TECHNISCHE UNIVERSITÄT
CHEMNITZ

Automatisiertes Lifelog Moment Retrieval basierend auf Bildsegmentierung und Ähnlichkeitsmaßen

Masterarbeit

Fakultät für Informatik
Professur Medieninformatik

Eingereicht von Stefan Taubert
Matrikelnummer: 369897
Chemnitz, den 10. Oktober 2019

Prüfer: Prof. Dr. Maximilian Eibl
Betreuer: M. Sc. Stefan Kahl

Taubert, Stefan
Automatisiertes Lifelog Moment Retrieval basierend auf Bildsegmentierung und Ähnlichkeitsmaßen
Masterarbeit, Fakultät für Informatik
Technische Universität Chemnitz, Oktober 2019

Zusammenfassung

Die vorliegende Arbeit stellt verschiedene Verfahren des automatisierten Lifelog Moment Retrievals vor. Dazu war im Rahmen eines Wettbewerbes ein Datenset mit Lifelog-Daten von 29 Tagen und jeweils zehn Test- und Trainingsanfragen gegeben. Nach der Darlegung bereits existierender Verfahren und der Analyse des Datensets wurden vier selbsterstellte Retrieval-Verfahren vorgestellt. Dabei fanden in der finalen Variante Wortvektoren Verwendung, um Ähnlichkeitswerten zwischen Anfragen und Bildern zu berechnen. Die Anzahl an Bildern wurde mit Hilfe einer auf Farbhistogrammen basierenden Bildsegmentierung verringert, welche 93,23% der Aufnahmen entfernte und dabei einen Cluster Recall von 89,37% erreichte. Die gegebenen Metadaten wurden um zusätzliche Annotationen mit Hilfe von Detectron und YOLOv3 ergänzt. Es resultierten 13 Labelarten, welche es ermöglichten, unterschiedliche Bereiche des Lifelogs zu beschreiben. Alle entwickelten Verfahren ließen sich komplett automatisiert auf generische Anfragen anwenden. Es erfolgte außerdem eine Evaluation für einen Ansatz, welcher auf maschinellem Lernen basierte. Der höchste F1 für die Trainingsanfragen betrug 33,13% und für die Testanfragen wurde ein maximaler F1 von 11,7% erzielt.

Alle Internetquellen wurden am 10. Oktober 2019 zum letzten Mal auf Aktualität geprüft.

Inhaltsverzeichnis

Inhaltsverzeichnis . . . . . **i**

Abbildungsverzeichnis . . . . . **iv**

Tabellenverzeichnis . . . . . **vi**

1 Motivation **1**
1.1 Gedächtnis 2
1.1.1 Kodierung 2
1.1.2 Speicherung 2
1.1.3 Erinnerung 3
1.1.4 Lifelogs als Gedächtnisverstärkung 3
1.2 Kritische Gesichtspunkte von Lifelogging 4
1.2.1 Rechtlicher Aspekt 4
1.2.2 Gesundheitlicher Aspekt 5
1.2.3 Unterschied von Daten und Wissen 5
1.3 ImageCLEF 5
1.4 Aufbau der Arbeit 5

2 State of the Art **6**
2.1 Bildverarbeitung 6
2.2 Bildsegmentierung 7
2.2.1 Ansätze 7
2.2.2 Bildvergleich 7
2.2.3 Clustering 8
2.2.4 Keyframe Selection 8
2.3 Metadatenverarbeitung 9
2.4 Anfragenverarbeitung 9
2.4.1 Interaktiver Ansatz 9
2.4.2 Automatischer Ansatz 9
2.5 Filterung 10
2.6 Retrieval 10
2.7 Diversifikation 11
2.7.1 Automatischer Ansatz 12
2.7.2 Human in the Loop Ansatz 12

3 Analyse des Lifelogs . . . 14
3.1 Autografische Bilder . . . 14
3.2 Persönliche Bilder . . . 16
3.3 Visuelle Konzepte . . . 17
3.4 Minutenbasierte Ereignisliste . . . 19
3.5 Trainingsanfragen . . . 22
3.5.1 Ground Truth . . . 22
3.5.2 Erste Anfrage . . . 23
3.5.3 Zweite Anfrage . . . 23
3.5.4 Vierte Anfrage . . . 24
3.5.5 Restliche Anfragen . . . 24
3.6 Testanfragen . . . 26
3.6.1 Erste Anfrage . . . 26
3.6.2 Vierte Anfrage . . . 26
3.6.3 Neunte Anfrage . . . 27
3.6.4 Restliche Anfragen . . . 27
3.7 Vergleich von Trainings- und Testanfragen . . . 28

4 Entwicklung eines Verfahrens . . . 29
4.1 Ressourcen . . . 29
4.1.1 Python als Entwicklungsumgebung . . . 29
4.1.2 GloVe Wortvektoren . . . 30
4.1.3 Objekterkennungssysteme, CNNs und Datensets . . . 30
4.2 Vorverarbeitung . . . 31
4.2.1 Definition der Labelarten . . . 31
4.2.2 Berechnung der Inversen Dokumentfrequenz . . . 33
4.2.3 Bildverarbeitung . . . 33
4.2.4 Anfragenverarbeitung . . . 37
4.3 Prediction . . . 38
4.3.1 Vektorvergleich . . . 39
4.3.2 Ähnlichkeitsberechnungen . . . 42
4.3.3 Auswahl repräsentativer Bilder . . . 43
4.3.4 Durchführung der XGB-basierten Prediction . . . 44
4.4 Postprocessing . . . 45
4.4.1 Sortierung . . . 45
4.4.2 Anordnung . . . 45

5 Evaluation . . . 46
5.1 Metriken . . . 47
5.1.1 CR@X . . . 47
5.1.2 P@X . . . 47
5.1.3 F1@X . . . 48
5.1.4 Überprüfung der Korrektheit der Implementierung . . . 48
5.2 Submissions . . . 48

5.3 Zufallsbasierte Modellarchitektur . . . 48
5.4 Regelbasierte Modellarchitektur . . . 49
5.5 Segmentbasierte Modellarchitektur . . . 50
5.5.1 Erster Durchgang . . . 51
5.5.2 Zweiter Durchgang . . . 52
5.5.3 Optimale Parameter-Einstellungen . . . 53
5.6 Vektorbasierte Modellarchitektur . . . 54
5.6.1 Funktionsweise . . . 54
5.6.2 Baseline-Experimente . . . 55
5.6.3 Experimente zur Segmentierung . . . 56
5.6.4 Experimente zur Auswahl der Konzepte . . . 56
5.6.5 Experimente zur Inversen Dokumentfrequenz . . . 57
5.6.6 Experimente zur Intensivierung . . . 58
5.6.7 Experimente zur Anordnung . . . 58
5.6.8 Experimente zum Vektorvergleich . . . 59
5.6.9 Experimente zur Labelartkombination . . . 59
5.6.10 Experimente mit maschinellen Lernen . . . 60
5.6.11 Schwächen der Modellarchitektur . . . 61
5.7 Poolbasierte Modellarchitektur . . . 61
5.7.1 Prototyp . . . 61
5.7.2 Parameter der Architektur . . . 62
5.7.3 Aufbau und Umsetzung der Experimente . . . 62
5.7.4 Auswertung der Experimente . . . 64
5.7.5 Experimente des Labelart Modelltyps . . . 65
5.7.6 Submission 1 . . . 66
5.7.7 Experimente des VK Modelltyps . . . 67
5.7.8 Submissions 2, 3, 4 und 5 . . . 68
5.7.9 Experimente mit XGBoost . . . 69
5.7.10 Submissions 6 und 7 . . . 70
5.7.11 Experimente des VK MBT Modelltyps . . . 70
5.7.12 Submissions 8 und 9 . . . 71
5.7.13 Experimente des Erweiterten VK Modelltyps . . . 72
5.7.14 Submissions 10 und 11 . . . 73
5.7.15 Experimente des Erweiterten VK MBT Modelltyps . . . 74
5.7.16 Submissions 12 und 13 . . . 75
5.7.17 Fortführung der Experimente des Erweiterten VK Modelltyps 75
5.7.18 Submission 14 . . . 77
5.8 Vergleich mit anderen Wettbewerbsteilnehmern . . . 80

6 Zusammenfassung und Ausblick . . . 82

Literaturverzeichnis . . . 85

Anhang . . . 94

Abbildungsverzeichnis

1.1 Beispiele für Lifelogging Kameras 2

3.1 Auszüge aus einigen autografischen Bildern 15
3.2 Durchschnittliche Anzahl an autografischen Bildern pro Minute . . . 15
3.3 Auszüge aus einigen persönlichen Bildern 16
3.4 Word Clouds für Attribute und Konzepte 18
3.5 Word Cloud für Kategorien 18
3.6 Bilder mit falsch annotierten Konzepten 19
3.7 Diagramme zur Herzrate und zum Blutzuckerspiegel 21
3.8 Diagramme zur Aktivität und zur Zeitzone 21
3.9 Auszüge aus fehlerhaften Metadaten 22
3.10 Ausgewählte Bilder zur ersten Trainingsanfrage 23
3.11 Ausgewählte Bilder zur zweiten Trainingsanfrage 24
3.12 Ausgewählte Bilder zur vierten Trainingsanfrage 25
3.13 Ausgewählte Bilder zur vierten Testanfrage 27

4.1 Veranschaulichung des Clustering Prozesses 34
4.2 Erster Schritt der Segmentierung 35
4.3 Zweiter Schritt der Segmentierung 36
4.4 Dritter Schritt der Segmentierung 36
4.5 Vorletzter Schritt der Segmentierung 36
4.6 Letzter Schritt der Segmentierung 36

5.1 Taxonomie und Relationen der Evaluationsbegriffe 46
5.2 Ergebnisse der Experimente der zufallsbasierten Modellarchitektur . . 49
5.3 Ergebnisse der Experimente der regelbasierten Modellarchitektur . . . 50
5.4 Ergebnisse 1. Durchgang der segmentbasierten Modellarchitektur . . . 51
5.5 Ergebnisse 2. Durchgang der segmentbasierten Modellarchitektur . . . 53
5.6 Ergebnisse des segmentbasierten Baseline-Modell 55
5.7 Ergebnisse der Experimente zur Segmentierung 56
5.8 Ergebnisse der Experimente zur Auswahl der Konzepte 57
5.9 Ergebnisse der Experimente zur Inversen Dokumentfrequenz 57
5.10 Ergebnisse der Experimente zur Intensivierung 58
5.11 Ergebnisse der Experimente zur Anordnung 59
5.12 Ergebnisse der Experimente zum Vektorvergleich 59
5.13 Ergebnisse der Experimente zur Labelartkombination 60
5.14 Aufbau der Experimente für die poolbasierte Modellarchitektur . . . 63

5.15 Übersicht über die Konfigurationen der Auswertung 65
5.16 Ergebnisse der Experimente des Labelart Modelltyps 66
5.17 Ergebnisse der 1. Submission 67
5.18 Ergebnisse der Experimente des VK Modelltyps 67
5.19 Ergebnisse der 2., 3., 4. und 5. Submission 69
5.20 Ergebnisse der Experimente mit XGBoost 69
5.21 Ergebnisse der 7. Submission 70
5.22 Ergebnisse der Experimente des VK MBT Modelltyps 71
5.23 Ergebnisse der 8. und 9. Submission 72
5.24 Ergebnisse der Experimente des Erweiterten VK Modelltyps 73
5.25 Ergebnisse der 10. und 11. Submission 73
5.26 Ergebnisse der Experimente des Erweiterten VK MBT Modelltyps . . 74
5.27 Ergebnisse der 12. und 13. Submission 75
5.28 Ergebnisse der fortführenden Experimente des Erw. VK Modelltyps . 76
5.29 Gewichtungen der Labelarten für die besten Trainings-Modelle 76
5.30 Ergebnisse der 14. Submission 77
5.31 Auszüge aus vorhergesagten Bildern des finalen Trainings-Modells . . 78
5.32 Auszüge aus vorhergesagten Bildern des finalen Test-Modells 79
5.33 Leaderboard des diesjährigen Wettbewerbes 81
5.34 Anzahl an Submissions je Teilnehmer 81

Tabellenverzeichnis

4.1 Überblick über die verwendeten Labelarten 32

5.1 Anzahl an erzeugten Segmenten nach der Segmentierung 52

1 Motivation

Ein Lifelog ist eine Ansammlung von strukturierten Daten aus dem Leben einer Person [40][30, S. 1]. Diese durch verschiedenste Sensoren erfassten Daten sind zum Beispiel die Herzrate, die elektrodermale Aktivität, die Stimmung, die Temperatur, der Blutzuckerspiegel, die geografischen Koordinaten oder Momentaufnahmen der Umgebung, in Form von Bildern [19, S. 53]. Viele dieser Sensoren befinden sich in heutigen Smartphones und Smartwatches, aber auch in spezialisierten Geräten, wie in tragbaren Kameras. Anhand von Bildern lassen sich durch Methoden maschinellen Lernens weitere Metadaten extrahieren, wie beispielsweise die Objekte, die auf dem Bild zu sehen sind oder, ob sich der Lifelogger in einem Gebäude oder im Freien befindet. Je mehr Daten zur Verfügung stehen, desto besser lässt sich im Allgemeinen das Leben des Lifeloggers analysieren. In Abbildung 1.1 sind tragbare Kameras aus den Jahren 1998 (Wearable Wireless Webcam), 2004 (Microsoft SenseCam [45]), 2006 (Handy als Lifeloggingkamera [98]) und 2013 (Narrative Clip) dargestellt [37].
Privat dient ein Lifelog als Speicher für Erinnerungen, um diese bei Bedarf erneut ins Gedächtnis zu rufen und, um diese mit anderen zu teilen. Weitere Beispiele sind, es herauszufinden, wann man zuletzt eine bestimmte Person gesehen hat oder verloren gegangene Gegenständen wiederzufinden [78, S. 72f.]. Zudem können sie nützlich sein, um Diskussionen erneut durchzugehen oder um die eigenen Gewohnheiten oder die Produktivität auf Arbeit zu analysieren [19, S. 52]. Eine wichtige Rolle spielen die Lifelogs bei Patienten mit leichter kognitiver Beeinträchtigung (MCI), Gedächtnisschwund oder neurodegenerativen Erkrankungen wie der Alzheimer-Krankheit [57]. Die Patienten können selbstständig den erlebten Tag und dessen wichtige Momente durchgehen und somit ihre Gedächtnisleistung verbessern.
Dritte, wie zum Beispiel Ärzte, können die Lifelogs ebenfalls verwenden, indem sie Auswertungen durchführen und analysieren, wie oft ein Patient Sport gemacht hat oder wie oft er etwas gegessen hat. Anhand dieser Daten könnte ein Arzt individuellere Empfehlungen abgeben. Psychologen könnten auch einige Erkenntnisse aus den Lifelogs gewinnen, indem sie die Verhaltensmuster eines Patienten untersuchen. Dinge, die der Lifelogger während seiner Einkäufe länger angesehen hat, eignen sich für Werbefirmen, um ein Werbeprofil zu erstellen.
Aufgrund der verschiedenen Verhaltensweisen von Menschen und der großen Datenmengen ist es schwierig diese effektiv zu katalogisieren, zu managen und eine generalisierte Auswertung dieser Daten zu erstellen [41]. Da das Verarbeiten von den großen Datenmengen per Hand jedoch sehr aufwändig ist, eignen sich vor allem Computersysteme für diesen Zweck. Das Ziel der vorliegenden Arbeit ist es, computergestützte Verfahren vorzustellen, welche gesammelte Daten aus dem Leben einer Person mit vordefinierten Abfragen auswerten können.

Abbildung 1.1: Dargestellt sind Lifelogging Kameras (v.l.n.r) aus den Jahren 1998, 2004, 2006 und 2013 [37].

1.1 Gedächtnis

Eine wesentliche Anwendung von Lifelogs besteht darin als Gedächtnisprothese und zur Gedächtnisverstärkung zu dienen [55, 76]. Daher lohnt es sich zunächst zu untersuchen, wie das menschliche Gedächtnis funktioniert, um zu wissen, an welchen Stellen Lifelogs ansetzen können, um dieses zu stärken. Das Gedächtnis hat die drei Phasen Kodierung, Speicherung und Erinnerung [35].

1.1.1 Kodierung

Die Kodierung eines Events umfasst faktenbasierte Repräsentationen, kontextuelle Verbindungen, Charakterisierungen des raumzeitlichen Kontextes und die Referenz zu der Person, die es erlebt hat [27]. Daher ist es für eine Person einfacher sich an ein Event zu erinnern, wenn sie sich in dem Raum befindet, in welchem dieses stattgefunden hat [83]. Besonders die Zeit und der Ort haben sich als die wichtigsten Faktoren des episodischen Gedächtnisses gezeigt [17]. Eine weitere Rolle spielt wie viel Aufmerksamkeit eine Person bei der Kodierung schenkt. Untersuchungen zeigten, dass die Personen, die einem Moment wenig Aufmerksamkeit gaben, weil sie sich beispielsweise auf mehrere Dinge gleichzeitig konzentrierten, sich signifikant schlechter an diesen erinnerten [65], wohingegen die Personen sich im Umkehrschluss an aufmerksamere Momente besser erinnern konnten [20].

1.1.2 Speicherung

Die eigenen Erfahrungen, welche auf den Sinnen basieren, lassen sich im Gedächtnis in atomare Events beziehungsweise Episoden einteilen und speichern, wobei diese unterschiedlicher Länge sein können [41]. Längere Episoden sind beispielsweise das Schauen von Fußball oder ein Meeting, wohingegen kürzere Episoden das Sehen eines Freundes auf der anderen Straßenseite oder das Wahrnehmen eines Blitzes

sein können. Diese Episoden lassen sich im Gedächtnis räumlich und zeitlich relativ zueinander anordnen [95] und durch Assoziationspunkte mit anderen Episoden verknüpfen und damit in den Kontext einordnen [20]. Studien zeigten, dass die Fähigkeit des Gedächtnisses, neue mit vorhandenen Events zu assoziieren, umso zuverlässiger funktioniert, desto besser bereits stattgefundene Events gespeichert und abrufbar sind [92].

1.1.3 Erinnerung

Das menschliche Gedächtnis ist stichwortgetrieben [85]. Diese Stichwörter oder Hinweise können zum Beispiel Bilder, Wörter, nichtverbale Geräusche, Emotionen, Stimmungen, Orte und Umgebungen sein [38, 4]. Beispielsweise kann ein Bild aus der eigenen Kindheit als Auslöser dienen, um sich an einen früheren Moment zu erinnern. Weiterhin können wir Reize schlechter abrufen als wahrnehmen [79]. Durch Bilder könnte sich eine Person bei der Suche nach einem bestimmten Ereignis besser an einen Reiz erinnern, welcher zu diesem Zeitpunkt eine Rolle spielte. Dieser Reiz kann dann als Stichwort dienen und dazu führen, dass die Person sich an das Ereignis erinnert. Weiterhin gibt es den sogenannten „Testing Effect“, welcher besagt, dass eine häufige Wiederholung von Erinnerungen, die spontane Erinnerungsfähigkeit an diese verbessert [76].
Studien anhand von Tagebüchern zeigen, dass es Erinnerungsprobleme zurück- und vorausblickender Natur gibt. Zurückblickende Probleme sind zum Beispiel, wenn sich Personen nicht an Namen, Telefonnummern, Orte oder Fakten über andere Leute erinnern können, wohingegen vorausblickende Probleme Situationen beschreiben, bei denen eine Person einen Termin vergisst oder vergisst was sie machen wollte oder was sie gesucht hat. Im Durchschnitt hatten die Studienteilnehmer zu 51% Probleme zurückblickender Natur und zu 38% vorausblickender Natur. [91, 55, 34]
Es existieren zwei Arten von Beeinflussungen (Interferenzen) für die Erinnerung an Events, welche dazu führen können, dass sich einen Person an bestimmte Events schlechter oder gar nicht erinnert. Die retroaktive Interferenz besagt, dass es umso schwieriger ist, gleichartige Events voneinander zu unterscheiden, desto öfter diese auftreten [85]. Bei der proaktiven Interferenz geht es darum, dass es, je mehr Events stattgefunden haben, umso schwieriger ist, sich aus diesen an ein bestimmtes Event zu erinnern [82].

1.1.4 Lifelogs als Gedächtnisverstärkung

Ein Lifelog sollte so viele Anhaltspunkte, die bei der Kodierung relevant sind, wie möglich speichern, um als gute Gedächtnishilfe zu dienen [41]. Das Gedächtnis lässt sich durch häufiges Ansehen der Lifelogs auf Grund des Testing Effects stärken, da es dadurch neue Momente im Speicherprozess besser mit vorhandenen verknüpfen kann. Eine weitere Ausnutzung des Testing Effects ist in diesem Zusammenhang das Retrieval-induzierte Vergessen, mit welchem Personen durch nicht-wiederholen

von bestimmten Erinnerungen diese aktiv vergessen können [5]. Lifelogs stellen zudem vor allem Stichworte dar, um sich an Momente zu erinnern. Ein Anschauen eines Lifelogs führte in [6, 99] dazu, dass sich sowohl gesunde, als auch Personen mit Gedächtnisstörungen besser an Momente erinnern konnten. Automatisch und manuell geschossene Bilder schnitten als Stichworte auf kurze Sicht gleichermaßen gut ab [77].
Lifelogs konnten einem 13-jährigen Jungen, welcher Probleme mit seinem episodischen Gedächtnis hatte, dabei helfen, sich an einen Spaziergang zu erinnern und die Lifelogs trugen vermutlich zu seiner Rehabilitation bei [68]. Die SenseCam wurde in [68] benutzt, um durch Bildtagebücher einer Frau zu helfen, die unter strenger Gedächtnisschwäche litt. Es stellte sich heraus, dass die Frau sich durch häufiges Anschauen der vergangenen Ereignisse an 80% dieser erinnern konnte. Im Vergleich zu handgeschriebenen, detaillierten Tagebüchern erwies sich die durch die SenseCam erzeugten Bildtagebücher für die Erinnerung als doppelt so effektiv [12]. Auch Patienten, die unter Amnesie litten, konnten sich im Vergleich zu handgeschriebenen Tagebüchern, besser an signifikante Momentdetails erinnern [99].

1.2 Kritische Gesichtspunkte von Lifelogging

Neben den genannten positiven Anwendungen von Lifelogs, gibt es jedoch auch einige rechtliche und gesundheitliche Aspekte zu beachten. Diese Aspekte finden im Folgenden nähere Betrachtung, genauso wie eine Erläuterung, weshalb Daten und Wissen nicht gleichzusetzen sind.

1.2.1 Rechtlicher Aspekt

Lifelogging bietet neben den Vorteilen auch einige Risiken und Probleme. Beispielsweise hackten sich im Jahr 2014 Angreifer in Apples iCloud ein[1] und stahlen persönliche Fotos. Aber auch versehentliches Teilen von privaten Bildern stellt ein Risiko dar. Neben den Hackern könnte auch die Landesbehörde potentiellen Zugriff erlangen, zum Beispiel in Rahmen der Suche nach Terroristen, Kinderschändern oder organisiertem Verbrechen [2] oder, um Steuerhinterziehungen, Geschwindigkeitsübertretungen, Versicherungs- und Sozialbetrüge zu identifizieren [41]. In diesem Zusammenhang steht auch die forciert Beweislastumkehr, bei welcher eine Person, um ihre Unschuld zu beweisen, ihre aufgezeichneten Lifelogs der Strafverfolgung zuspielen muss [2]. Ein mögliches Problem stellt zudem dar, dass einige Cafés und Restaurants Lifelogging bereits verboten haben [28, S. 6]. Weiterhin könnten Personen den Lifeloggern vorwerfen illegale Bilder von Kindern zu schießen [41].
Lichtbildwerke erhalten laut §§ 1, 2 Abs. 5 UrhG Urheberrechtsschutz, wobei nach § 7 der Urheber des Bildes der Lifelogger ist. Die geschossenen Bilder dürfen Lifelogger laut § 53 UrhG in vollem Umfang privat nutzen, für eine Veröffentlichung müssen laut § 22 KunstUrhG in der Regel die abgebildeten Personen einwilligen.

[1] `https://www.bbc.com/news/technology-29076899`

1.2.2 Gesundheitlicher Aspekt

Für Personen mit psychischen Problemen wie bipolarer und unipolarer Depression könnte ein psychologischer Hazard entstehen, wenn sich die Betroffenen erneut ihre wahrgenommenen Fehler, Beleidigungen, verpassten Möglichkeiten und sozialen Missstände anschauen, wie zum Beispiel, wenn sie auf Grund ihrer Krankheit andere Leute beleidigen oder verletzen und es ihnen im Anschluss leid tat [2]. Aber auch gesunde Leute könnten davon betroffen sein, erneut zu sehen, welche Dinge sie falsch gemacht haben [2, S. 65].

1.2.3 Unterschied von Daten und Wissen

Ein Trugschluss ist es, Daten mit Wissen und Fakten gleichzusetzen und daraus zu schlussfolgern, dass mehr Daten auch mehr Wissen bedeuten [62]. Der Grund ist, dass Daten einen signifikanten Interpretationsspielraum lassen und es daher schwer bis unmöglich ist, die Intentionen der Handlungen einer Person aus den Lifelogs herauszulesen, da vor allem der Kontext eine große Rolle spielt [66]. Falls eine Person den Lifelog an Andere freigibt und dafür gewisse private, wichtige Stellen herauslöscht, kann dies dazu führen, dass dieses Löschen die Wahrnehmung eines Momentes maßgeblich beeinflusst [39].

1.3 ImageCLEF

Die vorliegende Arbeit verwendet die von ImageCLEF im Rahmen eines Wettbewerbes zur Verfügung gestellten Lifelogs und Anfragen. ImageCLEF ist eine im Jahr 2003 aus CLEF (Cross-Language Education and Function) entstandene Plattform, bei der der Fokus auf sprachübergreifendem Informationsretrieval in Bezug auf Bildern liegt. Die Ziele der Plattform sind die Verbesserung der Analyse, Indizierung, Klassifikation und des Retrievals visueller Medien. Dafür finden jährlich Wettbewerbe statt, bei denen Teilnehmer aus akademischen Kreisen und der Industrie teilnehmen können. Dieses Jahr gab es vier verschiedene Wettbewerbe, wovon einer die Auswertung von Lifelogs als Ziel hatte (Lifelog Moment Retrieval) [25]. Für diesen reichte ich ein Paper [89] ein und nahm bereits im Jahr zuvor an der GeoCLEF Challenge teil [90].

1.4 Aufbau der Arbeit

Die Arbeit ist wie folgt aufgebaut. Zuerst erfolgt ein kurzer Abriss in den State of the Art, gefolgt von einer Analyse des gegebenen Datensets. In diesem Kapitel finden Bilder, Metadaten und Anfragen nähere Betrachtung. In Kapitel 4 wird das umfangreichste in dieser Arbeit entwickelte Verfahren vorgestellt und anschließend die durchgeführten Experimente inklusive deren Evaluation vorgetragen. Zum Schluss folgt eine Zusammenfassung und ein Ausblick auf zukünftige Ansätze.

2 State of the Art

Lifelog Moment Retrieval hat in den letzten Jahren immer mehr Aufmerksamkeit erhalten. In dem Wettbewerb vor zwei Jahren gab es nur einen Teilnehmer, letztes Jahr waren es sechs Teilnehmer und dieses Jahr nahmen insgesamt acht Teams teil [23, 24, 25].
Die Schwierigkeit dieser Aufgabe liegt in der Extraktion der relevanten Informationen und der Erkennung komplexer Muster aus den sehr großen Datenmengen und dem Vergleich dieser mit den aus den Anfragen extrahierten Begriffe [24]. Es folgt eine kurze Darstellung des State of the Arts der Herangehensweisen des Lifelog Moment Retrieval der letzten Jahre. Dazu gehören die Verfahren zur Vorverarbeitung der Bilder und der Anfragen, sowie die Möglichkeiten zur Bildsegmentierung. Weiterhin sind Filterungsverfahren, der Vergleich der Anfragen mit den Bildern und Diversifikation von Bedeutung.

2.1 Bildverarbeitung

Eine Vorverarbeitung der Bilder und Metadaten ist notwendig, um diese mit den Anfragen später vergleichen zu können. Um die Anzahl an Daten zu erweitern und den Inhalt der Bilder zu beschreiben, ist es sinnvoll, diese mit Bilderkennungsverfahren zu labeln, da die Metadaten wie Herzrhythmus oder Position weniger Relevanz haben, als das was auf den Bildern zu sehen ist.
Tsun-Hsien et al. verwendeten in ihrem vorgestellten Ansatz mehrere CNNs auf die Bilder an [88]. Dazu zählten Yolo-v2 [72] und Faster R CNN [74], welche jeweils auf das Microsoft COCO Datenset [59] und das Open Images Datenset [54] vortrainiert waren. Ersteres Datenset enthält 328.000 gelabelte Bilder und eignet sich besonders gut, um alltägliche Situationen zu beschreiben. In dem Open Images Datenset sind teilweise komplexe Szenen, mit durchschnittlich acht annotierten Labels, dargestellt. Weiterhin nutzten sie die Google Cloud Vision API[1] um weitere Lables zu annotieren und, um durch OCR Texte auf den Bildern herauszulesen. Regim Lab verwendeten ein angepasstes GoogLeNet [87] und AlexNet [53] für das Labeling.
Ein anderes Team setzte auf ein auf das Places365 Datenset [100] trainiertes VGG16 Netz [80], um den Bildern 365 Labels zuzuordnen, die die Umgebung beschreiben [31]. Um Nahrungsmittel zu klassifizieren, verwendeten sie außerdem ein auf das ImageNet Datenset [29] vortrainiertes InceptionV3 Netz [86], welches sie auf das Food101 Datenset [10] anpassten. Es gab auch ein Team, welches ein auf das

[1] `https://cloud.google.com/vision`

ImageNet Datenset und dem Places365 Datenset vortrainiertes ResNet152 [43] verwendete [64]. Das ImageNet Datenset enthält 1.000 verschiedene Labels, ist aber auf nur ein Objekt pro Bild begrenzt, das heißt es kann nur ein Label pro Bild annotieren.
Einen etwas anderen Ansatz verfolgte das HCMUS-Team 2018, indem es, unter Verwendung von dem auf das COCO Datenset trainierte NeuralTalk Netz [96, 51], für jedes Bild eine Beschreibung generierte [93]. Das Zählen von Personen auf den Bildern erwies sich auch als hilfreich [64]. Die Feststellung der Anzahl an Personen kann zum Beispiel durch das Histogram of Oriented Gradients Verfahren [21] oder mit Hilfe der Person Detection API von Sighthound[2] erfolgen.

2.2 Bildsegmentierung

Die Segmentierung von Bilder ermöglicht das Zusammenfassen von ähnlichen Bildern zu einer Sequenz und erleichtert später die Extraktion aller relevanten Momente, da pro Sequenz jeweils nur ein repräsentatives Bild gesucht wird. Die Verfahren zur Bildsegmentierung ist ähnlich zu den Verfahren für die Segmentierung von Videos in einzelnen Szenen [40, S. 15]. Dazu existierten zwei grundlegende Ansätze.

2.2.1 Ansätze

Der erste Ansatz ist eine Segmentierung anhand der Aufnahmezeitpunkte der Bilder. Dazu wird ein Zeitintervall festgelegt, welches zwei Bildsequenzen voneinander trennt [70, S. 208]. Dieser Ansatz ist nicht geeignet für die Bilder des Lifelogs, da diese kontinuierlich geschossen werden und die einzigen größeren Zeitintervalle lediglich die nächtliche Bettruhe und händisch entnommene Momente sind. Daher eignet sich lediglich die zweite inhaltsbasierte Herangehensweise, die auf der Berechnung von Distanzmaßen zwischen den einzelnen Bildern basiert [60]. Inhalte können beispielsweise die Metadaten der Bilder wie GPS oder MPEG-7 Descriptoren sein [13, 16]. Zudem können Bildpixel, Farbhistogramme oder Bildkanten, berechnet mit Hilfe von Canny-Edge-Detection [14], ebenfalls Inhalte beschreiben [33]. Durch Bilderkennungsalgorithmen wie zum Beispiel mit einem vortrainierten CaffeNet CNN [49] können zusätzliche Inhalte generiert werden. Diesen Ansatz verfolgte das Organisatoren-Team vor zwei Jahren bei dieser Challenge [103]. Da bei Lifelogs üblicherweise neben den Bildern weitere Daten gegeben sind, kann eine Bildsegmentierung ebenso unter Zuhilfenahme dieser Daten erfolgen [13].

2.2.2 Bildvergleich

Für den Vergleich der Inhalte existieren verschiedene Möglichkeiten, wie zum Beispiel die Berechnung der Manhattan-Distanz oder der euklidischen Distanz, wobei dadurch bei Lifelogs häufig das Phänomen auftritt, dass ein kurzzeitiges Verdecken

[2] `https://www.sighthound.com/docs/cloud/detection`

der Kamera oder ein kleiner Schwenker dazu führen, dass zwei aufeinanderfolgende Bilder nicht mehr ähnlich sind und deswegen in verschiedene Segmente aufgespalten werden [33, S. 4]. Daher eignet sich der Text Tiling Algorithmus von Hearst [44] besser, da bei ihm der Vergleich auf Blöcken von mehreren zusammenhängenden Bildern basiert. Ein Schwellenwert legt fest, wie ähnlich sich zwei Blocks sein dürfen, um diese in ein Segment zusammenzufassen. Sollte ein problematisches Bild auftreten, fällt dieses nicht so stark ins Gewicht und führt nicht zu dem Auftrennen in mehrere separate Segmente [33, S. 4].

FlowNet

Weiterhin ist zu beachten, dass bei Lifelogs des öfteren nur ein Bereich im Bild ändert, wenn zum Beispiel Videos auf dem Fernseher zu sehen sind oder während des Autofahrens sich nur der die Umgebung ändert. Um dieses Problem zu lösen, haben Truong et al. ein Verfahren entwickelt, welches den prozentualen Unterschied der Pixel zwischen zwei Bildern berechnet und nebenbei Bilder gruppiert, bei dem ein bestimmter Bereich im Bild konstant bleibt. Zudem fand das, mit Hilfe von gestapelten CNNs Bewegungsvektoren berechnende, FlowNet [46] mehrmals Anwendung, um durch die Berechnung des prozentualen Anteils an minimalen optischen Bewegungsvektoren auf Kamerabewegungen zu reagieren [94, 93].

2.2.3 Clustering

Die Segmente ergeben sich durch das Anwenden von unüberwachten hierarchischen agglomerativen Clustering [3, S. 132ff.], [9]. Hierbei ist jedoch zu beachten, dass das standardmäßige Clustering die Reihenfolge der Bilder, also die zeitliche Kohärenz, ignoriert. Dieses Problem löst sich durch das Anwenden eines Postprocessings oder durch die Verwendung von TCK-Means [60], einem zeiterzwungenen Clusteringverfahren basierend auf K-Means [61], bei dem weit auseinander liegende Frames während des Clusterings in der Kostenfunktion stärker bestraft wurden als nah beieinander liegende Frames. Bolanos et al. führte das Postprocessing mit einem Division and Fusion Ansatz durch, indem sie Events eines Clusters auftrennten, zwischen denen Events aus anderen Clustern lagen und im Anschluss Events wieder zusammenführten, die einen zeitlichen Abstand von einem zuvor bestimmten Schwellenwert hatten.

2.2.4 Keyframe Selection

Beim Lifelog Moment Retrieval ist jeweils nur ein Bild pro Segment relevant, da die anderen Bilder die gleiche Situation und daher auch den gleichen Moment darstellen. Um herauszufinden welches Bild am aussagekräftigsten ist, gibt es wiederum verschiedene Methoden wie zum Beispiel der Random Walk Algorithmus [69], bei dem das repräsentativste Bild dasjenige ist, welches am ähnlichsten zu allen anderen Bildern im Segment ist. Der Algorithmus erstellt dazu einen Graphen, dessen

Kantengewichte die Ähnlichkeiten der einzelnen Bilder sind, extrahiert daraus anschließend die assoziierte Matrix, berechnet deren Eigenvektoren und nimmt zum Schluss das Bild heraus, welches den größten Wert in dem ersten Eigenvektor der Matrix hat. Ein anderer Ansatz ist es, das Bild zu wählen, welches die geringste akkumulierte euklidische Distanz zu allen anderen Bildern hat. Es hat sich gezeigt, dass der Random Walk Algorithmus geringfügig besser abschneidet als die Berechnung der minimalen Distanz [9].

2.3 Metadatenverarbeitung

Die Metadaten umfassen meist sehr viele verschiedene Informationen. Die Extraktion der relevanten Informationen spielt dabei eine ebenso wichtige Rolle wie die Bildverarbeitung. Die Aufnahmezeitpunkte der Bilder lassen sich in die Tageszeitpunkte *morning*, *afternoon*, *evening* und *night* transformieren [64].

2.4 Anfragenverarbeitung

Für die Verarbeitung der Anfragen existieren grundsätzlich zwei Herangehensweisen, wobei es das Ziel beider ist, für die Suche bestimmte Kriterien aus den Anfragen zu extrahieren, um diese anschließend mit den Bildern und deren Metadaten vergleichen zu können.

2.4.1 Interaktiver Ansatz

Beim ersten, interaktiven Ansatz gibt es einen Human-in-the-loop Eingriff, das heißt der Anfragensteller kann zwischenzeitlich in die Verarbeitung eingreifen und bestimmte positive und negative Kriterien manuell definieren. Die Organisatoren implementierten dafür ein automatisches System, um die Kriterien zu extrahieren und verfeinerten diese anschließend manuell [103]. Im darauf folgenden Jahr entschied das Team LIFER [104, 102] einzusetzen, welches aus dem Lifelog sechs Daten, die verschiedene Facetten aus dem Leben einer Person widerspiegeln, extrahiert und anschließend mit einem speziellen Filtermechanismus entsprechende Bilder heraussucht. Die Idee, viele Facetten abzubilden hatten bereits 2005 Gemmel et al. [36] und 2012 Doherty et al [32] in ihren Systemen. Truong et al. definierten in ihrem Retrieval System vier Kriteriumgruppen für die ein Nutzer manuell aus einer Anfrage Kriterien definieren konnte. Dazu zählten der Ort, die Zeit, die Handlung und extra Kriterien wie Biometrie-Daten [94].

2.4.2 Automatischer Ansatz

Der zweite, automatische Ansatz besteht darin, die Kriterien ohne weitere Anpassung der Anfrage während des Suchprozesses unter Zuhilfenahme von Methoden der

natürlichen Sprachverarbeitung zu extrahieren. Das oben genannte Organisatoren-Team und das NLP-Lab-Team extrahierte dafür alle Wörter der Anfrage als potentielle Konzepte, die in den Bildern vorkommen können [104, 88]. Das Regim Lab-Team wandelte alle Wörter einer Anfrage mit Hilfe von Word Embeddings in einen numerischen Vektor und trainierte ein LSTM Netz in Verbindung mit einem SGDM Optimizer [1]. Ein anderes Team nutze ebenfalls Word Embeddings in Form eines Word2Vec Modells [63], um die vorhandenen Labels um ähnliche Labels zu erweitern [93], indem sie mit der Kosinus-Distanz naheliegende Labels anhand eines Vektorraums berechneten.

2.5 Filterung

Bilder mit nur wenig Informationsgehalt spielen eine untergeordnete Rolle, weshalb sie oft herausgefiltert werden. Das Multimedia Lab-Team und das VC-I2R-Team stellte die Stärke des Fokus' eines Bildes durch Anwenden der Laplace-Filterung fest, bestimmten dessen Varianz und entfernten alle Bilder unter einer Threshold auf Grund deren Unschärfe [31, 64]. Zudem umfasste diese Filterung Bilder mit größeren homogenen Bereichen, wobei sich die Farbvielfalt auch durch quantisierte RGB Werte aus den Histogrammen ablesen lässt. Es gilt, je häufiger die vorwiegend dominante Farbe auftritt, desto homogener ist das Bild [64].
Eine weitere Variante für die Feststellung des Fokus' ist die von den Organisatoren 2017 eingesetzte Berechnung der Wavelet Koeffizienten [97] und deren Vergleich mit einer Threshold, wobei Bilder die unterhalb dieser Threshold liegen als unscharf gelten [103] und daher keine weitere Verwendung finden. Um Bilder mit großen Objekten herauszufiltern, wandelten sie die Bilder in binäre Bilder um, extrahierten verbundene Komponenten, berechneten ihre Zentren und verbanden diese zu einem BLOB. Bei einer Abdeckung von über 50% des Bildes durch den BLOB, zählte dieses Bild als überdeckt und fand ebenfalls keine Weiterverwendung.

2.6 Retrieval

Die einfachste Form des Retrievals besteht darin, die aus den Anfragen extrahierten Labels mit den Labels der Metadaten zu vergleichen und diejenigen Bilder als relevant zu betrachten, welche Übereinstimmungen haben [103, 104].
Der Nachteil dieser Variante besteht jedoch in ihrem niedrigen Recall, weswegen Tsun-Hsien et al. eine weitere Methode implementierten, welche auf der BM25 [75] Metrik basierte [88]. Diese Metrik bezieht die Termfrequenz und Inverse Dokument Frequenz der Labels in den Vergleich ein und gewichtet damit seltene, aussagekräftige Labels stärker.
Eine weitere Variante des Teams bestand darin Word Embeddings, ähnlich wie in 2.4.2, zu verwenden, indem sie den Durchschnitt der Kosinus-Distanzen zwischen den Labels der Bilder und denen der Anfragen aggregierten. Dies hatte den Vorteil,

dass ähnliche Labels nicht direkt miteinander übereinstimmen mussten. Für die Berechnung der Ähnlichkeiten kamen mit Hilfe von fastText [50] vortrainierte Word Embeddings zum Einsatz, wobei sie Wörter, die nicht in dem Vokabular der Embeddings vorkamen, ignorierten. Neben fastText experimentieren sie zusätzlich mit Subword-Enriched-Word-Vectors [8], GloVe [71], ConceptNet Numberbatch [84] und Dependency-Based Word Embeddings [58]. Dabei fanden sie heraus, dass Word Embeddings, die zusätzliche kontextbasierte Informationen enthalten, wie zum Beispiel die syntaktische Abhängigkeit oder die lexikalische Onkologie, am besten funktionierten. Die Überprüfung von Ort und Zeit basierte auf der Natural Language API von Google[3].
Regim Lab erstellte eine invertierte Index-Matrix, indem sie zu jedem Label aller Labelarten den jeweiligen Score für die Bilder eines Tages zuordneten, in denen das Label vorkam, alle anderen Labels erhielten eine Null. Die aus dem oben genannten LSTM Netz extrahierten Labels der Anfragen verglichen sie mit den Labels aus der Matrix und suchten die entsprechenden Bilder heraus. Für das Retrieval kam bei ihnen zudem XQuery [7] zum Einsatz, eine Abfragesprache für XML-Datenbanken, mit welcher sie den Ort, die Aktivität und die Zeit aus den Sensordaten abfragten. Die beiden Ergebnisse aggregierten sie und sortierten sie absteigend nach den höchsten Scores.
Eine andere Variante implementierte das Team VC-I2R unter Verwendung von Vektoren [64]. Sie erzeugten für jeden Frame und für jede Anfrage einen Descriptor-Vektor, welcher mehrere Gewichte beinhaltete. Relevante Labels bekamen ein positives Gewicht und irrelevante Labels erhielten ein negatives Gewicht. Relevante Orte und Aktivitäten bekamen ebenfalls ein positives Gewicht, wobei irrelevante Orte oder Aktivitäten dazu führten, dass ein Frame komplett aus folgenden Betrachtungen ausschied. Frame Descriptoren für Anfragen, die Personen verlangten, erhielten zusätzlich ein positives Gewicht, falls diese eine entsprechende Anzahl an Personen beinhalteten. Der finale Relevanz Score berechnete sich aus dem Skalarprodukt der Frame- und Anfrage-Vektoren. Das Finden der optimalen Gewichte geschah heuristisch anhand von auf einem Trainingsset durchgeführten Versuchen. Verschiedene Experimente zeigten, dass das ausschließliche Verwenden von Metadaten durchschnittlich zu schlechteren Ergebnissen führte, als das Verwenden von ausschließlich visuellen Daten und eine Kombination aus beiden Daten am besten funktionierte. Es zeigte sich außerdem, dass das Zählen der Personen bei schlechten Lichtverhältnissen und unscharfen Bildern zu keinem besseren Ergebnis führte. Multimedia Lab baute ihr Modell auf den gerade genannten Ansatz auf und erweiterten es auf fünf verschiedene Arten von Frame- und Anfrage-Vektorpaaren [31].

2.7 Diversifikation

Die Diversifikation der Ergebnisse erfolgt optional nach dem Retrieval aller relevanten Bilder, um den Recall, also die Anzahl an unterschiedlichen Situationen, zu

[3] `https://cloud.google.com/natural-language`

erhöhen und um jeweils das relevanteste Bild pro Situation herauszunehmen. Strategien dafür bauen auf automatischen und Human in the Loop Ansätzen auf.

2.7.1 Automatischer Ansatz

Das Organizer Team gruppierte die relevanten Bilder mit Hilfe von einem hierarchischen, agglomerativen Clustering und sortierte die Cluster absteigend nach der Anzahl an Segmenten [103]. Danach extrahierten sie für jedes Segment eines Clusters das Bild, welches am nächsten zum Clustermittelpunkt lag.
Multimedia Lab setzte einen K-means Cluster-Ansatz ein, bei dem sie 5, 10, 25 oder 50 Cluster erstellten und dann jeweils das Bild mit dem höchsten Relevanz Score Round-Robin-artig in die Ergebnisliste übernahmen. Das Clustering basierte dabei auf einem zusammengesetzten Vektor aus den Histogramm of Oriented Gradients Features und Farbhistogrammfeatures. Die Autoren verglichen während des Einfügens der Bilder in die Ergebnisliste diese mit den bereits eingefügten Bildern und ignorierten ähnliche Bilder. Dies erwieß sich als nicht förderlich, da dadurch die Precision abnahm. Der Ähnlichkeitsvergleich basierte auf den beiden Metriken der mittleren quadratischen Abweichung (MSE) und dem Index der strukturellen Ähnlichkeit (SSIM). [31]

2.7.2 Human in the Loop Ansatz

Die einfachste Erweiterung des automatischen Ansatzes in einen interaktiven besteht darin, aus N der zurückgegebenen Bilder jeweils die relevantesten manuell auszuwählen [93]. Eine klassische Variante für einen Human in the Loop Ansatz ist die Verwendung des dichotomischen Relevanz Feedback Paradigmas [22, 103], bei dem der Nutzer die in 2.4.2 zurückgegebenen Ergebnisse in relevante und irrelevante Bilder aufteilt. Anschließend läuft die Sortierung der Ergebnisse wie folgend beschrieben ab. An erster Stelle in den Ergebnissen stehen die Cluster mit den meisten als relevant markierten Bildern. Sollten Cluster die gleiche Anzahl an relevanten Bildern enthalten, so stehen diejenigen mit den wenigsten irrelevanten Bildern weiter oben. Sollten in den Clustern die irrelevanten und relevanten Bilder gleich oft vorkommen, steht das Cluster, welches mehr Segmente hat, an erster Stelle.
In den letzten Jahren stellte sich heraus, dass diejenigen Verfahren, die diese Art an Relevanz Feedback einbezogen, bei einem kleinen X nicht unbedingt besser abschnitten, bei einem größeren X jedoch deutlich bessere Ergebnisse lieferten als die vollständig automatisierten Verfahren [103, 52, 64].
Eine weitere Variante ist das sogenannte Pseudo Relevanz Feedback [11], welches darauf basiert, dass die relevanten Ergebnisse innerhalb aller Ergebnisse weiter oben stehen als die irrelevanten Ergebnisse. Dass dem so ist, hat sich zum Beispiel bei Flickr's Retrieval System gezeigt [47, 48]. Dieser Ansatz ist nur pseudomäßig, da er keinen Nutzer benötigt, der die Ergebnisse überprüft und der relevante von irrelevanten Bildern unterscheidet. Das Pseudo Relevanz Feedback benötigt lediglich einen vordefinierten Wert, welcher angibt, wie viel Prozent der ersten N Bilder

als relevant gelten. Es folgt ein hierarchisches Clustering der Ergebnisse. Lediglich Cluster, die mindestens zur Hälfte aus relevanten Bildern bestehen, sind in späteren Betrachtungen von Bedeutung. Der Algorithmus entfernt im nächsten Schritt von den übrig gebliebenen Clustern die irrelevanten Bilder und sortiert die restlichen Bilder entsprechend ihrem ursprünglichen Rang. Anschließend entnimmt er aus jedem Cluster jeweils das Bild mit dem niedrigsten Rang. Dieser Vorgang wiederholt sich solange, bis in keinem Cluster mehr ein Bild enthalten ist. Es hat sich gezeigt, dass der zuvor vorgestellte Ansatz in Hinblick auf Cluster Recall und Precision besser abschneidet [22].

3 Analyse des Lifelogs

In dem Wettbewerb waren Lifelogs von zwei Personen gegeben, die aus minutengenau erfassten Bildern von tragbaren Kameras und Metadaten bestanden. Ich entschloss mich dazu, mich ausschließlich auf die erste Person zu konzentrieren, da alle Trainingsanfragen an sie gerichtet waren und eine Evaluation damit nur für diese Person möglich war, ohne eigenhändig Anfragen zu generieren.
Für jeden Tag waren Aufzeichnungen für jede Minute von 00:00 bis 23:59 gegeben. Daraus ergaben sich 1.440 mögliche Einträge pro Tag. Alle Aufzeichnungen wurden im Zeitraum vom einschließlich 3. bis 31. Mai 2018 gesammelt, also innerhalb von 29 Tagen. Der Datenbestand enthielt vier Teile:

- **Autografische Bilder**: mit der tragbaren Kamera geschossene Bilder
- **Persönliche Bilder**: mit dem Smartphone geschosse Bilder
- **Visuelle Konzepte**: Begriffsdatenbank, die visuelle Konzepte enthielt
- **Minutenbasierte Ereignisliste**: minutenbasiertes Log, welches zusätzliche Metadaten beinhaltete

Zusätzlich waren jeweils zehn Trainings- und Testanfragen gegeben. Zu jeder Trainingsanfrage waren außerdem Lösungs-Cluster („Ground Truth“) gegeben, welche alle relevanten Bilder pro Moment enthielten. Dabei genügte es später, nur ein Bild aus jedem Cluster zu finden.

3.1 Autografische Bilder

Jede Person hatte im Zeitraum der Datenerhebung eine tragbare Kamera umhängen, die automatisch mehrere Bilder pro Minute schoss („autografische Bilder“). Die Bilder waren tageweise gruppiert, wobei insgesamt 63.696 Bilder existierten, die eine Speichergröße von ca. zehn Gigabyte einnahmen. Die autografischen Bilder hatten alle eine Auflösung von 1024x768 Pixel und waren damit nicht sehr hochauflösend. Dadurch war es schwieriger die Objekte auf den Bildern mit Hilfe von neuronalen Netzen korrekt zu annotieren.
In Abbildung 3.1 sind einige repräsentative Bilder aus dem Datenset aufgelistet. Die Auszüge zeigen, dass es neben scharfen Bildern auch verzerrte und verdunkelte Bilder gab. Außerdem waren die Arme der Person häufig zu sehen oder die Kamera war verdeckt. Zudem ist erkennbar, dass Gesichter aus Anonymitätsgründen verpixelt wurden.

(a) Normal (b) Gesicht unkenntlich (c) Arme im Bild

(d) Unscharf (e) Dunkel (f) Kamera verdeckt

Abbildung 3.1: Auszüge aus den autografischen Bildern zeigen, dass es neben scharfen Bildern (a) auch verzerrte (d) und verdunkelte (e) Bilder gab. Außerdem waren die Arme des Lifeloggers häufig zu sehen (e, f) oder die Kamera war verdeckt (f). Bei b) ist erkennbar, dass Gesichter aus Anonymitätsgründen verpixelt wurden.

Das Diagramm in Abbildung 3.2 zeigt die durchschnittliche Anzahl an Aufnahmen pro Minute über den gesamten Zeitraum vom 3. bis 31. Mai 2018. Es ist erkennbar, dass der Tag des Lifeloggers im Schnitt von 6 bis 23 Uhr ging, da in dieser Zeit die meisten Bilder existierten. Weiterhin fällt auf, dass auch in der Zeit der Nachtruhe Bilder entstanden sind. Eine Ursache dafür war, dass der Lifelogger am 26. Mai in der Nacht von Shanghai nach Dublin geflogen ist.

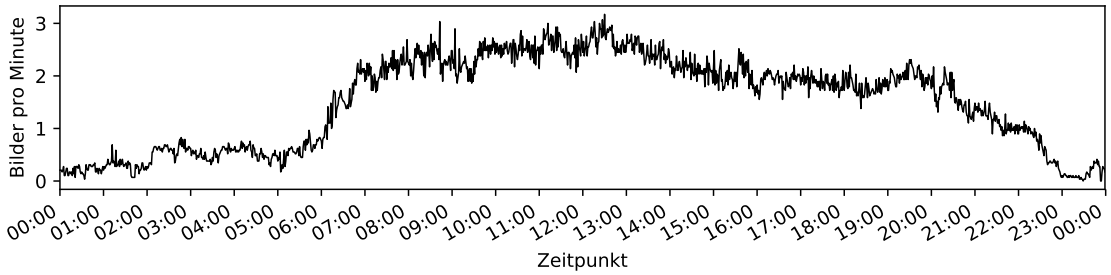
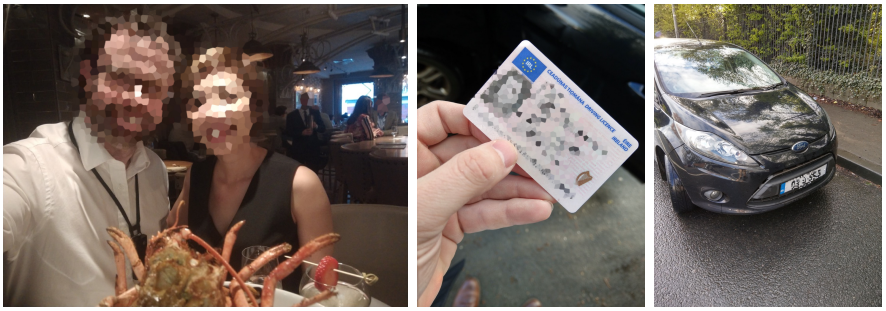


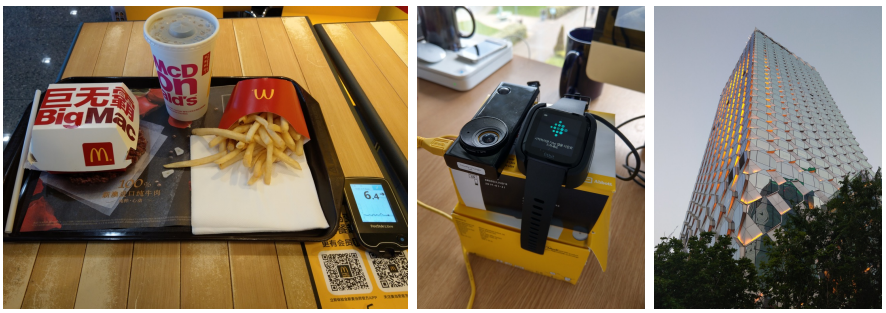
Abbildung 3.2: Das Diagramm zeigt die Aufnahmen pro Minute für den gesamten Zeitraum. Es ist erkennbar, dass die Person durchschnittlich gegen 6 Uhr aufstand und 23 Uhr schlafen ging.

3.2 Persönliche Bilder

Neben den automatisch geschossenen Bildern nahm die Person 405 Bilder per Hand in der selben Auflösung wie in 3.1 mit ihrem Smartphone (HTC U11) oder ihrer tragbaren Kamera auf, welche 108 Megabyte Speicherplatz einnahmen. Auszüge aus diesen Bildern in Abbildung 3.3 zeigen, dass Gesichter ebenfalls unkenntlich waren, die Bildqualität jedoch im Vergleich zu den automatisch geschossenen Bildern viel besser war, da es keine unscharfen Bilder gab. Auffallend ist auch, dass es sehr viele Fotos von Mahlzeiten gab (geschätzt über 90%) und dass bei diesen häufig ein Blutzuckermessgerät mit abgebildet war. Ausweise, Auto-Kennzeichen, Führerschein, Personalausweis, Kreditkarten und sonstige persönliche Dinge wurden ebenfalls verpixelt. Das vierte Bild zeigt die tragbare Kamera und die Smartwatch, die der Lifelogger trug. Sehenswürdigkeiten wie das Gebäude auf dem letzten Bild nahm der Lifelogger ebenfalls auf.



(a) Personen (b) Dokument (c) Kennzeichen



(d) Mahlzeit (e) Kamera (f) Gebäude

Abbildung 3.3: Auszüge aus den persönlichen Bildern zeigen, dass Gesichter (a), persönliche Dokumente (b) und Auto-Kennzeichen (c) unkenntlich waren und die Bildqualität sehr gut war. Fotos von Mahlzeiten, wie in (d), kamen häufig vor. Bild (e) zeigt die getragene Kamera.

Unter genauer Betrachtung stellte sich heraus, dass die Ersteller des Datensets vergaßen, einige der Bilder ins Log zu übernehmen und dass in den vorgegebenen Lösungs-Clustern kein einziges persönliches Bild existierte. Daraus ließ sich schließen, dass diese in den Clustern der Testanfragen ebenfalls irrelevant waren.

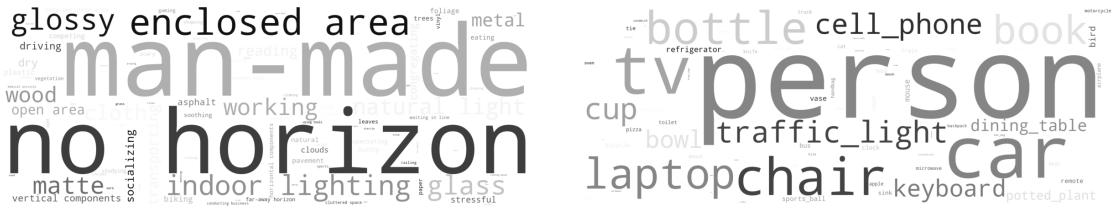
3.3 Visuelle Konzepte

Die Visuellen Konzepte waren in Form einer CSV-Tabelle gegeben, welche zu jedem geschossenen Bild durch CNNs annotierte Begriffe, Konzepte und Attribute inklusive ihrer Konfidenz-Scores enthielt. Es existierten die folgenden Spalten:

- **Id**: Die Spalte *image_id* enthielt den eindeutigen Bezeichner für ein Bild. Eingeschlossen waren alle Bilder der tragbaren Kamera und des Smartphones (mit „_cam"-Zusatz), z.B. *u1_20180503_0617_i00* oder *u1_20180503_0619_cam_i00*.
- **Pfad**: Die Spalte *image_path* enthielt den relativen Pfad zum entsprechenden Bild im Dateiverzeichnis, z.B. *u1_photos/2018-05-03 07.28.26.jpg* oder *2018_05_03/B00001383_21I6X0_20180503_072356E.JPG*.
- **Attribute**: Die Spalten *attribute_top[01-10]* enthielt die zehn wahrscheinlichsten der insgesamt 97 Attribute, die mit Hilfe des auf das SUNattribute Datenset [67] trainierte Place CNNs [101] vorhergesagt wurden, z.B. *no horizon*, *man-made* oder *glossy*.
- **Kategorien**: Die Spalten *category_top[01-05]* und *category_top[01-05]_score* enthielt die wahrscheinlichsten fünf der insgesamt 360 Kategorien inklusive ihrer Scores, die mit Hilfe des auf das Places 365 Datenset [100] trainierte Places CNNs [101] vorhergesagt wurden, z.B. *beauty_salon* mit Score: *0,245* oder *drugstore* mit Score: *0,310*.
- **Konzepte**: Die Spalten *concept_class_top[01-25]*, *concept_score_top[01-25]* und *concept_bbox_top[01-25]* enthielten die Namen, Scores und Bounding Boxen der wahrscheinlichsten 25 Objekte eines Bildes, die mit Hilfe des auf das COCO Datenset [59] trainierte Faster R CNNs [74] vorhergesagt wurden. Dabei konnten beliebige der insgesamt 76 verfügbaren Konzepte auch mehrfach zuwiesen sein. Die Bounding Boxen stellten die Koordinaten der Ecken des umgebenden Rechtecks des betroffenen Objektes auf dem Bild dar. Dabei waren die Zahlen in dem Format *x y Breite Höhe* angegeben, wobei x und y die linke untere Ecke angaben. Ein Beispiel ist *person* mit Score: *0.911197* und Bounding Box: *600.052734 614.484375 420.957764 151.570190*.

Ausschließlich die Konzepte durften leere Einträge haben, da logischerweise nicht in jedem Bild ein Objekt dargestellt sein musste. Es fällt außerdem auf, dass die Kategorien im Vergleich zu den Konzepten insgesamt sehr niedrige Scores hatten.
In den Abbildung 3.5 und 3.4 sind Word Clouds aller Labels aus den verschiedenen Visual Concepts dargestellt. Dabei sind die höher frequenten Labels in einer größeren Schrift abgebildet.
Die häufigsten Attribute waren *man-made* und *no horizon*, welche über 63.000 mal und damit im Durchschnitt bei jedem zehnten Bild auftraten. Am seltensten hingegen war das Attribut *digging* mit nur zwei Vorkommen.
Bei den Konzepten war das Label *person* mit einer Rate von 44,43% überdurchschnittlich häufig annotiert. Der Grund dafür war, dass die beiden Arme des Lifeloggers oft im Bild waren und das CNN diese jeweils als Person labelte (siehe erstes

Bild in Abbildung 3.6). Auf Platz zwei der häufigsten Konzepte folgte mit einem großen Abstand *car* mit 11,62% Auftrittsrate. Das seltenste Konzept war *bear* mit nur einem Vorkommen, dabei handelte es sich jedoch um ein falsches Label.



(a) Word Cloud Attribute

(b) Word Cloud Konzepte

Abbildung 3.4: Das häufigste Attribut war *man-made*. Bei den Konzepten trat das Label *person* mit einer Rate von 44,43% überdurchschnittlich häufig auf. Der Grund dafür war, dass die beiden Arme des Lifeloggers oft im Bild waren und das CNN diese jeweils als Person labelte.

Die häufigsten Kategorien waren *airplane_cabin* und *car_interior* mit 7,75% beziehungsweise 5,05% Vorkommen. Dabei war das Label *airplane_cabin* häufig falsch annotiert, weil es ähnlich mit *car_interior* war. Die Ursache für die häufigen autobezogenen Label Annotationen lag darin, dass der Lifelogger fast täglich ca. 30 Minuten zur Arbeit und wieder zurück fuhr. Elf Kategorien traten jeweils nur einmal auf, wie beispielsweise *creek* (Bach), *pasture* (Weide) oder *pond* (Teich).

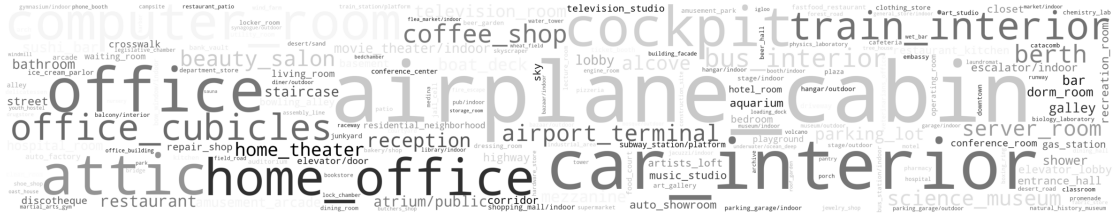
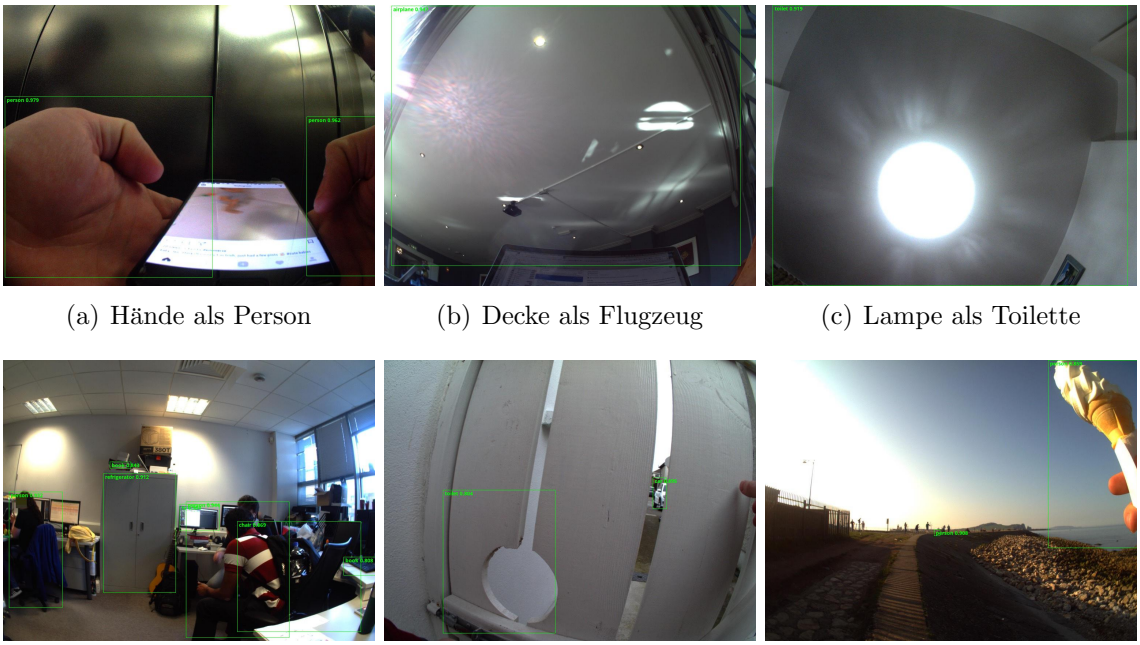


Abbildung 3.5: Die häufigsten Kategorien waren *airplane_cabin* und *car_interior* mit 7,75% beziehungsweise 5,05% Vorkommen. Elf Kategorien traten nur einmal auf, wie beispielsweise *creek*, *pasture* oder *pond*.

Es fällt auf, dass die Word Clouds der Kategorien deutlich überfüllter waren als die der Attribute oder Konzepte. Der Grund dafür war die deutlich höhere Anzahl an Labels (siehe Anhang).
In Abbildung 3.6 auf der nächsten Seite sind einige falsch annotierte Bilder mit den Bouding Boxen der Labels (grün umrandet) und ihrem Score dargestellt. Beispielsweise wurde eine Zimmerdecke zu 94,7% als Flugzeug und eine Lampe zu 91,9% sowie ein Zaun zu 80,4% als Toilette erkannt. Außerdem funktionierte die Erkennung von Personen nicht zuverlässig, da ein Eis als Person gelabelt wurde. Weiterhin war ein Schrank mit einem Score von über 91% falsch als Kühlschrank annotiert.



(a) Hände als Person (b) Decke als Flugzeug (c) Lampe als Toilette

(d) Schrank als Kühlschrank (e) Zaun als Toilette (f) Eis als Person

Abbildung 3.6: Die Bilder zeigen, dass die Annotation der Konzepte nicht zuverlässig funktionierte. Beispielsweise wurde eine Zimmerdecke als Flugzeug (b) und eine Lampe (c) beziehungsweise ein Zaun (e) als Toilette erkannt. Ein Eis war als Person gelabelt (f) und ein Schrank als Kühlschrank (d). Außerdem galten Hände als einzelne Personen (a).

Die Erkenntnis daraus war, dass die Vorhersage der Konzepte nicht verlässlich funktionierte. Eine Lösung dieses Problems besteht darin, entweder eine Threshold festzulegen, ab welcher ein Konzept als zutreffend galt und/oder in der erneuten Annotation mit einem anderen CNN.

3.4 Minutenbasierte Ereignisliste

Die minutenbasierte Ereignisliste bestand aus einer Aufzeichnung aller stattgefundenen Ereignisse in Form eines Logs. In ihr standen Metadaten, sowie Verweise zu den visuellen Konzepten (siehe vorherige Sektion). Es lagen die folgenden Spalten vor:

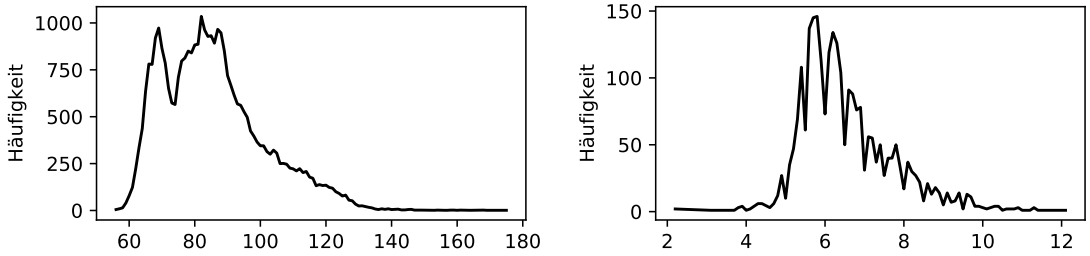
- **Id**: Die Spalte *minute_ID* enthielt einen eindeutigen Bezeichner für jede Minute, z.B. *u1_20180503_0016*.
- **Zeitpunkt**: Die Spalte *utc_time* enthielt die Zeitangabe für die Minute im UTC-Format, z.B. *20180503_0016_UTC*.
- **Lokale Zeit**: Die Spalte *utc_time* enthielt die lokale Zeit, z.B. *20180503_0116*.

- **Zeitzone**: Die Spalte **time_zone** enthielt die mit Hilfe des Smartphones ausgelesene Zeitzone, z.B. *Europe/Dublin*, *Europe/Berlin* oder *Asia/Shanghai*.
- **Breitengrad**: Die Spalte *lat* enthielt den Breitengrad der geografischen Koordinaten, z.B. *53.428072*.
- **Längengrad**: Die Spalte *lon* enthielt den Längengrad der geografischen Koordinaten, z.B. *-6.224823*.
- **Aktivität**: Die Spalte *activity* enthielt eine der beiden Aktivitäten *transport* oder *walking*.
- **Schritte**: Die Spalte *steps* enthielt die Anzahl der Schritte, z.B. *0* oder *180*.
- **Kalorien**: Die Spalte *calories* enthielt die Menge verbrannter Kalorien (kcal), z.B. *3.28579998016*.
- **Ortsbezeichnung**: Die Spalte *name* enthielt den Namen für den Ort, z.B. *Home*, *Costa Coffee* oder *Collins Park*.
- **Song**: Die Spalte *song* enthielt den Titel des Liedes, welches in dieser Minute gelaufen ist, z.B. *Fire and Rain* oder *Angel Of Mercy*. Die Titel hat der Last.FM Music Tracker durch sogenanntes „Scrobbling“ aufgezeichnet[1].
- **Blutzuckerspiegel**: Die Spalte *historic_glucose (mmol/L)* enthielt den durch einen Fingerstich gemessenen Glukoseanteil im Blut in der Einheit mmol/L, z.B. *2.2*, *9* oder *12.1*.
- **Gescannter Blutzuckerspiegel**: Die Spalte *historic_glucose (mmol/L)* enthielt den Glukoseanteil im Blut in der Einheit mmol/L. Gemessen wurde dieser alle 15 Minuten durch ein tragbares Messgerät (FreeStyle Libre[2]), das den Blutzuckerspiegel per Sensor an der Rückseite des Oberarms auslas, z.B. *3.9*, *9* oder *10.7*.
- **Herzrate**: Die Spalte *heart_rate* enthielt die Herzrate, z.B. *56*, *100* oder *175*.
- **Distanz**: Die Spalte *distance* enthielt die zurückgelegte Distanz in Metern, z.B. *0*, *0.00430000014603* oder *0.0207000002265*.
- **Autographer Bilder**: Die Spalten *img[00-19]_id* enthielt die Ids der Autographer Bilder (höchstens 20), z.B. *u1_20180521_1940_i00*.
- **Persönliche Bilder**: Die Spalten *cam[00-14]_id* enthielt die Ids der persönlichen Bilder (höchstens 15), z.B. *u1_20180508_1650_cam_i00*.

Die Felder *Id*, *Zeitpunkt* und *Lokale Zeit* waren obligatorisch, bei den anderen Feldern waren leere Angaben möglich. Die Kalorien, Schritte und die Herzrate zeichnete der *FitBit Fitness Tracker Versa*[3] auf und die Orte und Aktivitäten erfasste eine Handy-App namens *Moves*[4].

[1] `https://www.last.fm/de/about/trackmymusic`
[2] `https://www.drwf.org.uk/news-and-events/news/scanning-device-monitoring-blood-sugar-levels-be-made-available-nhs`
[3] `https://www.fitbit.com/de/versa`
[4] `https://www.moves-app.com`

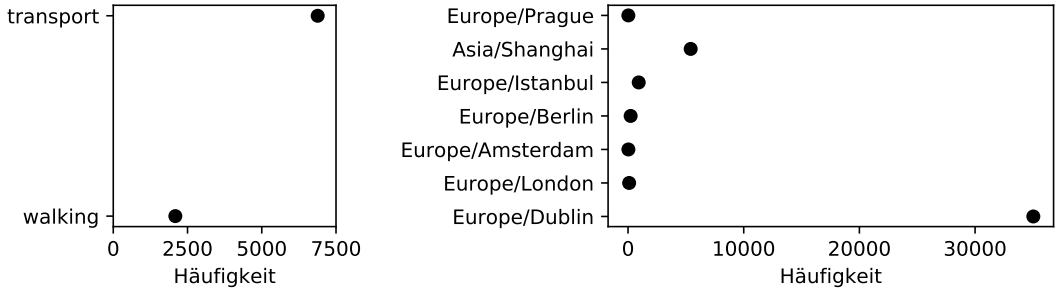
In den nachstehenden Abbildungen 3.7 und 3.8 sind einige Auszüge aus den gegebenen Metadaten der Ereignisliste dargestellt. Die Herzrate verhielt sich im Normalbereich und hatte zwei Spitzen bei ca. 70 und 85 Schlägen pro Minute, wobei die Person eine maximale Herzrate von 175 hatte. Der Blutzuckerspiegel betrug durchschnittlich 6 mmol/L und erreichte maximal 12,1 mmol/L. Die Daten waren demnach verwertbar, da sie sinnvolle Werte enthielten.



(a) Herzrate (b) Blutzuckerspiegel

Abbildung 3.7: Die Herzrate hatte zwei Spitzen bei ca. 70 und 85 Schlägen pro Minute, wobei die Person eine maximale Herzrate von 175 hatte. Der Blutzuckerspiegel betrug durchschnittlich 6 mmol/L und erreichte maximal 12,1 mmol/L.

Bei den Werten der Aktivitäten ist zu sehen, dass der Lifelogger sehr häufig mit einem Verkehrsmittel fuhr, dabei ist jedoch zu beachten, dass es insgesamt 32.787 leere Einträge gab. Im letzten Diagramm ist zu sehen, dass sich die Person vorwiegend in Dublin aufhielt und zwischenzeitlich in Shanghai war. Die anderen Zeitzonen traten sehr selten auf, da die Person sich in diesen jeweils nur auf Grund des Fluges zwischen Dublin und Shanghai befand.



(a) Aktivität (b) Zeitzone

Abbildung 3.8: Bei den Werten der Aktivitäten ist zu sehen, dass der Lifelogger häufig mit einem Verkehrsmittel fuhr. Im letzten Diagramm ist zu sehen, dass sich die Person vorwiegend in Dublin aufhielt und zwischenzeitlich in Shanghai war.

Unter den Metadaten befanden sich einige Ungenauigkeiten und Fehler. Das erste Bild in Abbildung 3.9 zeigt eindeutig die Abfahrtstafel eines Flughafens. Die Metadaten enthielten jedoch die Information, dass sich die Person zu diesem Zeitpunkt an der *East China Normal University* befand.

(a) Flughafen (b) Transport anstatt Walking (c) Transport ist falsch

Abbildung 3.9: Unter den Metadaten befanden sich einige Ungenauigkeiten und Fehler. Bild (a) zeigt eindeutig die Abfahrtstafel eines Flughafens, die Metadaten enthielten jedoch als Ort *East China Normal University*. Auf den beiden anderen Bildern läuft und sitzt die Person, die Metadaten enthielten indessen die Aktivität *transport*.

Auf den beiden anderen Bildern läuft die Person zu ihrem Auto und sitzt in einem Café, die Metadaten enthielten indessen als Aktivität *transport*. Bei dem zweiten Bild stieg die Person erst zwei Aufnahmen später in das Auto ein und fuhr damit, sodass hier eine Ungenauigkeit vorlag.

3.5 Trainingsanfragen

Die Trainingsanfragen dienten zur späteren Evaluation des entwickelten Verfahrens. Zu allen Trainingsfragen waren verschiedene Cluster definiert, aus denen das zu entwickelnde Verfahren jeweils mindestens ein Bild identifizieren sollte. Drei repräsentative Anfragen verschiedener Schwierigkeitsstufen finden im Folgenden nähere Betrachtung. Dabei sind die Anfragen nach diesem Schema formuliert: **Titel**: Beschreibung. *Narrativ*. Alle Anfragen waren in englischer Sprache dargeboten.

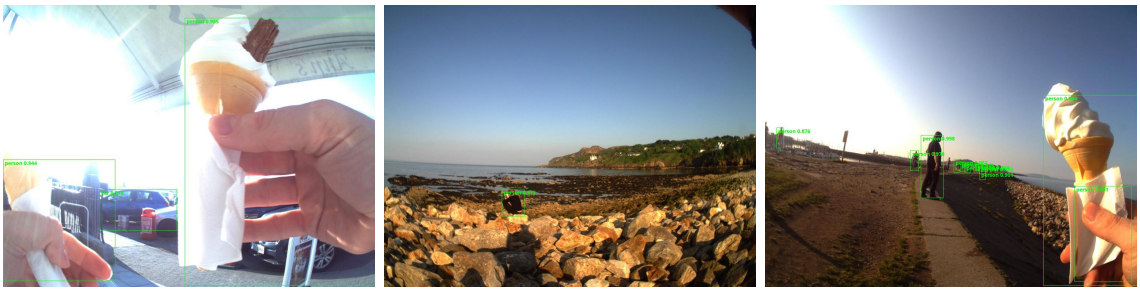
3.5.1 Ground Truth

Für jede Trainingsanfrage existierte eine Liste der gesuchten Cluster inklusive aller sich darin befindenden Bilder. Ein Cluster beschrieb einen Moment inklusive einiger Bilder davor und danach. Die Cluster waren in einer CSV-Datei gespeichert, in welcher zu jeder Anfrage-Id alle Cluster-Ids einschließlich ihren Bezeichnungen, wie zum Beispiel *Canteen*, *KFC* oder *book_reviewing*, zugewiesen wurden. In einer zweiten Tabelle waren für jede Anfrage die möglichen Bild-Ids zu jeder Cluster-Id gegeben, von welchen das Verfahren mehrheitlich mindestens ein Bild finden sollte.

3.5.2 Erste Anfrage

Icecream by the Sea: Find the moment when u1 was eating an icecream beside the sea. *To be relevant, the moment must show both the ice cream with cone in the hand of u1 as well as the sea clearly visible. Any moments by the sea, or eating an ice cream which do not occur together are not considered to be relevant.*

Der Auszug aus dem einzigen verfügbaren Lösungs-Cluster für das erste Thema (siehe Abbildung 3.10) zeigt, dass Eis und Strand nicht gleichzeitig auf dem Bild sein mussten, da auch die umgebenden Bilder relevant waren. Es ist auch erkenntlich, dass das CNN das Eis nicht identifizieren konnte, sondern dieses als *person* labelte. Die Ursache lag daran, dass das Label *icecream* nicht in dem COCO Datenset vorkam. Der einzige Anhaltspunkt war die existierende Kategorie *ice_cream_parlor* (Eisdiele), welche aber nur dem ersten Bild zugewiesen war.



(a) Bild mit Eis ohne Strand (b) Bild mit Strand ohne Eis (c) Bild mit Eis und Strand

Abbildung 3.10: Der Auszug aus dem einzigen verfügbaren Cluster für das erste Thema zeigt, dass das CNN das Eis nicht identifizieren konnte (a, c), sondern dieses als *person* labelte und dass Eis und Strand nicht gleichzeitig auf dem Bild sein mussten (a, b).

3.5.3 Zweite Anfrage

Having food in a restaurant: Find the moment when u1 was eating food or drinking in a restaurant. *U1 was eating food in a restaurant while away from home. Any kinds of dishes are relevant. Only Drinking coffee and have dessert in a cafe won't be relevant.* (Anmerkung: Schreibfehler wurden übernommen.)

Zur zweiten Anfrage waren 14 Cluster definiert. Auf der nächsten Seite sind Auszüge aus drei repräsentativen Clustern dargestellt (Abbildung 3.11). Es ist erkennbar, dass Begriffe wie *pizza* und *dining table* gelabelt wurden. Dadurch war sehr leicht festzustellen, dass es sich um Essen handelte. Die Anfrage richtete sich jedoch an Essen in einem Restaurant, dies ließ sich aber anhand der Begriff ebenfalls leicht ableiten. Alle passenden Kategorien für diese Anfrage waren *fastfood_restaurant*, *restaurant*, *restaurant_kitchen*, *restaurant_patio* (Terrasse), *sushi_bar*, *dining_room*, *dining_hall*, *food_court* und *pizzeria*. Zutreffende Attribute waren *waiting in line* oder *eating* und mögliche Konzepte waren *wine_glass*, *bowl*, *dining_table* oder *pizza*.

(a) DCU Kantine (b) Chinesisches Restaurant (c) McDonalds

Abbildung 3.11: Es ist erkennbar, dass Begriffe wie *pizza* und *dining table* gelabelt wurden (a). Dadurch war es sehr leicht festzustellen, dass es sich um Essen handelte. Der Ort als Restaurant ließ sich anhand anderer Begriffe ebenfalls leicht ableiten.

Weiterhin waren im DCU Kantinen-Cluster Bilder von mehreren Tagen vorhanden. Das war entgegen der Erwartung, dass Momente aus zusammenhängenden Bildern bestehen und erschwerte die Entwicklung eines Verfahrens enorm. Das Verfahren musste also teilweise Momente als einen einzigen Moment ansehen, obwohl sie erwiesenermaßen an unterschiedlichen Tagen stattfanden. Ohne die Beachtung dieses Umstands sank der Cluster Recall deutlich ab, wenn das Verfahren für diese Anfrage nur Bilder von mehreren Momenten aus der Kantine fand.

3.5.4 Vierte Anfrage

Photograph of a Bridge: Find the moment when u1 was taking a photo of a bridge. *U1 was walking on a pedestrian street and stopped to take a photo of a bridge. Moments when u1 was walking on a street without stopping to take a photo of a bridge are not relevant. Any other moment showing a bridge when a photo was not being taken are also not considered to be relevant.*

Bei der vierten Anfrage war nur ein Cluster definiert, welches drei Bilder enthielt. Die Bilder sind in Abbildung 3.12 aufgelistet. Diese Anfrage stellte damit die schwerste aller Trainingsanfragen dar. Wie zu sehen ist, war das dritte Bild das relevanteste. Hier gab es ein weiteres Mal das Problem, dass das Label *bridge* nicht in dem COCO Datenset vorkam und daher gar nicht gelabelt sein konnte. Eine passende Kategorie war *arch* (Bogen), und die Attribute *sunny* und *clouds* ließen zwar auf das letzte Bild schließen, trafen aber auch auf viele andere Bilder zu.

3.5.5 Restliche Anfragen

Es folgen die sieben restlichen Anfragen für das Training. Bis auf die achte Anfrage handelte es sich immer um Momente, die in der Regel häufiger als einmal auftraten. Es gab demnach zwei unterschiedliche Anfragearten: Momente, die üblicherweise nur selten auftreten und alltägliche Momente. Zu ersteren gehörten die erste, vierte

(a) Brücke schwer erkennbar (b) Bild ohne Brücke (c) Relevanteste Bild

Abbildung 3.12: Diese Anfrage stellt die schwierigste aller Trainingsanfragen dar, da lediglich diese drei Bilder gesucht waren und es kein Label im COCO Datenset gab, welches eine Brücke beschrieb.

und achte Anfrage, wie z.B. das Fotografieren einer bestimmten Brücke. Die anderen Anfragen richteten sich eher an Momente des alltäglichen Lebens, wie Kochen, Einkaufen und das Ansehen von Videos.

3. **Watching Videos**: Find the moment when u1 was watching video when using other digital devices. *To be relevant, u1 must be watching videos in any location and any digital devices can be considered. For example: TV machine, tablet, mobile phone, laptop, desktop computer.* Clusteranzahl: 31
5. **Grocery shopping**: Find the moment when u1 was shopping for food in a grocery shop. *To be considered relevant, u1 must be clearly in a grocery shop and bought something from the it.* Clusteranzahl: 12
6. **Playing a Guitar**: Find the moment when U1 or a man is playing guitar in view. *Any use of guitars indoors could be considered relevant. Any type of Guitar could be considered as relevant.* Clusteranzahl: 2
7. **Cooking**: Find moments when u1 was cooking food. *The moments shows U1 was cooking food at any places are relevant.* Clusteranzahl: 11
8. **Car Sales Showroom**: Find the moments when u1 was in a car sales showroom. *u1 visited a car sales showroom a few times. Relevant moments show u1 indoors in a car sales showroom, either looking at cars or waiting for a salesman sitting at a table. Any moments looking at cars while outside of a showroom are not considered relevant.* Clusteranzahl: 2
9. **Public transportation**: Find the moments when U1 is taking the public transportation in any countries. *To be considered relevant,the U1 must take a public transportation to other place. The moments that the U1 is driving a car is not relevant.* Clusteranzahl: 5
10. **Paper or book reviewing**: Find all moments when u1 was reading a paper or book. *To be relevant, the paper or book must be visible in front of U1 and sometimes U1 use a pen to mark on the paper or book.* Clusteranzahl: 7

3.6 Testanfragen

Neben den Trainingsanfragen waren weitere zehn Testanfragen gegeben, deren Cluster allerdings nur die Organisatoren der Challenge besaßen. Daraus folgt, dass keine eigene, automatisierte Evaluation möglich war, wobei die Ergebnisse sich aber trotzdem optisch grob einschätzen ließen.

3.6.1 Erste Anfrage

In a Toyshop: Find the moment when u1 was looking at items in a toyshop. *To be considered relevant, u1 must be clearly in a toyshop. Various toys are being examined, such as electronic trains, model kits and board games. Being in an electronics store, or a supermarket, are not considered to be relevant.*

Bei der ersten Anfrage handelte es sich um einen speziellen Moment, der höchstwahrscheinlich nur einmal im gesamten Zeitraum auftrat. Der Vorteil dieser Anfrage war, dass das Label *toyshop* in den Kategorien vorkam. Außerdem gab es das Attribut *shopping*, welches ähnlich zu Shop war. Aus diesen Gründen sollte diese Anfrage leicht abzuarbeiten sein.

3.6.2 Vierte Anfrage

Watching Football: Find the moments when either u1 or u2 was watching football on the TV. *To be considered relevant, either u1 or u2 must be indoors and watching football on a television. Watching any other TV content is not considered relevant.*

Die vierte Anfrage drehte sich um das Schauen von Fußball auf einem Fernseher. Das Label *tv* existierte in den Konzepten, wurde aber häufig falsch annotiert, wie in den ersten beiden Bildern von Abbildung 3.13 zu sehen ist. In dieser Anfrage kam auch das erste und einzige mal die zweite Person vor, aber wie bereits erwähnt, lohnte es sich wegen dieser einen Stelle nicht, den kompletten Lifelog dieser Person zu analysieren. Der Fernseher war jedoch nur ein Teil der Anfrage, in dem Narrativ hieß es, dass alle Sendungen abseits von Fußball nicht relevant waren. Die Schwierigkeit bestand deshalb darin, zusätzlich das Fußballspiel als solches zu identifizieren. Dafür eigneten sich die Kategorien *stadium/football* und *football_field*, das Konzept *sports_ball*, sowie das Attribut *sports*. Es war jedoch fraglich, ob diese Labels auch den entsprechenden Bildern zugewiesen wurden, da der Fernseher häufig ziemlich klein auf den Bildern abgelichtet war (siehe letztes Bild in Abbildung 3.13). Zudem wurde er häufig überbelichtet und es war nur anhand der grünen Farbe schlussfolgerbar, dass es sich um ein Fußballspiel handelte. Bei dem letzten Bild wurden folgende Kategorien zugeordnet: *recreation_room*, *home_office*, *office*, *attic* (Dachboden) und *art_studio*. Keine dieser ließen auf das Schauen von Fußball schließen. Die zugewiesenen Attribute waren *enclosed area*, *no horizon*, *indoor lighting*, *working*, *wood*, *glass*, *cloth*, *glossy* und *matte*. Auch keines dieser Attribute deutete auf die gesuchte Anfrage hin, womit sie als ziemlich schwierig einzuschätzen war.

(a) Drei Monitore als TV (b) Lampe als TV (c) Gesuchtes Bild

Abbildung 3.13: Das erste Bild zeigt, dass Computermonitore (a) und selbst Lampen (b) häufig als TV gelabelt waren. Bild (c) stellt ein gesuchtes Bild dar, der Fernseher war jedoch ziemlich klein und es gab kein zugewiesenes Label, welches das Fußballspiel korrekt beschrieb.

3.6.3 Neunte Anfrage

Wearing a red plaid shirt: Find the moment when U1 was wearing a red plaid shirt. *To be relevant, the user1 was wearing a red plaid shirt in a day life.*

Die neunte Anfrage stellte die schwierigste Testanfrage dar, da ein rotes Hemd gesucht war, das der Lifelogger getragen hatte. Da die Kamera immer vom Körper weg zeigte, ließ sich dieser Moment nur finden, indem der Lifelogger im Spiegel zu sehen war oder die Arme der Person im Bild waren. Die Erkennung eines karierten Musters stellte eine zusätzliche Schwierigkeit dar. Als Attribut eignete sich *cloth*, aber sonst existierte kein Label, welches das Hemd in irgendeiner Weise beschrieb.

3.6.4 Restliche Anfragen

Es folgen die restlichen Testanfragen. Es war auffällig, dass sich zwei Anfragen um Kaffee drehten, dabei sollten die Bilder von Anfrage fünf in den gesuchten Momenten von Anfrage sieben auch zu finden sein. Die Auswertung des Narrativs der siebten Anfrage stellte das Verfahren vor das gleiche Problem wie bei der neunten Anfrage, da sich Kleidungen sehr schlecht identifizieren ließen. Das Fahren mit dem Auto in Anfrage zwei sollte aufgrund der vorhandenen vielen zutreffenden Labels wie *traffic_light*, *car* oder *car_interior* leicht zu finden sein. Dass es sich jedoch um die Heimfahrt handelte und der Moment damit nachmittags stattfand, war schwierig automatisiert herauszulesen. Die dritte Anfrage stellte eine der einfacheren Anfragen dar, da es das Konzept *refrigerator* gab.

2. **Driving home**: Find any moment when u1 was driving home from the office. *Moments which show u1 is driving home from the office is relevant. Driving from other place and to other place are not relevant.*
3. **Seeking Food in a Fridge**: Find the moments when u1 was looking inside a refrigerator at home. *Moments when u1 is at home and looking inside a*

refrigerator are considered relevant. Moments when eating food or cooking in the kitchen are not considered relevant.

5. **Coffee time**: Find the moment when u1 was having coffee in a cafe. *To be considered relevant, u1 must be in a cafe and having coffee alone or with another individual.*
6. **Having breakfast at home**: Find the moment when u1 was having breakfast at home. *U1 was having breakfast at home and the breakfast time must be from 5:00 am until 9:00 am.*
7. **Having coffee with two person**: Find the moment when u1 was having coffee with two person. *Find the moment when u1 was having coffee with two person. One was wearing blue shirt and the other one was wearing white cloth. Gender is not relevant.*
8. **Using smartphone outside**: Find the moment when u1 was using smartphone when he was walking or standing outside. *To be considered relevant, u1 must be clearly using a smartphone and the location is outside*
10. **Having a meeting in China**: Find all moments when u1 was attending a meeting in China. *To be relevant, the user1 must be in China and was having a meeting with others.*

3.7 Vergleich von Trainings- und Testanfragen

Trainings- und Testanfragen sollten sehr ähnlich sein, weil ein Verfahren, das zwar für die Trainingsanfragen gut funktioniert, nicht zwangsläufig genauso gut für die Testanfragen funktionieren muss. Unterscheiden sich die Anfragen stark voneinander, lassen sich keine Rückschlüsse darüber ziehen, inwieweit die Ursache darin liegt, dass das Verfahren zu stark an die Trainingsanfragen adaptiert ist.
Bei dem Vergleich von Trainings- und Testanfragen fiel auf, dass sie sich vom Detailgrad unterschieden. Beispielsweise war in der zehnten Testanfrage ein Moment in einem anderen Land gesucht („in China“). Das Land spielte bei den Testanfragen jedoch gar keine Rolle (vgl. 9.: „in any countries“). Weiterhin war bei der siebten Testanfrage, nach einer bestimmten Anzahl an Personen gesucht („with two person“), in den Trainingsanfragen gab es allerdings keine Anfragen bei der eine bestimmen Anzahl an Konzepten gesucht war.
In beiden Anfragearten gab es auch Gemeinsamkeiten. Beispielsweise spielte die Aktivität in Trainingsanfrage neun („transportation“) und in Testanfrage acht („walking outside“) eine Rolle. Auch die Suche nach einer Art von Shopping kam in beiden Anfragearten vor (vgl. Trainingsanfrage fünf: „grocery shop“ und Testanfrage eins: „toyshop“). Weiterhin waren digitale Geräte beiderseits relevant (vgl. Trainingsanfrage drei: „using digital devices“ und Testanfrage vier: „on the TV“). Genauso war die Nahrungsaufnahme in beiden Anfragearten wichtig (vgl. Trainingsanfragen sieben: „cooking food“ und zwei: „eating food“ und Testanfrage sechs: „having breakfast“).

4 Entwicklung eines Verfahrens

In diesem Kapitel folgt die Vorstellung des Verfahrens zur automatisierten Anfragenverarbeitung, welches auf Bildsegmentierung und Ähnlichkeitswerten basierte. In diesem berechnete der Computer Ähnlichkeitswerte zwischen den gegebenen Anfragen und Bildern inklusive ihrer Metadaten. Dabei geschah an erster Stelle eine Extraktion von Labels, die diverse Bereiche des Lifelogs beschrieben, um diese im Anschluss mit den Tokens aus den Anfragen zu vergleichen. Zusätzlich erfolgte eine Bildsegmentierung, um ähnliche Momente zusammen zu gruppieren und, um die Gesamtanzahl an Bildern, durch das Entfernen von dünn besetzten Segmenten, zu reduzieren.

4.1 Ressourcen

Für die Entwicklung der Verfahren kamen verschiedenste Ressourcen zum Einsatz, wobei der Fokus dabei auf freier Software lag. In dieser Sektion folgen kurz die verwendeten CNNs inklusive der Datensets und die GloVe-Wortvektoren.

4.1.1 Python als Entwicklungsumgebung

Das Verfahren ist in Python implementiert. Diese Sprache wurde ausgewählt, da sie sich gegenüber anderen Programmiersprachen wie R, MATLAB oder C# vor allem im Bereich der Datenanalyse und neuronalen Netze sehr gut eignet und eine starke Community hat. Die Menge an Code ist zudem deutlich geringer als der Code von z.B. Java oder C++. Ein weiterer Vorteil von Python ist, dass es interpretierbar ist. Somit lassen sich Module als Einheit ausführen, ohne dass das restliche Projekt fehlerfrei sein muss. Nicht zuletzt gibt es sehr viele hilfreiche Bibliotheken, wie NumPy[1], SciPy[2], Scikit-Learn[3], Pandas[4] oder matplotlib[5], die den Umgang mit den großen Mengen an Daten klar erleichtern.
Für die Ausführung des Quellcodes kam ein Intel Core i5-7500T Prozessor in Kombination mit einer NVIDIA GeForce GTX 1070 Mobile Grafikkarte zum Einsatz. Letztere führte gegenüber des Prozessors zu einer deutlichen Beschleunigung der Annotation weiterer Labels.

[1] `http://numpy.org`
[2] `http://scipy.org`
[3] `http://scikit-learn.org`
[4] `http://pandas.pydata.org`
[5] `http://matplotlib.org`

4.1.2 GloVe Wortvektoren

GloVe ist ein unüberwachter Lernalgorithmus, welcher 2014 in Stanford entwickelt wurde. Er ordnet Terme in Form von Vektoren im Raum an, wobei kontextuell zusammengehörende Wörter nah beieinander liegen. Die Terme sind entweder Wörter oder Phrasen, welche eine unabhängige Dimension im Vektorraum bekommen. Dadurch ist es möglich, jeden Text als Vektor darzustellen. [81, 71]
Für das Verfahren kamen vortrainierte GloVe Wortvektoren von Common Crawl zum Einsatz, welche 300 Dimensionen und ein Vokabular von 2,2 Millionen Termen besaßen. Zu allen Termen, die nicht in dem Vokabular vorkommen, lässt sich keine Ähnlichkeit bestimmen.

Kosinus-Ähnlichkeit und Kosinus-Distanz

Ein Maß, welches die Ähnlichkeiten zwischen den Termen angibt, ist die Kosinus-Ähnlichkeit. Sie berechnet sich aus dem Winkel zwischen den Wortvektoren eines Terms mit denen eines anderen Terms. Hat der Kosinus den Wert 1 bedeutet das, dass die beiden Token die gleiche Richtung haben und sehr ähnlich sind. Bei dem Wert 0 stehen die entsprechenden Vektoren orthogonal zueinander und bei dem Wert -1 sind sie entgegengesetzt und haben daher eine gegenteilige Bedeutung. Der Wert liegt immer im Intervall $[-1, 1]$. Die Kosinus-Distanz berechnet sich aus der Kosinus-Ähnlichkeit, durch deren Subtraktion von 1, und liegt damit im Intervall [0, 2]. Je kleiner die Kosinus-Distanz ist, desto ähnlicher sind sich zwei Terme.

4.1.3 Objekterkennungssysteme, CNNs und Datensets

Für die Annotation von weiteren Labels kamen diverse Objekterkennungssysteme zum Einsatz. Dazu zählte YOLOv3 [73], welches einen mAP von 57,9% auf das COCO-Datenset hatte. Ein weiterer Vorteil war, dass es möglich war, das Modell in Richtung Geschwindigkeit oder Genauigkeit ohne erneutes Training anzupassen. YOLOv3 basiert auf dem Neural Network Framework *Darknet*, welches in C und CUDA implementiert wurde. Mit YOLO war es möglich, Modelle zu verwenden, welche auf diverse Datensets vortrainiert wurden. Die Datensets die in dem vorgestellten Verfahren Anwendung fanden, waren das das ImageNet 1000 Datenset [29], das Open Images Datenset [54] und das COCO-Datenset [59].
Das ImageNet 1000 Datenset gibt es seit 2009 und es basiert auf über 14 Millionen per Hand annotierten Bildern aus verschiedenen Situationen, welchen ein Label von 1000 verfügbaren Labels zugewiesen wurde. Das heißt, es handelt sich jeweils um nur ein Konzept pro Bild, wie zum Beispiel das Label *cash machine*. Google entwickelte 2016 das Open Images Datenset, welches nun in der fünften Variante verfügbar ist. In diesem existieren 600 Klassen, wobei es hierbei möglich ist, dass mehrere Labels pro Bild zugewiesen sind, wenn zum Beispiel ein Hund und eine Katze auf einem Bild zu sehen sind, werden beide Tiere gelabelt. Das COCO Datenset steht für Common Objects in Context und wurde 2014 von Microsoft entworfen und enthält 80

verschiedene Objekt-Kategorien und 91 Kategorien für andere Dinge in über 200.000 gelabelten Bildern, bei denen bis zu 5 Labels pro Bild annotiert sind. Das heißt, hier sind wie im Open Images Datenset, im Gegensatz zum ImageNet Datenset, mehrere Labels pro Bild möglich.
Neben YOLO wurde zudem das in Python geschriebene Software System *Detectron* eingesetzt, welches 2018 von Facebooks AI Research Team entworfen wurde. Der benutzte Objekterkennungsalgorithmus basierte auf Mask R-CNN [42], welches wiederum eine Erweiterung des Faster R-CNNs [74] war. Mit Hilfe von Detectron wurden, unter Zuhilfenahme eines vortrainierten Modells, ebenfalls Labels aus dem COCO Datenset annotiert.

4.2 Vorverarbeitung

Die Vorverarbeitung bestand darin, die Labelarten zu definieren, die die Daten aus dem Lifelog beschreiben. Zu diesen wurde zusätzlich die Inverse Dokument Frequenz (IDF) berechnet. Außerdem wurden die Bilder und die Anfragen verarbeitet, damit sie in der darauffolgenden Prediction miteinander verglichen werden konnten.

4.2.1 Definition der Labelarten

Die Grundlage für alle Versuche bildeten Auszüge aus den in 3.3 und 3.4 beschriebenen Labelarten, die unterschiedliche Bereiche des Lifelogs beschreiben. Zusätzlich wurden weitere Labels durch CNNs annotiert und einige gegebene Labels generalisiert, um eine Konzeptvielfalt zu erreichen. Es folgt eine Beschreibung der verwendeten Labelarten.

Beschreibung des Zustandes

Für die Charakterisierung des Zustandes existierte ein Labelart, welcher die Ortsbewegung beschrieb. Er basierte auf den gegebenen Aktivitäts-Metadaten der minutenbasierten Ereignisliste. Leere Einträge wurden ignoriert, sodass sich für die Zustandsbeschreibung die beiden möglichen Labels *transport* und *walking* ergaben.

Beschreibung der Zeit

Die Zeit ließ sich zwar durch die lokale Zeit minutengenau beschreiben, diese Beschreibung wäre jedoch nur nützlich gewesen, wenn Anfragen verfasst worden wären, die nach einer genauen Uhrzeit gefragt hätten. Da dies nicht der Fall war, bot es sich an, Tageszeiten aus der lokalen Zeit zu generieren. Es ergaben sich die vier Labels: *morning* (von 4 bis 12 Uhr), *afternoon* (von 12 bis 17 Uhr), *evening* (von 17 bis 22 Uhr) und *night* (von 22 bis 4 Uhr).

Beschreibung des Ortes

Bildobjekte und die Umgebung gehörten zur Ortsbeschreibung, genauso wie die Ortsnamen der minutenbasierten Ereignisliste und die Städtenamen.
Um die Bildobjekte zu annotieren, wurden die gegebenen Konzepte verwendet, welche die Labels des COCO Datensets enthielten. Mit Hilfe von YOLOv3 und Detectron wurden, wie bereits erwähnt, weitere Labels aus dem COCO Datenset, dem Open Images Datenset und dem ImageNet 1000 Datenset annotiert. Die Intention dieser zusätzlichen Annotation war es, eine detailliertere Bestimmung der zu sehenden Objekte zu erhalten.
Es zeigte sich, dass die annotierten COCO Labels sich geringfügig unterschieden. Eine Generalisierung transformierte die Labels einheitlich um, z.B.: *tvmonitor* → *tv*, *aeroplane* → *airplane* oder *diningtable* → *dining table*. Dieser Schritt war optional, bot sich jedoch an, da sich die drei auf dem COCO Datenset basierenden Labelarten dadurch später besser vergleichen ließen.
Die gegebenen Attribute und Kategorien wurden für die Beschreibung der Umgebung benutzt, da sie auf dem Places 365 Datenset [100] beruhten. Die Kategorien ließen sich in die Kernkategorien und die beiden Labels *indoor* bzw. *outdoor* aufspalten. Beispielsweise wurde das Label *diner outdoor* somit zur Kernkategorie *diner*.
Die Ortsbezeichnung wurde in häufige Orte generalisiert, um festzustellen wo sich der Lifelogger namentlich befand. Es wurden die folgenden 15 Orte definiert: *airport*, *bakery*, *bar*, *cafe*, *college*, *embassy*, *home*, *hotel*, *railway station*, *restaurant*, *shopping centre*, *solicitor*, *store*, *university*, *work*. Die Zuordnung zu diesen Orten geschah indem geschaut wurde, welches dieser Labels in der Ortsbezeichnung enthalten war. Zum Beispiel wurde die Ortsbezeichnung *East China Normal University* zu *university* gemappt. Der Ort *costa coffee* wurde in einer späteren Variante hinzugefügt, da sich zwei der Testanfragen um das Trinken von Kaffee handelten und daher dieser Ort besonders relevant war. Weiterhin wurde die Stadt aus der Zeitzone (z.B. *Europe/Amsterdam*) extrahiert, wodurch sich die sieben Stadt-Labels: *Prague*, *Shanghai*, *Istanbul*, *Berlin*, *Amsterdam*, *London* und *Dublin* ergaben.

Resultierende Labelarten

Es folgt eine Übersicht über alle eingesetzten Labelarten (siehe Tabelle 4.1). Es ist ersichtlich, dass für die Beschreibung des Ortes die meisten Labels verfügbar waren, wohingegen für den Zustand nur zwei und für die Zeit nur vier Labels existierten. Viele der Labels besaßen zudem Scores, welche die Konfidenz pro Label angaben. Für diese Labelarten bestand die Option *threshold*, welche festlegte, ab welcher Konfidenz ein Label als zutreffend galt. Je größer die Konfidenz gewählt wurde, desto weniger Labels gingen hervor. Die konkreten Labels zu den Labelarten sind im Anhang (6) aufgelistet.

Tabelle 4.1: Überblick über die verwendeten Labelarten, der Anzahl an Labels und ob Scores zugewiesen waren. Es ergaben sich insgesamt 13 Labelarten und 1.541 mögliche Labels, welche Ort, Zustand und Zeit beschrieben.

Labelart	**Labelanzahl**	**inkl. Scores**
Aktivitäten	2	nein
Tageszeiten	4	nein
Attribute	97	nein
Kategorien	360	ja
- Kernkategorien	347	ja
- Indoor/Outdoor	2	ja
Gegebene Konzepte	76	ja
Yolo Konzepte	72	ja
Detectron Konzepte	79	ja
Open Images Konzepte	53	ja
ImageNet Konzepte	426	ja
Orte	16	nein
Städte	7	nein

4.2.2 Berechnung der Inversen Dokumentfrequenz

Für die Labels wurde zudem die Inverse Dokumentfrequenz (IDF) bestimmt, um sie später im Vergleich mit den Anfragen zu gewichten. Die IDF war direkt abhängig von der *threshold*, da vereinzelte Labels häufig falsch annotiert waren und daher hohe IDF-Werte mit steigender *threshold* seltener wurden.

4.2.3 Bildverarbeitung

Die Verarbeitung der Bilder bestand in einer optionalen Segmentierung und der darauffolgenden Erzeugung von Bild-Vektoren für jede Labelart.

Segmentierung

Das vorgestellte Verfahren verfügte optional über eine ähnlich wie in 2.2 beschriebene Bildsegmentierung basierend auf anhand von Farbhistogrammen berechneten Distanzmaßen. Die Idee dahinter war, dass die aufeinanderfolgenden Bilder der tragbaren Kamera aufgrund der gleichen Aufnahmeposition ähnlich waren. Die persönlichen Fotos waren für die Segmentierung irrelevant, da sich die Kamera zum Aufnahmezeitpunkt dieser an variierenden Positionen befand.
Die ersten beiden Schritte stellten die tageweise Gruppierung der autografischen Bilder und die anschließende Berechnung der Farbhistogramme dar.

Es folgte ein unüberwachtes hierarchisches agglomeratives Clustering (siehe 2.2.3). Für den intercluster Vergleich fand die Linkage Methode *average* Anwendung, welche den Durchschnitt der Unterschiede zwischen den Histogrammen aus Cluster A mit denen von Cluster B berechnet. Die Unterschiede berechneten sich aus der Euklidischen Distanz der Histogramme. Verschiedene weitere Vergleichsgrundlagen stellten sich in Experimenten als ungeeignet heraus. Darunter fielen das Berechnen der Euklidischen Distanz der Bildpixel, bei dem die Verschiebung des Bildes um einen Pixel jedoch zu einem viel zu großen Unterschied und damit zu keinem erfolgreichen Clustering führte. Die gleiche Problematik trat bei den mit Hilfe des Canny Edge Detection Algorithmus' extrahierten Bildkanten auf, sodass sich diese ebenfalls nicht als Grundlage für das Clustering eigneten.

Beispiel Clustering Zur Veranschaulichung soll das Dendogramm in Abbildung 4.1 dienen, welches des Ergebnis eines beispielhaften Clusteringvorgangs zeigt. Gegeben sind die fünf Histogramme 0 bis 4 mit den Werten 87, 25, 41, 99 und 29. Jede dieser fünf Zahlen repräsentiert dabei ein Histogramm (vereinfacht; in der Praxis besteht ein Histogramm aus einem 256-dimensionalen Zahlenarray). Jede Zahl befindet sich in einem eigenen Cluster, z.B. (1) oder (3). Diese lassen sich auf Grund der niedrigen euklidischen Distanz zu einem übergeordneten Cluster zusammenfassen z.B. (1, 4) oder (0, 3). Der Abstand zwischen (1) und (4) ist in diesem Beispiel $29 - 25 = 4$ (siehe Y-Achse). Im nächsten Schritt kombiniert der Algorithmus das entstandene Cluster mit Cluster (2), da der durchschnittliche Abstand von (2) zu (1, 4) den Wert $(41 - 25 + 41 - 29) \div 2 = (16 + 12) \div 2 = 14$ hat und damit geringer als der Abstand zum Cluster (0, 3) ist. Das Zusammenfügen geschieht solange, bis kein Cluster mehr übrig ist.

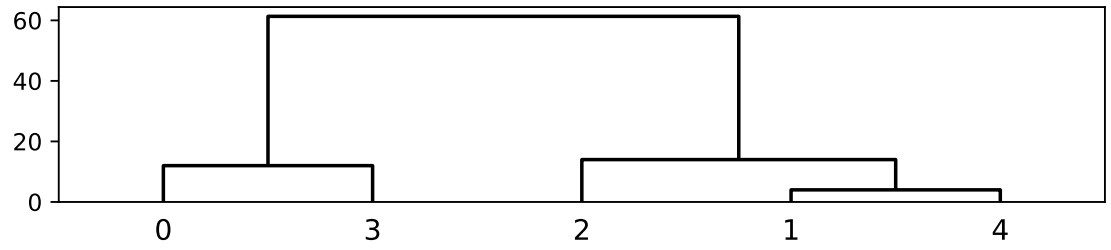


Abbildung 4.1: Das Dendogramm soll den Prozess des Clusterings veranschaulichen. Gegeben sind die Histogramme 0 bis 4 mit den Werten 87, 25, 41, 99 und 29. Jede dieser fünf Zahlen repräsentiert dabei ein Histogramm. Der Algorithmus fasst ähnliche Cluster in ein übergeordnetes Cluster zusammen, solange bis kein Cluster mehr übrig ist. Der Vergleich von Clustern geschieht durch Bilden des durchschnittlichen Abstandes der Bilder eines Clusters mit denen des anderen Clusters.

Der nächste Schritt bestand darin, die Cluster abzuflachen, sodass es keine übergeordneten Cluster mehr gab und jedes Bild nur einem Cluster zugewiesen war.

Diejenigen Histogramme, die eine euklidische Distanz unter einer zuvor definierten Threshold hatten, kamen in ein gemeinsames Cluster. In dem obigen Beispiel ist der Wert 30 als Threshold sinnvoll, wodurch sich nach dem genannten Schritt die beiden Cluster (0, 3) und (2, 1, 4) ergeben würden.
Ein noch existierendes Problem war, dass die Bilder der Histogramme aus den neuen Clustern nicht notwendigerweise zusammenhängend sein mussten. So waren zum Beispiel die Momente, bei denen die Person früh zur Arbeit und abends nach Hause fuhr, in dem gleichen Cluster. Es folgte deshalb die Einteilung der Bilder eines Clusters in einzelne Segmente, welche aus zusammenhängenden Bildern bestanden. Angewandt auf das Beispiel, würden sich dadurch die zwei Cluster ((0), (3)) und ((1, 2), (4)) ergeben, welche aus jeweils zwei Segmenten bestehen.
Um die Anzahl an Segmenten zu reduzieren, ließen sich diejenigen Segmente entfernen, die weniger als eine zuvor definierten Menge an Bildern enthielten. Bei Segmenten mit nur einem Bild handelte es sich häufig um Kameraverdeckungen oder -schwenkern.
Nach diesem Schritt konnte optional ein erneutes Durchführen der vorherigen Schritte, durch erneutes Clustering, Abflachen und Segmentieren aller aus dem letzten Schritt resultierenden Bilder, erfolgen, allerdings mit einer größeren Threshold. Der Sinn dahinter war es, durch eine erste niedrige Threshold unscharfe Bilder direkt zu entfernen, um die ähnlichen Bilder anschließend mit der größeren Threshold in ein gemeinsames Segment zusammenzufügen, um schlussendlich die Segmentanzahl erneut zu reduzieren.
Eine letzte Option ermöglichte es, naheliegende Segmente zusammenzufügen, welche zum Beispiel durch einen Kameraschwenker voneinander getrennt wurden. Dazu wurde eine Bilderanzahl definiert, die zwischen dem letzten Bild eines Segmentes und dem ersten Bild des darauffolgenden Segmentes maximal sein durfte.

Veranschaulichung

Es folgt eine Veranschaulichung der Segmentierung ohne die optionale, zweite Filterung. In der Abbildung 4.2 ist der Schritt nach dem Einlesen der Bilder dargestellt. Die drei Graustufen symbolisieren dabei die Ähnlichkeiten, zum Beispiel sind Bilder 1, 2, 5 und 6 ähnlich zueinander. Im nächsten Schritt (siehe Abbildung 4.3) wurden die Bilder in sieben Cluster eingeteilt. Anschließend, in Abbildung 4.4 ist das Ergebnis des dritten Schritts zu sehen. Die Cluster wurden abgeflacht und in Segmente eingeteilt. Im vorletzten Schritt wurden alle schwach besetzten Segmente entfernt (siehe Abbildung 4.5). Die letzte Abbildung 4.6 stellt alle finalen Segmente dar.

Abbildung 4.2: Zuerst wurden die Bilder einlesen. Die drei Graustufen symbolisieren Ähnlichkeiten, zum Beispiel sind Bilder 1, 2, 5 und 6 ähnlich.

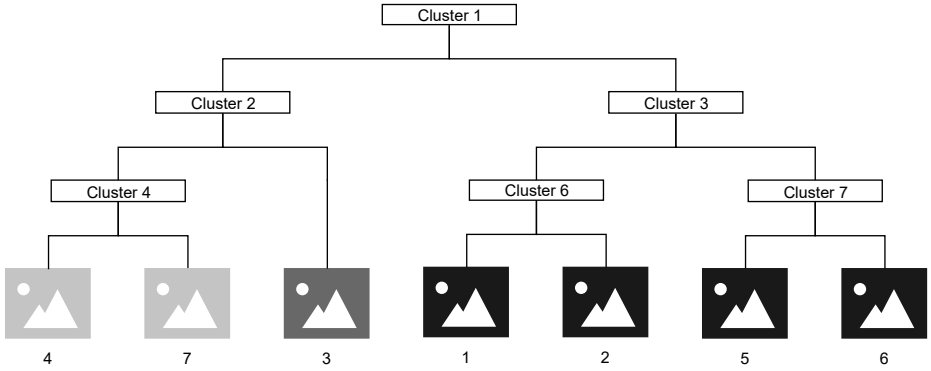


Abbildung 4.3: Dargestellt ist der zweite Schritt nach dem Clustering.

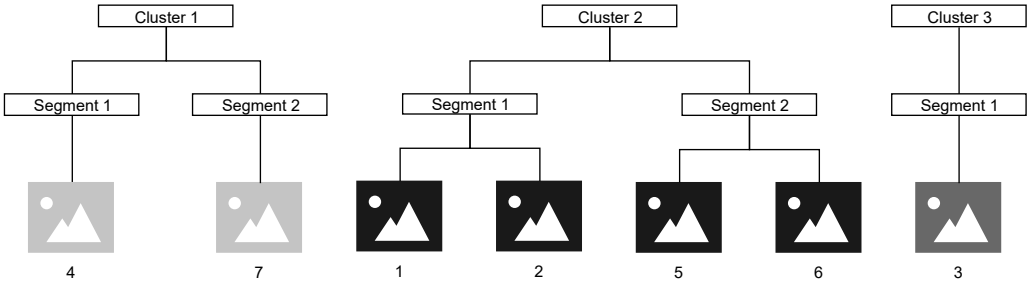


Abbildung 4.4: Die Abbildung zeigt die Bilder nach dem Abflachen und Umwandeln der Cluster in Segmente.

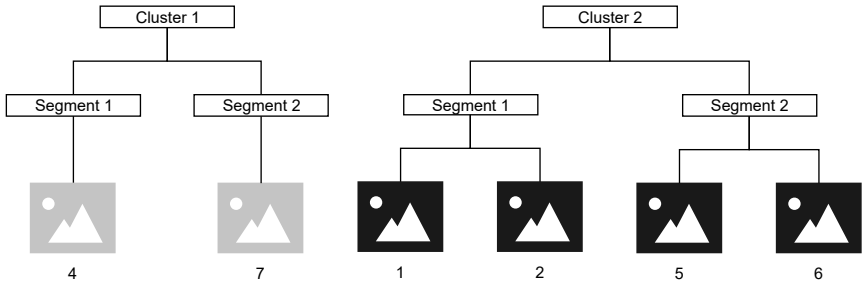


Abbildung 4.5: Im vorletzten Schritt wurden alle schwach besetzte Segmente entfernt (Cluster 3 - Segment 1).

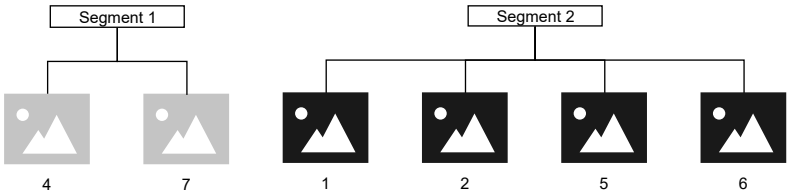


Abbildung 4.6: Zuletzt wurden Segmente mit wenig Abstand zueinander zusammengefügt und aus den Clustern extrahiert.

Vektorerzeugung

Für jedes Bild bzw. Segment wurde pro Labelart ein Vektor berechnet, dessen Dimension gleich der Anzahl an jeweils möglichen Labels entsprach. Der i-te Eintrag des Vektors repräsentierte das i-te Label aus den sortierten möglichen Labels einer Labelart.
Die bildbasierte und segmentbasierte Erzeugung lief wie folgt unterschiedlich ab.
Das Verfahren suchte bei der bildbasierten Vektorerzeugung zu jedem Bild die zugewiesenen Labels heraus und wies dem Vektor der zugehörigen Labelart, an den entsprechenden Positionen bei Annotation eine Eins und andernfalls eine Null zu. Für die Labelarten mit Scores wurde dieser anstelle der Eins übernommen.
Bei der segmentbasierten Vektorerzeugung legte ein Wert fest, welche Bilder einbezogen werden sollten. Bei der Option *first* oder *last* nahm der Algorithmus jeweils das erste bzw. das letzte Bild aus einem Segment heraus und bildete den Vektor wie bei der bildbasierten Erzeugung. Die dritte Option *all* unterschied sich insofern von den beiden anderen, dass der Computer jedes Bild eines Segmentes der Reihe nach durchging und dabei das Maximum des Scores eines Labels mit dem bereits im Vektor stehenden Wert bildete und dieses dann übernahm. Falls kein Wert vorhanden war, übernahm er den Score direkt.

4.2.4 Anfragenverarbeitung

Die Verarbeitung der Anfragen bestand darin, diese zuerst zu tokenisieren, optional danach diese zu clustern und schlussendlich in Vektoren zu transformieren.

Definition der Anfrage

Der erste Schritt bestand darin, festzulegen, was als Anfrage gelten sollte. Es konnte zwischen dem Titel, der Beschreibung und dem Narrativ gewählt werden. Außerdem war es möglich alle drei Teile der Anfrage zusammen zu kombinieren oder manuell eine Anfrage zu definieren.

Tokenisierung

Im nächsten Schritt erfolgte die Tokenisierung der Anfrage. Hierbei wurde zuerst die Satzzeichen entfernt, das heißt vorwiegend Kommas und Punkte, aber auch Sonderzeichen wie Doppelpunkte oder Semikolons. Anschließend wurden alle Wörter in Kleinbuchstaben transformiert, um sie zu vereinheitlichen, da im Titel viele Wörter groß und in der Beschreibung fast alle Wörter klein geschrieben wurden. Mit Hilfe der NLTK Bibliothek wurden die Tokens extrahiert und einige dieser herausgefiltert. Die Filterung umfasste Stoppwörter, Duplikate, anfragetypische Wörter und den Wortvektoren unbekannte Token, die nicht im Vokabular waren. Als anfragetypische Wörter galten irrelevante Wörter, die keinen Mehrwert für den Inhalt boten: *find*, *moment*, *moments*, *u1* und *u2*. Zum Schluss wurden alle übrig gebliebenen Token zur besseren Übersicht aufsteigend sortiert.

Die Ergebnisse der tokenisierten Anfragen sind im Anhang unter 6 zu finden. Die Ergebnisse für die Trainingsanfragen zeigen, dass die Beschreibung Token enthält, die zu keiner Labelart ähnlich sind, wie beispielsweise *view*, *taking*, *using* oder *beside*. Unter den Token der Testanfragen gab es ebenfalls Token, die keine klare Repräsentation in einer Labelart hatten, wie zum Beispiel *using*, *two*, *items*, *plaid* oder *red*.

Tokenclustering

Die Tokens wunden optional Clustern zugewiesen, um ähnlich Tokens zu gruppieren und damit Abfragen mit Oder-Verknüpfungen besser zu verstehen. Das Clustering erfolgte ähnlich zu dem der Bildsegmentierung, wobei es hier andere Parameter gab. Mit Hilfe der Wortvektoren konnte die Kosinus-Distanz zwischen zwei Tokens berechnet werden. Sie wurde genutzt, um die gebildeten Cluster abzuflachen, indem alle Tokens in ein Segment kamen, welche eine maximale Kosinus-Distanz von einer angegebenen Threshold hatten.

Beispiel In der zweiten Trainingsanfrage „Find the moment when u1 was eating food or drinking in a restaurant.“ wurden die Beschreibungstoken *eating*, *food*, *drinking* und *restaurant* extrahiert. Ein Bild mit einem Getränk, welches in einem Restaurant geschossen wurde, würde ohne Clustering zwar ähnlich zu den letzten beiden Tokens sein, aber würde trotzdem schlechter bewertet werden, als ein Bild welches zudem noch Essen zeigt. Der Grund dafür ist, dass das nicht vorhandene Essen auf dem ersten Bild zu einer niedrigeren Ähnlichkeit führt, wenn davon ausgegangen wird, dass die der Durchschnitt der Ähnlichkeiten gebildet wird. Da in der Frage aber *eating* und *food* oder-verknüpft sind, sollten die beiden Bilder gleich ähnlich sein. Durch das Berechnen der maximalen Ähnlichkeit pro Cluster lässt sich dies realisieren. Es könnten sich beispielsweise zwei Cluster bilden mit (eating, food, drinking) und (restaurant).
Mehr zu der Ähnlichkeitsberechnung folgt in der nächsten Sektion.

Vektorerzeugung

Im letzten Schritt wurden Vektoren für jedes Token, in der gleichen Länge und im gleichen Stil wie die Bild-Vektoren, also entsprechend der Anzahl an Labels pro Labelart, gebildet. Die Werte in dem Vektor waren die mit Hilfe der Wortvektoren berechnete Kosinus-Ähnlichkeiten zwischen dem Token und jedem der Labels. Wurden für eine Anfrage beispielsweise vier Token extrahiert, dann ergaben sich für diese Anfrage vier Token-Vektoren pro Labelart.

4.3 Prediction

Bei der Prediction wurden die zuvor erzeugten Vektoren miteinander verrechnet, sodass sich ein Ähnlichkeitswert zwischen jedem Bild zu jeder Anfrage ergab. Ein

Zwischenschritt zum finalen Ähnlichkeitsmaß bestand darin, Ähnlichkeitsmaße pro Bild und Token einer Anfrage für jede Labelart zu berechnen. Auf diese Methode wird zuerst eingegangen. Im Anschluss folgt die Erklärung für die Verrechnung der einzelnen Ähnlichkeitsmaße zu einem übergreifenderen Ähnlichkeitsmaß. Für die Beschreibung der Methodik werden einige mathematische Variablen für ein besseres Verständnis aufgestellt.
Die Labelarten lassen sich durch P und Q_p beschreiben:

$$\begin{aligned} P &:= \text{Anzahl an Labelarten} \\ Q_p &:= \text{Anzahl an Labels der } p\text{-ten Labelart}, p \in \{1, \dots, P\} \end{aligned}$$

Die Anzahl der Anfragen, ihre Token und die Token-Vektoren lassen sich mathematisch wie folgt ausdrücken:

$$\begin{aligned} M &:= \text{Anzahl an Anfragen} \\ O_m &:= \text{Anzahl an Token der } m\text{-ten Anfrage}, m \in \{1, \dots, M\} \\ T_{p,o_m} &:= \text{Token-Vektor für die } p\text{-te Labelart und das } o\text{-te Token der } m\text{-ten Anfrage}, \\ &\quad p \in \{1, \dots, P\}, m \in \{1, \dots, M\}, o_m \in \{1, \dots, O_m\}, T_{p,o_m} \in [-1, 1]^{Q_p} \end{aligned}$$

Für die Bilder/Segmente lassen sich die Variablen N und $B_{p,n}$ definieren:

$$\begin{aligned} N &:= \text{Anzahl an Bildern} \\ B_{p,n} &:= \text{Bild-Vektor für die } p\text{-te Labelart und das } n\text{-te Bild} \\ &\quad p \in \{1, \dots, P\}, n \in \{1, \dots, N\}, B_{p,n} \in [0, 1]^{Q_p} \end{aligned}$$

4.3.1 Vektorvergleich

Nachdem die Vektoren für die Bilder und die Anfrage-Tokens berechnet waren, ließ sich zwischen diesen ein Ähnlichkeitsmaß berechnen. Die Berechnung erfolgte für jeden Bild-Vektor mit jedem Token-Vektor einer Anfrage für jeweils alle Labelarten. Das heißt zum Beispiel würden mit 13 Labelarten, für eine Anfrage mit vier Tokens, pro Bild $13 \times 4 = 52$ Ähnlichkeitsmaße berechnet werden. Für 10 Anfragen mit durchschnittlich 4 Tokens und 63.696 Bildern hätten sich $52 \times 10 \times 63.696 = 33.121.920$ Ähnlichkeitsmaße ergeben.

Berechnung der Ähnlichkeitsmaße

Zuerst wurden pro Labelart IDF-Vektoren gebildet, die die IDF Werte der Labels enthielten. Alle drei Vektorarten (Token-Vektoren, Bild-Vektoren und IDF-Vektoren) hatten damit die gleiche Dimension.

$$I_p := \text{IDF-Vektor für die Labels der } p\text{-ten Labelart}, p \in \{1, \dots, P\}, I_p \in [1, \infty)^{Q_p}$$

Der nächste Schritt bestand darin, diejenigen Einträge der Vektoren aller Vektorarten zu ignorieren, bei denen an der jeweiligen Stelle des Bild-Vektors eine Null, also kein Label zugewiesen war. Die verkleinerten Vektoren erhalten im Folgenden eine Tilde und sind zwangsläufig zusätzlich vom n-ten Bild abhängig:

$$\begin{aligned} R_{p,n} &:= \text{Anzahl an Nicht-Null-Einträgen aus } B_{p,n} \\ \widetilde{B}_{p,n} &:= B_{p,n} \text{ ohne die Null-Einträge, } \widetilde{B}_{p,n} \in (0,1]^{R_{p,n}} \\ \widetilde{T}_{p,o_m,n} &:= T_{p,o_m} \text{ ohne die Null-Eintragsindizies aus } B_{p,n}, \\ &\quad\ \widetilde{T}_{p,o_m,n} \in [-1,1]^{R_{p,n}} \\ \widetilde{I}_{p,n} &:= \text{IDF-Vektor ohne die Null-Eintragsindizies aus } B_{p,n}, \\ &\quad\ \widetilde{I}_{p,n} \in [1,\infty)^{R_{p,n}} \end{aligned}$$

Im dritten Schritt ließ sich durch eine Option angeben, dass die Scores der Bild-Vektoren ignoriert und auf Eins aufgerundet werden sollten. Dies war nur sinnvoll bei Bild-Vektoren für Labelarten, die einen Score besaßen, da andernfalls die Vektoren bereits nur aus Einsen bestanden.

$$\widetilde{B}_{p,n} \leftarrow \lceil \widetilde{B}_{p,n} \rceil$$

Um die negativen Werte in den Token-Vektoren zu entfernen, wurden alle Werte mit Eins addiert und anschließend durch Zwei dividiert. Dadurch lagen die Werte im Intervall [0, 1], wobei Token und Label bei Eins am ähnlichsten waren.

$$\widetilde{T}^{(i)}_{p,o_m,n} \leftarrow \frac{\widetilde{T}^{(i)}_{p,o_m,n} + 1}{2}, i \in \{1,\dots,R_{p,n}\}, \widetilde{T}_{p,o_m,n} \in [0,1]^{R_{p,n}}$$

Danach wurde ein Differenz-Vektor durch das Berechnen der absolute Differenz von den Bild-Vektoren mit den Token-Vektoren gebildet. Je kleiner die Werte in diesem Vektor waren, desto ähnlicher waren Bild und Token.

$$D_{p,o_m,n} := |\widetilde{B}_{p,n} - \widetilde{T}_{p,o_m,n}|$$

Für den Differenz-Vektor ließ sich optional mit Hilfe des IDF-Vektors ein Boosting-Vektor berechnen, welcher Ähnlichkeiten und Unähnlichkeiten je nach IDF durch Multiplikation verstärkte. Dazu definierte eine Threshold, bis zu welcher Ähnlichkeit der Wert des Differenz-Vektors verbessert (= verkleinert) bzw. ab welcher Unähnlichkeiten der Wert des Differenz-Vektors verschlechtert (= vergrößert) werden sollte. Jeder Eintrag des Differenz-Vektors wurde mit der Threshold verglichen und bei unterschreiten der Threshold wurde in den Boosting-Vektor das Reziproke der dem Label entsprechenden IDF geschrieben, andernfalls wurde kein Reziproke gebildet, sondern die IDF wie sie war übernommen.

$$t := \text{Boosting-Threshold}, t \in (0,1)$$

$$A_{p,o_m,n} := \begin{pmatrix} a_1 \\ \vdots \\ a_{R_{p,n}} \end{pmatrix}, a_i = \begin{cases} \frac{1}{\widetilde{I}^{(i)}_{p,n}} : & D^{(i)}_{p,o_m,n} < t \\ \widetilde{I}^{(i)}_{p,n} : & \text{sonst} \end{cases}, i \in \{1, \ldots, R_{p,n}\}$$

War das Boosting nicht gewünscht, bestand der Boosting-Vektor aus Einsen.

$$A_{p,o_m,n} := \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} \in \mathbb{R}^{R_{p,n}}$$

Der Vektor wurde in diesem Schritt jedoch noch nicht an den Differenz-Vektor multipliziert, da zuerst ein polynomiales Boosting erfolgte (nächster Schritt). Ein Beispiel soll die spätere Multiplikation verdeutlichen: Für ein seltenes Label mit einem IDF von Fünf wurde eine hohe Ähnlichkeit von $0,1$ zum Beispiel zu $0,1 \div 5 = 0,02$ verkleinert, wohingegen ein häufiges Label mit einem IDF von Zwei nur eine Verkleinerung auf $0,1 \div 2 = 0,05$ erfuhr. Andersherum wurde jedoch auch für das seltene Label eine Unähnlichkeit von $0,9$ auf $0,9 \times 5 = 4,5$ verstärkt, wohingegen für das häufigere Label lediglich eine Vergrößerung auf $0,9 \times 2 = 1,8$ erfolgte.
Nach der Berechnung des Boosting-Vektors bestand eine weitere Option darin, Ähnlichkeiten und Unähnlichkeiten polynomial zu verändern. Dazu war der Parameter g festzulegen:

$$g := \text{polynomialer Boosting-Faktor}$$
$$D^{(i)}_{p,o_m,n} \leftarrow (D^{(i)}_{p,o_m,n})^g$$

Durch das Setzen von g auf beispielsweise die Zahl Zwei ließen sich die Werte des Differenz-Vektors je näher sie an $0,5$ lagen, umso stärker verringern. Der Wert $0,5$ wurde beispielsweise um $0,25$ auf $0,25$ verringert, wohingegen die Werte $0,1$ bzw. $0,9$ nur um $0,09$ auf $0,01$ bzw. $0,81$ verringert wurden.
Im Anschluss erfolgte die Bildung des Hadamard-Produktes zwischen dem Vektor und dem Boosting-Vektor.

$$D_{p,o_m,n} \leftarrow D_{p,o_m,n} \circ A_{p,o_m,n}$$

Im darauf folgenden Schritt wurde die Norm des Differenz-Vektors gebildet, welche ein Maß für die Ähnlichkeit darstellte. Je größer die Dezimalzahl war, desto unähnlicher waren sich Token und Bild. Die Norm lag jedoch nicht im Intervall von $[0,1]$, weshalb sie daraufhin durch Bilden eines Teilers in das genannte Intervall transformiert wurde. Der Teiler bestand aus der Norm des mit g potenzierten

IDF-Vektors. Sie stimmte mit der maximal möglichen Norm des Boosting-Vektors überein, nämlich in dem Fall, wenn alle Werte größer gleich der Boosting-Threshold waren. Da im ursprünglichen Differenz-Vektor alle Werte im Intervall $[0, 1]$ lagen, konnten sich durch die Multiplikation mit dem Boosting-Vektor keine größeren Vektor-Werte als die des Boosting-Vektors bilden und demnach konnte die Norm nicht größer als die des Boosting-Vektors sein. Da durch die optionale Potenzierung die Werte des Differenz-Vektors jedoch abnahmen, war es sinnvoll, diese Potenzierung vor dem Bilden der Norm ebenfalls auf die IDF-Vektoren anzuwenden. Die finale Ähnlichkeit berechnete sich durch Abziehen des zuvor gebildeten Quotienten von 1, da es sich bei diesem um ein Unähnlichkeitsmaß handelte.

$$\widetilde{I}^{(i)}_{p,n} \leftarrow (\widetilde{I}^{(i)}_{p,n})^g$$

$K_{n,o_m,p}$ = Ähnlichkeitsmaß für das n-te Bild und dem o-ten Token der m-ten Anfrage für die p-te Labelart, $n \in \{1, \ldots, N\}$, $m \in \{1, \ldots, M\}, o_m \in \{1, \ldots, O_m\}, p \in \{1, \ldots, P\}$,

$$K_{n,o_m,p} := 1 - \frac{\|D_{p,o_m,n}\|}{\|\widetilde{I}_{p,n}\|}, K_{n,o_m,p} \in [0, 1], \text{je größer der Wert, desto höher ist die Ähnlichkeit}$$

4.3.2 Ähnlichkeitsberechnungen

Der nächste Schritt war, die Ähnlichkeitsmaße zu verrechnen, sodass für jedes n-te Bild/Segment zu jeder m-ten Anfrage ein Wert existierte, der die Ähnlichkeit angab. Dies geschah mit Hilfe einer Ähnlichkeitsmatrix $T_{m,n}$ pro Bild-Anfragen-Paar, welche alle zugehörigen Ähnlichkeitsmaße enthielt. Die Zeilen standen für die p-te Labelart und die Spalten repräsentierten das o_m-te Token.

$$(T_{m,n})_{ij} := K_{n,j,i}, m \in \{1, \ldots, M\}, n \in \{1, \ldots, N\}, i \in \{1, \ldots, P\}, j \in \{1, \ldots, O_m\}$$

Für die Token wurden im nächsten Schritt optional Gewichte berechnet. Ziel war es, in die Ähnlichkeitsberechnung einzubeziehen, wie gut eine Labelart für bestimmte Token geeignet war. Da zum Beispiel das Token *eating* mit der *Städte*-Labelart wenig zu tun hatte, sollte sie für dieses Token nichts zur Berechnung beitragen. Die Gewichte wurden durch das Bilden aller Kosinus-Ähnlichkeiten zwischen einem Token und allen Labels einer Labelart berechnet. Die größtmögliche Ähnlichkeit wurde als Gewicht genommen. Da es sein konnte, dass diese negativ war, wurde das Gewicht in diesem Fall auf Null gesetzt.

$(W_m)_{ij}$:= maximale Kosinus-Ähnlichkeit zwischen i-ter Labelart und j-ten Token, $m \in \{1, \ldots, M\}, i \in \{1, \ldots, P\}, j \in \{1, \ldots, O_m\}, (W_m)_{ij} \in [-1, 1]$

$$(W_m)_{ij} \leftarrow \max((W_m)_{ij}, 0), (W_m)_{ij} \in [0, 1]$$

Weiterhin ließen sich die Labelarten optional manuell, tokenübergreifend gewichten. Die Gewichte wurden in einem Vektor zusammengefasst:

$$X_i := \text{Gewicht für die } i\text{-te Labelart}, i \in \{1, \ldots, P\}, X_i \in [0, 1]$$

Sollte jeweils keine automatische/manuelle Gewichtung erfolgen, bestand die Matrix bzw. der Vektor aus Einsen:

$$(W_m)_{ij} := \begin{pmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{pmatrix} \in \mathbb{R}^{P \times O_m}, X_i := \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} \in \mathbb{R}^P$$

Die beiden Gewichtsarten ließen sich nun durch Multiplikation in die Ähnlichkeitsmatrix integrieren:

$$(T_{m,n})_{ij} \leftarrow X_i \cdot (W_m)_{ij} \cdot (T_{m,n})_{ij}, m \in \{1, \ldots, M\}, i \in \{1, \ldots, P\}, \\ j \in \{1, \ldots, O_m\}, (T_{m,n})_{ij} \in [0, 1]$$

Anhand der resultierenden Matrix ließen sich drei verschiedene Ähnlichkeitsmaße berechnen. Der erste Wert war der Durchschnittswert der Matrix, das heißt, es wurde die durchschnittliche Ähnlichkeit berechnet. Bei der zweiten Variante wurden die Durchschnitte der Zeilenmaxima gebildet, das heißt, es wurde für jede Labelart die größte Ähnlichkeit über alle Token herausgesucht und der Durchschnitt berechnet. Die letzte Variante bestand darin, die Durchschnitte der Spaltenmaxima zu bilden, es wurde also die größte Ähnlichkeit eines Tokens über alle Labelarten extrahiert und der Durchschnitt berechnet.
Eine Alternative stellte die Verwendung der in 4.2.4 beschriebenen Tokencluster dar. Dafür wurden die drei Ähnlichkeitsmaße für jedes Cluster separat berechnet und anschließend wurde aus diesen entweder der Durchschnitt oder als zweite Option das Maximum berechnet.
Es resultierte ein Ähnlichkeitsmaß pro Bild/Segment und Anfrage:

$$S_{n,m} := \text{Ähnlichkeitsmaß für das } n\text{-te Bild und der } m\text{-ten Anfrage} \\ n \in \{1, \ldots, N\}, m \in \{1, \ldots, M\}$$

4.3.3 Auswahl repräsentativer Bilder

Unter der Verwendung der Segmentierung, war bis zu diesem Schritt zu jedem Segment ein Ähnlichkeitsmaß berechnet. Da ein Segment jedoch in der Regel aus mehreren Bildern bestand, war es sinnvoll, ein repräsentatives Bild des Segmentes auszuwählen, da mit hoher Sicherheit alle Bilder die gleiche Situation darstellten und nur ein Bild pro Cluster, also pro Moment, gefordert war. Dazu existierten die Optionen, entweder jeweils das erste Bild, oder das letzte Bild, oder alle Bilder eines Segmentes auszuwählen.

4.3.4 Durchführung der XGB-basierten Prediction

Ein Ansatz, welcher auf maschinellem Lernen basierte, bestand darin, XGBoost [18] für die Vorhersagen zu verwenden. Dazu waren ein Trainings-, Evaluierungs- und Testset notwendig. Das Trainingsset wurde zum Trainieren benutzt, wobei in jedem Trainingsschritt anhand des Evaluierungssets festgestellt wurde, ob XGBoost sich zu stark an die Daten anpasste (Overfitting).

Berechnung der Ähnlichkeiten

Zuerst wurde jeweils für die Trainings- und Testanfragen ein Modell erstellt, welches die Ähnlichkeitsmaße berechnete (wie in 4.3.1 beschrieben). Danach wurden die Ähnlichkeitsmaße für beide Modelle wie in 4.3.2 verrechnet, wobei hier nur die ersten beiden Varianten *Zeilendurchschnitte* oder *Zeilenmaxima* möglich waren. Aus diesen Werten wurde jedoch kein Gesamtdurchschnitt berechnet, sondern sie fanden in Form eines Zeilen-Ähnlichkeits-Vektors weitere Verarbeitung. Die Dimension des Vektors entsprach demzufolge der Anzahl an Labelarten.

Erzeugung von Datensets

Die beiden Modelle wurden in einem dritten Modell weiterverwendet, indem aus den Ähnlichkeitsmaßen Trainings-, Evaluierungs- und Testsets erzeugt wurden. Für jede Anfrage und für jedes Cluster wurden die Ids der Bilder aus der Ground Truth herausgesucht und dem Trainings- bzw. Evaluierungsset zugewiesen. Dabei definierte eine Dezimalwert im Intervall [0, 1], wie viel Prozent der Positives sich im Evaluierungsset befinden sollten. Anschließend filterte das Verfahren bei dem segmentbasierten Ansatz diejenigen Bilder heraus, die nicht in den auserwählten Bildern (siehe 4.3.3) vorkamen. Danach wurden zu jeder Anfrage irrelevante Bilder herausgesucht, welche in keinem Cluster vorkamen. Dabei war es möglich, dass Bilder aus Clustern von anderen Anfragen eingeschlossen waren. Weiterhin ließ sich durch eine Threshold festlegen, wie viel der irrelevanten Bilder in die Datensets übernommen werden sollten, da es sich um eine sehr große Menge handelte und sich das Verhältnis zwischen Positives und Negatives dadurch besser ausgleichen ließ. Die Positives und Negatives wurden danach kombiniert und es wurden zu den entsprechenden Bild-Ids die Zeilen-Ähnlichkeits-Vektoren zugewiesen. Diese dienten als späteres Trainings- bzw. Evaluierungsset. Zu den Positives wurde eine Eins und zu den Negatives eine Null als Zielvariable (Y) zugewiesen. Nachdem die Datensets für das Training erzeugt wurden, folgte die Erstellung des Testsets. Dazu wurden die aus dem zweiten Modell berechneten Ähnlichkeitsmaße genommen.

Training

Da alle Datensets erstellt waren, begann das Training basierend auf logistischer Regression und für alle Anfragen auf einmal. Die Trainingsparameter wurden festgelegt. Es ließ sich definieren, wie tief der Algorithmus die Verzweigungen (Bäume)

erzeugen sollte, und nach wie vielen vergeblichen Versuchen der Erhöhung des Evaluationsscores das Training beendet werden sollte. Dieser Score berechnete sich aus dem Logloss der Predictions.

Prediction

Die eigentliche Prediction folgte im Anschluss, indem zuerst das Evaluierungsset predictet wurde, um weitere Evaluationsmetriken zu berechnen. Danach lief die Prediction für das Testset ab. In beiden Fällen wurde jede Anfrage für sich predictet. Das Ergebnis war ein Wert im Intervall $[0,\ 1]$ für jedes Bild-Anfrage-Paar, welcher die Ähnlichkeit angab.

4.4 Postprocessing

Im Postprocessing wurden die Ähnlichkeitswerte absteigend sortiert und optional tageweise gestaffelt.

4.4.1 Sortierung

Da die berechneten Ähnlichkeitswerte pro Bild-Anfrage-Paar unsortiert waren, fand in der Nachverarbeitung eine absteigende Sortierung statt. Die Predictions mit den höchsten Ähnlichkeiten standen damit an erster Stelle.

4.4.2 Anordnung

Eine weitere Option bestand darin, die absteigend sortierten Ergebnisse tageweise aufzuteilen, sodass immer nur die x relevantesten Bilder pro Tag in die Submission übernommen wurden. Dies hatte den Zweck, den Cluster-Recall zu erhöhen, da einige der Anfragen auf Situationen abzielten, die üblicherweise nur einmal pro Tag stattfinden, wie zum Beispiel das Nachhausefahren von Arbeit oder das Anschauen von Gegenständen in einem Spielzeugladen.

5 Evaluation

In diesem Kapitel wird eine Übersicht über alle entwickelten Verfahren gegeben und die Vorgehensweise für die Evaluation dieser Verfahren erläutert. Es folgt eine kurze taxonomische Beschreibung der in diesem Kapitel verwendeten Begriffe „Modellarchitektur“, „Modelltyp“ und „Modell“.
Eine Modellarchitektur beschreibt ein systematisches Verfahren, welches zu einer Menge von Anfragen eine Menge von relevanten Bildern zurückgibt. Für eine Architektur können Parameter definiert sein, deren Werte sich in einem bestimmten Bereich befinden dürfen. Modelltypen grenzen eine Modellarchitektur durch die konkrete Festlegung von Werten für eine Teilmenge an Parametern ein. Ein Beispiel für diese Eingrenzung sind die Festlegung von Gewichten für bestimmte Labelarten, sodass sich ein Modelltyp ergibt, der nur Kategorielabels einbezieht. Dadurch ist es möglich, generelle Aussagen über Modelle zu treffen. Ein Modell stellt eine konkrete Instanz dar, bei der jedem Parameter einer Modellarchitektur ein eindeutiger Wert zugewiesen wurde. Modelle verarbeiten Anfragen entsprechend ihrer Parameterwerte und geben relevante Bilder zurück. Jedes Modell lässt sich einem Modelltypen zuordnen.
Ein Experiment umfasst die Ausführung eines Modells und die anschließende Berechnung der Evaluations-Metriken („Scores“) anhand der zurückgegebenen Bilder („Predictions“). Jedes Experiment ist ein Teil einer Experimentreihe, welche für einen bestimmten Modelltyp durchgeführt wurde. Die Wahl der Modellparameter geschah dafür fast ausschließlich in Form einer Grid Search. Die Begriffe sind im nachstehenden Diagramm noch einmal in Relation gesetzt.

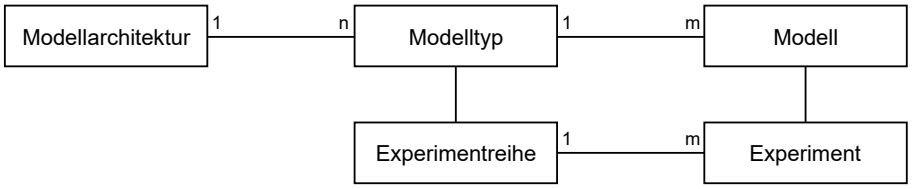


Abbildung 5.1: Das Diagramm stellt die Relationen zwischen den verschiedenen Begriffen der Evaluation dar. Ein Modell stellt eine konkrete Instanz eines Verfahrens dar, wobei jedem Parameter ein eindeutiger Wert zugeordnet wurde. Diese Modelle lassen sich genau einem Modelltyp zuordnen, welcher für eine Teilmenge der Parameter festgelegte Werte besitzt. Modelltypen sind alle Bestandteil einer Modellarchitektur, welche ein systematisches Verfahren beschreibt. Für jeden Modelltyp gab es eine Reihe von Experimenten.

5.1 Metriken

Für die Evaluation der Predictions existierten drei Metriken: der Cluster Recall@X (CR@X), die Precision@X (P@X) und der F1@X [25]. Diese wurden zur Evaluation der Verfahren in Bezug auf die Trainings- und Testanfragen genutzt. Es folgt eine Beschreibung der Metriken. Das @X steht dabei für die ersten X zurückgegebenen Ergebnisse pro Anfrage für eine Submission. Falls nicht näher angegeben, beziehen sich Angaben über die Metriken in den Experimenten immer auf $X = 10$.

5.1.1 CR@X

Der Cluster Recall ist ein Diversitätsmaß und gibt an, wie viele verschiedene Cluster der Ground Truth unter den besten X Ergebnissen repräsentiert sind. Dazu wird jedes Cluster durchgegangen und geschaut, ob mindestens ein Bild unter den Ergebnissen ist. Falls dies zutrifft wird eine lokale Zählvariable iteriert, welche die Anzahl an getroffenen Cluster angibt. Zum Schluss wird das Minimum aus der Anzahl an getroffenen Clustern und X gebildet und mit dem Minimum aus der Anzahl an Clustern und X dividiert:

$$\text{CR@X} = \frac{\min(\text{Anzahl getroffene Cluster}, \text{X})}{\min(\text{Anzahl an Clustern}, \text{X})}, \text{CR@X} \in [0, 1], \text{X} > 0$$

Der Grund für die Berechnung des Minimums im Nenner ist, dass es keine Rolle spielt, wie viele weitere Cluster getroffen wurden. Die Minimum-Funktion im Zähler stellt sicher, dass es bei Anfragen, die weniger als X Cluster haben, trotzdem möglich ist, einen CR@X von Eins zu erreichen. Die Reihenfolge der Cluster ist demzufolge irrelevant.
Der Cluster Recall spielt eine wichtige Rolle bei Momenten, die häufiger als einmal aufgetreten sind. Dazu zählen Momente wie das Essen im Restaurant oder das Benutzen des Smartphones.

5.1.2 P@X

Die Precision ist ein Maß darüber, wie viele relevante Ergebnisse unter den besten X Ergebnissen sind. Dazu wird für jedes Bild geprüft, ob es in einem Cluster der Ground Truth vorkommt. Die Precision berechnet sich aus der Anzahl an richtigen Bildern dividiert durch X. Hierbei kann es sein, dass die Anzahl an Bildern in der Ground Truth kleiner als X ist. In diesem Fall lässt sich keine Precision von Eins erzielen.

$$\text{P@X} = \frac{\text{Anzahl relevante Bilder}}{\text{X}}, \text{P@X} \in [0, 1], \text{X} > 0$$

Eine hohe Precision ist besonders relevant bei Momenten, die nur einmal auftreten. Solche Momente sind beispielsweise das Fotografieren einer Brücke oder das Eisessen am Strand.

5.1.3 F1@X

Der F1 stellt das harmonische Mittel aus dem Cluster Recall und der Precision dar. Es berechnet sich wie folgt:

$$\text{F1@X} = 2 \cdot \frac{\text{P@X} \cdot \text{CR@X}}{\text{P@X} + \text{CR@X}} \in [0, 1], \text{X} > 0$$

Diese Metrik stellt die relevanteste der drei Metriken dar, weil sie die zuvor genannten Metriken gleichermaßen einbezieht. Daher spielte sie für die Evaluation eine wesentliche Rolle.

5.1.4 Überprüfung der Korrektheit der Implementierung

Für die beschriebenen Metriken existierte keine Python-Bibliothek, weshalb diese eigenhändig implementiert wurden. Um sicherzustellen, dass die Implementierungen korrekt funktionierten, wurde neben Unit-Tests eine Art Modell erstellt, welches ausschließlich auf der Ground Truth basierte. Es selektierte pro Trainingsanfrage aus jedem Cluster das erste Bild und fügte es in der Ergebnisliste hinzu. Es ergaben sich für alle drei Metriken 100%. In der Konsequenz war die Korrektheit der Implementierung sichergestellt.

5.2 Submissions

Die CSV-Datei für die Testanfragen, welche durch das Verfahren generiert und anschließend durch Hochladen auf das Portal der Challenge evaluiert wurde, stellt eine Submission dar. Sie bestand aus den drei Spalten Anfrage-Id, Bild-Id und Konfidenz-Score. Letztere gab die Ähnlichkeit zwischen Anfrage und Bild im Intervall $[0, 1]$ an. Die Reihenfolge der Zeilen war dabei relevant. Insgesamt war es möglich, bis zu 15 Submissions einzureichen. Die Evaluation der Submissions erfolgte durch die drei Metriken CR@10, P@10 und F1@10.
Dabei war nicht klar, ob die Veranstalter der Challenge die Metriken wie oben beschrieben implementierten und es hätte möglich sein können, dass sich bei Anfragen mit weniger als zehn Ground Truth Clustern keine 100% Cluster Recall erreichen ließ oder, dass die Precision für Anfragen mit weniger als zehn Bildern trotzdem 100% sein konnte.

5.3 Zufallsbasierte Modellarchitektur

Die erste Modellarchitektur wies jeder Anfrage N zufällige Bilder aus dem Datenset zu. Es wurden drei Modelltypen für $N = 5, 10$ und 50 definiert. Die Experimentreihe bestand dabei aus 150 Durchläufen mit jeweils unterschiedlichen Bildern. Die Ergebnisse sind in Abbildung 5.2 dargestellt.

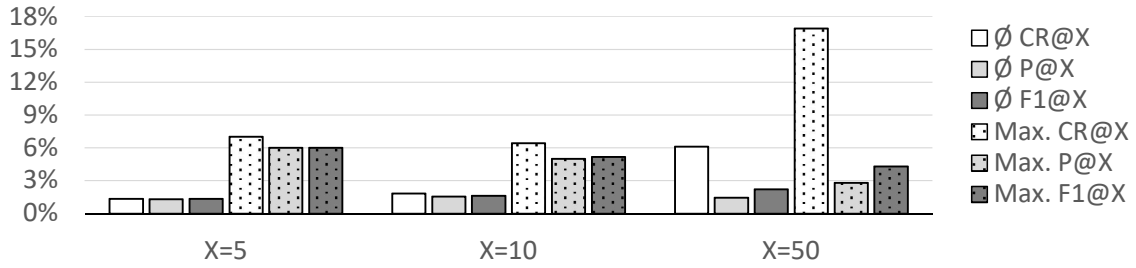


Abbildung 5.2: Die Ergebnisse der Experimente der zufallsbasierten Modellarchitektur zeigen, dass im Durchschnitt nur sehr niedrige Scores erreicht wurden. Der höchste Cluster Recall betrug 16,90% und im Schnitt wurde ein F1 von 1,62% erreicht.

Es zeigte sich, dass die durchschnittlichen Scores sehr niedrig waren und dass sich die Scores mit größer werdenden N erwartungsgemäß leicht verbesserten, da die Wahrscheinlichkeit stieg, dass sich relevante Bilder in den Ergebnissen befanden. Der höchste Cluster Recall betrug 16,90% und im Schnitt wurde ein F1 von 1,62% erreicht. Dieser F1-Score stellte die absolute Untergrenze für jedes folgende Verfahren dar. Der hohe Cluster Recall lässt sich darin erklären, dass wie oben beschrieben (siehe 5.1.1) das Minimum aus der Anzahl an getroffenen Clustern und X berechnet wurde und die maximale Anzahl der Cluster bei 31 lag. Durch die längere Liste an Ergebnissen existierte eine höhere Wahrscheinlichkeit ein Cluster zu treffen.

5.4 Regelbasierte Modellarchitektur

Um die gegebenen Metadaten erstmals mit einzubeziehen, wurde eine regelbasierte Modellarchitektur erstellt. Zum Zeitpunkt der Erstellung war die Tokenisierung der Anfragen noch nicht relevant, da es sich nur um einen ersten Prototypen handeln sollte, dessen Fokus auf dem Vergleich der Anfragen mit den Bildern lag. Deshalb wurden die Token manuell aus den Anfragen definiert (siehe Anhang 6).
Das Verfahren las alle gegebenen visuellen Konzepte ein (*Attribute*, *Kategorien* und *gegebene Konzepte*) und ersetzte Schräg- und Unterstriche durch Leerzeichen. Anschließend wurden diese tokenisiert und diejenigen Kategorien ignoriert, von denen mindestens ein Token nicht im Vokabular der Wortvektoren vorkam, da andernfalls kein Vergleich zu den Anfrage-Token möglich war. Danach wurden für jede Anfrage pro Labelart passende Labels herausgesucht. Dies geschah, indem zu jedem Anfrage-Token pro Labelart eine Liste an Labels mit absteigender Ähnlichkeit erstellt wurde. Aus dieser Liste extrahierte das Modell die ersten N Labels, welche somit die relevantesten Labels je Labelart darstellten.
Es folgte der Vergleich der relevanten Labels mit den Anfragen. Dazu wurde die Regel aufgestellt, dass alle Bilder in die Submission übernommen wurden, deren *Attribute*, *Konzepte* und *Kategorien* mindestens M relevante Labels enthielten.
Da es sich nur um einen Prototypen handelte, wurde nur ein Modelltyp mit $N = 2$

und dafür ein Modell mit $M = 1$ definiert. Das entsprechende Experiment erzielte für die Trainingsanfragen das folgende Resultat, dargestellt in Abbildung 5.3.

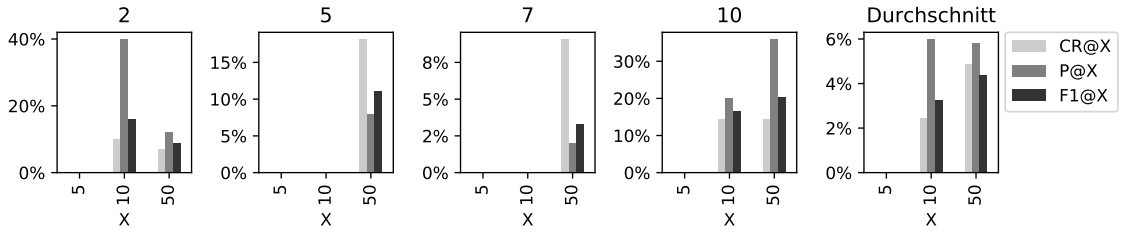


Abbildung 5.3: Das Ergebnis für das einzige Modell der regelbasierten Modellarchitektur zeigt, dass für die 80% der Trainingsanfragen unter den besten 10 Ergebnissen kein relevantes Bild gefunden wurde.

Für die Anfragen 1, 3, 4, 6, 8 und 9 wurden keine korrekten Bilder predictet. Der höchste F1@50 mit 20,45% bzw. F1@10 mit 16,67% ließ sich bei der zehnten Anfrage erreichen. Unter den ersten fünf Bildern wurde für keine Anfrage ein relevantes Bild zurückgegeben. Durchschnittlich ergab sich ein F1@50 von 4,39% und ein F1@10 von 3,27%. Dies stellte eine kleine Verbesserung um 1,65% zum Zufalls-Modell dar. Die Ursache für den niedrigen F1 lag darin, dass alle drei Labelarten für ein Zutreffen der Regel getroffen werden mussten. Damit schieden bereits alle Bilder aus, bei denen kein Konzept gelabelt war. Des weiteren spielten die Konfidenz-Scores für die Labels keinen Rolle, sodass falsche Klassifikationen mit niedrigen Scores ungeprüft als korrekte Annotation angenommen wurden.

5.5 Segmentbasierte Modellarchitektur

Nachdem der Segmentierungsalgorithmus (siehe 4.2.3) implementiert wurde, ließen sich durch eine segmentbasierte Modellarchitektur optimale Segmentierungs-Parameter finden.
Für die Segmentierung existierten die folgenden Parameter:

- *min_img*: definierte die minimale Anzahl an Bildern pro Segment.
- *threshold*: definierte, bis zu welcher Threshold zwei angrenzende Bilder sich dem gleichen Segment zuordnen ließen.
- *merge_dist*: definierte die Anzahl an Bildern, welche zwischen zwei Segmenten eines Clusters liegen durften, damit diese zusammengefasst wurden. Der Wert -1 war eine Ausnahme und gab an, dass kein Zusammenfügen stattfinden sollte.
- *repr_sel*: definierte, welche Bilder der Segmente als repräsentative Bilder zählten. Mögliche Werte waren entweder *first* oder *last* für das jeweils erste oder das letzte Bild eines Segmentes.

Das Verfahren führte zuerst die Segmentierung durch und extrahierte danach die repräsentativen Bilder aus den Segmenten. Die Labelarten spielten dabei keine Rolle, da es nur darum ging, die Bilder in möglichst wenige Segmente einzuteilen und dabei einen hohen Cluster Recall zu erreichen. Durch einen weiteren Parameter *use_step_two* ließ sich festlegen, ob eine zweite Segmentierung erfolgen sollte. Daraus ergaben sich zwei Modelltypen, je nachdem, welchen Wert der Parameter *use_step_two* hatte. Auf den Modelltyp mit der zweiten Segmentierung wird im Folgenden nicht weiter eingegangen, da es den Rahmen der Arbeit sprengen würde. Die Evaluation erfolgte, indem der Cluster Recall über alle Bilder errechnet wurde. Die Reihenfolge der Bilder war somit irrelevant, es ging in den Experimenten nicht darum, eine hohe Precision und damit einen hohen F1-Score zu erzielen.
Es wurden nacheinander zwei Experimentreihen durchgeführt, da eine Optimierung aller Parameter auf einmal zu einer zu großen Anzahl an Experimenten geführt hätte.

5.5.1 Erster Durchgang

Im ersten Durchgang sollte der Parameter *min_img* und *repr_selection* optimiert werden. Daher war das Setup für den ersten Durchgang das folgende:

- *threshold*: von 10.000 bis 50.000 in 1.000er Schritten
- *min_img*: 2, 3, 4
- *merge_dist*: -1
- *repr_selection*: *first*, *last*

Es ergaben sich 246 unterschiedliche Experimente, deren Ergebnisse in Abbildung 5.4 aufgelistet sind.

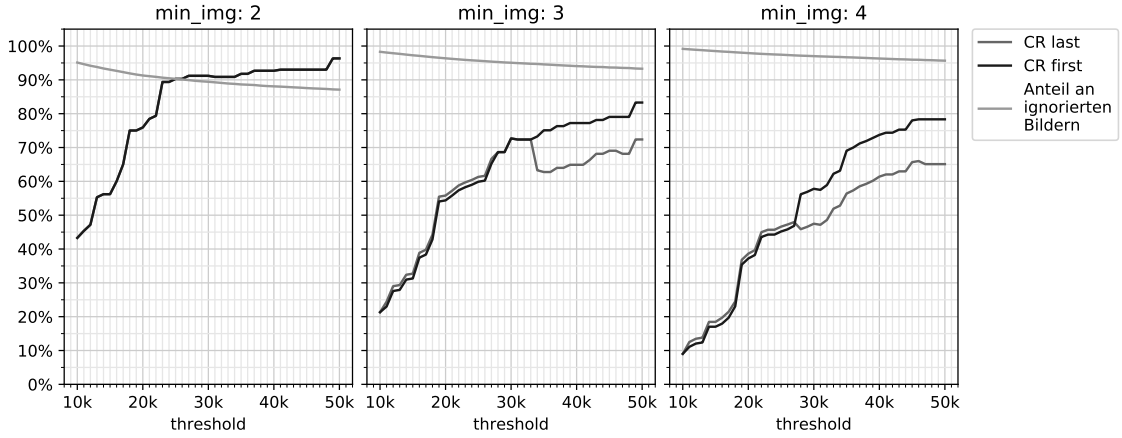


Abbildung 5.4: Es ist erkennbar, dass der Cluster Recall bei steigender *threshold* durchschnittlich zunahm und die Anzahl an ignorierten Bildern abnahm. Der höchste Recall von 96,34% wurde mit einer minimalen Anzahl von zwei Bildern erreicht.

Es ist erkennbar, dass der Cluster Recall bei steigender *threshold* durchschnittlich zunahm und dass er bei der Selektion des jeweils ersten Bildes eines Segmentes gegenüber des letzten Bildes höher war. Die Anzahl an Bildern wurde um zwischen 87,11% und 99,16% reduziert, wobei sie mit steigender *threshold* zunahm, also weniger Bilder ignoriert wurden. Der Recall war im Durchschnitt mit einer minimalen Anzahl von zwei Bildern am höchsten. Er erreichte maximal 96,34% bei einer *threshold* von 50.000, wobei sich bereits bei einer *threshold* von 23.000 ein Recall von 89,37% erreichen ließ.
In Tabelle 5.1 ist zu sehen, dass bei den beiden anderen *min_img*-Werten zu wenige Segmente entstanden, sodass zu viele relevante Bilder herausgefiltert wurden und der Cluster Recall in der Folge niedriger war. Die maximale Segmentanzahl halbierte sich annähernd mit jeder Erhöhung des Parameters *min_img*.

Tabelle 5.1: Die Tabelle zeigt die minimale und maximale entstandene Anzahl an Segmenten je Parameter-Wert für *min_img*. Es ist zu sehen, dass sich die maximale Segmentanzahl mit jeder Werterhöhung ungefähr halbierte. Je niedriger die Anzahl an Segmenten war, desto geringer war auch der resultierende Cluster Recall.

min_img	**Segmentanzahl Min.**	**Max.**
2	3.112	8.210
3	1.087	4.273
4	535	2.759

Dementsprechend ergaben sich mit *min_img* = 2 und *repr_selection* = *first* die optimalen Parameterwerte.

5.5.2 Zweiter Durchgang

In der zweiten Versuchsreihe bestand das Ziel darin, die Parameter *threshold* und *merge_dist* zu optimieren. Dazu wurden 300 Durchläufe mit folgenden Parametereinstellungen durchgeführt:

- *threshold*: von 23.000 bis 50.000 in 3.000er Schritten
- *min_img*: 2
- *merge_dist*: von 0 bis 29
- *repr_selection*: *first*

Die Ergebnisse sind in Abbildung 5.5 dargestellt. Sie zeigen, dass die Auswirkungen des Parameters *merge_dist* auf den Cluster Recall mit steigender *threshold* zunahmen. Annähernd konstant verlief der Recall bis zur *threshold* 32.000. Dort erreichte er einen Wert von ca. 90%. Die Anzahl an ignorierten Bildern stieg mit der Erhöhung

des Parameters *merge_dist* für jede *threshold*. Ab einer *threshold* von 35.000 gab es immer einen Wert für den Parameter *merge_dist*, ab dem der Cluster Recall um ca. 10% schlagartig abnahm. Bei dem Zusammenfügen von Segmenten, bei denen kein Bild dazwischen lag, ließ sich zwar der höchste Recall erzielen, die Anzahl an Bildern wurde allerdings am wenigsten reduziert.

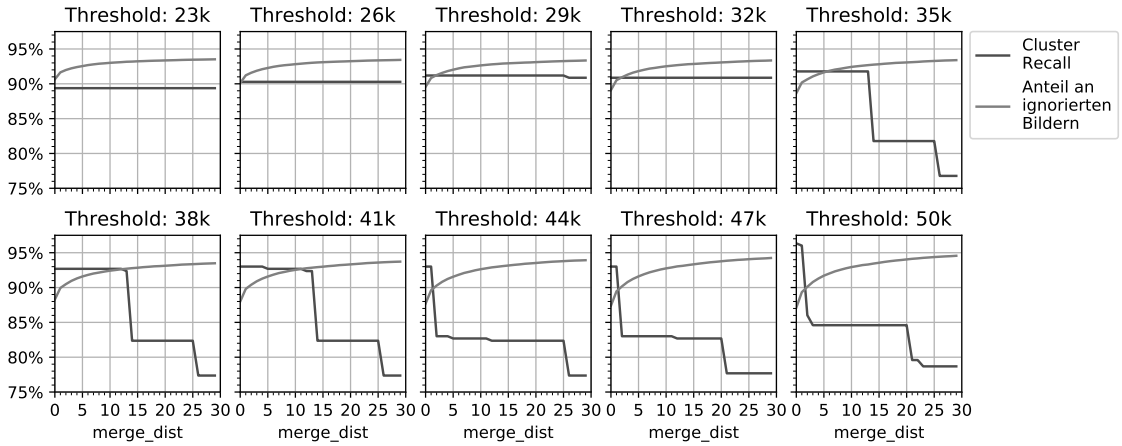


Abbildung 5.5: Der Cluster Recall verlief annähernd konstant bis zu einer *threshold* von 32.000, wobei die Anzahl an ignorierten Bilder mit der Erhöhung der *merge_dist* zunahm. Es fällt auf, dass es ab einer *threshold* von 35.000 eine Parametereinstellung gab, bei der der Recall schlagartig abnahm.

Anhand des letzten Durchlaufes zeigte sich, dass eine Threshold von 23.000 bis 32.000 optimal war, da es in diesem Bereich keine schlagartigen Einbrüche des Cluster Recalls gab. Ab einer *merge_dist* von 15 ließen sich die Bildanzahl nur noch marginal verringern, weshalb dieser Wert als optimaler Wert gewählt wurde. Dies entsprach einem Abstand von ca. sieben Minuten zwischen zwei Segmenten, welcher auch realistisch schien. Da die *threshold* so klein wie möglich sein sollte, damit unähnliche Bilder nicht in einem Segment landeten, wurde der Wert 23.000 als Optimalwert gewählt.

5.5.3 Optimale Parameter-Einstellungen

Es ergaben sich die folgenden optimalen Parametereinstellungen für die Segmentierung:

- *threshold*: 23.000
- *min_img*: 2
- *merge_dist*: 15
- *repr_selection*: *first*

Diese Einstellungen wurden lediglich anhand der Trainingsanfragen optimiert, weshalb eine ähnliche Performanz auf den Testanfragen nicht garantiert war. Die Herangehensweise stellte jedoch die sinnvollste Strategie für die Wahl der Parameter dar, da wie bereits erwähnt nur 15 Submissions für die Testanfragen möglich waren. Schlussendlich konnten durch diese Einstellungen 4.311 Segmente aus den 63.696 Bildern gebildet werden, welche 21.523 Bilder enthielten. Dies entsprach einer Reduktion der Bildanzahl um 66,21%. Durch das Wählen der jeweils ersten Bilder aus den Segmenten konnte die Anzahl der Bilder auf 4.311 verringert werden, was einer Reduktion um erneut 79,98% entsprach. Die Verringerung der Bildanzahl betrug schlussendlich 93,23%, wobei eine Verringerung des Cluster Recalls um 10,63% auf 89,37% erfolgte.

5.6 Vektorbasierte Modellarchitektur

Die vektorbasierte Modellarchitektur stellte die erste Architektur dar, welche die Bildsegmentierung mit einbeziehen konnte. In dieser Sektion wird das Verfahren erklärt und alle durchgeführten Experimente werden dargelegt.

5.6.1 Funktionsweise

Zuerst wurden die Segmente mit den optimalen Parametereinstellungen aus der vorherigen Sektion optional extrahiert. Nachdem die Segmente geformt wurden, bildete der Algorithmus für jede Anfrage und für jedes Bild bzw. Segment einen Vektor, welcher aus den Vektoren der *Attribute*, *Kategorien* und *Konzepte* zusammengesetzt war. Ein Parameter legte fest, welche der Konzepte Verwendung finden sollten. Die Auswahl bestand zwischen den *gegebenen Konzepten*, den *Yolo Konzepten* und den *Detectron Konzepten*. Dadurch ließ sich später in der Evaluation herausfinden, welche Konzeptart am besten geeignet war.
Im Anschluss bestand die Möglichkeit, die resultierenden Vektoren zu intensivieren. Dazu gab es die Parameter n und m. Der Parameter n gab die Faktorisierung und der Parameter m die Potenzierung der Vektoreinträge an.
Danach bestand die Option, die Inversen Dokument Frequenzen der Labels einzubeziehen. In diesem Falle wurden die Vektoreinträge mit den IDF-Werten multipliziert.
Anschließend folgte der Vergleich von Bild- und Anfrage-Vektoren. Dies geschah entweder durch das Bilden der Differenz und der anschließenden Berechnung der euklidischen Norm oder durch das Bilden der Kosinus-Distanz und dem anschließenden Abziehen von Eins. In beiden Fällen gab der resultierende Wert die Ähnlichkeit an, wobei sich Anfrage und Bild/Segment umso ähnlicher waren, je kleiner der Wert war.
Im nächsten Schritt war es optional möglich, mit Hilfe von XGBoost ein Modell mit den Ähnlichkeiten zu trainieren und relevante Bilder zu predicten.
In der Nachverarbeitung wurden die Ähnlichkeitsmaße aufsteigend sortiert und optional angeordnet, wie in 4.4.2 beschrieben.

Anhand dieser Parameter war es möglich, acht Modelltypen zu definieren, bei denen jeweils alle bis auf einen Parameter den gleichen Wert hatten. Dazu wurde eine Baseline-Konfiguration definiert, welche die Standard-Werte für die einzelnen Parameter festlegte. Es wurden zudem zwei übergeordnete Modelltypen definiert, wovon der eine bildbasiert und der andere segmentbasiert war.

5.6.2 Baseline-Experimente

Zuerst wurden zwei Baseline-Experimente für die beiden übergeordneten Modelltypen ausgeführt, welche die Grundlage für alle weiteren Experimente darstellten. Die beiden entsprechenden Modelle benutzten die folgenden Baseline-Konfigurationen:

- Konzepte: *gegebenen Konzepte*
- Einbezug der IDF: *false*
- Anordnung: *false*
- Vergleichsverfahren: *euclidean*
- Representatives Bild: *first*

Für das bildbasierten Baseline-Modell wurde ein F1 von 10,12% und ein F1@50 von 16,47% erreicht.
Die Ergebnisse für das segmentbasierte Model sind in Abbildung 5.6 einsehbar. Sie zeigen, dass keine relevanten Bilder für die Anfragen 4, 6, 8 und 9 gefunden wurden und die zweite Anfrage den höchsten F1 erreichte. Es ist auffällig, dass der Cluster Recall für die erste Anfrage 100% beträgt. Die Ursache dafür liegt darin, dass es nur ein Cluster für die Anfrage gab. Da nur ein Bild gefunden wurde, beträgt die Precision 10% und es resultierte deshalb für die erste Anfrage ein F1 von 18,18%. Der durchschnittliche F1 lag bei 18,67%, wobei der F1@50 16,43% betrug.

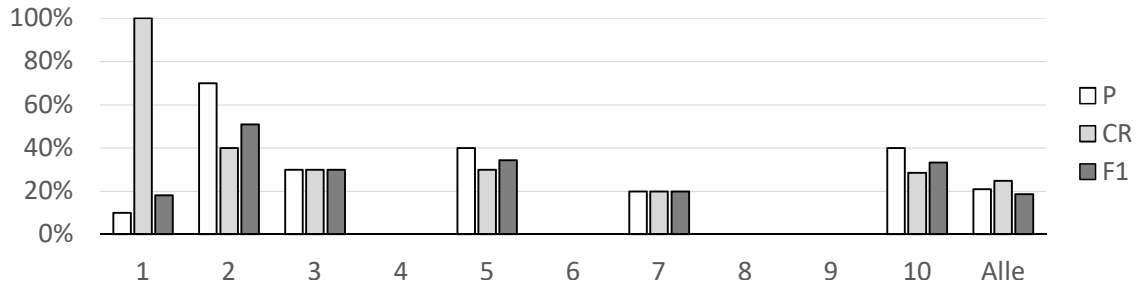


Abbildung 5.6: Die Ergebnisse des segmentbasierten Baseline-Modells zeigen, dass keine relevanten Bilder für die Anfragen 4, 6, 8 und 9 gefunden wurden. Der durchschnittliche F1 lag bei 18,67%.

Es ist ersichtlich, dass der F1 kleiner als die Precision und der Cluster Recall war. Dies konnte bei der Berechnung des Durchschnittes auftreten und stellt keinen Fehler dar.

5.6.3 Experimente zur Segmentierung

Für den segmentbasierten Modelltyp existierte ein Modelluntertyp, bei dem der Parameter *img_sel* variabel war. Dieser gab an, welche Bilder das Verfahren für die Berechnung des Segment-Vektors auswählen sollte. Es wurden die vier Werte *first*, *last*, *first_last* und *all* ausprobiert.
Die Experimente zeigten, dass es keine Rolle spielte, welche Bilder für die Berechnung der Segment-Vektoren genommen wurden, da alle F1-Scores den gleichen Wert hatten.
Der F1@50 war niedriger als der F1@10, was daran lag, dass es verhältnismäßig weniger relevante Bilder unter die Top 50 Ergebnisse schafften.

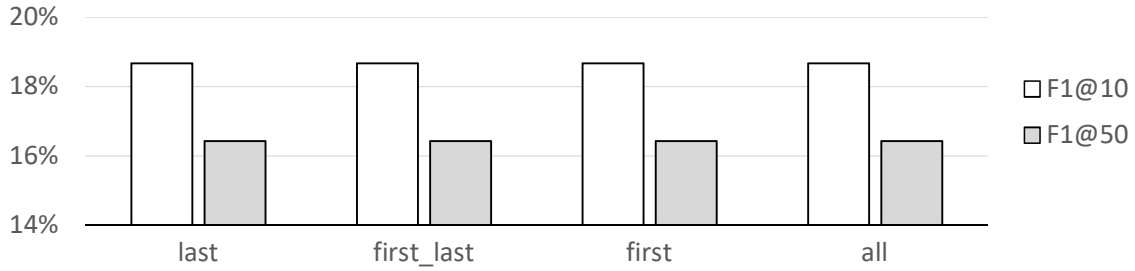


Abbildung 5.7: Die Experimente haben gezeigt, dass es keine Rolle spielte, welche Bilder für die Berechnung der Segment-Vektoren genommen wurden, da alle F1-Maße den gleichen Wert hatten. Der F1@50 war bei beiden Variationen niedriger als der F1@10.

Der Baseline-Score ließ sich demgemäß nicht verbessern, indem die Wahl der repräsentativen Bilder eines Segmentes geändert wurde.

5.6.4 Experimente zur Auswahl der Konzepte

Für beide Modelltypen wurde jeweils ein Modelluntertyp definiert, welcher entweder die *gegebenen*, die *Yolo* oder die *Detectron Konzepte* mit den *Attributen* und *Kategorien* zu einem Vektor kombinierte. Das Ziel war es, herauszufinden, welche Konzepte bessere Ergebnisse lieferten.
Die Ergebnisse sind in der nachstehenden Abbildung 5.8 zu sehen. Es hat sich herausgestellt, dass die *Yolo Konzepte* für den bildbasierten Modelltyp am besten funktionierten (F1 mit 11,99%), wobei für den segmentbasierten Typ die *gegebene Konzepte* zu dem höchsten F1 führten (gleicher F1 wie im Baseline-Versuch). Es ist zudem erkennbar, dass in dem bildbasierten Modelltyp der F1@50 höher als der F1@10 war. Die Ursache dafür lag daran, dass mehr potentiell relevante Bilder existierten, die die Precision erhöhten, welche mitunter den gleichen Moment darstellten. Da der Algorithmus jedoch eine hohe Variation unter den ersten 10 Ergebnissen liefern sollte, spielte der F1@50 eine untergeordnete Rolle.

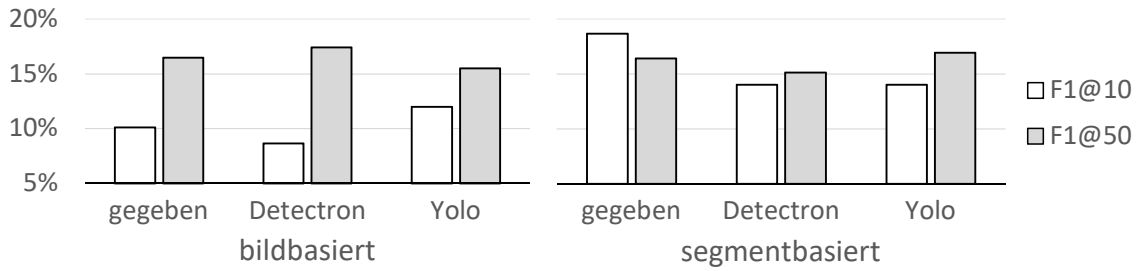


Abbildung 5.8: Bei den Konzeptwahl-Experimenten hat sich herausgestellt, dass die *Yolo Konzepte* für den bildbasierten Modelltyp am besten funktionierten (F1 mit 11,99%), wobei für den segmentbasierten Typ die *gegebene Konzepte* zu dem höchsten F1 führten.

Die segmentbasierte Baseline ließ sich nicht verbessern, die bildbasierte hingegen schon auf 11,99%.

5.6.5 Experimente zur Inversen Dokumentfrequenz

Um festzustellen, welche Auswirkung die Einberechnung der Inversen Dokumentfrequenz hatte, wurden für beide Modelltypen jeweils zwei Modelluntertypen mit und ohne Einberechnung der IDF-Werte erstellt und Experimente dazu durchgeführt.
In Abbildung 5.9 ist zu sehen, dass die IDF-Maße den F1 des bildbasierten Modelltyps um 0,9% auf 11,02% erhöhte. Bei dem segmentbasierten Typ führte die Verwendung der IDF-Werte zu einer Verringerung des F1 um 8,40%.

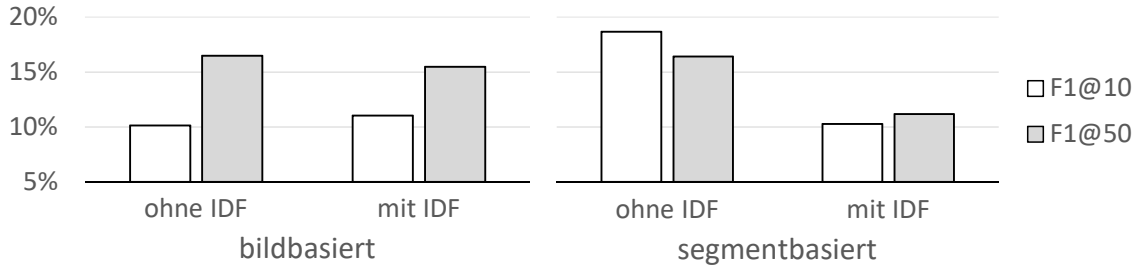


Abbildung 5.9: Die IDF-Versuche ergaben, dass die IDF-Maße den F1 des bildbasierten Modelltyps um 0,9% erhöhte. Bei dem segmentbasierten Ansatz führte der IDF zu einer Verringerung des F1 um 8,40%.

Es lässt sich schlussfolgern, dass das Einbeziehen der IDF-Maße nicht sinnvoll war. Die Ursache dafür lag darin, dass Ähnlichkeiten bei seltenen Labels bestraft wurden, da sie eine hohe IDF hatten und durch die Multiplikation zu einem größeren Wert führten. Sinnvoller wäre es jedoch gewesen, wenn sich ein kleinerer Wert ergeben hätte.

5.6.6 Experimente zur Intensivierung

Bei den Experimenten zur Intensivierung wurden die Faktoren n und m pro Modelltyp verschieden variiert:

- n: von 1 bis 5
- m: von 1 bis 5

Es resultierten 25 Versuche. Durch die beiden Variablen wurden für z.B. den Wert 2, alle Ähnlichkeiten unter 0,5 relativ zueinander verkleinert und über 0,5 vergrößert. Die Ergebnisse zeigten, dass der Faktor n zu keiner Änderung im Score führte, sondern dass ausschließlich m den Score beeinflusste. Daher sind in Abbildung 5.10 nur die F1-Scores in Abhängigkeit von m dargestellt. Es ist erkennbar, dass das bildbasierte Modell bei einer Potenzierung um den Wert 2 eine Verbesserung um weniger als ein Prozent auf 12,82% zum Baseline-Modell ergab. Für den segmentbasierten Modelltyp ließ sich keine Verbesserung erzielen.

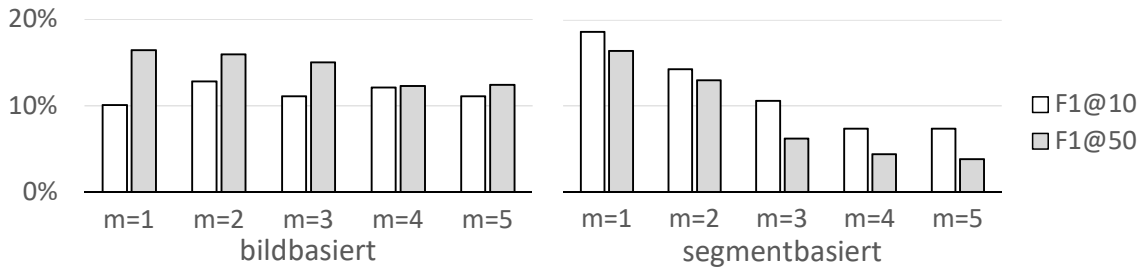


Abbildung 5.10: Die Ergebnisse der Intensivierungs-Experimente zeigen, dass das bildbasierte Modell mit der Potenzierung um den Wert Zwei eine Verbesserung um weniger als ein Prozent zum Baseline-Modell darstellte. Für den segmentbasierten Modelltyp ließ sich keine Verbesserung erzielen.

Als Konklusion lässt sich schlussfolgern, dass die Intensivierung keine große Verbesserung erzielen konnte.

5.6.7 Experimente zur Anordnung

In dieser Versuchsreihe wurde untersucht, welche Auswirkung die Wahl der besten x Bilder pro Tag auf den Score hatte. Dazu wurden entweder 1, 2, 3 oder 4 Bilder pro Tag selektiert.
In Abbildung 5.11 ist zu sehen, dass das Wählen des jeweils besten Bildes pro Tag für beide Modelltypen zum höchsten F1 führte. Je mehr Bilder genommen wurden, desto schlechter wurde der F1.

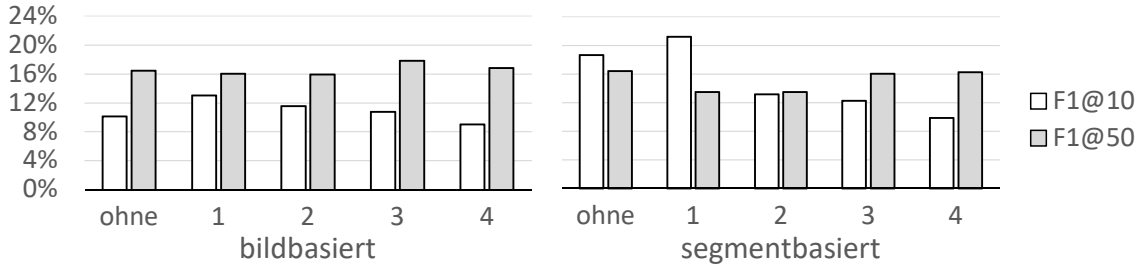


Abbildung 5.11: In den Ergebnissen der Anordnungs-Experimenten ist zu sehen, dass die Selektion des jeweils besten Bildes pro Tag für beide Modelltypen zum höchsten F1 führte.

Die Selektion des jeweils besten Bildes pro Tag konnte den Score des bildbasierten Baseline-Modells um 2,93% auf 13,05% und den des segmentbasierten Baseline-Modells um 2,54% auf 21,21% verbessern. Höhere Werte für x waren nicht förderlich.

5.6.8 Experimente zum Vektorvergleich

Um festzustellen, wie die beiden Vergleichsmethoden *cosine* (Kosinus-Distanz) und *euclidean* (euklidische Distanz) funktionierten, wurden beide Methoden für beide Modelltypen untersucht.
In Abbildung 5.12 ist erkennbar, dass der Vergleich mittels der euklidischen Distanz deutlich höhere F1-Werte erzielte.

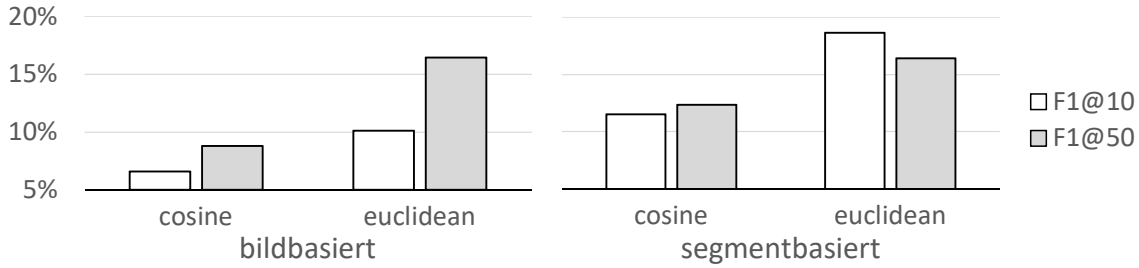


Abbildung 5.12: Die Resultate der Experimente zum Vektorvergleich zeigen, dass der Vergleich mit der euklidischen Distanz deutlich bessere F1-Werte erzielte.

Es ließ sich folglich keine Verbesserung zur Baseline erzielen.

5.6.9 Experimente zur Labelartkombination

In der letzten Versuchsreihe sollte geprüfte werden, zu welchen Ergebnissen verschiedene Kombinationen an Labelarten führen. Dazu wurden die *gegebene Konzepte*, die *Attribute* und die *Kategorien* in allen Möglichkeiten miteinander kombiniert.

Abbildung 5.13 stellt die Ergebnisse dar. Für den segmentbasierten Modelltyp ließ sich der F1 nicht erhöhen. Für den bildbasierten Typ konnte die ausschließliche Benutzung der Kategorien den Score um 13% auf 23,57% verbessern. Die Konzepte schnitten bei beiden Ansätzen am schlechtesten ab. Dies konnte daran liegen, dass nicht jedes Bild ein zugeordnetes Konzept besaß.

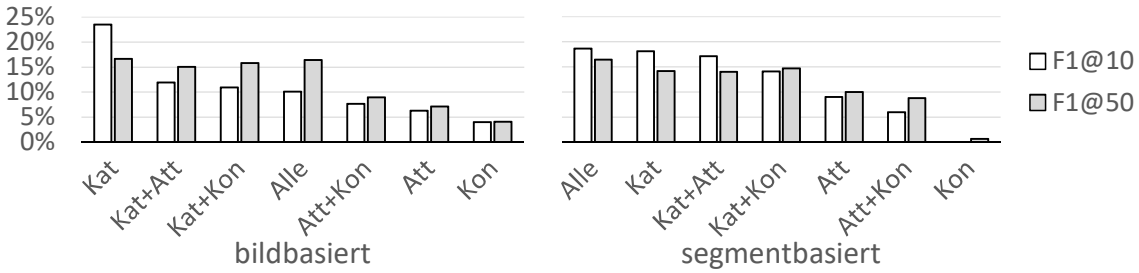


Abbildung 5.13: Die Kombinations-Versuche zeigten, dass sich der F1 für den segmentbasierten Modelltyp nicht erhöhen ließ. Für den bildbasierten Typ konnte die ausschließliche Benutzung der Kategorien den Score auf 23,57% verbessern.

Der Baseline-Score für den bildbasierten Modelltyp ließ sich über das Doppelte verbessern.

Weiterführende Experimente

Nachdem sich gezeigt hatte, dass die Kategorien den Score so stark verbesserten, wurden alle relevanten bildbasierten Experimente erneut durchlaufen und es zeigte sich, dass der Score durch das Potenzieren mit 3 erneut von 23,57% um 0,45% auf 24,02% verbessert wurde. Dieser Score stellte das absolute Maximum aus allen Experimenten dar.

5.6.10 Experimente mit maschinellen Lernen

Für den bildbasierten Modelltyp wurde ein auf maschinellen Lernen basierender Modelluntertyp definiert, welcher folgende zusätzliche Parameter besaß:

- *train_size_gt*: gibt den Anteil an Positiven und Negativen für das Trainingsset an (Baseline: 20%).
- *max_depth*: gibt die maximale Tiefe der erzeugten Bäume an (Baseline: 2)
- *cut_negatives*: gibt den Anteil an zu ignorierenden Negativen an (Baseline: 0%)

Der Parameter *max_depth* war ein spezieller XGBoost Parameter. Für diesen Modelltyp wurden alle vorherigen Experimente durchgeführt.
Da die detaillierte Auflistung der Ergebnisse den Rahmen sprengen würde, folgen die wichtigsten Erkenntnisse. Der Baseline-Versuch erreichte einen F1 von 41,91% und

einen F1@50 von 45,30%. Diese Score war sehr hoch, aber auf das Lernen im Training zurückzuführen. Es zeigte sich, dass die Intensivierungen und das Einbeziehen der IDF keine Auswirkungen auf den Score hatten. Die *Detectron Konzepte* führten zu einer Verbesserung auf 51,32% und durch die Selektion der besten vier Bilder pro Tag ließ sich der Score ebenfalls verbessern. Außerdem führte eine höhere Baumtiefe immer zu einem besseren Score. Dieser Umstand war jedoch nicht verwunderlich, da XGBoost die Ähnlichkeitswerte durch tiefere Bäume besser auswendig lernen konnte. Weiterhin führte das Entfernen von 40% der irrelevanten Bilder zu einer Verbesserung auf 49,63%. Die optimale Trainingsset-Größe stellte sich mit 70% heraus, bei der sich eine Verbesserung des Scores um über 10% auf 52,07% erzielen ließ. Dies stellte den maximalen Score dar, welcher erreicht wurde. Die verbesserten Scores wurden demnach hauptsächlich durch Overfitting generiert. Daher ist dieser Ansatz nicht für die produktive Anwendung von neuen, unbekannten Anfragen geeignet.

5.6.11 Schwächen der Modellarchitektur

Ein Problem, welches sich durch die Aneinanderreihung der Vektoren ergab, war, dass die Labelarten mit vielen Labels eine höhere Gewichtung erhielten. Weiterhin war es nicht möglich, die Konfidenz-Scores der annotierten Labels zu filtern, um falsche Predictions zu ignorieren. Darin könnte auch die Ursache für den Unterschied zwischen den verschiedenen Scores der Konzeptarten liegen. Der IDF wurde, wie bereits erwähnt, inkorrekt einbezogen, da nicht zwischen Ähnlichkeiten und Unähnlichkeiten unterschieden wurde.

5.7 Poolbasierte Modellarchitektur

Aus den Erkenntnissen der vektorbasierten Modellarchitektur wurde die poolbasierte Modellarchitektur erzeugt. Diese kombinierte verschiedene Labelarten aus einem Pool miteinander und ließ dadurch eine gleichmäßige Gewichtung der Labelarten zu. Außerdem bezog sie die IDF-Maße korrekt ein. Der Vergleich zwischen Anfragen und Bildern wurde komplexer, sodass es z.B. möglich war, die einzelnen Token der Anfragen separat auszuwerten.

5.7.1 Prototyp

Bis es zu der finalen Architektur kam, gab es einen Prototypen, für welchen ebenfalls Experimente gemacht wurden. Die Auswertung dieses Prototyps folgt an dieser Stelle nicht, da diese sich mit einigen Ergebnisses der finalen Architektur deckt und zu weit führen würde. Die Funktionsweise des Prototypen bestand darin, für jede Labelart ein eigenes Modell zu erstellen und anschließend mit Hilfe eines übergeordneten Modells diese Modelle zu kombinieren. Der Grund für die Aufteilung in mehrere Modelle war, dass es möglich sein sollte, jede Labelart für sich zu optimieren. In der

poolbasierten Modellarchitektur war es durch das Setzen von Parametern möglich, jede Labelart für sich zu optimieren.

5.7.2 Parameter der Architektur

Mit Hilfe der poolbasierten Modellarchitektur wurden neben den Trainingsanfragen auch alle Testanfragen verarbeitet. Die Funktionsweise wurde bereits in Kapitel 4 beschrieben. Folgende Variablen bezeichneten die einzelnen Modellparameter:

- *use_seg*: Segmentierung nutzen → *true/false*
- *use_tc*: Tokenclustering nutzen → *true/false*
- *use_reord*: nur das beste Ergebnis pro Tag nehmen → *true/false*
- *use_weights*: Token pro Labelart gewichten → *true/false*
- *query_src*: Anfrage-Quelle → *description/titel/narrative*
- *comp_method*: Vergleichsmethode → *mean/max*
- *tc_comp_method*: Vergleichsmethod für das Tokenclustering → *mean/max*
- *img_sel*: Wahl der Vektorerzeugungsbilder eines Segmentes → *first/last/all*
- *repr_sel*: Wahl der repräsentativen Bilder eines Segmentes → *first/last/all*

Für jede Begriffsart mussten diese Parameter festgelegt sein:

- *weight*: Gewichtung
- *threshold*: Threshold, ab der eine Annotation als zutreffend gilt
- *use_ceiling*: Aufrunden der Konfidenz-Scores → *true/false*
- *p*: Potenzierung der Konfidenz-Scores
- *use_idf*: IDF Maße einbeziehen → *true/false*

Es existierten noch weitere Parameter, welche jedoch in den Experimenten nicht optimiert wurden, wie zum Beispiel die Parameter der Segmentierung, und deshalb auch nicht mit aufgelistet sind.

5.7.3 Aufbau und Umsetzung der Experimente

Für die Experimente der poolbasierten Modellarchitektur wurde strategisch vorgegangen. Das Ziel war es herauszufinden, wie bestimmte Kombinationen von Labelarten miteinander harmonierten. Deshalb wurden verschiedene Modelltypen gebildet, welche differierende Gewichtungen der Labelarten enthielten. In Abbildung 5.14 sind die verschiedenen Modelltypen aufgelistet, zu denen Experimente gemacht wurden. Die Typbezeichnungen suggerieren dabei, welche Labelarten enthalten waren.
Zum Beispiel enthielt der VK MBT Modelltyp alle Labelarten aus den visuellen Konzepten („VK“) und der minutenbasierten Ereignisliste/Tabelle („MBT“), wobei die Labelarten der zusätzlichen Konzepte eine Gewichtung von Null erhielten.

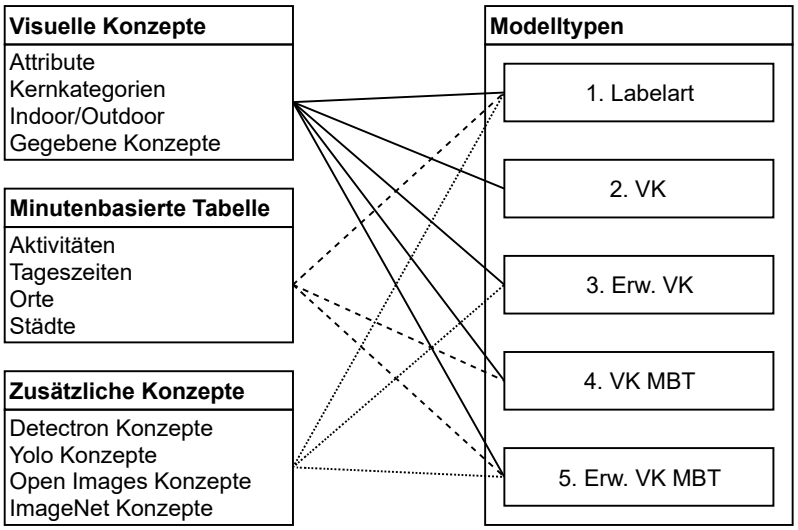


Abbildung 5.14: Die Übersicht stellt alle Modelltypen dar, mit welchen Experimente durchgeführt wurden. Verschiedene Gruppen von Labelarten wurden miteinander kombiniert, um später Schlüsse über die Auswirkungen auf den F1 schließen zu können. Die Grundlage bildeten die Optimierungen der Labelarten, da diese bei den späteren Modellen verwendet wurden.

Zuerst wurden die Labelarten alle jeweils für sich optimiert. Danach wurden Modelle erstellt, welche nur die Labelarten der visuellen Konzepte verwendeten (VK Modelltyp). Dieser Modelltyp wurde im dritten Schritt um die zusätzlich generierten Labelarten erweitert (Erweiterter VK Modelltyp). Der vierte Typ setzte sich aus allen gegebenen Labelarten zusammen (VK MBT Modelltyp) und wurde im letzten Schritt ebenfalls um die zusätzlichen Labelarten erweitert (Erweiterter VK MBT Modelltyp). Durch diesen Experimentaufbau ließen sich später Schlüsse über die Performanz der verschiedenen Kombinationen der Labelarten ziehen.

Umsetzung der Experimente

Ein Experiment bestand darin, ein Modell auf die Trainingsanfragen anzuwenden und die Ergebnisse der drei Metriken für X = 0, 5, 10, 20, 30, 40 und 50 auf jede Anfrage einzeln und für alle Anfragen zusammen zu ermitteln. Bei X = 0 wurden die Metriken auf alle zurückgegebenen Bilder angewandt.
Für jede Experimentreihe wurden zuvor für jeden Parameter die zu testenden Werte definiert. Im Rahmen einer Grid Search wurde für jede mögliche Kombination anschließend ein Modell erstellt und ausgeführt. Da die Grid Search bei einem Modelltyp mit vielen Labelarten sehr lange dauerte, wurde zur Optimierung der Durchführung eine spezielle Methodik angewandt, welche nachstehend beschrieben steht.

Für jeden Parameter wurde der Zeitpunkt in der Abarbeitung des Modells identifiziert, welcher angab in welchen konkreten Schritt der Parameter zu einer Veränderung im Modell führte. Alle Schritte davor mussten nicht erneut abgearbeitet werden. Bei den Experimenten wurden alle durchzuführenden Parameterkombinationen so sortiert, dass die Parameter mit späten Auswirkungen vor den Parametern mit frühen Auswirkungen abgearbeitet wurden. Beispiele für Parameter, welche sehr früh zu Änderungen im Modell führten, waren die Definition der Anfrage-Quelle oder die Threshold für die Segmentierung, wohingegen der Parameter *use_reord* erst am Ende relevant war. Durch diese Methodik war es möglich, Grid Searches in kurzer Zeit durchzuführen.

5.7.4 Auswertung der Experimente

Der Fokus lag bei allen Experimenten darauf, die besten Modelle eines Modelltyps in Bezug auf den bild- und segmentbasierten Ansatz zu identifizieren. Dabei wurde jeweils zwischen der ungeordneten und der angeordneten Variante (nur das relevanteste Bild pro Tag) unterschieden. Es ergaben sich vier Modelluntertypen mit folgende Grundkonfigurationen (Gk):

- Gk-00: *use_seg*: *false*, *use_reord*: *false*
- Gk-01: *use_seg*: *false*, *use_reord*: *true*
- Gk-10: *use_seg*: *true*, *use_reord*: *false*
- Gk-11: *use_seg*: *true*, *use_reord*: *true*

Für jede Gk wurde in der Auswertung zwischen zwei speziellen Modellkonfigurationen unterschieden. Die erste Konfiguration („Best“) war diejenige, welche den höchsten F1 erzielte. Weiterhin sollte es möglich sein, Rückschlüsse über die Kombinationen der verschiedenen Labelarten (Modelltypen) ziehen zu können. Deshalb wurde eine zweite Konfiguration („Baseline“) definiert, welche folgende Parameterwerte besaß:

- *use_tc*: *false*
- *use_weights*: *false*
- *query_src*: *description*
- *comp_method*: *mean*

Die anderen Parameter hatten diejenigen Werte, welche zum höchsten F1 beitrugen. Anhand dieser Konfigurationen wurde nach jeder Experimentreihe mindestens ein Test-Modell erstellt, gegebenenfalls leicht angepasst, auf die Testanfragen angewandt und die Submission eingereicht. Daher folgen im weiteren Verlauf nach jeder Experimentbeschreibung die jeweiligen resultierenden Submissions.

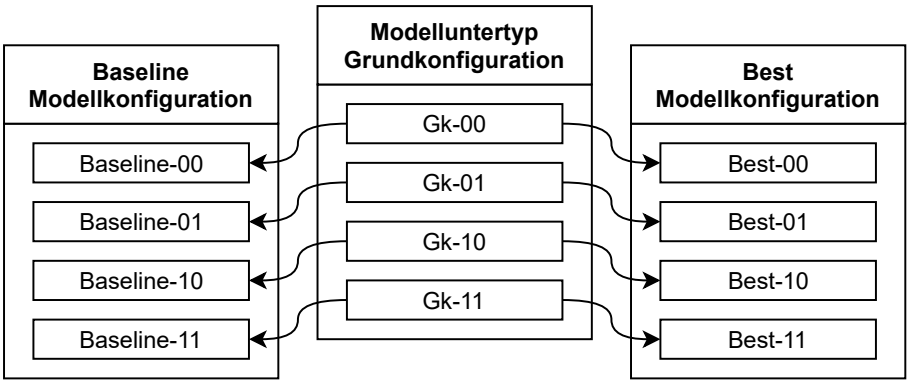


Abbildung 5.15: In dieser Abbildung ist eine Übersicht über die Konfigurationen der Auswertung dargestellt. Für jeden Modelltyp existierten vier Modelluntertypen, aus denen jeweils die Modelle mit der besten Modellkonfiguration und der Baselinekonfiguration herausgesucht wurden. Die Evaluation erfolgte pro Modelltyp anhand der Ergebnisse dieser acht Modelle.

Jedes Test-Modell wurde zusätzlich auf die Trainingsanfragen angewandt, um einen direkten Vergleich zwischen beiden Anfragetypen zu erhalten. Die Evaluation der Test-Modelle erfolgt außerdem anhand der vorherigen eingereichten Modelle.

5.7.5 Experimente des Labelart Modelltyps

Die Grundlage für die späteren Experimente bildeten optimierte Labelarten. Daher wurde jede Labelart mit Hilfe einer Grid Search für jede Grundkonfiguration in folgenden Parameterwerten optimiert:

- *use_ceiling: true/false*
- *use_idf: true/false*
- *img_sel: first/all*
- *p: 1/2/3*
- *threshold: 0/0,9/0,95/0,99*

Eine Ausnahme stellte die Labelart *Kategorien* dar, da sie sehr niedrige Konfidenz-Scores hatten. Für sie wurden die Thresholds *0/0,01/0,05* und *0,1* verwendet.
Diagramm 5.16 stellt die höchsten F1-Scores pro Labelart für den bild- und segmentbasierten Ansatz dar. Auf eine komplette Analyse für alle acht Evaluations-Modelle je Labelart wird im Folgenden auf Grund der Menge an Informationen verzichtet. Für die Labelarten *Städte*, *Aktivitäten* und *Orte* wurden im bildbasierten Ansatz keine relevanten Bilder gefunden. Die Ursache liegt in der ungenauen Beschreibungsfähigkeit einer Szene durch die Labels. Auf der anderen Seite konnten die *Kategorien* einen hohen F1 von 23,21%, einen CR von 25,33% und eine Precision von 26% erreichen.

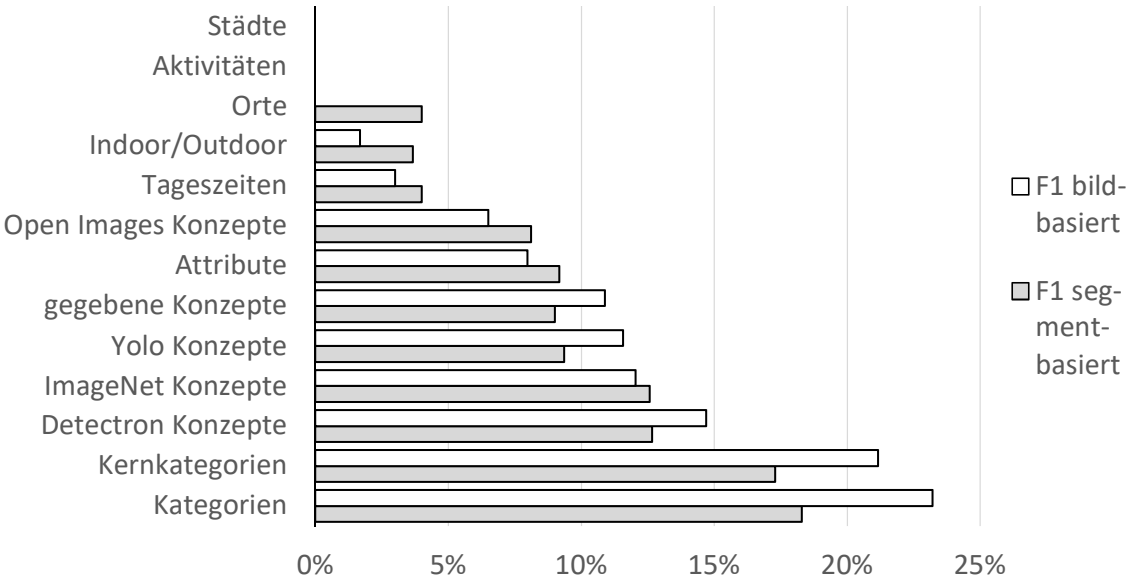


Abbildung 5.16: Das Diagramm stellt die höchsten F1-Scores pro Labelart für den bild- und segmentbasierten Ansatz dar. Für die Labelarten *Städte*, *Aktivitäten* und *Orte* wurden im bildbasierten Ansatz keine relevanten Bilder gefunden. Die *Kategorien* konnten einen hohen F1 von 23,21% erreichen.

Der höchste CR wurde durch die Kernkategorien mit 30,33% erzielt. Die Scores des segmentbasierten Ansatzes verhielten sich ähnlich zu denen des bildbasierten Ansatzes. Es ergaben sich folgende optimale Parameterkonfigurationen für die überwiegende Anzahl an Labelarten: *use_ceiling: false*, *use_idf: false* und *p: 1*. Die *Open Images Konzepte* und *Kategorien* profitierten durch ein Aufrunden und die *ImageNet Konzepte* erreichten den höchsten F1, indem nur Labels mit größer gleich 99% Konfidenz-Score genommen wurden. Die *Yolo Konzepte* performten am besten bei einer *threshold* von 90%. Zwischen den bild- und segmentbasierten Ansätzen unterschieden sich teilweise die optimalen *thresholds* pro Labelart. Für die *Open Images Konzepte* war es sinnvoll, den Parameter p auf den Wert 2 zu setzen. Ebenso resultierte der Wert 3 für die beiden *Indoor/Outdoor* Labels in dem höchsten F1. Für den Parameter *img_sel* war es für vier Labelarten irrelevant, welcher Wert genommen wurde, für weitere vier war *all* der beste Wert und die restlichen fünf erzielten mit dem Wert *first* die besten Resultate.

5.7.6 Submission 1

Da die *Kategorien* einen hohen F1 erzielten, wurde ein Test-Modell erstellt, welches auf dem entsprechenden Baseline-00 Modell basierte, und auf die Testanfragen angewandt („Kategorie Modell“). Diagramm 5.17 zeigt die Ergebnisse im Vergleich zu den Trainingsanfragen. Es ist ersichtlich, dass es in allen drei Metriken große Differenzen gab.

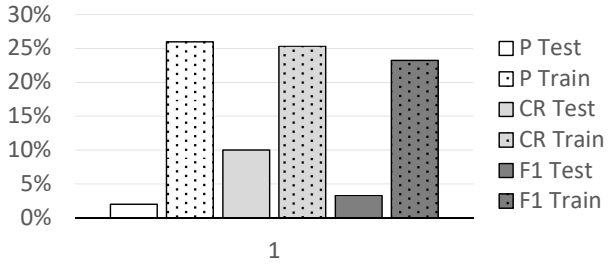


Abbildung 5.17: Das Diagramm zeigt die Ergebnisse des ersten Test-Modells auf beide Anfragetypen. Es ist ersichtlich, dass es in allen drei Metriken eine große Differenz gab. Die Precision betrug 2%, der Recall lag bei 10% und der F1 belief sich auf 3,3%. Dies entsprach einem Unterschied von 19,9% zwischen den F1-Metriken.

Die Precision betrug 2%, der Recall lag bei 10% und der F1 belief sich auf 3,3%. Dies entsprach einem Unterschied von 19,9% zwischen Trainings- und Test-F1.
Dieses Ergebnis war unerwartet, da es bei diesem Modell kein Overfitting gab, weil es sich um die Baselinekonfigurationen handelte. Es ließ sich vermuten, dass bei allen folgenden Modellen ebenfalls ein großer Unterschied zwischen den Metriken erkennbar sein würde, da die Konzepte der Testanfragen offensichtlich stark von denen der Trainingsanfragen abwichen.

5.7.7 Experimente des VK Modelltyps

Nachdem die optimalen Parameter für die Labelarten feststanden, wurden Experimente zum VK Modelltyp gemacht, welcher nur die gegebenen Labelarten aus den visuellen Konzepten enthielt. Der erste Modelltyp, welcher nur aus den Kategorielabels bestand, wurde demnach um die zwei anderen gegeben Labelarten erweitert. Die Experimente wurden wieder in Form einer Grid Search durchgeführt. Abbildung 5.18 zeigt die Ergebnisse der Experimente. Das Anordnen der Ergebnisse führte bei beiden Ansätzen zu einer signifikanten Erhöhung des F1 um ca. 3 bis 4%.

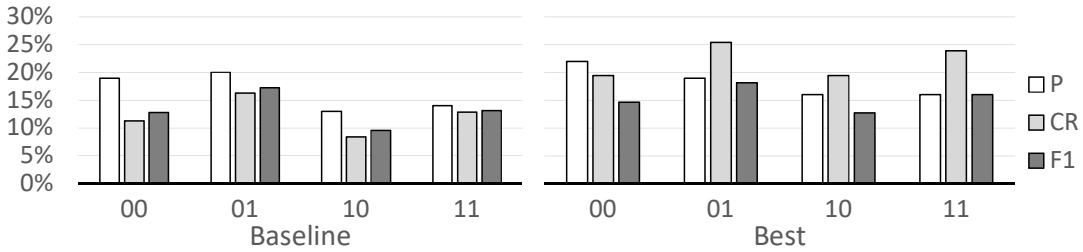


Abbildung 5.18: In den Ergebnissen der Experimente des VK Modelltyps ist erkennbar, dass der bildbasierte Ansatz gegenüber dem segmentbasierten zu einem besseren Score führte. Die Baseline-Modelle schnitten nur etwas schlechter als die besten Modelle ab.

Es ist erkennbar, dass der bildbasierte Ansatz gegenüber dem segmentbasierten Ansatz zu einem besseren Score führte. Es ließ sich bestenfalls ein F1 von 18,15% erreichen, wobei der Cluster Recall bei 25,43% und die Precision bei 19% lag. Für den segmentbasierten Ansatz ließ sich ein maximaler F1 von 15,97% erreichen. Dabei betrug der Cluster Recall 23,86% und die Precision 16%.
Die Baseline-Modelle schnitten nur etwas schlechter als die besten Modelle ab. Weiterhin war ersichtlich, dass bei ersteren die Precision höher als der Cluster Recall war und dass bei den besten Modellen fast immer das Gegenteil der Fall war. Für die Trainingsanfragen ließ sich demzufolge im Vergleich zum Kategorie Modell keine Verbesserung in den Metriken erzielen.

5.7.8 Submissions 2, 3, 4 und 5

Die vier Baseline-Modelle aus den Experimenten des VK Modelltyps wurden in etwas veränderter Form verwendet um die Testanfragen zu verarbeiten. Es wurden dabei nicht die besten Modelle verwendet, weil fortgeschrittenere Funktionalitäten wie die Einbeziehung von Tokengewichten erst in späteren Submissions folgen sollten und die Unterschiede zwischen den Metriken der Baseline-Modelle und der besten Modelle nur gering waren.

- Baseline-00 → Submission 2
- Baseline-01 → Submission 3
- Baseline-10 → Submission 4
- Baseline-11 → Submission 5

Die Ergebnisse der vier Modelle bildeten die Baseline für alle folgenden Testläufe. Diese ließen sich dadurch in Hinblick auf die besten Konfigurationen evaluieren. Die Evaluation der Baseline-Konfigurationen von den noch folgenden Modelltypen fand nicht statt, da es sich um doppelt so viele Submissions gehandelt hätte und die Anzahl an Submissions auf 15 begrenzt war.
Für den segmentbasierten Ansatz wurde die Threshold für die *gegebene Konzepte* Labelart wie folgt angepasst:

- Submission 4: 99%
- Submission 5: 95%

Der Grund für diese Anpassung war, dass die entsprechenden Thresholds bei den Optimierungsversuchen von 5.7.5 im segmentbasierten Ansatz jeweils den höchsten F1 erreichten. Für die bildbasierten Modelle wurde die Threshold von 90% beibehalten.
In Abbildung 5.19 sind die Ergebnisse der vier Submissions aufgelistet. Es zeigte sich, dass für den segmentbasierten Ansatz die Unterschiede zwischen den Metriken viel geringer waren als die des bildbasierten Ansatzes. Ersterer erreichte einen maximalen Test-F1 von 3,9%, wobei letzterer einen F1 von 9% erlangte und damit besser

abschnitt, obwohl es in den Ergebnissen der Optimierungsversuche umgekehrt gewesen war. Der Cluster-Recall überbot stets die Precision. Weiterhin ist erkennbar, dass das Anordnen für den segmentbasierten Ansatz zu einer kleinen Verbesserung um 0,4% führte (für beide Anfrage-Arten) und dass im bildbasierten Ansatz das Anordnen nur bei den Trainingsanfragen zu einem höheren F1 führte.

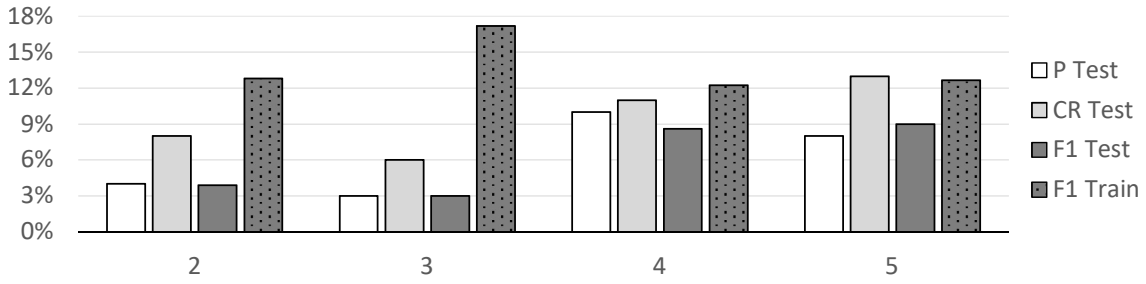


Abbildung 5.19: Die Ergebnisse des auf die Testanfragen angewandten VK Modelltyps zeigten, dass für den segmentbasierten Ansatz die Unterschiede zwischen den Metriken viel geringer waren als die des bildbasierten Ansatzes. Für den ersten Ansatz wurde ein maximaler Test-F1 von 3,9% erreicht, wobei letzterer einen F1 von 9% erlangte und damit besser abschnitt.

Für die Testanfragen resultierte eine Steigerung des F1 gegenüber dem Kategorie Modell um 5,7% auf 9%.

5.7.9 Experimente mit XGBoost

Um zu sehen, wie der maschinelle Ansatz für die Trainingsanfragen performte, wurden zu Gk-00 und Gk-10 des VK Modelltyps Experimente durchgeführt. Da der Hauptfokus auf den Modellen ohne maschinellem Lernen lag, wurden die beiden anderen Gk aus Zeitgründen außen vor gelassen.
Es zeigte sich in den Ergebnissen der Experimente (siehe Abbildung 5.20), dass der bildbasierte Ansatz auch hier für die Trainingsanfragen besser funktionierte.

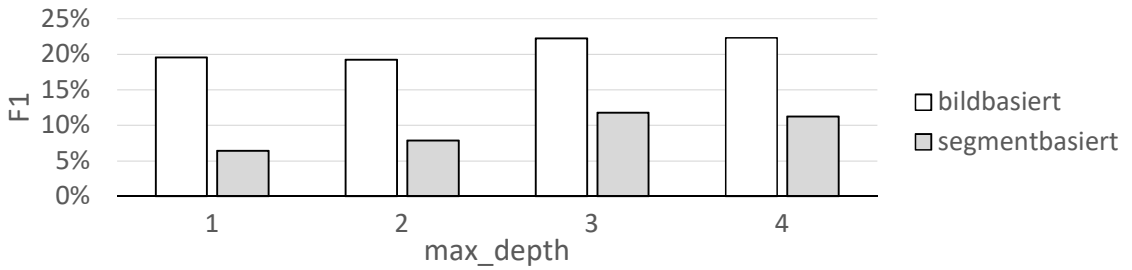


Abbildung 5.20: Die Ergebnisse der Experimente mit XGBoost zeigten ähnliche Resultate wie die des Vektor-Modells aus 5.6.10. Der bildbasierte Ansatz funktionierte deutlich besser als der segmentbasierte Ansatz.

Es ist erkennbar, dass je größer der Parameter *max_depth* gewählt wurde, desto besser war in der Regel der F1 (genauso wie bei 5.6.10). Der maximale bildbasierte F1 betrug 22,35% und der maximale segmentbasierte F1 lag bei 11,78%. Auf eine detailliertere Auswertung wird an dieser Stelle verzichtet, da diese den Rahmen der Arbeit sprengen würde. Es ließ sich somit auch hier keine Verbesserung im F1 erzielen. Weiterhin fiel auf, dass die F1-Scores gegenüber dem XGB Modelltyp der vektorbasierten Modellarchitektur viel niedriger waren. Die Ursache hierfür lag wahrscheinlich an der anderen Art der Ähnlichkeitsberechnung zwischen Bildern und Anfragen.

5.7.10 Submissions 6 und 7

Beide XGBoost Modelltypen wurden mit *max_depth = 3* auf die Testanfragen angewandt. Es ergaben sich die beiden Submissions:

- Gk-00 → Submission 6
- Gk-10 → Submission 7

Für den bildbasierten Ansatz wurde kein relevantes Bild gefunden. Das war zu erwarten gewesen, da bereits in den Versuchen aus 5.6.10 geschlussfolgert wurde, dass der maschinelle Ansatz zu Overfitting führte. Umso überraschender war es, dass der segmentbasierte Ansatz einen F1 von 4,6% erzielte (siehe Diagramm 5.21). Vermutlich waren in den Segmenten viele relevante Bilder, von denen zufällig einige von XGBoost selektiert wurden.

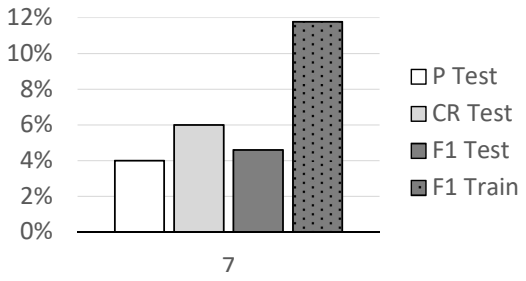


Abbildung 5.21: Für den maschinellen bildbasierten Ansatz wurde wie erwartet über alle Testanfragen kein relevantes Bild gefunden. Umso überraschender war es, dass der segmentbasierte Ansatz einen F1 von 4,6% erzielte.

Der Test-F1 wurde demnach nicht verbessert und es lässt sich schlussfolgern, dass maschinelles Lernen nicht für diese Art von Challenges geeignet ist.

5.7.11 Experimente des VK MBT Modelltyps

Der VK Modelltyp wurde um die Labelarten der minutenbasierten Tabelle zum VK MBT Modelltyp erweitert. Damit wurden nur gegebene Daten verwendet und auf

die zusätzlich generierten Labelarten verzichtet. Das Ziel bestand darin, herauszufinden, ob eine Zuhilfenahme weiterer Labelarten, die andere Aspekte des Lebens des Lifeloggers beschreiben, zu einer Verbesserung im Score führen.
Dazu wurde erneut eine Grid-Search mit den gleichen Parameterwerten durchgeführt, um die besten Parameterkonfigurationen zu identifizieren. Die Ergebnisse sind in der nachstehenden Abbildung 5.22 dargestellt. Dabei sind neben den Metriken auch die Unterschiede zu den Ergebnissen des VK Modelltyps dargestellt. In den Baseline-Konfigurationen ergab sich eine starke Verringerung in allen Metriken für den bildbasierten Ansatz. Zudem ließ sich nur in dem Best-00 Modell eine kleine Verbesserung von 0,05% erzielen. Das beste bildbasierte Modell erreichte einen F1 von 15,73%, wobei das beste segmentbasierte Modell einen F1 von 12,54% erlangte.

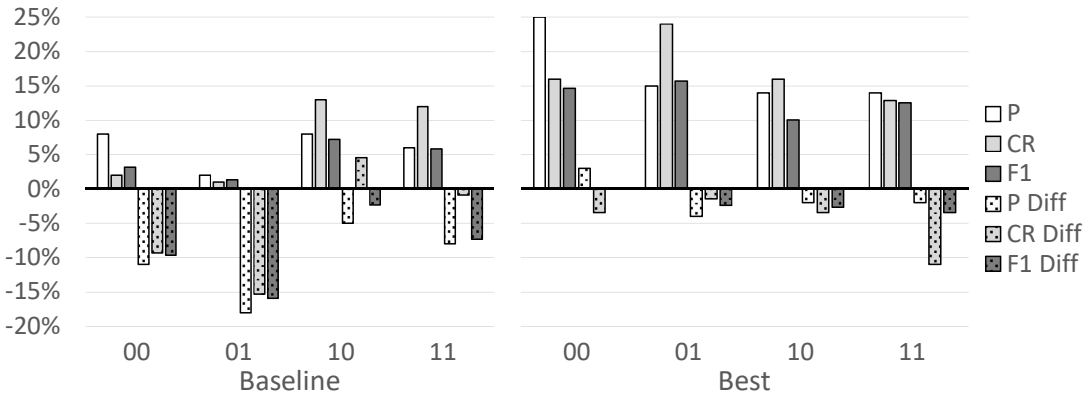


Abbildung 5.22: Die Ergebnisse der Experimente des VK MBT Modelltyps zeigen, dass es in den Baseline-Konfigurationen eine starke Verringerung in allen Metriken für den bildbasierten Ansatz gab. Das beste bildbasierte Modell erreichte einen F1 von 15,73%, wobei das beste segmentbasierte Modell einen F1 von 12,54% erlangte.

Die Wahl der Anfragequelle auf das Narrativ funktionierte für die Trainingsanfragen für diesen Modelltyp am besten, aber die Ursache dafür war mit hoher Sicherheit entweder Overfitting oder Zufall, da es zu viele Token pro Anfrage waren, die für sich betrachtet keinen Sinn machten. Die einzige Verbesserung ließ sich für die Trainingsanfragen nur für die Gk-00 mit einem F1 von 15,73% erzielen.

5.7.12 Submissions 8 und 9

Folgende Modelle aus den Experimenten des VK MBT Modelltyps wurden in abgewandelter Weise für die Verarbeitung der Testanfragen verwendet:

- Best-00 → Submission 8
- Best-10 → Submission 9

Der Grund dafür, dass Best-10 anstatt Best-11 verwendet wurde, obwohl es einen niedrigeren F1 erzielte, lag darin, dass es in den Submissions 4 und 5 keinen großen Unterschied im Score gab. Außerdem wurden die Modelle angepasst, indem nicht das Narrativ, sondern die Beschreibung als Anfrage-Quelle gewählt wurde und kein Tokenclustering benutzt wurde.
Die Ergebnisse der beiden Submissions in Abbildung 5.23 zeigen, dass es einen großen Unterschied in den Metriken zwischen Test- und Trainingsanfragen gab. Die Metriken für die Testanfragen waren außerdem ziemlich niedrig. Die achte Submission erzielte einen F1 von 1,7%, wobei die neunte einen F1 von 1,4% erreichen konnte.

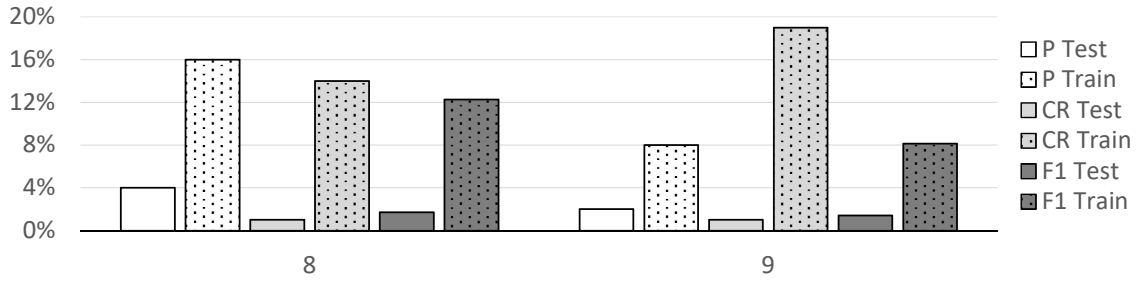


Abbildung 5.23: Die Ergebnisse für Submission 8 und 9 zeigten, dass es einen großen Unterschied in den Metriken zwischen Test- und Trainingsanfragen gab. Die achte Submission erzielte einen F1 von 1,7%, wobei die neunte einen F1 von 1,4% erreichen konnte. Damit ließ sich durch das Hinzunehmen der Labelarten aus der minutenbasierten Tabelle keine Verbesserung in den Scores erzielen.

Die beiden Submissions zeigten, dass sich durch das Hinzunehmen der Labelarten aus der minutenbasierten Tabelle keine Verbesserung in den Scores für die Testanfragen erzielen ließ.

5.7.13 Experimente des Erweiterten VK Modelltyps

In den Experimenten des VK MBT Modelltyps zeigte sich, dass die Hinzunahme der Labelarten aus der minutenbasierten Tabelle zu einer Verschlechterung im F1 führte. Deshalb wurde ein Modelltyp erstellt, welcher die gegebenen Labelarten der visuellen Konzepte um die zusätzlich generierten Labelarten erweiterte und die der minutenbasierten Tabelle ignorierte (Erweiterter VK Modelltyp).
Wie in den beiden vorherigen Versuchsreihen wurde wieder eine Grid Search mit den gleichen Parameterwerten durchgeführt und ausgewertet. In Abbildung 5.24 sind die Ergebnisse der Metriken inklusive der Unterschiede zur ersten Versuchsreihe (VK Modelltyp) dargestellt. Es ist wieder eine große Verringerung der Metriken in den Baseline-Modellen sichtbar. Die besten Modelle konnten den F1 vor allem für den bildbasierten Ansatz verbessern. Dort ließ sich ein maximaler F1 von 26,16% erreichen, wobei die Precision 35,29% betrug.

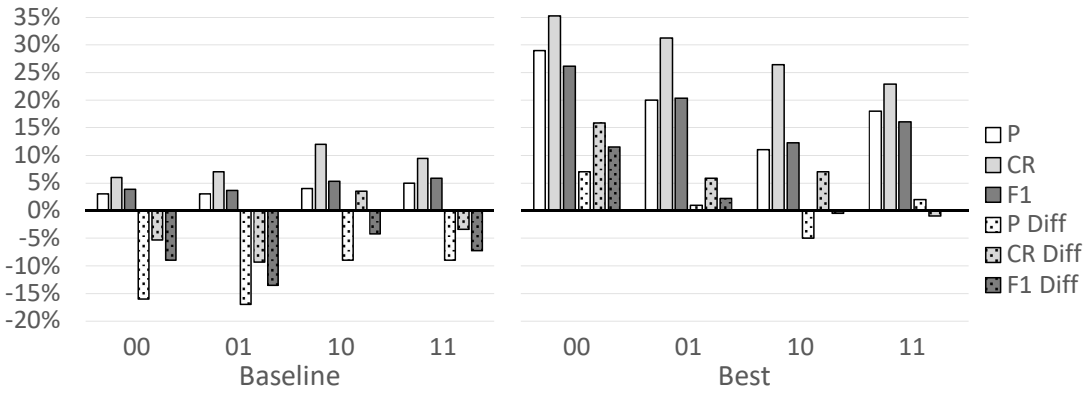


Abbildung 5.24: Die Ergebnisse der Experimente des Erweiterten VK Modelltyps zeigten, dass es wieder eine starke Verringerung in den Baseline-Metriken gab. Es ließ sich ein maximaler F1 von 26,16% erreichen.

Es fällt auf, dass in jedem Modell der Cluster Recall höher als die Precision war und dass der bildbasierte Ansatz in den besten Modellen besser funktionierte. Es stellte sich heraus, dass der Titel als Anfragequelle für Gk-00, Gk-01 und Gk-11 am besten funktionierte. Der F1 für den bildbasierten Ansatz ließ sich folglich auf 26,16% verbessern. Für den segmentbasierten Ansatz wurde kein höherer F1 erzielt.

5.7.14 Submissions 10 und 11

Für die beiden besten Modelle des Erweiterten VK Modelltyps wurde je Ansatz ein Test-Modell erstellt, bei dem alle Parameter gleich blieben, aber auf Tokenclustering verzichtet wurde.

- Best-00 → Submission 10
- Best-11 → Submission 11

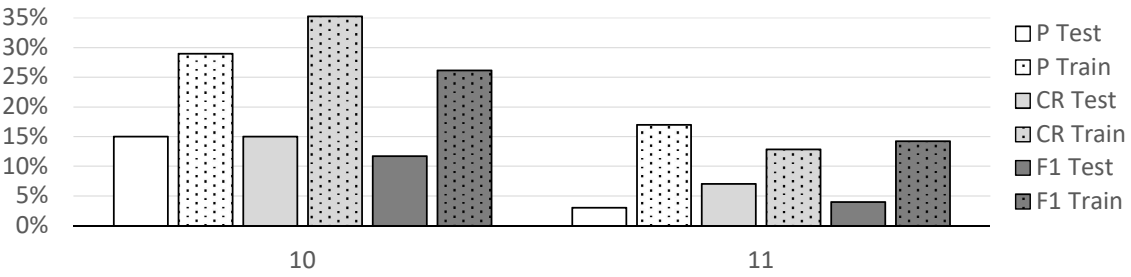


Abbildung 5.25: Bei den beiden Submissions 10 und 11 gab es wieder einen größeren Abstand zwischen den einzelnen Metriken. Submission 10 erreichte einen F1 von 11,7%, wobei die elfte Submission einen F1 von 4% erzielte.

Abbildung 5.25 zeigt die Ergebnisse. Es ist erkennbar, dass es wieder einen größeren Abstand zwischen den einzelnen Metriken gab. Submission 10 erreichte einen F1 von 11,7% bei einem Cluster Recall von 35,29% und die elfte Submission erzielte einen F1 von 4%. Der bildbasierte Ansatz schnitt demnach, wie in den Experimenten, besser ab. Es fällt auf, dass die Precision für den segmentbasierten Ansatz ziemlich niedrig war.
Folglich ließ sich der F1 für den bildbasierten Ansatz auf 11,7% verbessern, wobei der des segmentbasierten Ansatzes nicht verbessert wurde.

5.7.15 Experimente des Erweiterten VK MBT Modelltyps

Der letzte Modelltyp bestand aus allen verfügbaren Labelarten und deckte damit die meisten Bereiche des Lifelogs ab (siehe 4.2.1). Es wurde wieder eine Grid Search durchgeführt. Im nachstehenden Diagramm 5.26 sind die Ergebnisse der Experimente aufgelistet. Das Diagramm ähnelt Diagramm 5.24, hat aber durchschnittlich etwas niedrigere Scores. Der höchste F1 für den bildbasierten Ansatz betrug 21,62%, wobei der höchste F1 für den segmentbasierten Ansatz 13,94% war. Für die bildbasierten Baseline-Modelle wurden wieder die meisten Einbußen im Score gemacht. Die besten bildbasierten Modelle erzielten eine kleine Verbesserung gegenüber denen des VK Modelltyps. Es zeigte sich, dass die Verwendung des Tokenclusterings zu keinem besseren Score in dem bildbasierten Ansatz führte.

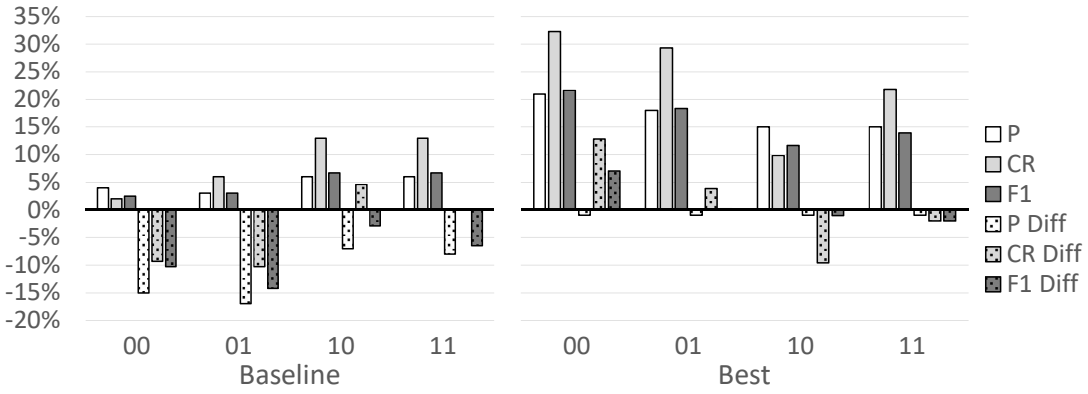


Abbildung 5.26: Die Ergebnisse der Experimente des Erweiterten VK MBT Modelltyps ähnelten denen des Erweiterten VK Modelltyps, aber es wurden insgesamt niedrigere Scores erzielt. Der höchste F1 für den bildbasierten Ansatz betrug 21,62%, wobei dieser für den segmentbasierten Ansatz bei 13,94% lag.

Es lässt sich schlussfolgern, dass dieser Ansatz für die Trainingsanfragen in keinem der Scores eine Verbesserung erzielen konnte. Die zusätzlichen Labelarten konnten die schwache Performanz der Labelarten aus der minutenbasierten Tabelle nicht ausgleichen.

5.7.16 Submissions 12 und 13

Aus den beiden Modellen Best-00 und Best-11 des Erweiterten VK MBT Modelltyps wurden wieder zwei Test-Modelle erstellt:

- Best-00 → Submission 12
- Best-11 → Submission 13

Die Ergebnisse auf die Testanfragen sind im folgenden Diagramm 5.27 dargelegt. Es lässt sich erneut eine große Differenz zwischen den Metriken beider Anfragetypen beobachten. Der segmentbasierte Ansatz erreichte sehr niedrige Ergebnisse für die Testanfragen mit einem F1 von 1,1%. Dies entsprach dem niedrigsten Score nach dem XGB Modell in RUN 7. Für den bildbasierten Ansatz ließ sich ein F1 von 8,7% erreichen. In beiden Submissions war die Precision für die Trainingsanfragen unter allen Metriken am höchsten.

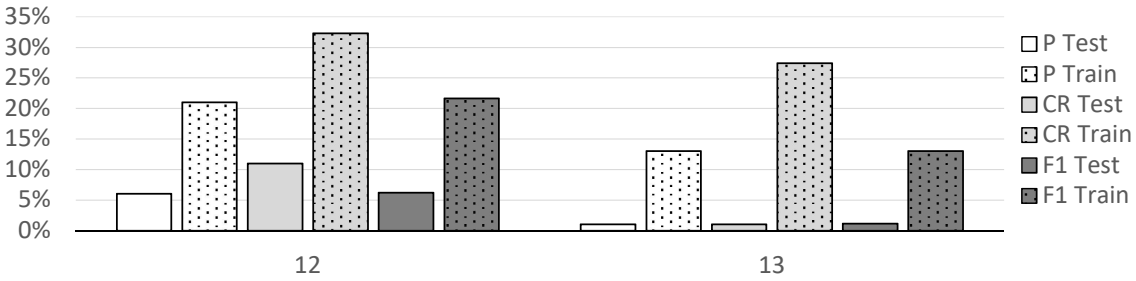


Abbildung 5.27: In den Ergebnissen der 12. und 13. Submission lässt sich eine große Differenz zwischen den Metriken der beiden Ansätze beobachten. Der segmentbasierte Ansatz erreichte sehr niedrige Ergebnisse für die Testanfragen mit einem F1 von 1,1%.

Demnach war es nicht möglich, den F1 zu verbessern, weil die Hinzunahme der Labelarten der minutenbasierten Tabelle erneut wie in 5.7.12 zu einer deutlichen Verschlechterung in den Metriken führte.

5.7.17 Fortführung der Experimente des Erweiterten VK Modelltyps

Da sich herausstellte, dass der bildbasierte Erweiterte VK Modelltyp den höchsten F1 unter allen Modelltypen für beide Anfragetypen erzielte, wurde eine Random Search mit 1.000 Versuchen durchgeführt, bei der die einzelnen Labelarten der Modelle unterschiedlich gewichtet wurden. Die Zuweisung erfolgte dabei zufällig anhand von 10% Schritten zwischen 0% und 100%. Nachdem die Suche beendet war, ergaben sich drei Gewichtskonfigurationen für die jeweils höchste Precision mit 41%, den höchsten Cluster Recall mit 36,29% und den höchsten F1 mit 33,10% (siehe Abbildungen 5.28 und 5.29). Diese drei Scores stellten die höchsten aller Versuche für die Trainingsanfragen dar.

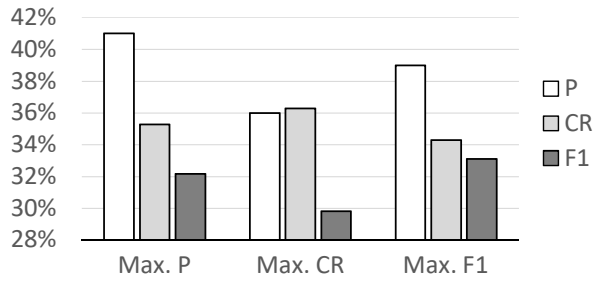


Abbildung 5.28: Das Diagramm stellt die Werte der Metriken für die jeweils besten Modelle mit der höchsten Precision (41%), mit dem höchsten Cluster Recall (36,29%) und dem höchsten F1 (33,10%) dar. Die hohen Scores wurden durch eine Gewichtung der Labelarten erzielt.

Anhand der Gewichte wurde eine manuelle Anpassung vorgenommen, um eine Konfiguration zu finden, die die Precision und den F1 gleichzeitig erhöhen sollte. Die vier Labelarten *Indoor/Outdoor, Kernkategorien, ImageNet Konzepte* und *Detectron Konzepte* erwiesen sich als die wertvollsten Labelarten und bekamen eine manuelle Gewichtung von 1, da sie mit mindestens 80% Gewichtung zur höchsten Precision beitrugen und diese ausschlaggebend für einen hohen F1 war. Der Cluster Recall spielte eine untergeordnete Rolle, da er einen F1 unter 30% erzielte. Die *Detectron Konzepte* performten als beste Labelart unter den auf dem COCO Datenset basierenden Labelarten. Die *Yolo Konzepte* trugen maßgeblich zur höchsten Precision bei und bekamen deshalb die Gewichtung von 50%. Die *gegebenen Konzepte* zeichneten sich als irrelevant ab und erhielten keine Gewichtung. Die *Attribute* und *Open Images Konzepte* sollten die anderen Labelarten ergänzen und erhielten eine Gewichtung von 50%.

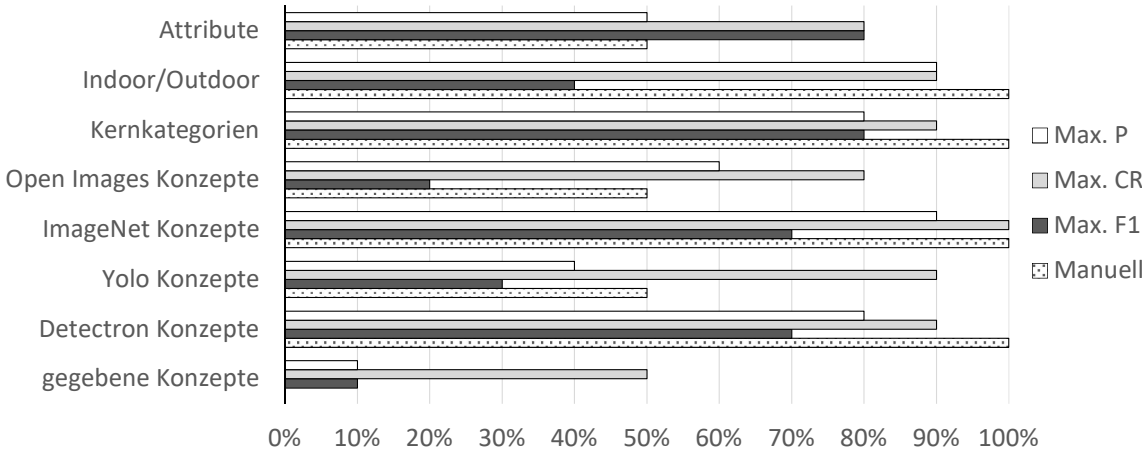


Abbildung 5.29: In dem Diagramm sind die zu den besten Modellen entsprechenden Gewichtungen der Labelarten dargestellt. Anhand von diesen Gewichten wurde eine manuelle Konfiguration erstellt, welche vor allem die Precision und den F1 noch weiter erhöhen sollte.

Mit Hilfe der manuellen Gewichte wurde erneut eine Grid Search durchgeführt, um die optimalen Parameterkonfigurationen zu finden. Es stellte sich heraus, dass sich lediglich ein F1 von 30,60% erreichen ließ, welcher damit schlechter als der zuvor erzielte höchste F1 war.
Die manuelle Anpassung war demnach nicht sinnvoll. Der F1 für die Trainingsanfragen ließ sich somit für den bildbasierten Ansatz auf 33,10% erhöhen.

5.7.18 Submission 14

In der finalen Submission wurde versucht, den Score zu verbessern, indem die Gewichte des Modells mit dem höchsten F1 aus der Random Search des Erweiterten VK Modells genommen wurden. Die Wahl des Modelltyps fiel trotz des schlechteren Scores gegenüber dem Erweiterten Modelltyp auf den Erweiterten VK MBT Modelltyp, wobei die Gewichte der MBT Labels auf Eins gesetzt wurden, um alle gleich einzubeziehen, in der Hoffnung dass mehr Metadaten die Testanfragen besser verarbeiten würden als weniger. Es stellte sich jedoch heraus, wie in der nachstehenden Abbildung zu sehen ist, dass dem nicht so war. Zwar erreichte der F1 für die Trainingsanfragen einen besseren Wert als in der zehnten Submission, aber für die Testanfragen resultierte lediglich ein F1 von 8,7%. Die Ursache dafür war die niedrige Precision und das daraus entstandene hohe Ungleichgewicht im Vergleich zum Cluster Recall. Mit 21% war dieser der höchste, welcher in allen Test-Modellen erzielt wurde.

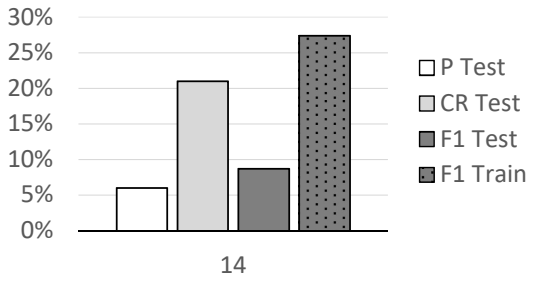


Abbildung 5.30: Die letzte Submission führte zu den höchsten Cluster Recall für die Testanfragen mit 21%. Ein neuer Bestwert für den F1 wurde hingegen nicht gefunden.

Damit ergab sich kein neuer Highscore für den F1 der Testanfragen. Es wäre womöglich besser gewesen, den Erweiterten VK Modelltyp zu verwenden und die Submission einzureichen.

Auszüge aus den zurückgegebenen Bildern

Da sich dennoch hohe F1-Scores ergaben, sind im Folgenden einige Auszüge aus den zurückgegebenen Bildern der Trainings- und Test-Modelle für die einzelnen Anfragen dargestellt (siehe Abbildungen 5.31 und 5.32).

Trainingsanfragen Für die erste Trainingsanfrage wurden sieben relevante Bilder des einzigen Clusters gefunden, wodurch sich ein Cluster Recall von 100% und eine Precision von 70% ergab. Bei zwei anderen Bildern war zwar das Meer zu sehen, aber es handelte sich um einen anderen Moment, bei dem der Lifelogger kein Eis aß. Für die zweite Anfrage wurde überraschenderweise kein relevantes Bild gefunden. Die Ursache dafür lag darin, dass dem Token „having“ kein sinnvolles Label zugeordnet werden konnte. Es handelte sich bei allen Top 20 Bildern um den gleichen Moment, bei dem der Lifelogger durch eine Einkaufsmeile lief. Das Modell fand für die dritte Anfrage unter den Top 20 Bilder ausschließlich relevante Bilder, bei dem der Lifelogger abends vor seinem Fernseher saß. Dabei wurden sechs unterschiedliche Momente gefunden, wodurch sich eine Precision von 100% und ein Cluster Recall von 60% ergab. Der Grund für diese hohen Scores war, dass das Label „tv“ in allen drei COCO Konzept Labelarten vorkam und annotiert wurde. Bei der vierten, schwierigsten Trainingsanfrage wurde kein relevantes Bild gefunden. Immerhin wurden zwei Bilder mit abgebildeten Brücken und ein Bild, bei dem der Lifelogger einen Hafen fotografiert, zurückgegeben.

Anfr. 1: zweites Bild | Anfr. 3: erstes Bild | Anfr. 4: erstes Bild | Anfr. 5: elftes Bild

Anfr. 7: erstes Bild | Anfr. 8: erstes Bild | Anfr. 10: zweites Bild | Anfr. 10: fünftes Bild

Abbildung 5.31: Auszüge aus den zurückgegebenen Bildern des finalen Trainings-Modells verdeutlichen, dass das Verfahren häufig keine relevanten Bilder fand, weil ungenaue Token im Titel vorkamen. Insgesamt wurden bei sechs Anfragen relevante Bilder zurückgegeben.

Für die fünfte Anfrage gab es auch keine relevanten Bilder unter den Top 10 Ergebnissen. Allerdings stellten die Bilder 11-20 alle einen relevanten Moment dar, bei dem der Lifelogger an der Wursttheke Fleisch kaufte. Für die nächste, sechste Anfrage wurde kein Bild gefunden, auf dem der Lifelogger Gitarre spielte. Die Ursache lag höchstwahrscheinlich daran, dass sich das Token „playing“ wieder keinen sinnvollen Labels zuordnen ließ. Bei der siebten Anfrage, bei der Momente gefunden werden sollten, in denen der Lifelogger kochte, wurden drei relevante Momente zurückgege-

ben. Die anderen Bilder zeigten Momente, in denen gegessen wurde oder bei denen der Lifelogger sich einen Ofen in einem Elektronikladen angesehen hatte. Für die achte Trainingsanfrage waren sechs relevante Bilder unter den Top 10 Ergebnissen, die anderen zeigten die Momente, bei denen die Person mit dem Auto fuhr. Unter den sechs Bildern kamen alle zwei möglichen Momente vor, wodurch sich ein Cluster Recall von 100% und eine Precision von 60% ergab. Der Grund für diese hohen Scores ist wieder, dass es das Label „auto showroom“ unter den Kategorien gab. Bei der vorletzten Anfrage wurde kein relevantes Bild gefunden, da wieder ein irrelevantes Token („public“) im Titel vorkam, welches sich nicht zuordnen ließ. Für die zehnte Anfrage wurden fünf relevante Bilder ausgegeben, welche zwei unterschiedliche Momente darstellten. Ein anderer Moment wurde zusätzlich ausgegeben, in dem der Lifelogger zwar ein Buch in der Hand hatte, aber lediglich Geld darin versteckte.

Testanfragen Die Auszüge aus den Predictions für die Testanfragen ergaben, dass für die Anfragen 1, 3, 6, 7 und 8 mindestens ein relevantes Bild gefunden wurde. Das erste Bild der ersten Anfrage enthielt korrekterweise ein Spielwarengeschäft. Neben diesem wurden zwei weitere relevante Bilder gefunden. Die restlichen Bilder waren aus einem Baumarkt, einem Supermarkt oder aus einem Kiosk. Dies bedeutete, dass die Spezifikation nach Spielzeugen in den Modellen scheiterte.


Anfr. 1: erstes Bild


Anfr. 2: erstes Bild


Anfr. 3: erstes Bild


Anfr. 6: fünftes Bild


Anfr. 7: siebentes Bild


Anfr. 8: zweites Bild


Anfr. 9: erstes Bild

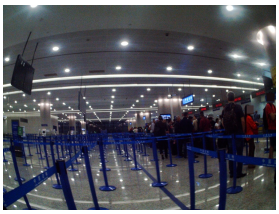

Anfr. 10: erstes Bild

Abbildung 5.32: Auszüge aus den zurückgegebenen Bildern des finalen Test-Modells zeigen, dass für die Hälfte der Anfragen mindestens ein relevantes Bild gefunden wurde. Oft beschrieben die zurückgegebenen Bilder nur einen Moment, weshalb der Cluster Recall niedrig war.

Bei der zweiten Anfrage wurden ausschließlich Bilder aus Sicht eines Autofahrers zurückgegeben, welche aber alle am selben Tag und auf dem Weg zur Arbeit und nicht von der Arbeit weg geschossen wurden. Für die dritte Anfrage wurden drei Bilder unter den zehn besten Bildern gefunden, welche den Kühlschrank von innen

zeigten. Bei den anderen Bildern handelte es sich zwar um Bilder aus der Küche, aber es war kein Kühlschrank abgebildet. Unerwarteterweise stellten die Bilder 11-20 alle den Kühlschrank von innen dar und waren damit relevant. Wäre X = 20 gewählt wurden, hätte hier die Precision deutlich zugenommen, von 30% auf 65%. Für die vierte Anfrage, bei welcher Momente herausgesucht werden sollte, bei dem der Lifelogger Fußball schaute, wurden keine relevanten Bilder gefunden, sondern vier Bilder, auf denen ein Laptop zu sehen war, zurückgegeben, welche wahrscheinlich auf das Token „watching“ ansprachen. Das Modell fand für die fünfte Anfrage ebenfalls kein relevantes Bild, bei dem ein Kaffee abgebildet war. Die Ursache dafür war, dass „time“ ein Token des Titels war, welchem sich keine sinnvollen Labels zuordnen ließen. Bei der sechste Anfrage wurden acht relevante Bilder gefunden, welche jedoch alle einen Moment darstellten. Somit war die Precision hoch, aber der Recall nur bei 10%. Bei der siebten Anfrage befanden sich drei relevante Bilder unter den Top 10, welche alle den gleichen Moment darstellten. In den anderen Bildern war der Lifelogger zwar in einer Kantine, aber trank keinen Kaffee. In Bildern 11 bis 20 befand sich noch ein weiterer relevanter Moment. Bei der achten Anfrage ist unter den ersten 20 zurückgegebenen Bildern immer das Handy des Lifeloggers zu sehen, von denen aber nur drei Bilder draußen aufgenommen wurden, welche aber alle verschiedene Momente darstellten. Für die vorletzte Anfrage wurde kein Bild gefunden, bei dem der Lifelogger ein rotes kariertes Hemd trug, sondern nur ein Bild auf dem ein gestreiftes Hemd und ein gestreifter Schirm zu sehen war. Diese Anfrage stellte die schwierigste unter allen dar, da die Kamera die Bilder abgewandt vom Lifelogger schoss und daher das Hemd nur anhand der zu sehenden Arme erkannt werden hätte können. Das Modell lieferte für die letzte Anfrage ausschließlich Bilder für den Moment, bei dem der Lifelogger am Flughafen in China war. Dieser Moment stellte also kein Meeting dar. Das Verfahren schlussfolgerte fälschlicherweise, dass die Menschen auf dem Bild in einem Meeting waren.

5.8 Vergleich mit anderen Wettbewerbsteilnehmern

In Abbildung 5.33 sind die Ergebnisse der Teilnehmer des diesjährigen Wettbewerbes dargestellt. Der in dieser Arbeit vorgestellte Ansatz erreichte den 6. Platz. Insgesamt nahmen 36 Teams teil, wobei jedoch nur 8 davon mindestens eine Submission einreichten. Die Organisatoren zählten nicht mit in die offizielle Statistik ein, da sie die korrekten Bilder zu den Testanfragen besaßen. Das beste Verfahren entwickelte die Ho Chi Minh City University of Science (HCMUS) mit einen F1 von 61%. Dabei ist jedoch zu beachten, dass dieses Team bereits im letzten Jahr an dem Wettbewerb teilnahm und es daher schon auf einem vorhandenen Verfahren aufbauen konnte. Weiterhin lassen sich die Ansätze nicht direkt miteinander vergleichen, da sie unterschiedliche Herangehensweisen verwendeten. Das HCMUS Team erreichte den hohen F1 unter anderem, weil sie einen Human in the Loop Ansatz verwendeten und Wörter aus den Anfragen teilweise manuell extrahierten [56]. Das Verfahren der vorliegenden Arbeit basierte jedoch ausschließlich auf einem automatisierten Ansatz.

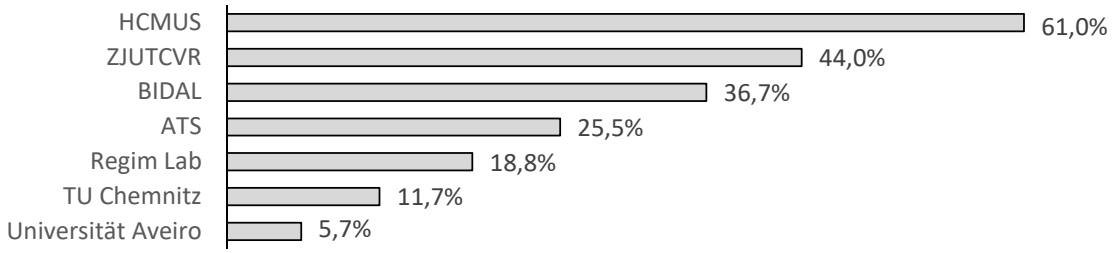


Abbildung 5.33: Das Leaderboard des diesjährigen Wettbewerbes zeigt, dass mit dem in dieser Arbeit entwickelten Verfahren der 6. Platz erreicht wurde. Das beste Team war HCMUS, welches einen F1 von 61% erzielte.

Das nachstehende Diagramm in Abbildung 5.34 stellt die Anzahl an Submissions je Team dar. Es fällt auf, dass die besten drei Teams jeweils maximal drei Submissions einreichten. Das Ziel der ImageCLEF war jedoch ein Ausprobieren vieler verschiedener Herangehensweisen, um dadurch Schlüsse ziehen zu können, welche Ansätze zielführend sind. Werden nur drei Submissions getätigt, lässt sich kaum etwas über die Auswirkungen von Parameteränderungen aussagen. Deshalb war das Ziel dieser Arbeit vor allem, so viele verschiedene Labelartkombinationen und Parametereinstellungen wie möglich auszuprobieren.

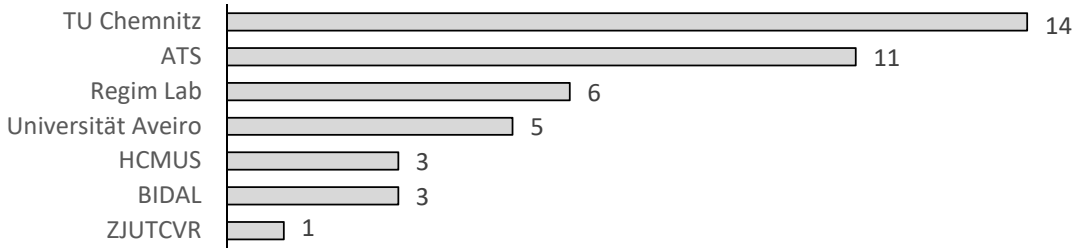


Abbildung 5.34: Das Diagramm stellt die Anzahl an Submissions je Teilnehmer dar. Es ist auffällig, dass viele Teilnehmer nur wenige Submissions tätigten.

Abschließend lässt sich sagen, dass Human in the Loop Ansätze wie erwartet besser als die automatischen Ansätze abschnitten und dass die maximal erreichten 61% F1 immer noch Luft nach oben für bessere Verfahren des computergestützten Lifelog Moment Retrievals lassen.

6 Zusammenfassung und Ausblick

Lifelogging spielt eine große Rolle im medizinischen Sektor, da es unter anderem Menschen mit einer Gedächtnisschwäche die Erinnerung an 80% ihrer vergangenen Ereignisse ermöglicht. Außerdem können sich Menschen mit Amnesie durch Lifelogs besser an signifikante Momentdetails erinnern.
Da das manuelle Auswerten von Lifelogs aufgrund der großen Datenmengen viel Zeit an Anspruch nimmt und sich durch einen Computer Verarbeitungsschritte schneller als per Hand ausführen lassen, gibt es eine hohe Nachfrage nach computergestützten Verfahren zur Verarbeitung von Lifelogs. Das zeigt sich auch in einer steigenden Anzahl an Data Science Challenges. Einige Beispiele sind die seit 2017 stattfindende Lifelogging Challenge von ImageCLEF, der Lifelogging Task der NTCIR[1], den es auch seit 2017 gibt oder die Lifelog Search Challenge (LSC) [26] des ICMR (ACM International Conference on Multimedia Retrieval), welche seit 2018 jährlich stattfindet[2].
Daher existieren bereits viele Verfahren zur computergestützten Verarbeitung von Lifelogs. Allein bei ImageCLEF wurden bis zum jetzigen Zeitpunkt 17 Paper eingereicht, welche Verfahren für Lifelogs zur Anfragenverarbeitung vorstellten. Diese Verfahren wurden in der vorliegenden Arbeit analysiert. Es stellte sich heraus, dass diese Verfahren sich aus den Komponenten Bildsegmentierung, Metadatenverarbeitung, Anfragenverarbeitung, Filterung, Retrieval und Diversifikation zusammensetzten. Für jede dieser Komponenten waren zahlreiche Methoden verfügbar, zum Beispiel bestand die Bildsegmentierung aus der Extraktion von Bildinformationen, dem Vergleich von Bildern im Rahmen eines hierarchischen Clusterings und aus der Keyframe Selection.
In dieser Arbeit wurden mehrere Verfahren entwickelt und evaluiert. Dafür war ein Datenset gegeben, welches aus 63.696 Bildern bestand, zu welchen die Verfahren zehn Test- und Trainingsanfragen verarbeiten sollten. Alle Bilder waren bereits annotiert und zusätzliche Metadaten, wie beispielsweise die Herzrate, der Blutzuckerspiegel oder der Ortsname waren ebenfalls gegeben. Es zeigte sich, dass es einige Fehlannotationen im Datenset gab, zum Beispiel wurde für ein Bild am Flughafen der Ort „East China Normal University“ annotiert oder ein Eis als „person“ gelabelt. Außerdem beinhaltete die Ground Truth Cluster mit Bildern aus mehreren Momenten. Dies konnte bei bestimmten Umständen zu einer Verringerung des Cluster Recalls führen. Die Test- und Trainingsanfragen hatten viele Gemeinsamkeiten, unterschieden sich aber teilweise im Detailgrad, sodass zum Beispiel das Land eines

[1] http://ntcir-lifelog.computing.dcu.ie/
[2] http://lsc.dcu.ie/

gesuchten Moments nur bei den Testanfragen ein Rolle spielte. Weiterhin beinhalteten einige Anfragen Begriffe, die in den gegebenen Annotationen nicht vorkamen, wie beispielsweise das Label „icecream“. Daher wurden entsprechende Labels mit Software Systemen wie Detectron und Yolo zusätzlich annotiert.
Python wurde als Entwicklungsumgebung benutzt, da es sich im Data Science Bereich sehr gut eignete. Es wurde eine zufallsbasierte, regelbasierte, segmentbasierte, vektorbasierte und poolbasierte Modellarchitektur entworfen und mit Hilfe der Metriken Precision, Cluster Recall und F1 evaluiert. Für jede Architektur wurden Probleme identifiziert und es wurde versucht, diese in der darauf folgenden Architektur zu beheben.
Die erste, zufallsbasierte Modellarchitektur erzielte einen durchschnittlichen F1 von 1,62%. Dieser Wert zählte als unterste Schranke für alle folgenden Versuche, da kein Verfahren schlechter als der Zufall sein sollte.
In der darauffolgenden, regelbasierten Modellarchitektur wurde der F1 auf 3,27% erhöht, wobei dies immer noch einen niedrigen Wert darstellte. Es wurden neben den Labelarten aus den gegebenen visuellen Konzepten die zusätzlich genierten Labelarten *Detectron Konzepte* und *Yolo Konzepte* verwendet. Die Hauptprobleme dieser Architektur waren, dass Bilder ohne Konzepte ignoriert wurden und es keine Möglichkeit gab, die Konfidenz-Scores pro Labelart anzupassen, um inkorrekte Annotationen herauszufiltern.
Die dritte, segmentbasierte Modellarchitektur wurde nur zur Optimierung der Bildsegmentierung genutzt und verwendete daher keine der Labelarten. In zwei Durchgängen in insgesamt 546 Experimenten wurden die besten Parameter identifiziert. Am Ende resultierten 4.311 Segmente und die ursprüngliche Bildanzahl ließ sich um 93,23% verringern, wobei der Cluster Recall immer noch 89,37% betrug.
Die Segmentierung wurde anschließend in der vektorbasierten Modellarchitektur genutzt, bei welcher sieben Parameter mit Hilfe von 181 Experimenten für den bildbasierten und segmentbasierten Ansatz optimiert wurden. Für ersteren ließ sich ein maximaler F1 von 24,02% erreichen, wobei die Kategorien unter allen Labelarten am besten funktionierten. Für den segmentbasierten Ansatz ließ sich ein F1 von 21,21% erreichen. Es wurde zudem ein Ansatz mit maschinellem Lernen implementiert, welcher einen maximalen F1 von 52,07% auf die Trainingsanfragen erzielte. Dieser hohe Wert war jedoch auf Overfitting zurückzuführen. Die Schwächen der Architektur lagen darin, dass die Labelarten ungleichmäßig gewichtet wurden, dass die Konfidenz-Scores ebenfalls nicht gefiltert wurden und dass die Inverse Dokumentfrequenz inkorrekt einbezogen wurde.
Die finale, komplexeste Architektur basierte auf einer Kombination von verschiedensten Labelarten. Es wurden Labelarten erzeugt, welche den Zustand, die Zeit und den Ort beschrieben. Es ergaben sich 13 Labelarten mit insgesamt 1.541 Labels. Aus diesen wurden verschiedene Modelltypen mit variierenden Gewichten je Labelart erstellt. Für jeden Typ wurden mehrere Experimente durchgeführt, wobei in der Evaluation jeweils zwischen dem bild- und segmentbasierten Ansatz und einer Variante mit und ohne Selektion des besten Bildes pro Tag unterschieden wurde. Außerdem erfolgte die Evaluation anhand einer Baseline-Konfiguration und der

jeweils besten Modelle. Der segmentbasierte Ansatz schnitt mit einem maximalen F1 von 16,06% für die Trainingsanfragen und 9% für die Testanfragen ab. Es ergab sich für den bildbasierten Ansatz ein Trainings-F1 von 33,13% und ein Test-F1 von 11,7%. Der entsprechende Modelltyp verwendete ausschließlich Labelarten aus Objektannotationen, woraus sich schließen ließ, dass diese am relevantesten waren. Es gab häufig einen großen Unterschied zwischen den Scores beider Anfragetypen. Dies lag vermutlich an den Unterschieden in den Anfragen, an der Möglichkeit, dass die Veranstalter des Wettbewerbes die Metriken auf eine andere Weise implementierten oder daran, dass das Verfahren auf die Trainingsanfragen optimiert wurde. Bei genauer Analyse der zurückgegebenen Bilder stellte sich heraus, dass viele irrelevante Bilder in der finalen Ergebnisliste landeten, weil in den Anfragen Token enthalten waren, die sich keinen klaren Labels zuordnen ließen, wie z.B. die Token „time“, „playing“ oder „public“. Die auf maschinellem Lernen basierenden Ansätze führten zu starkem Overfitting, von denen einer zu einem F1 von 0% für die Testanfragen resultierte. Das Zurückgeben der jeweils besten Bilder pro Tag für die segmentbasierten Ansätze führte immer zu einer Erhöhung des F1. Zudem ermöglichte die Angabe einer Threshold für die Konfidenz-Scores die effektive Identifikation der zutreffenden Labels pro Labelart. Die IDF wurde in diesem Verfahren gegenüber der vorherigen Modellarchitektur korrekt einberechnet, führte aber dennoch zu keiner Verbesserung.
Das wichtigste Feature war das Gewichten der Token je Labelart, da sich durch dieses die Labelarten je nach Kontext unterschiedlich einberechnen ließen, was immer zu einer Erhöhung des F1 führte. Das Clustern der Token erhöhte den F1 nicht, die Ursache dafür kann aber auch darin begründet sein, dass die Anfragen wenige Oder-Verknüpfungen enthielten. Der Titel stellte sich als Anfrage-Quelle am geeignetsten heraus, da er die meisten relevanten Token enthielt und das Potenzieren der Ähnlichkeiten führte nur bei wenigen Labelarten zu einer Verbesserung. Das Bilden des Durchschnittes der höchsten Token-Ähnlichkeiten je Labelart führte ebenfalls immer zu einer Verbesserung im Score.
Schlussendlich war der bildbasierte Ansatz in jeder Modellarchitektur besser als der segmentbasierte Ansatz. Die geringe Anzahl an Trainingsanfragen machte es schwierig, ein allgemeines, automatisiertes Verfahren zu entwickeln, da es mit Ansteigen der möglichen Parameter schnell zum Overfitting kam.
Zukünftige Verbesserungen könnten darin bestehen, neben den Ähnlichkeitswerten auch Unähnlichkeitswerte zu berechnen. So könnte zum Beispiel die Annotation eines bestimmten Labels dazu führen, dass ein Bild komplett aus der Liste der potentiellen Bildern entfernt wird. Weiterhin kann es sinnvoll sein, mehr Metadaten zu nutzen, um dadurch Momente besser beschreiben zu können. Relevante Metadaten könnten beispielsweise die Herzrate oder der Blutzuckerspiegel sein, zum anderen könnten CNNs noch weitere Annotationen durchführen. Mit Open Pose [15] ließen sich beispielsweise Hände des Lifeloggers identifizieren, welche es dann ermöglichten, falsche „person“-Annotationen herauszufiltern. Das Entfernen irrelevanter Token stellt eine weitere Option dar, welche zu einer Verbesserung des Verfahrens führen könnte.

Literaturverzeichnis

[1] Abdallah, F.B., Feki, G., Ezzarka, M., Ammar, A.B., Amar, C.B.: Regim Lab Team at ImageCLEF Lifelog Moment Retrieval Task 2018 S. 10 (2018)

[2] Allen, A.L.: Dredging up the Past: Lifelogging, Memory, and Surveillance. The University of Chicago Law Review S. 28 (2008)

[3] Anderberg, M.R.: Cluster Analysis for Applications: Probability and Mathematical Statistics: A Series of Monographs and Textbooks, vol. 19. Academic press (2014)

[4] Anderson, J.R., Reder, L.M.: The fan effect: New results and new theories. Journal of Experimental Psychology: General 128(2), S. 186 (1999)

[5] Anderson, M.C., Bjork, R.A., Bjork, E.L.: Remembering Can Cause Forgetting: Retrieval Dynamics in Long-Term Memory S. 25 (1994)

[6] Berry, E., Kapur, N., Williams, L., Hodges, S., Watson, P., Smyth, G., Srinivasan, J., Smith, R., Wilson, B., Wood, K.: The use of a wearable camera, SenseCam, as a pictorial diary to improve autobiographical memory in a patient with limbic encephalitis: A preliminary report. Neuropsychological Rehabilitation 17(4-5), S. 582–601 (2007)

[7] Boag, S., Chamberlin, D., Fernández, M.F., Florescu, D., Robie, J., Siméon, J., Stefanescu, M.: XQuery 1.0: An XML query language (2002)

[8] Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching Word Vectors with Subword Information. Transactions of the Association for Computational Linguistics 5, S. 135–146 (Dez 2017)

[9] Bolanos, M., Mestre, R., Talavera, E., Giro-i-Nieto, X., Radeva, P.: Visual summary of egocentric photostreams by representative keyframes. In: 2015 IEEE International Conference on Multimedia & Expo Workshops (ICMEW). S. 1–6. IEEE, Turin, Italy (Jun 2015)

[10] Bossard, L., Guillaumin, M., Van Gool, L.: Food-101 – Mining Discriminative Components with Random Forests. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) Computer Vision – ECCV 2014, vol. 8694, S. 446–461. Springer International Publishing, Cham (2014)

[11] Boteanu, B., Mironica, I., Ionescu, B.: Hierarchical clustering pseudo-relevance feedback for social image search result diversification. In: 2015 13th International Workshop on Content-Based Multimedia Indexing (CBMI). S. 1–6. IEEE, Prague, Czech Republic (Jun 2015)

[12] Browne, G., Berry, E., Kapur, N., Hodges, S., Smyth, G., Watson, P., Wood, K.: SenseCam improves memory for recent events and quality of life in a patient with memory retrieval difficulties. Memory 19(7), S. 713–722 (Okt 2011)

[13] Byrne, D., Lavelle, B., Doherty, A.R., Jones, G.J., Smeaton, A.F.: Using bluetooth and GPS metadata to measure event similarity in SenseCam Images. In: Information Sciences 2007, S. 1454–1460. World Scientific Publishing (Jul 2007)

[14] Canny, J.: A Computational Approach to Edge Detection S. 20 (1987)

[15] Cao, Z., Hidalgo, G., Simon, T., Wei, S.E., Sheikh, Y.: OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. arXiv:1812.08008 [cs] (Dez 2018)

[16] Chang, S.F., Sikora, T., Purl, A.: Overview of the MPEG-7 standard. IEEE Transactions on Circuits and Systems for Video Technology 11(6), S. 688–695 (Jun 2001)

[17] Chen, C., Oakes, M., Tait, J.: Browsing personal images using episodic memory (time+ location). In: European Conference on Information Retrieval. S. 362–372. Springer (2006)

[18] Chen, T., Guestrin, C.: XGBoost: A Scalable Tree Boosting System. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16. S. 785–794. ACM Press, San Francisco, California, USA (2016)

[19] Colonna, L.: Legal and regulatory challenges to utilizing lifelogging technologies for the frail and sick. International Journal of Law and Information Technology 27(1), S. 50–74 (Mär 2019)

[20] Craik, F.I., Lockhart, R.S.: Levels of processing: A framework for memory research. Journal of Verbal Learning and Verbal Behavior 11(6), S. 671–684 (Dez 1972)

[21] Dalal, N., Triggs, B.: Histograms of Oriented Gradients for Human Detection. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). vol. 1, S. 886–893. IEEE, San Diego, CA, USA (2005)

[22] Dang-Nguyen, D.T., Piras, L., Giacinto, G., Boato, G., Natale, F.G.B.D.: Multimodal Retrieval with Diversification and Relevance Feedback for Tourist

Attraction Images. ACM Transactions on Multimedia Computing, Communications, and Applications 13(4), S. 1–24 (Aug 2017)

[23] Dang-Nguyen, D.T., Piras, L., Riegler, M., Boato, G., Zhou, L., Gurrin, C.: Overview of ImageCLEFlifelog 2017: Lifelog Retrieval and Summarization S. 23 (2017)

[24] Dang-Nguyen, D.T., Piras, L., Riegler, M., Zhou, L., Lux, M., Gurrin, C.: Overview of ImageCLEFlifelog 2018: Daily Living Understanding and Lifelog Moment Retrieval S. 19 (2018)

[25] Dang-Nguyen, D.T., Piras, L., Riegler, M., Zhou, L., Lux, M., Tran, M.T., Le, T.K., Ninh, V.T., Gurrin, C.: Overview of ImageCLEFlifelog 2019: Solve My Life Puzzle and Lifelog Moment Retrieval S. 17 (2019)

[26] Dang-Nguyen, D.T., Schoeffmann, K., Hurst, W.: LSC2018 Panel - Challenges of Lifelog Search And Access S. 2

[27] Davies, G.M., Thomson, D.M.: Memory in Context: Context in Memory. John Wiley & Sons (1988)

[28] Davies, N., Friday, A., Clinch, S., Sas, C., Langheinrich, M., Ward, G., Schmidt, A.: Security and Privacy Implications of Pervasive Memory Augmentation. IEEE Pervasive Computing 14(1), S. 44–53 (Jan 2015)

[29] Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A Large-Scale Hierarchical Image Database S. 8 (2009)

[30] Dodge, M., Kitchin, R.: 'Outlines of a World Coming into Existence': Pervasive Computing and the Ethics of Forgetting. Environment and Planning B: Planning and Design 34(3), S. 431–445 (Jun 2007)

[31] Dogariu, M., Ionescu, B.: Multimedia Lab @ ImageCLEF 2018 Lifelog Moment Retrieval Task S. 13 (2018)

[32] Doherty, A.R., Pauly-Takacs, K., Caprani, N., Moulin, C.J.A., O'Connor, N.E., Smeaton, A.F.: Experiences of Aiding Autobiographical Memory using the SenseCam S. 32 (2012)

[33] Doherty, A.R., Smeaton, A.F., Lee, K., Ellis, D.P.W.: Multimodal Segmentation of Lifelog Data S. 18 (2007)

[34] Eldridge, M., Sellen, A., Bekerian, D.: Memory Problems at Work: Their Range, Frequency and Severity S. 36 (1992)

[35] Eysenck, M.W.: Principles of Cognitive Psychology. Psychology Press (2001)

[36] Gemmell, J., Aris, A., Lueder, R.: Telling Stories with Mylifebits. In: 2005 IEEE International Conference on Multimedia and Expo. S. 1536–1539. IEEE, Amsterdam, The Netherlands (2005)

[37] Glogger: LifeGlogging cameras 1998, 2004, 2006 and 2013 (2013)

[38] Goddard, L., Pring, L., Felmingham, N.: The effects of cue modality on the quality of personal memories retrieved. Memory 13(1), S. 79–86 (Jan 2005)

[39] Grudin, J.: Desituating Action: Digital Representation of Context. Human–Computer Interaction 16(2-4), S. 269–286 (Dez 2001)

[40] Gurrin, C., Smeaton, A.F., Doherty, A.R.: LifeLogging: Personal Big Data. Foundations and Trends® in Information Retrieval 8(1), S. 1–125 (2014)

[41] Harvey, M., Langheinrich, M., Ward, G.: Remembering through lifelogging: A survey of human memory augmentation. Pervasive and Mobile Computing 27, S. 14–26 (Apr 2016)

[42] He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proceedings of the IEEE International Conference on Computer Vision. S. 2961–2969 (2017)

[43] He, K., Zhang, X., Ren, S., Sun, J.: Deep Residual Learning for Image Recognition. arXiv:1512.03385 [cs] (Dez 2015)

[44] Hearst, M.A.: TextTiling: A Quantitative Approach to Discourse Segmentation S. 10 (1993)

[45] Hodges, S., Williams, L., Berry, E., Izadi, S., Srinivasan, J., Butler, A., Smyth, G., Kapur, N., Wood, K.: SenseCam: A Retrospective Memory Aid. In: Dourish, P., Friday, A., Hutchison, D., Kanade, T., Kittler, J., Kleinberg, J.M., Mattern, F., Mitchell, J.C., Naor, M., Nierstrasz, O., Pandu Rangan, C., Steffen, B., Sudan, M., Terzopoulos, D., Tygar, D., Vardi, M.Y., Weikum, G. (eds.) UbiComp 2006: Ubiquitous Computing, vol. 4206, S. 177–193. Springer Berlin Heidelberg, Berlin, Heidelberg (2006)

[46] Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., Brox, T.: FlowNet 2.0: Evolution of Optical Flow Estimation with Deep Networks. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). S. 1647–1655. IEEE, Honolulu, HI (Jul 2017)

[47] Ionescu, B., Popescu, A., Lupu, M., Gînscă, A.L., Boteanu, B., Müller, H.: Div150Cred: A social image retrieval result diversification with user tagging credibility dataset. In: Proceedings of the 6th ACM Multimedia Systems Conference on - MMSys '15. S. 207–212. ACM Press, Portland, Oregon (2015)

[48] Ionescu, B., Radu, A.L., Menéndez, M., Müller, H., Popescu, A., Loni, B.: Div400: A social image retrieval result diversification dataset. In: Proceedings of the 5th ACM Multimedia Systems Conference on - MMSys '14. S. 29–34. ACM Press, Singapore, Singapore (2014)

[49] Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: Convolutional Architecture for Fast Feature Embedding. arXiv:1408.5093 [cs] (Jun 2014)

[50] Joulin, A., Grave, E., Bojanowski, P., Mikolov, T.: Bag of Tricks for Efficient Text Classification. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers. S. 427–431. Association for Computational Linguistics, Valencia, Spain (2017)

[51] Karpathy, A., Fei-Fei, L.: Deep Visual-Semantic Alignments for Generating Image Descriptions S. 10 (2015)

[52] Kavallieratou, E.: Retrieving Events in Life Logging S. 11 (2018)

[53] Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. Communications of the ACM 60(6), S. 84–90 (Mai 2017)

[54] Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J., Krasin, I., Pont-Tuset, J., Kamali, S., Popov, S., Malloci, M., Duerig, T., Ferrari, V.: The Open Images Dataset V4: Unified image classification, object detection, and visual relationship detection at scale. arXiv:1811.00982 [cs] (Nov 2018)

[55] Lamming, M., Brown, P., Carter, K., Eldridge, M., Flynn, M., Louie, G., Robinson, P., Sellen, A.: The design of a human memory prosthesis. The Computer Journal 37(3), S. 153–163 (1994)

[56] Le, N.K., Nguyen, D.H., Nguyen, V.T., Tran, M.T.: Lifelog Moment Retrieval with Advanced Semantic Extraction and Flexible Moment Visualization for Exploration S. 17 (2019)

[57] Lee, M.L., Dey, A.K.: Lifelogging memory appliance for people with episodic memory impairment. In: Proceedings of the 10th International Conference on Ubiquitous Computing - UbiComp '08. S. 44. ACM Press, Seoul, Korea (2008)

[58] Levy, O., Goldberg, Y.: Dependency-Based Word Embeddings. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). S. 302–308. Association for Computational Linguistics, Baltimore, Maryland (2014)

[59] Lin, T.Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C.L., Dollár, P.: Microsoft COCO: Common Objects in Context. arXiv:1405.0312 [cs] (Mai 2014)

[60] Lin, W.H., Hauptmann, A.: Structuring continuous video recordings of everyday life using time-constrained clustering. In: Chang, E.Y., Hanjalic, A., Sebe, N. (eds.) Electronic Imaging 2006. S. 60730D. San Jose, CA (Jan 2006)

[61] MacQueen, J.: Some methods for classification and analysis of multivariate observations. In: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability. vol. 1, S. 281–297. Oakland, CA, USA (1967)

[62] Marx, G.: Some Information Age Techno-Fallacies. Journal of Contingencies and Crisis Management 11(1), S. 25–31 (Mär 2003)

[63] Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed Representations of Words and Phrases and their Compositionality S. 9 (2013)

[64] del Molino, A.G., Mandal, B., Lin, J., Lim, J.H., Subbaraju, V., Chandrasekhar, V.: VC-I2R@ ImageCLEF2017: Ensemble of deep learned features for lifelog video summarization (2017)

[65] Naveh-Benjamin, M., Craik, F.I., Guez, J., Dori, H.: Effects of divided attention on encoding and retrieval processes in human memory: Further support for an asymmetry. Journal of Experimental Psychology: Learning, Memory, and Cognition 24(5), S. 1091 (1998)

[66] Owen, T.: What the tesla affir tells us about data journalism. TOW Center Blog, http://towcenter. org/what-the-tesla-affair-tells-us-about-data-journalism (2013)

[67] Patterson, G., Hays, J.: SUN attribute database: Discovering, annotating, and recognizing scene attributes. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition. S. 2751–2758. IEEE, Providence, RI (Jun 2012)

[68] Pauly-Takacs, K., Moulin, C.J.A., Estlin, E.J.: SenseCam as a rehabilitation tool in a child with anterograde amnesia. Memory 19(7), S. 705–712 (Okt 2011)

[69] Pearson, K.: The problem of the random walk. Nature 72(1867), S. 342 (1905)

[70] Peitgen, H.O., Jürgens, H., Saupe, D.: Chaos and Fractals: New Frontiers of Science (2012), oCLC: 890007077

[71] Pennington, J., Socher, R., Manning, C.: Glove: Global Vectors for Word Representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). S. 1532–1543. Association for Computational Linguistics, Doha, Qatar (2014)

[72] Redmon, J., Farhadi, A.: YOLO9000: Better, Faster, Stronger. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). S. 6517–6525. IEEE, Honolulu, HI (Jul 2017)

[73] Redmon, J., Farhadi, A.: YOLOv3: An Incremental Improvement. arXiv:1804.02767 [cs] (Apr 2018)

[74] Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence 39(6), S. 1137–1149 (Jun 2017)

[75] Robertson, S.: The Probabilistic Relevance Framework: BM25 and Beyond. Foundations and Trends® in Information Retrieval 3(4), S. 333–389 (2010)

[76] Roediger, H.L., Karpicke, J.D.: Test-Enhanced Learning: Taking Memory Tests Improves Long-Term Retention. Psychological Science 17(3), S. 249–255 (Mär 2006)

[77] Sellen, A., Fogg, A., Aitken, M., Hodges, S., Rother, C., Wood, K.: Do Life-Logging Technologies Support Memory for the Past? An Experimental Study Using SenseCam S. 10 (2007)

[78] Sellen, A.J., Whittaker, S.: Beyond total capture: A constructive critique of lifelogging. Communications of the ACM 53(5), S. 70 (Mai 2010)

[79] Shepard, R.N.: Recognition memory for words, sentences, and pictures. Journal of Verbal Learning and Verbal Behavior 6(1), S. 156–163 (Feb 1967)

[80] Simonyan, K., Zisserman, A.: Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv:1409.1556 [cs] (Sep 2014)

[81] Singhal, A.: Modern Information Retrieval: A Brief Overview S. 9 (2001)

[82] Slamecka, N.J.: Proactive inhibition of connected discourse. Journal of experimental psychology 62(3), S. 295 (1961)

[83] Smith, S.M., Vela, E.: Environmental context-dependent eyewitness recognition. Applied Cognitive Psychology 6(2), S. 125–139 (1992)

[84] Speer, R., Chin, J., Havasi, C.: ConceptNet 5.5: An Open Multilingual Graph of General Knowledge S. 8 (2017)

[85] Surprenant, A.M., Neath, I.: Principles of Memory. Psychology Press (2013)

[86] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the Inception Architecture for Computer Vision. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). S. 2818–2826. IEEE, Las Vegas, NV, USA (Jun 2016)

[87] Szegedy, C., Wei Liu, Yangqing Jia, Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). S. 1–9. IEEE, Boston, MA, USA (Jun 2015)

[88] Tang, T.H., Fu, M.H., Huang, H.H., Chen, K.T., Chen, H.H.: Visual Concept Selection with Textual Knowledge for Understanding Activities of Daily Living and Life Moment Retrieval S. 12 (2018)

[89] Taubert, S., Kahl, S., Kowerko, D., Eibl, M.: Automated Lifelog Moment Retrieval based on Image Segmentation and Similarity Scores S. 17 (2019)

[90] Taubert, S., Mauermann, M., Kahl, S., Kowerko, D., Eibl, M.: Species Prediction based on Environmental Variables using Machine Learning Techniques. In: CLEF (Working Notes) (2018)

[91] Terry, W.S.: Everyday forgetting: Data from a diary study. Psychological reports 62(1), S. 299–303 (1988)

[92] Thompson, C.P., Skowronski, J.J., Larsen, S.F., Betz, A.L.: Autobiographical Memory: Remembering What and Remembering When. Psychology Press (2013)

[93] Tran, M.T., Dinh-Duy, T., Truong, T.D., Vo-Ho, V.K., Luong, Q.A., Nguyen, V.T.: Lifelog Moment Retrieval with Visual Concept Fusion and Text-based Query Expansion (2018)

[94] Truong, T.D., Dinh-Duy, T., Nguyen, V.T., Tran, M.T.: Lifelogging Retrieval based on Semantic Concepts Fusion. In: Proceedings of the 2018 ACM Workshop on The Lifelog Search Challenge - LSC '18. S. 24–29. ACM Press, Yokohama, Japan (2018)

[95] Tulving, E., et al.: Episodic and semantic memory. Organization of memory 1, S. 381–403 (1972)

[96] Vinyals, O., Toshev, A., Bengio, S., Erhan, D.: Show and tell: A neural image caption generator. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). S. 3156–3164. IEEE, Boston, MA, USA (Jun 2015)

[97] Wang, Z., Simoncelli, E.P.: Reduced-reference image quality assessment using a wavelet-domain natural image statistic model. In: Rogowitz, B.E., Pappas, T.N., Daly, S.J. (eds.) Electronic Imaging 2005. S. 149. San Jose, CA (Mär 2005)

[98] Weber, K.: Mobile Devices, Virtual Presence, and Surveillance: Questions Concerning Epistemology and Some New Challenges for Privacy and Data Protection. SSRN Electronic Journal (2011)

[99] Woodberry, E., Browne, G., Hodges, S., Watson, P., Kapur, N., Woodberry, K.: The use of a wearable camera improves autobiographical memory in patients with Alzheimer's disease. Memory 23(3), S. 340–349 (Apr 2015)

[100] Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., Torralba, A.: Places: A 10 Million Image Database for Scene Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 40(6), S. 1452–1464 (Jun 2018)

[101] Zhou, B., Lapedriza, A., Xiao, J., Torralba, A., Oliva, A.: Learning Deep Features for Scene Recognition using Places Database S. 9 (2014)

[102] Zhou, L., Hinbarji, Z., Dang-Nguyen, D.T., Gurrin, C.: LIFER: An Interactive Lifelog Retrieval System. In: Proceedings of the 2018 ACM Workshop on The Lifelog Search Challenge - LSC '18. S. 9–14. ACM Press, Yokohama, Japan (2018)

[103] Zhou, L., Piras, L., Riegler, M., Boato, G., Dang-Nguyen, D.T., Gurrin, C.: Organizer Team at ImageCLEFlifelog 2017: Baseline Approaches for Lifelog Retrieval and Summarization S. 11 (2017)

[104] Zhou, L., Piras, L., Riegler, M., Lux, M., Dang-Nguyen, D.T., Gurrin, C.: An Interactive Lifelog Retrieval System for Activities of Daily Living Understanding S. 9 (2018)

Anhang

Nachstehend sind die Ergebnisse der Tokenisierung und die Labels aller Labelarten aufgelistet.

Ergebnisse der Tokenisierung

T Titeltoken
B Beschreibungstoken
N Narrativtoken
M Manuelle Token (nur bei Trainingsanfragen)

Trainingsanfragen

1. Anfrage
   - T icecream, sea
   - B beside, eating, icecream, sea
   - N clearly, cone, considered, cream, eating, hand, ice, must, occur, relevant, sea, show, together, visible, well
   - M eating, icecream, sea
2. Anfrage
   - T food, restaurant
   - B drinking, eating, food, restaurant
   - N away, cafe, coffee, dessert, dishes, drinking, eating, food, home, kinds, relevant, restaurant, wont
   - M drinking, eating, food, restaurant
3. Anfrage
   - T videos, watching
   - B devices, digital, using, video, watching
   - N computer, considered, desktop, devices, digital, example, laptop, location, machine, mobile, must, phone, relevant, tablet, tv, videos, watching
   - M video, watching
4. Anfrage
   - T bridge, photograph
   - B bridge, photo, taking
   - N also, bridge, considered, pedestrian, photo, relevant, showing, stopped, stopping, street, take, taken, walking, without
   - M bridge, photo, taking

5. Anfrage
   - T grocery, shopping
   - B food, grocery, shop, shopping
   - N bought, clearly, considered, grocery, must, relevant, shop, something
   - M food, grocery, shop, shopping
6. Anfrage
   - T guitar, playing
   - B guitar, man, playing, view
   - N considered, could, guitar, guitars, indoors, relevant, type, use
   - M guitar, playing
7. Anfrage
   - T cooking
   - B cooking, food
   - N cooking, food, places, relevant, shows
   - M cooking, food
8. Anfrage
   - T,B car, sales, showroom
   - N car, cars, considered, either, indoors, looking, outside, relevant, sales, salesman, show, showroom, sitting, table, times, visited, waiting
   - M car, sales, showroom
9. Anfrage
   - T public, transportation
   - B countries, public, taking, transportation
   - N car, considered, driving, must, place, public, relevant, take, transportation
   - M public, taking, transportation
10. Anfrage
    - T book, paper, reviewing
    - B book, paper, reading
    - N book, front, mark, must, paper, pen, relevant, sometimes, use, visible
    - M book, paper, reading

Testanfragen

1. Anfrage
   - T toyshop
   - B items, looking, toyshop
   - N board, clearly, considered, electronic, electronics, examined, games, kits, model, must, relevant, store, supermarket, toys, toyshop, trains, various
2. Anfrage
   - T driving, home
   - B driving, home, office
   - N driving, home, office, place, relevant, show

3. Anfrage
   - T food, fridge, seeking
   - B home, inside, looking, refrigerator
   - N considered, cooking, eating, food, home, inside, kitchen, looking, refrigerator, relevant
4. Anfrage
   - T football, watching
   - B either, football, tv, watching
   - N considered, content, either, football, indoors, must, relevant, television, tv, watching
5. Anfrage
   - T coffee, time
   - B cafe, coffee
   - N alone, another, cafe, coffee, considered, individual, must, relevant
6. Anfrage
   - T,B breakfast, home
   - N 500, 900, breakfast, home, must, time
7. Anfrage
   - T,B coffee, person, two
   - N blue, cloth, coffee, gender, one, person, relevant, shirt, two, wearing, white
8. Anfrage
   - T outside, smartphone, using
   - B outside, smartphone, standing, using, walking
   - N clearly, considered, location, must, outside, relevant, smartphone, using
9. Anfrage
   - T,B plaid, red, shirt, wearing
   - N day, life, plaid, red, relevant, shirt, user1, wearing
10. Anfrage
    - T china, meeting
    - B attending, china, meeting
    - N china, meeting, must, others, relevant, user1

Verfügbare Labels je Labelart

Nachstehend sind alle verfügbaren Labels je Labelart aufgelistet.

Aktivitäten (2)

transport, walking

Städte (7)

Prague, Shanghai, Istanbul, Berlin, Amsterdam, London, Dublin

Tageszeiten (4)

morning, afternoon, evening, night

Orte (16)

airport, bakery, bar, cafe, college, costa coffee, embassy, home, hotel, railway station, restaurant, shopping centre, solicitor, store, university, work

Attribute (97)

aged, asphalt, bathing, biking, boating, brick, camping, carpet, cleaning, climbing, cloth, clouds, cluttered space, cold, competing, concrete, conducting business, congregating, digging, dirt, dirty, diving, driving, dry, eating, enclosed area, exercise, far-away horizon, fencing, fire, flowers, foliage, gaming, glass, glossy, grass, hiking, horizontal components, ice, indoor lighting, leaves, man-made, matte, medical activity, metal, moist, natural, natural light, no horizon, ocean, open area, paper, pavement, plastic, playing, praying, railing, railroad, reading, research, rock, rugged scene, running water, rusty, sand, scary, semi-enclosed area, shingles, shopping, shrubbery, smoke, snow, socializing, soothing, spectating, sports, sterile, still water, stressful, studying, sunny, swimming, symmetrical, tiles, touring, training, transporting, trees, using tools, vegetation, vertical components, vinyl, waiting in line, warm, wire, wood, working

Gegebene Konzepte (76)

airplane, apple, backpack, banana, baseball_bat, baseball_glove, bear, bed, bench, bicycle, bird, boat, book, bottle, bowl, broccoli, bus, cake, car, carrot, cat, cell_phone, chair, clock, couch, cow, cup, dining_table, dog, donut, elephant, fire_hydrant, fork, frisbee, giraffe, handbag, horse, hot_dog, keyboard, kite, knife, laptop, microwave, motorcycle, mouse, orange, oven, parking_meter, person, pizza, potted_plant, refrigerator, remote, sandwich, scissors, sheep, sink, skateboard, spoon, sports_ball, stop_sign, suitcase, surfboard, teddy_bear, tennis_racket, tie, toilet, toothbrush, traffic_light, train, truck, tv, umbrella, vase, wine_glass, zebra

Detectron Konzepte (79)

airplane, apple, backpack, banana, baseball bat, baseball glove, bear, bed, bench, bicycle, bird, boat, book, bottle, bowl, broccoli, bus, cake, car, carrot, cat, cell phone, chair, clock, couch, cow, cup, dining table, dog, donut, elephant, fire hydrant, fork, frisbee, giraffe, hair drier, handbag, horse, hot dog, keyboard, kite, knife, laptop, microwave, motorcycle, mouse, orange, oven, parking meter, person, pizza, potted plant, refrigerator, remote, sandwich, scissors, sheep, sink, skateboard, skis, spoon, sports ball, stop sign, suitcase, surfboard, teddy bear, tennis racket, tie, toaster, toilet, toothbrush, traffic light, train, truck, tv, umbrella, vase, wine glass, zebra

Yolo Konzepte (72)

aeroplane, apple, backpack, banana, baseball bat, bear, bed, bench, bicycle, bird, boat, book, bottle, bowl, broccoli, bus, cake, car, carrot, cat, cell phone, chair, clock, cow, cup, diningtable, dog, donut, elephant, fire hydrant, fork, frisbee, handbag, horse, hot dog, keyboard, knife, laptop, microwave, motorbike, mouse, orange, oven, parking meter, person, pizza, pottedplant, refrigerator, remote, sandwich, scissors, sink, skateboard, sofa, spoon, sports ball, stop sign, suitcase, surfboard, teddy bear, tennis racket, tie, toilet, toothbrush, traffic light, train, truck, tvmonitor, umbrella, vase, wine glass, zebra

Open Images Konzepte (53)

Animal, Bed, Bicycle wheel, Billiard table, Bird, Building, Cake, Car, Carnivore, Clothing, Coffee, Coffee cup, Computer keyboard, Computer monitor, Dog, Door, Drink, Flag, Food, Footwear, Furniture, Home appliance, Human face, Jeans, Kitchenware, Land vehicle, Laptop, Mammal, Man, Mirror, Musical instrument, Office building, Person, Picture frame, Plant, Sculpture, Sports equipment, Stairs, Table, Tableware, Tea, Telephone, Television, Tool, Tree, Trousers, Van, Vehicle, Vehicle registration plate, Weapon, Wheel, Window, Woman

Kategorien (360)

airfield, airplane_cabin, airport_terminal, alcove, alley, amphitheater, amusement_arcade, amusement_park, apartment_building/outdoor, aquarium, aqueduct, arcade, arch, archaelogical_excavation, archive, arena/hockey, arena/performance, arena/rodeo, army_base, art_gallery, art_school, art_studio, artists_loft, assembly_line, athletic_field/outdoor, atrium/public, attic, auditorium, auto_factory, auto_showroom, badlands, bakery/shop, balcony/exterior, balcony/interior, ball_pit, ballroom, bamboo_forest, bank_vault, banquet_hall, bar, barn, barndoor, baseball_field, basement, basketball_court/indoor, bathroom, bazaar/indoor, bazaar/outdoor, beach, beach_house, beauty_salon, bedchamber, bedroom, beer_garden, beer_hall, berth, biology_laboratory, boardwalk, boat_deck, boathouse, bookstore, booth/indoor, botanical_garden, bow_window/indoor, bowling_alley, boxing_ring, bridge, building_facade, bullring, burial_chamber, bus_interior, bus_station/indoor, butchers_shop, butte, cafeteria, campsite, campus, canal/natural, canal/urban, candy_store, canyon, car_interior, carrousel, castle, catacomb, cemetery, chalet, chemistry_lab, childs_room, church/indoor, church/outdoor, classroom, clean_room, cliff, closet, clothing_store, coast, cockpit, coffee_shop, computer_room, conference_center, conference_room, construction_site, corn_field, corral, corridor, cottage, courthouse, courtyard, creek, crevasse, crosswalk, dam, delicatessen, department_store, desert/sand, desert/vegetation, desert_road, diner/outdoor, dining_hall, dining_room, discotheque, doorway/outdoor, dorm_room, downtown, dressing_room, driveway, drugstore, elevator/door, elevator_lobby, elevator_shaft, embassy, engine_room, entrance_hall, escalator/indoor,

excavation, fabric_store, farm, fastfood_restaurant, field/cultivated, field/wild, field_road, fire_escape, fire_station, fishpond, flea_market/indoor, florist_shop/indoor, food_court, football_field, forest/broadleaf, forest_path, forest_road, formal_garden, fountain, galley, garage/indoor, garage/outdoor, gas_station, gazebo/exterior, general_store/indoor, general_store/outdoor, gift_shop, glacier, golf_course, greenhouse/indoor, greenhouse/outdoor, grotto, gymnasium/indoor, hangar/indoor, hangar/outdoor, harbor, hardware_store, hayfield, heliport, highway, home_office, home_theater, hospital, hospital_room, hot_spring, hotel/outdoor, hotel_room, house, hunting_lodge/outdoor, ice_cream_parlor, ice_floe, ice_shelf, ice_skating_rink/indoor, ice_skating_rink/outdoor, iceberg, igloo, industrial_area, inn/outdoor, islet, jacuzzi/indoor, jail_cell, japanese_garden, jewelry_shop, junkyard, kasbah, kennel/outdoor, kindergarden_classroom, kitchen, lagoon, lake/natural, landfill, landing_deck, laundromat, lawn, lecture_room, legislative_chamber, library/indoor, library/outdoor, lighthouse, living_room, loading_dock, lobby, lock_chamber, locker_room, mansion, manufactured_home, market/indoor, market/outdoor, marsh, martial_arts_gym, mausoleum, medina, mezzanine, moat/water, mosque/outdoor, motel, mountain, mountain_path, mountain_snowy, movie_theater/indoor, museum/indoor, museum/outdoor, music_studio, natural_history_museum, nursery, nursing_home, oast_house, ocean, office, office_building, office_cubicles, oilrig, operating_room, orchard, orchestra_pit, pagoda, palace, pantry, park, parking_garage/indoor, parking_garage/outdoor, parking_lot, pasture, patio, pavilion, pet_shop, pharmacy, phone_booth, physics_laboratory, picnic_area, pier, pizzeria, playground, playroom, plaza, pond, porch, promenade, pub/indoor, racecourse, raceway, raft, railroad_track, rainforest, reception, recreation_room, repair_shop, residential_neighborhood, restaurant, restaurant_kitchen, restaurant_patio, river, rock_arch, roof_garden, rope_bridge, ruin, runway, sandbox, sauna, science_museum, server_room, shed, shoe_shop, shopfront, shopping_mall/indoor, shower, ski_resort, ski_slope, sky, skyscraper, slum, snowfield, soccer_field, stable, stadium/baseball, stadium/football, stadium/soccer, stage/indoor, stage/outdoor, staircase, storage_room, street, subway_station/platform, supermarket, sushi_bar, swimming_hole, swimming_pool/indoor, swimming_pool/outdoor, synagogue/outdoor, television_room, television_studio, temple/asia, throne_room, ticket_booth, topiary_garden, tower, toyshop, train_interior, train_station/platform, tree_farm, tree_house, trench, tundra, underwater/ocean_deep, utility_room, valley, vegetable_garden, veterinarians_office, viaduct, village, vineyard, volcano, volleyball_court/outdoor, waiting_room, water_park, water_tower, waterfall, wave, wet_bar, wheat_field, wind_farm, windmill, yard, youth_hostel, zen_garden

ImageNet Konzepte (426)

Band Aid, CD player, Chesapeake Bay retriever, Chihuahua, Crock Pot, Dungeness crab, Granny Smith, Kerry blue terrier, Loafer, Old English sheepdog, Petri dish, Weimaraner, Windsor tie, abacus, abaya, academic gown, accordion, acorn squash, aircraft carrier, airliner, airship, ambulance, ant, armadillo, ashcan, backpack, badger, bagel, bakery, balloon, banana, banjo, bannister, barbell, barber chair,

barbershop, barn, barrel, barrow, bathtub, beacon, beagle, beaker, bearskin, beer bottle, beer glass, bell cote, bell pepper, bicycle-built-for-two, binder, binoculars, birdhouse, boathouse, bobsled, bolo tie, bonnet, book jacket, bookcase, bookshop, bow, brass, breakwater, breastplate, bucket, buckeye, buckle, bullet train, bullet-proof vest, burrito, butcher shop, cab, caldron, can opener, candle, cannon, canoe, car mirror, cardigan, carousel, cash machine, cassette, cassette player, cauliflower, cello, cellular telephone, chain, chain mail, cheeseburger, chest, chime, china cabinet, chocolate sauce, cinema, cleaver, cocktail shaker, coffee mug, coffeepot, coil, combination lock, computer keyboard, conch, confectionery, consomme, container ship, convertible, corkscrew, corn, cornet, cowboy boot, cowboy hat, cradle, crash helmet, crayfish, crib, crossword puzzle, crutch, cucumber, cuirass, cup, custard apple, dam, desk, desktop computer, diaper, digital clock, digital watch, dining table, dishwasher, disk brake, dome, dough, drum, drumstick, dugong, dumbbell, eggnog, electric fan, electric guitar, entertainment center, envelope, espresso, espresso maker, feather boa, file, fire engine, fire screen, fireboat, flagpole, flute, football helmet, forklift, fountain, fountain pen, four-poster, freight car, frying pan, fur coat, gar, garbage truck, gas pump, gasmask, gazelle, geyser, go-kart, goblet, golden retriever, golf ball, golfcart, gondola, gong, grand piano, greenhouse, grocery store, groenendael, guacamole, hair slide, hair spray, hand blower, hand-held computer, handkerchief, hard disc, harmonica, harp, hermit crab, home theater, honeycomb, hot pot, hotdog, hourglass, iPod, ice cream, iron, isopod, jackfruit, jean, jersey, jigsaw puzzle, jinrikisha, joystick, kimono, knee pad, knot, lab coat, ladle, lakeside, lampshade, laptop, lens cap, library, lighter, limousine, lotion, loudspeaker, loupe, lumbermill, mailbag, mailbox, marimba, mashed potato, mask, matchstick, maze, measuring cup, meat loaf, medicine chest, menu, microphone, microwave, milk can, minibus, minivan, mitten, mixing bowl, mobile home, modem, monitor, moped, mortar, mortarboard, mosquito net, motor scooter, mountain bike, mouse, mousetrap, moving van, mushroom, nematode, nipple, notebook, obelisk, ocarina, odometer, oil filter, orange, organ, oscilloscope, oxygen mask, packet, paddlewheel, palace, paper towel, parallel bars, park bench, parking meter, passenger car, patio, pay-phone, pedestal, photocopier, pick, picket fence, pickup, pier, pill bottle, pillow, pineapple, pinwheel, plane, planetarium, plastic bag, plate, plate rack, plunger, pole, police van, poncho, pool table, pop bottle, pot, „potters wheel“, power drill, pretzel, printer, prison, projectile, projector, puffer, punching bag, quill, quilt, racer, radiator, rain barrel, recreational vehicle, red wine, reel, refrigerator, remote control, restaurant, rifle, rocking chair, rotisserie, rubber eraser, rugby ball, rule, safe, safety pin, saltshaker, sandal, scale, school bus, scoreboard, screen, screw, screwdriver, seashore, seat belt, sewing machine, shoe shop, shoji, shopping cart, shower cap, shower curtain, ski mask, sleeping bag, slide rule, sliding door, slot, snowplow, soap dispenser, sock, solar dish, soup bowl, space bar, space heater, spaghetti squash, spatula, speedboat, spider web, spotlight, steam locomotive, steel arch bridge, steel drum, stethoscope, stole, stopwatch, stove, strainer, street sign, streetcar, stretcher, studio couch, submarine, suit, sulphur-crested cockatoo, sundial, sunglass, sunglasses, suspension bridge, sweatshirt, swing, switch, syringe, table lamp, tarantula, teapot, television,

tennis ball, thatch, theater curtain, thimble, tick, toaster, tobacco shop, toilet seat, toilet tissue, torch, toyshop, traffic light, trailer truck, tray, trench coat, trimaran, triumphal arch, tub, turnstile, umbrella, unicycle, upright, vacuum, vault, velvet, vending machine, viaduct, violin, waffle iron, wall clock, wallet, wardrobe, warplane, washbasin, washer, water bottle, water jug, water tower, web site, wig, window screen, window shade, wine bottle, wing, wok, wool, worm fence, zebra, zucchini

TECHNISCHE UNIVERSITÄT
CHEMNITZ

Studentenservice – Zentrales Prüfungsamt

Selbstständigkeitserklärung

Name: Taubert Vorname: Stefan geb. am: 30.07.1994 Matr.-Nr.: 369897	**Bitte beachten:** 1. Bitte binden Sie dieses Blatt am Ende Ihrer Arbeit ein.

Selbstständigkeitserklärung*

Ich erkläre gegenüber der Technischen Universität Chemnitz, dass ich die vorliegende Masterarbeit selbstständig und ohne Benutzung anderer als der angegebenen Quellen und Hilfsmittel angefertigt habe.

Die vorliegende Arbeit ist frei von Plagiaten. Alle Ausführungen, die wörtlich oder inhaltlich aus anderen Schriften entnommen sind, habe ich als solche kenntlich gemacht.

Diese Arbeit wurde in gleicher oder ähnlicher Form noch nicht als Prüfungsleistung eingereicht und ist auch noch nicht veröffentlicht.

Datum: 10.10.2019 Unterschrift: .....

* Statement of Authorship

I hereby certify to the Technische Universität Chemnitz that this thesis is all my own work and uses no external material other than that acknowledged in the text.

This work contains no plagiarism and all sentences or passages directly quoted from other people's work or including content derived from such work have been specifically credited to the authors and sources.

This paper has neither been submitted in the same or a similar form to any other examiner nor for the award of any other degree, nor has it previously been published.