiv:1908.08345*, 2019.

[48] Yuanhua Lv and ChengXiang Zhai. Lower-bounding term frequency normalization. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 7–16, 2011.

[49] Brian P. Mc Cune, Richard M. Tong, Jeffrey S. Dean, and Daniel G. Shapiro. Rubric: A system for rule-based information retrieval. *IEEE Transactions on Software Engineering*, (9):939–945, 1985.

[50] Carlo Meghini, Fabrizio Sebastiani, Umberto Straccia, and Costantino Thanos. A model of information retrieval based on a terminological logic. In *Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 298–307, 1993.

[51] Bhaskar Mitra, Fernando Diaz, and Nick Craswell. Learning to match using local and distributed representations of text for web search. In *Proceedings of the 26th International Conference on World Wide Web*, pages 1291–1299, 2017.

[52] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. Ms marco: A human generated machine reading comprehension dataset. In *CoCo@ NIPS*, 2016.

[53] Kyosuke Nishida, Itsumi Saito, Atsushi Otsuka, Hisako Asano, and Junji Tomita. Retrieve-and-read: Multi-task learning of information retrieval and reading comprehension. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 647–656, 2018.

[54] Rodrigo Nogueira, Wei Yang, Kyunghyun Cho, and Jimmy Lin. Multi-stage document ranking with bert. *arXiv preprint arXiv:1910.14424*, 2019.

[55] Rodrigo Nogueira, Zhiying Jiang, and Jimmy Lin. Document ranking with a pretrained sequence-to-sequence model. *arXiv preprint arXiv:2003.06713*, 2020.

[56] Christopher Olston and Marc Najork. *Web crawling*. Now Publishers Inc, 2010.

[57] Bing Pan, Helene Hembrooke, Thorsten Joachims, Lori Lorigo, Geri Gay, and Laura Granka. In google we trust: Users' decisions on rank, position, and relevance. *Journal of computer-mediated communication*, 12(3):801–823, 2007.

[58] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018.

[59] Telmo Pires, Eva Schlinger, and Dan Garrette. How multilingual is multilingual bert? *arXiv preprint arXiv:1906.01502*, 2019.

[60] Martin F Porter. An algorithm for suffix stripping. *Program*, 1980.

[61] Yifan Qiao, Chenyan Xiong, Zhenghao Liu, and Zhiyuan Liu. Understanding the behaviors of bert in ranking. *arXiv preprint arXiv:1904.07531*, 2019.

[62] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding with unsupervised learning. 2018.

[63] Filip Radlinski and Thorsten Joachims. Query chains: learning to rank from implicit feedback. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 239–248, 2005.

[64] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL `http://jmlr.org/papers/v21/20-074.html`.

[65] Razieh Rahimi, Azadeh Shakery, and Irwin King. Axiomatic analysis of cross-language information retrieval. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 1875–1878, 2014.

[66] Daniël Rennings, Felipe Moraes, and Claudia Hauff. An axiomatic approach to diagnosing neural ir models. In *European Conference on Information Retrieval*, pages 489–503. Springer, 2019.

[67] Stephen Robertson and Hugo Zaragoza. *The probabilistic relevance framework: BM25 and beyond.* Now Publishers Inc, 2009.

[68] Corby Rosset, Bhaskar Mitra, Chenyan Xiong, Nick Craswell, Xia Song, and Saurabh Tiwary. An axiomatic approach to regularizing neural ranking models. In *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval*, pages 981–984, 2019.

[69] Hinrich Schütze, Christopher D Manning, and Prabhakar Raghavan. *Introduction to information retrieval*, volume 39. Cambridge University Press Cambridge, 2008.

[70] Shuming Shi, Ji-Rong Wen, Qing Yu, Ruihua Song, and Wei-Ying Ma. Gravitation-based model for information retrieval. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 488–495, 2005.

[71] Catarina Silva and Bernardete Ribeiro. The importance of stop word removal on recall values in text categorization. In *Proceedings of the International Joint Conference on Neural Networks, 2003.*, volume 3, pages 1661–1666. IEEE, 2003.

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

[73] Simon Tong, Uri Lerner, Amit Singhal, Paul Haahr, and Steven Baker. Locating meaningful stopwords or stop-phrases in keyword-based retrieval systems, October 22 2019. US Patent 10,452,718.

[74] Cornelis J Van Rijsbergen. A new theoretical framework for information retrieval. In *Acm Sigir Forum*, volume 21, pages 23–29. ACM New York, NY, USA, 1986.

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

[76] Michael Völske, Alexander Bondarenko, Maik Fröbe, Matthias Hagen, Benno Stein, Jaspreet Singh, and Avishek Anand. Towards axiomatic explanations for neural ranking models. *arXiv preprint arXiv:2106.08019*, 2021.

[77] Soroush Vosoughi, Prashanth Vijayaraghavan, and Deb Roy. Tweet2vec: Learning tweet embeddings using character-level cnn-lstm encoder-decoder. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 1041–1044, 2016.

[78] W John Wilbur and Karl Sirotkin. The automatic identification of stop words. *Journal of information science*, 18(1):45–55, 1992.

[79] Peter Willett. The porter stemming algorithm: then and now. *Program*, 2006.

[80] Hao Wu and Hui Fang. Relation based term weighting regularization. In *European Conference on Information Retrieval*, pages 109–120. Springer, 2012.

[81] Wei Yang, Kuang Lu, Peilin Yang, and Jimmy Lin. Critically examining the"neural hype"weak baselines and the additivity of effectiveness gains from neural ranking models. In *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval*, pages 1129–1132, 2019.

[82] Xue Ying. An overview of overfitting and its solutions. In *Journal of Physics: Conference Series*, volume 1168. IOP Publishing, 2019.

[83] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021.

[84] Dell Zhang, Robert Mao, Haitao Li, and Joanne Mao. How to count thumb-ups and thumb-downs: user-rating based ranking of items from an axiomatic perspective. In *Conference on the Theory of Information Retrieval*, pages 238–249. Springer, 2011.

[85] Wei Zheng and Hui Fang. Query aspect based term weighting regularization in information retrieval. In *European Conference on Information Retrieval*, pages 344–356. Springer, 2010.