INSTITUT FÜR INFORMATIK
Datenbanken und Informationssysteme

Universitätsstr. 1 D–40225 Düsseldorf

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Analyse von Online-Partizipationsverfahren: Automatisierte Verschlagwortung von Textbeiträgen

Markus Brenneis

Masterarbeit

Beginn der Arbeit:	19. Mai 2018
Abgabe der Arbeit:	19. November 2018
Gutachter:	Prof. Dr. Stefan Conrad
	Prof. Dr. Stefan Harmeling

Erklärung

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

Düsseldorf, den 19. November 2018

Markus Brenneis

Zusammenfassung

Diese Arbeit beschäftigt sich mit der Analyse von Online-Partizipationsverfahren, bei denen sich Bürger mit dem Verfassen von Textbeiträgen an politischen Prozessen beteiligen können. Diese Beteiligungsverfahren laufen in der Regel unter einem Oberthema, z. B. Verbesserungen für den Radverkehr, wobei die einzelnen Beiträge genaueren Kategorien zugewiesen werden können.

Im Rahmen dieser Arbeit wurden Machine-Learning-Verfahren entwickelt, um für die Benutzer der Partizipationsplattform und für die Auswerter eine automatische Kategorisierung von Beiträgen zu ermöglichen. Diese Aufgaben sind im Rahmen der Auswertung eines Partizipationsverfahrens mit mehreren tausend Beiträgen sehr zeitaufwändig und auch fehleranfällig. Ferner weisen Benutzer der Plattform in einigen Fällen ihren eigenen Beiträgen auch falsche Kategorien zu, sodass hier ein automatischer Kategorisierungsvorschlag hilfreich ist.

Das Multi-Class-Problem der Kategorisierung von Textbeiträgen wurde dabei mithilfe von Features basierend auf Charakter-N-Grammen und logistischer Regression so gut gelöst, dass die Performance mit einer menschlichen Baseline vergleichbar ist. Dabei ist es auch möglich, einen Klassifikator zu verwenden, der auf einem bereits beendeten Partizipationsverfahren mit derselben Labelmenge trainiert worden ist.

Außerdem wurde das Multi-Label-Problem der Verschlagwortung untersucht, bei der Textbeiträgen mehrere Schlagwörter zugeordnet werden können. Die untersuchten Algorithmen liefern gute Ergebnisse, insbesondere wenn auf eine bereits vorher erfolgte Kategorisierung zurückgegriffen werden kann. Zum Teil werden auch Schlagwörter vorgeschlagen, die sinnvoll, aber von den Benutzern ursprünglich nicht gewählt worden sind.

Schließlich wurde auch untersucht, ob automatisiert die Themen eines Partizipationsverfahren aus seinen Beiträgen extrahiert werden können. Da die Themen der Beiträge aber oft nah beieinander liegen, konnten mit etablierten unüberwachten Verfahren keine guten Ergebnisse erzielt werden.

Inhaltsverzeichnis

1 Einleitung und Motivation **1**

1.1 Online-Partizipationsverfahren 1

1.2 Begriffsdefinitionen 2

1.3 Ziele der Arbeit 2

2 Datensätze **5**

2.1 Raddialoge 6

2.2 Bürgerhaushalt 9

2.3 Mängelmelder Braunschweig 10

2.4 Nahverkehrsplan Ulm 10

2.5 Leitbild Bad Godesberg 12

2.6 Bürgerbudget Wuppertal 12

3 Single-Label-Klassifikation **15**

3.1 Baseline-Modelle 15

3.2 Verbesserungen des Baseline-Klassifikators 17

3.3 Evaluation anderer klassischer Machine-Learning-Verfahren 22

3.4 Evaluation Graph-basierter Verfahren 26

3.5 Evaluation künstlicher neuronaler Netze 29

3.6 Analyse falsch kategorisierter Textbeiträge 37

3.7 Weitere Optimierungen 43

3.8 Finale Evaluation 51

3.9 Praxisrelevante Experimente 52

3.10 Zusammenfassung 57

4 Multi-Label-Klassifikation **59**

4.1 Verwendete Evaluationsmaße 59

4.2 Evaluation von Multi-Label-Verfahren 60

4.3 Evaluation von Hierarchie-basierten Verfahren 67

4.4 Zusammenfassung 69

5 Kategorienbildung durch automatisierte Themenextraktion **71**

5.1 Verwendete Topic-Modeling-Algorithmen 71

5.2 Evaluation der Modelle 72
5.3 Zusammenfassende Beurteilung 78

6 Zusammenfassung und Future Work **79**

A Anhang **81**
A.1 Datensatzeigenschaften 81
A.2 Gridsearch-Parameter 86
A.3 Lernkurven 86
A.4 Konfusionsmatrizen für hierarchische Einfachverschlagwortung 88
A.5 Evaluation der Topic-Modeling-Algorithmen 90

Literaturverzeichnis **97**

Abbildungsverzeichnis **101**

Tabellenverzeichnis **103**

1 Einleitung und Motivation

Zunächst wird erklärt, was Online-Partizipationsverfahren sind, welchen Zweck sie haben und inwiefern Automatisierung bei solchen Partizipationsverfahren sinnvoll ist. Anschließend werden wichtige Begriffe im Zusammenhang mit dieser Arbeit definiert und die Ziele der Arbeit dargestellt.

1.1 Online-Partizipationsverfahren

Als Online-Partizipationsverfahren werden Online-Plattformen bezeichnet, auf denen in Form von Diskussionen und Abstimmungen an Entscheidungsprozessen mitgewirkt werden kann. (Liebeck et al., 2017) In dieser Arbeit liegt der Fokus auf politischen Online-Partizipationsverfahren, die von Städten und Gemeinden durchgeführt werden, um ihre Bürger in politische Entscheidungsprozesse einzubeziehen. Mögliche Themen solcher Partizipationsverfahren sind Vorschläge für die Haushaltsplanung, das Melden von Mängeln oder Vorschläge zur Verbesserung des Radverkehrs.

Je nach Verfahren können Bürger selbst Vorschläge oder Beiträge verfassen oder es gibt von der Stadtverwaltung vorgegebene Vorschläge. Die Vorschläge sind von den Verfassern oft vorgegebenen Kategorien zuzuordnen, um einerseits die Suche nach bestehenden Beiträgen und andererseits die Auswertung zu vereinfachen. Andere Benutzer der Plattform haben oft die Möglichkeit, bestehende Vorschläge zu kommentieren oder zu bewerten, was dazu beiträgt, doppelte Vorschläge zu vermeiden und bei der abschließenden Auswertung wissen zu können, welche Vorschläge von den meisten Bürgern als wichtig angesehen werden.

Die Verfahren sind in der Regel top-down organisiert, gehen also von den Städten aus und werden von diesen ausgewertet. Bei Verfahren mit mehreren tausend Kommentaren ist die Auswertung mit einem erheblichen Kosten- und Zeitaufwand verbunden. Insbesondere bei Verfahren zur Haushaltsplanung ist aber eine zeitnahe Auswertung der Beiträge erforderlich, sodass eine zumindest zum Teil automatisierte Auswertung hilfreich ist.

Es gibt verschiedene Aufgaben, bei denen eine maschinelle Unterstützung im Rahmen eines Online-Partizipationsverfahrens nützlich ist. Im Rahmen der Auswertung könnten automatisch Themen identifiziert werden, wenn es während des Verfahrens nicht schon eine Kategorisierung der Beiträge gab oder diese unzureichend ist. Sofern bereits eine Kategorisierung vorliegt, könnten falsch kategorisierte Beiträge erkannt und richtig zugeordnet werden. Während das Verfahren noch läuft, wäre es möglich, dem Benutzer beim Schreiben eines Beitrags passende Kategorien vorzuschlagen, um eine mögliche falsche Kategorisierung zu vermeiden. Ferner könnten Nutzer auf schon vorhandene Beiträge mit demselben oder einem ähnlichen Thema hingewiesen werden, um unerwünschte Duplikate zu vermeiden.

1.2 Begriffsdefinitionen

Als *Kategorie* wird ein Begriff bezeichnet, der das Hauptthema eines Textbeitrages beschreibt. Jeder Textbeitrag ist genau einer Kategorie zugeordnet. Ein Beispiel für eine Menge von Kategorien ist {Ampeln, Beschilderung, Parken}; der Satz *Es gibt zu wenige Anwohnerparkplätze und zu viele Falschparker.* würde zur Kategorie *Parken* gehören. Das Zuordnen einer Kategorie zu einem Textbeitrag ist ein *Multiclass-Problem*.

Ein *Schlagwort* (oder *Tag*) ist ebenfalls ein Begriff, der das Thema eines Textbeitrags beschreibt, allerdings können einem Beitrag beliebig viele Schlagwörter zugeordnet sein. Eine Schlagwortmenge könnte z. B. {Ampel defekt, unklare Beschilderung, verdreckte Schilder, Falschparker, Parkplatzangebot} sein; der oben genannte Satz erhielte die Schlagwörter *Falschparker* und *Parkplatzangebot*. Die Zuordnung von Schlagwörtern zu einem Beitrag ist ein *Multilabel-Problem*.

Mit *Label* oder *Klasse* werden allgemein die Ausgabe oder erwartete Ausgabe (Ground Truth) eines Klassifikators in Zusammenhang mit supervised Learning bezeichnet, was je nach Zusammenhang eine Kategorie oder ein Schlagwort ist.

Ein *Thema* ist der gedankliche Mittelpunkt eines Textes. Ein Textbeitrag kann mehrere Themen umfassen und mehrere Beiträge dieselben Themen ansprechen. Als Themen werden in dieser Arbeit die Ausgaben von Topic-Modeling-Verfahren bezeichnet. Das automatische Finden von Themen erfolgt mit Hilfe von unüberwachtem Lernen. Im Gegensatz zu Kategorien und Schlagwörtern gibt es bei Themen keine Ground-Truth-Labels.

Als *Verfahren* werden Algorithmen des Maschinellen Lernens bezeichnet. (Online-)Partizipationsverfahren werden immer explizit als *Partizipationsverfahren* bezeichnet.

1.3 Ziele der Arbeit

Im Rahmen dieser Arbeit sollen verschiedene Verfahren entwickelt werden, die einerseits Nutzern bei der Kategorisierung ihrer Textbeiträge und andererseits bei der Auswertung eines abgeschlossenen Partizipationsverfahrens helfen sollen. Folgende Anwendungsmöglichkeiten stehen im Fokus:

- Vorschlagen von passenden Kategorien beim Schreiben eines Textbeitrags
- Hierarchische und nicht-hierarchische Verschlagwortung bereits kategorisierter Textbeiträge
- Automatische Extraktion von Themen eines Partizipationsverfahrens aus dessen Textbeiträgen

Als Grundlage für überwachte Verfahren sollen hierbei sowohl „Live-Daten“ aus einem laufenden Partizipationsverfahren als auch gelabelte Daten abgeschlossener Partizipationsverfahren dienen können. Die Performance des entwickelten Machine-Learning-Verfahrens sollte vergleichbar mit der eines Menschen sein und darüber hinaus so generisch

sein, dass es auf möglichst vielen verschiedenen Online-Partizipationsdatensätzen angewendet werden kann und gute Ergebnisse liefert.

Im nächsten Abschnitt werden zunächst die Datensätze vorgestellt, die für die Evaluation der Verfahren zur Verfügung stehen. Anschließend wird die Entwicklung eines Verfahrens zur Kategorisierung von Textbeiträgen beschrieben und evaluiert. Im folgenden Kapitel wird auf die automatische Verschlagwortung von Beiträgen eingegangen und im vorletzten Kapitel wird die Kategorienbildung durch automatische Themenextraktion untersucht. Abschließend werden die Ergebnisse zusammengefasst und mögliche weitere Forschungsrichtungen aufgezeigt.

Partizipationsverfahren	#Beiträge	#Kategorien	#Schlagwörter
Raddialoge	3167	8	32
Bonn	2330	8	32
Köln	375	8	24
Moers	462	8	27
Bürgerhaushalt	4049		
Bonn	1405	7 / 3	
Bonn 2011	1015	7 / 3	68
Bonn 2015	335	7 / 3	
Bonn 2017	55	7 / 3	
Köln	2644	3	
Köln 2012	594	5 / 3	
Köln 2013	592	2	
Köln 2015	631	2	
Köln 2016	827	2	
Mängelmelder Braunschweig	2576	12	
Nahverkehrsplan Ulm	1013	8	
Bad Godesberg	556	16	
Bürgerbudget Wuppertal	261		12

Tabelle 1: Zusammenfassung der Charakteristika der Datensätze; bei Datensätzen mit zwei verschiedenen Kategorienmengen sind die Größen beider Mengen angegeben

2 Datensätze

Die im Rahmen dieser Arbeit untersuchten Verfahren wurden auf verschiedenen Datensätzen evaluiert. Die Datensätze stammen größtenteils von Webseiten zu politischen Online-Partizipationsverfahren, bei denen von Benutzern Beiträge zu bestimmten Themen verfasst werden können. Die Benutzer haben dabei meistens ihre Beiträge genau einer Kategorie aus einer vorgegebenen Menge an Kategorien zuordnen müssen. Sofern nicht anders angegeben, wurden die Datensätze mithilfe existierender oder selbst geschriebener Webcrawler[1] zusammengestellt.

Zu Beginn der Arbeit wurden alle Datensätze in einen Trainings- (80 %) und Test-Datensatz (20 %) aufgeteilt. Der Test-Datensatz wurde nur für die abschließende Evaluation der Modelle benutzt, insbesondere nicht zum Bestimmen von Hyperparametern.

Im Folgenden werden alle verwendeten Datensätze beschrieben und es wird auf ihre jeweiligen Besonderheiten, z. B. im Hinblick auf Klassen-Imbalancen, eingegangen. Tabelle 1 gibt einen Überblick über die Eigenschaften der Datensätze.

[1] `https://github.com/Liebeck/OnlineParticipationDatasets`

2.1 Raddialoge

Die Städte Bonn[2], Köln[3] und Moers[4] haben im Jahr 2017 jeweils *Raddialoge* angeboten, bei denen Bürger den Städten Hinweise geben konnten, wie das Radfahren in der jeweiligen Stadt bzw. dem jeweiligen Bezirk verbessert werden kann. Die Benutzer haben ihre Beiträge dabei zu Kategorien wie *Radwegqualität* oder *Radverkehrsführung* zugeordnet. Während des Raddialogs und der Auswertung wurden zum Teil die Kategorien der Beiträge von den Moderatoren geändert, wenn die Kategorie unpassend für die Auswertung erschien. Dabei wurde folgendes Verfahren verwendet:

1. Sofern die vom Benutzer gewählte Kategorie sinnvoll ist, wurde diese behalten.
2. Wenn im Beitrag mehrere Themen angesprochen werden, dann bestimmt das Thema die Kategorie, das im Titel genannt wird oder im Beitrag im Vordergrund steht.
3. Wenn der Beitrag einen Vorschlag und eine Problembeschreibung enthält, gewinnt die Kategorie, die zum Vorschlag gehört. (Beispiel: Wird eine bessere Beschilderung zur Verbesserung der Radverkehrsführung vorgeschlagen, ist die Kategorie *Beschilderung*.)

Insgesamt wurde bei knapp 75 % der Beiträge die vom Benutzer gewählte Kategorie beibehalten.

Die Beiträge sind ungleichmäßig auf die acht Kategorien verteilt. In allen Raddialogen sind mindestens 40 % der Beiträge der Kategorie *Radverkehrsführung* zugeordnet. Die vier kleinsten Kategorien sind *Beleuchtung*, *Beschilderung*, *Fahrradparken* und *Sonstiges*. Abbildung 1 zeigt die Verteilung der Beiträge auf die Kategorien.

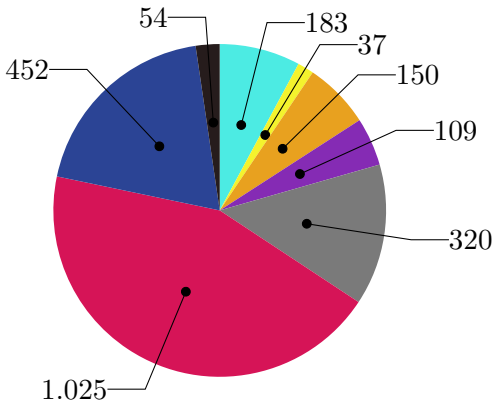
In der Auswertung der Raddialoge wurden die Beiträge außerdem mit einem oder mehreren Schlagwörtern, wie *Beleuchtung fehlt* oder *zu geringe Breite*, versehen. Obwohl jedes Schlagwort thematisch zu einer Kategorie gehört, ist es möglich, dass demselben Beitrag Schlagwörter verschiedener Kategorien zugeordnet werden. Tabelle 33 im Anhang gibt einen Überblick über die in den Raddialogen vergebenen Schlagwörter. Das am häufigsten vergebene Schlagwort ist *Vorschlag für neuen Radweg*, das knapp 20 % aller Beiträge tragen.

Von der Möglichkeit, einem Beitrag mehr als ein Schlagwort zuzuordnen, wird nur in 17 % der Fälle Gebrauch gemacht; der größte Teil der Beiträge hat nur genau ein Schlagwort. Wie in Abbildung 2 zu sehen ist, gibt es auch einen Beitrag ohne Schlagwörter, welcher später bei der Betrachtung der automatischen Verschlagwortung ignoriert worden ist. Durchschnittlich haben die Beiträge knapp 1,2 Schlagwörter.

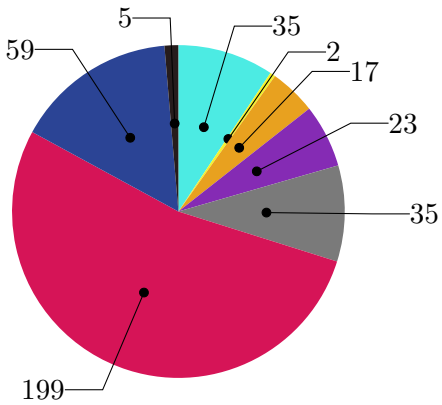
In Abbildung 3 sind beispielhaft Beiträge aus dem Bonner Raddialog mit den von den Benutzern und Moderatoren zugewiesenen Kategorien und Schlagwörtern dargestellt.

Die Beiträge konnten von anderen Nutzern kommentiert werden. Diese Kommentare wurden im Rahmen dieser Arbeit nicht betrachtet, da die zu entwickelnden Verfahren

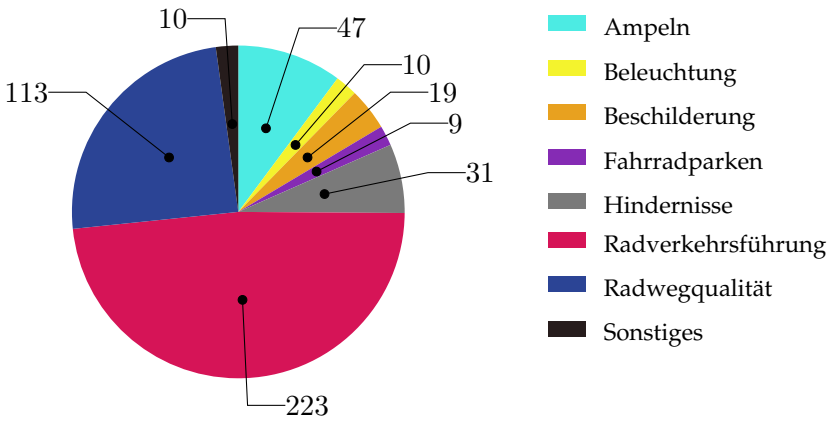
[2] Bonner Rad-Dialog, `https://raddialog.bonn.de/`
[3] Ehrenfelder Raddialog, `https://www.raddialog-ehrenfeld.koeln/`
[4] Raddialog der Stadt Moers, `https://raddialog.moers.de/`



(a) Raddialog Bonn, insgesamt 2330 Beiträge



(b) Raddialog Köln, insgesamt 375 Beiträge



(c) Raddialog Moers, insgesamt 462 Beiträge

Abbildung 1: Verteilung der Beiträge der Raddialoge auf die vorgegebenen Kategorien

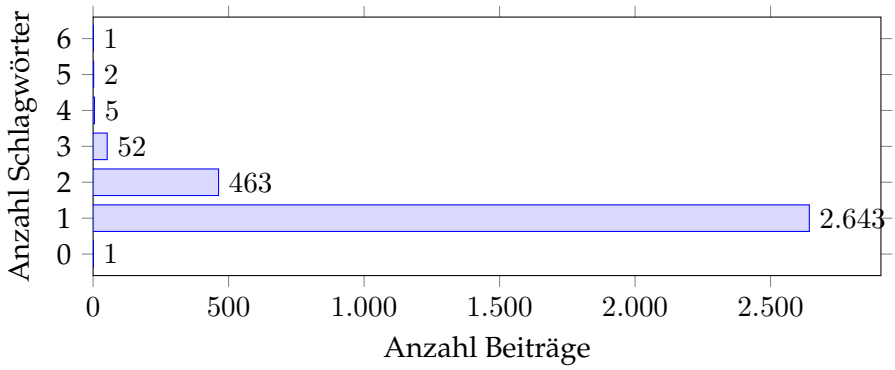


Abbildung 2: Übersicht über die Anzahl der Schlagwörter, die pro Raddialog-Beitrag vergeben wurden

Beschilderung ändern
Hier gibt es Schild "Radfahrer nehmt Rücksicht auf Fußgänger". Bitte dieses Schild ändern in "Radfahrer und Fußgänger nehmt bitte gegenseitig Rücksicht aufeinander". Denn es muss z. B. auch nicht sein, dass Fußgängergruppen den Weg in voller Breite blockieren und wenn man dann klingelt, weil man vorbeifahren möchte, angepöbelt wird mit Verweis auf dieses Schild.

`https://www.raddialog.bonn.de/dialoge/bonner-rad-dialog/beschilderung-aendern`, abgerufen am 19. Juli 2018

(a) von Benutzer gewählte Kategorie: *Beschilderung*, von Moderation gewählte Kategorie: *Radverkehrsführung*, Schlagwörter: *regelwidriges Verhalten*

Zu viele fussgänger zwischen den vielen Berufspendlern und fehlende Beleuchtung sowie Markierung
diese Strecke wird von sehr vielen Berufspendlern genutzt, die von Siegburg und Sankt Augustin nach Bonn müssen. Es ist im Winter morgens stockdunkel und dabei ist das ein beliebter Hundehalterweg, da er am Feldrand liegt. Das ist super gefährlich für alle Beteiligten. Hier wäre es wünschenswert, dass es einen reinen Radweg gibt, sowie eine Beleuchtung.

`https://www.raddialog.bonn.de/dialoge/bonner-rad-dialog/zu-viele-fussgaenger-zwischen-den-vielen-berufspendlern-und-fehlende`, abgerufen am 19. Juli 2018

(b) von Benutzer gewählte Kategorie: *Sonstiges*, von Moderation gewählte Kategorie: *Radverkehrsführung*, Schlagwörter: *Vorschlag für neuen Radweg*, *Beleuchtung fehlt*

Rad-/Fußgängerweg zu schmal
Der Rad-/Fußweg stadteinwärts von der Ampel bis zur (schlecht einsehbaren) Abbiegung in den Bröltalbahnweg ist viel zu schmal und derzeit von der Baustelle Stadttor Beuel häufig durch Baustellenfahrzeuge blockiert, sodass man auf die Straße ausweichen muss. Der Radweg darf eigentlich nur stadteinwärts befahren werden, wird aber häufig auch stadtauswärts befahren. Insgesamt müsste der Rad-/Fußgängerweg verbreitert werden und in beide Richtungen befahrbar sein dürfen.

`https://www.raddialog.bonn.de/dialoge/bonner-rad-dialog/rad-fussgaengerweg-zu-schmal`, abgerufen am 8. Oktober 2018

(c) von Benutzer gewählte Kategorie: *Radverkehrsführung*, von Moderation gewählte Kategorie: *Radverkehrsführung*, Schlagwörter: *Radweg beidseitig befahren*, *zu geringe Breite*

Abbildung 3: Beispiele für Beiträge zum Raddialog Bonn

auf möglichst vielen Arten von Online-Partizipationsverfahren anwendbar sein sollen, also auch solchen, bei denen es keine Kommentarmöglichkeit gibt. Weiterhin sollen Kategorievorschläge schon dann zur Verfügung stehen, wenn der Beitrag formuliert wird, aber zu diesem Zeitpunkt gibt es noch keine Kommentare, die als Eingabe verwendet werden könnten.

Da allen Raddialogen im Wesentlichen dieselbe Menge an Schlagwörtern und Kategorien zugrunde liegt, können die drei einzelnen Datensätze auch zu einem großen Datensatz zusammengefasst werden, der im Folgenden *Raddialoge* genannt wird.

2.2 Bürgerhaushalt

Von den Städten Bonn[5] und Köln[6] wurden in den vergangenen Jahren Partizipationsverfahren zum Thema *Bürgerhaushalt* durchgeführt, in dessen Rahmen die Bürger der Städte Vorschläge zum Haushalt der Stadt machen konnten. Dabei sind pro Jahr und Stadt zwischen 55 und 1015 Vorschläge eingegangen.

In allen Jahren wurden die Beiträge von den Bürgen allgemeinen Kategorien wie *Sparvorschlag* und *Ausgabevorschlag* zugeordnet, in manchen Jahren zusätzlich auch genaueren Kategorien wie *Bildung und Soziales*, sodass es zu diesen Beiträgen zwei Kategorien gibt; von diesen wurden die genaueren Kategorien verwendet, wenn sie vorhanden waren. Außerdem war es möglich, Beiträgen keine Kategorie zuzuordnen; diese Beiträge wurden im Rahmen dieser Arbeit nicht verwendet. Wie sich die Beiträge auf die unterschiedlichen Kategorien verteilen, ist in Tabelle 2 und Abbildung 21 im Anhang zu sehen.

Beim Bürgerhaushalt Bonn 2011 sind die Textbeiträge ferner mit Schlagwörtern versehen. Es gibt insgesamt 68 verschiedene Schlagwörter wie *ÖPNV* und *Klimaschutz* und die Schlagwörter *Sparvorschlag* und *Einnahmeerhöhung*, die direkt mit den Kategorien zusammenhängen. Eine Übersicht über die verwendeten Schlagwörter ist im Anhang in Tabelle 34.

Die meisten Beiträge haben genau drei Schlagwörter bekommen. Die Häufigkeit der Schlagwortvergabe ist in Abbildung 22 im Anhang dargestellt. Durchschnittlich hat ein Textbeitrag 2,5 Schlagwörter.

Wegen der sehr geringen Anzahl von 55 Beiträgen beim Bürgerhaushalt Bonn 2017 wurde dieser Datensatz nie alleine evaluiert, sondern nur in Kombination mit den anderen Bonner Datensätzen, die dieselben Labels verwenden; dieser kombinierte Datensatz wird in dieser Arbeit mit *Bürgerhaushalt Bonn* bezeichnet.

[5] Bürgerhaushalt Bonn 2011, `http://bonn-packts-an-2011.de/www.bonn-packts-an.de/dito/forumc0d2.html`, offline am 02.08.2018; Bonn packt's an – Bürgerdialog zum Haushalt 2015/2016, `https://www.bonn-macht-mit.de/node/84`; Bonn packt's an – Bürgerdialog zum Haushalt 2017/2018, `https://www.bonn-macht-mit.de/node/866`

[6] Bürgerhaushalt Köln 2012, `https://buergerhaushalt.stadt-koeln.de/2012/`; Kölner Bürgerhaushalt 2013/2014, `https://buergerhaushalt.stadt-koeln.de/2013/`; Kölner Bürgerhaushalt 2015, `https://buergerhaushalt.stadt-koeln.de/2015/`; Kölner Bürgerhaushalt 2016, `https://buergerhaushalt.stadt-koeln.de/2016/`

Kategorie	Bonn 2011	Bonn 2015	Bonn 2017
Bildung und Soziales	113	38	5
Finanzen und Beteiligungen	122	46	6
Freizeit und Sport	65	27	5
Kultur und Veranstaltungen	130	40	7
Sonstiges	46	23	4
Verkehr-Bauen-Umwelt	285	75	14
Verwaltung und Bürgerservice	243	39	3
–	11	47	11
Ausgabevorschlag	n/a	19	9
Einnahmevorschlag	263	76	9
Sonstige	119	n/a	n/a
Sparvorschlag	633	182	22
–	n/a	58	15
$\sum$	1015	335	55

Tabelle 2: Verteilung der Beiträge der Bonner Bürgerhaushalte auf die Kategorien

2.3 Mängelmelder Braunschweig

Beim Mängelmelder Braunschweig[7] können verschiedene Mängel in der Stadt Braunschweig, die zu einer der 12 vorgegebenen Kategorien passen, gemeldet werden. Dabei ist es möglich, den Beiträgen auch Bilder hinzuzufügen.

Zum Zeitpunkt des Crawlens gab es beim Mängelmelder 2576 Beiträge, wovon jeweils etwa ein Viertel den beiden größten Kategorien *Straßen-, Radweg- und Gehwegschäden* und *Straßenbeleuchtung/Laterne defekt* zugeordnet sind. Abbildung 4 gibt eine Übersicht über die Größe aller Kategorien.

2.4 Nahverkehrsplan Ulm

Ende 2016 hat die Stadt Ulm den Bürgerdialog Bus & Straßenbahn[8] durchgeführt, in dessen Rahmen Bürger vorgegebene Vorschläge zur Umgestaltung des Nahverkehrsnetzes in Ulm kommentieren konnten. Dazu gab es 1117 Kommentare, die nach Beendigung des Verfahrens von der Firma Zebralog, die diesen Datensatz freundlicherweise zur Verfügung gestellt hat, 8 Kategorien zugeordnet worden sind.

Die größte Kategorie ist *Bedienungshäufigkeit* mit 263 Kommentaren, die kleinste *Haltestellen* mit 22 Kommentaren. 104 Kommentare sind Moderationskommentare, welche bereits während des Dialogs als solche gekennzeichnet worden sind und deshalb im Rahmen dieser Arbeit nicht betrachtet wurden. Die vollständige Verteilung der Beiträge auf die Kategorien ist in Abbildung 5 zu sehen.

Anders als bei den anderen betrachteten Onlinepartizipationsverfahren wurden beim Nahverkehrsplan Ulm 17 Diskussionsthemen vorgegeben, zu denen die Benutzer Kom-

[7] Mängelmelder, `https://www.mitreden.braunschweig.de/node/1358`
[8] Zukunftsstadt Ulm, `https://www.zukunftsstadt-ulm.de/node/2986`

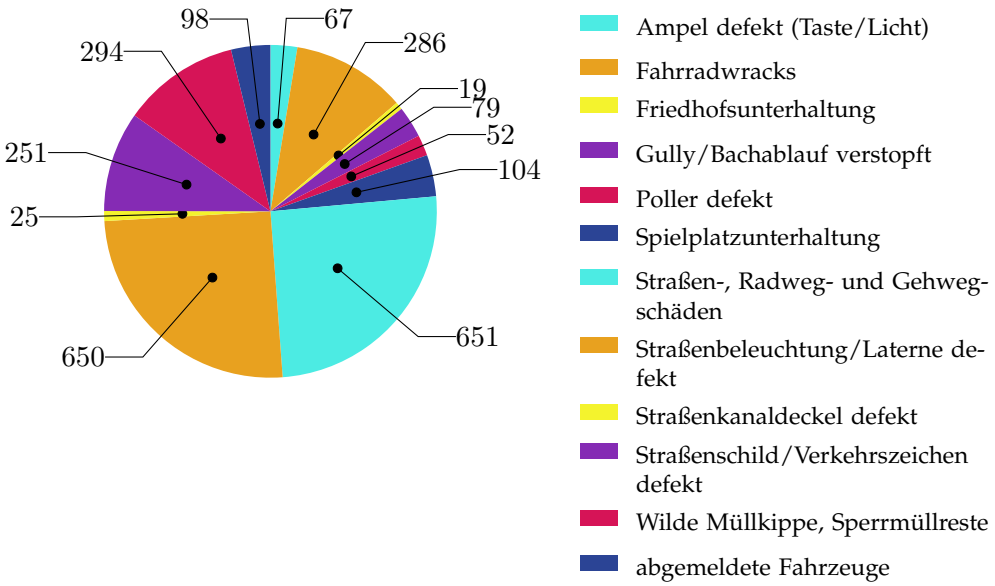


Abbildung 4: Verteilung der Beiträge des Mängelmelder Braunschweig auf die Kategorien

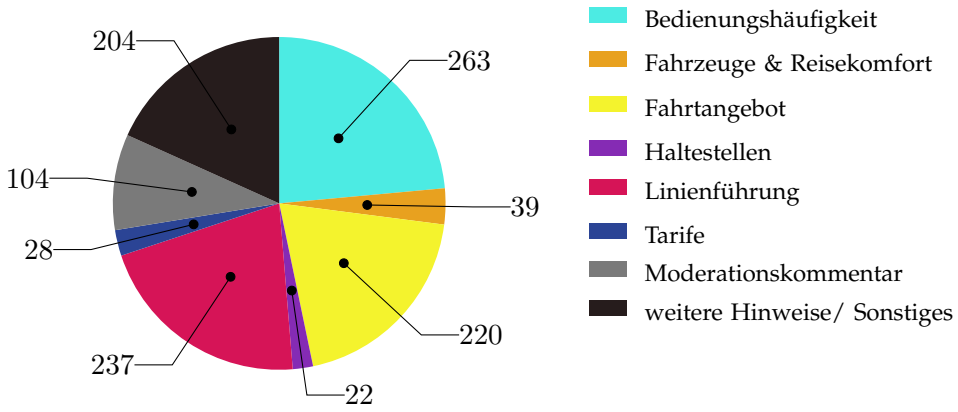


Abbildung 5: Verteilung der Beiträge zum Nahverkehrsplan Ulm auf die Kategorien

mentare abgeben konnten und andere Kommentare kommentieren konnten. Insgesamt gab es 498 Top-Level-Kommentare. Da auch die übrigen 619 Kommentare auf Kommentare meistens einen eindeutigen thematischen Bezug haben, wurden alle Kommentare bis auf die Moderationskommentare für die Evaluation der getesteten Verfahren verwendet.

2.5 Leitbild Bad Godesberg

Im Jahr 2018 hat die Stadt Bonn eine Onlinebeteiligung zum *Leitbild Bad Godesberg*[9] durchgeführt, bei der Vorschläge zur zukünftigen Gestaltung des Stadtbezirks Bad Godesberg gemacht werden konnten. Insgesamt gab es 556 Beiträge und 16 Kategorien wie *Kulturangebote verbessern* und *Sauberkeit verbessern*. Die größte Kategorie ist *Sonstiges* mit knapp ein Viertel der gemachten Vorschläge. 9 Beiträge sind keiner Kategorie zugeordnet. Alle Kategorien sind im Anhang in Abbildung 23 dargestellt.

2.6 Bürgerbudget Wuppertal

Im Jahr 2017 hat die Stadt Wuppertal ein Budget von 150.000 € zur Realisierung von Ideen von Bürgern, die dem Wohl der Bürger dienen, zur Verfügung gestellt. Im Rahmen der Onlinebeteiligung *Bürgerbudget Wuppertal*[10] wurden 261 Beiträge abgegeben, denen Schlagwörtern aus einer vorgegebenen Menge von 12 Schlagwörtern zugeordnet und die mit weiteren Informationen wie eine Kostenschätzung versehen werden konnten. Rund die Hälfte der Beiträge haben die Schlagwörter *Freizeit und Kultur* und *Gemeinschaft*. 23 Beiträge haben kein zugewiesenes Schlagwort, 49 haben genau ein Schlagwort und 4 Beiträge haben alle 12 Schlagwörter.

In Tabelle 3 ist dargestellt, wie oft welches Schlagwort zugewiesen wurde. Abbildung 6 zeigt, wie viele Schlagwörter pro Beitrag vergeben wurden. Ein Beitrag hat im Durchschnitt 4,2 Schlagwörter.

[9] Onlinebeteiligung zum Leitbild Bad Godesberg, `https://www.bonn-macht-mit.de/dialoge/onlinebeteiligung-zum-leitbild-bad-godesberg`

[10] Bürgerbudget Wuppertal, `https://buergerbudget.wuppertal.de/`

Schlagwort	#Beiträge
Arbeit	18
Bildung	61
Einkommen	10
Engagement	63
Freizeit und Kultur	150
Gemeinschaft	130
Gesundheit	68
Infrastruktur (Verkehr und Nahversorgung)	78
Sicherheit	34
Umwelt	71
Wohnen	35
Zufriedenheit	125

Tabelle 3: Häufigkeiten der beim Bürgerbudget Wuppertal vergebenen Schlagwörter

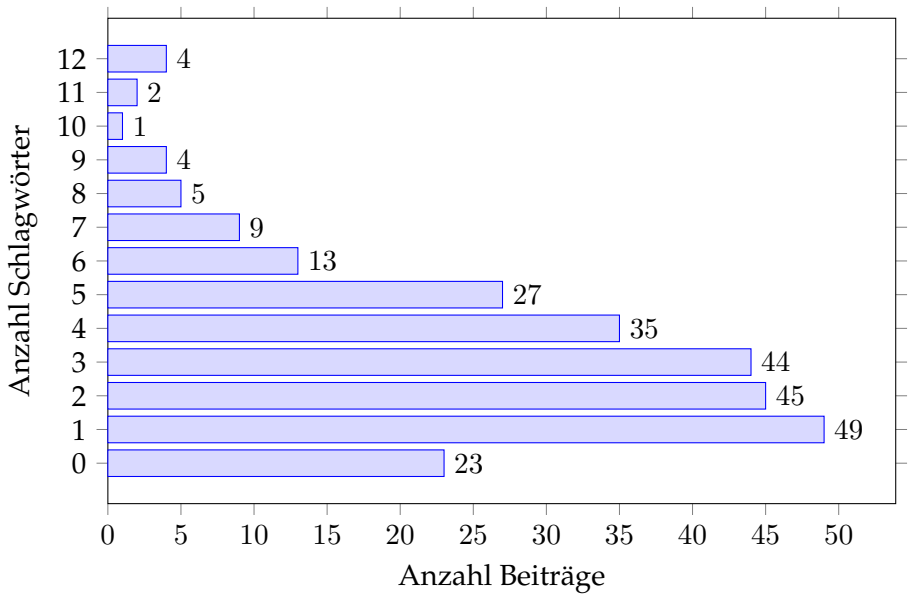


Abbildung 6: Übersicht über die Anzahl der Schlagwörter, die pro Beitrag zum Bürgerbudget Wuppertal vergeben wurden

3 Single-Label-Klassifikation

Dieses Kapitel beschäftigt sich mit der Entwicklung und Evaluation von supervised Verfahren, die Textbeiträgen eine Kategorie zuweisen, also ein Multiclass-Problem lösen. Ausgehend von einer Baseline werden die Modelle weiterentwickelt, auf den verschiedenen Datensätzen evaluiert und Stärken und Schwächen diskutiert. Wo verfügbar wird außerdem mit einer menschlichen Baseline verglichen.

Zur Beurteilung der Klassifikatoren wird das Macro-F_1-Maß verwendet. Die Accuracy ist ungeeignet, da es in den Datensätzen Klassenimbalancen gibt, sodass die Accuracy einen Klassifikator, der meistens nur die dominierende Kategorie vorhersagt, zu gut bewerten würde. Weil in der Anwendung sowohl die richtige Zuordnung zu einer Kategorie (Precision) als auch das Wiedererkennen von Kategorien (Recall) wichtig ist, ist das F_1-Maß als harmonisches Mittel von Precision und Recall eine gute Wahl. Durch die Macro-Durchschnittsbildung werden dabei alle Klassen als gleich wichtig angesehen.

Die angegebenen Performance-Werte wurden, wenn nicht anders angegeben, mithilfe von zehnfacher stratifizierter Kreuzvalidierung auf den Trainingsdaten ermittelt. Am Ende des Kapitels werden die Performance auf den Testdatensätzen angegeben sowie praxisrelevante Experimente dargestellt.

3.1 Baseline-Modelle

Zu Beginn wurde ein einfacher Klassifikator als Baseline evaluiert, mit dessen Performance komplexere Klassifikatoren verglichen werden können. Dazu wurde ein k-Nearest-Neighbor-Klassifikator (k-NN) mit $k = 5$ und der Cosinus-Metrik gewählt, der als Eingabe die konkatenierten Titel und Inhalte der Textbeiträge in einer Bag-of-Words-Darstellung erhält.

Dieser Klassifikator erreicht auf dem Raddialog-Datensatz einen Macro-F_1-Score von 31 %. Aus der Konfusionsmatrix in Abbildung 7 wird deutlich, dass der Klassifikator die meisten Beiträge der dominanten Kategorie *Radverkehrsführung* zuordnet. Außerdem wird kein Beitrag der Kategorie *Sonstiges* korrekt klassifiziert.

Das Problem der Bevorzugung dominanter Kategorien wird auch auf den anderen Datensätzen deutlich. So werden beim Bürgerhaushalt Köln 2013 viele Beiträge falsch der dominanten Kategorie *Sparvorschlag* zugeordnet, beim Bürgerhaushalt Köln 2015 der dominanten Kategorie *Ausgabevorschlag*.

Bei den Raddialogen stehen neben den finalen Kategorien auch die ursprünglichen Kategoriezuordnungen der Plattform-Benutzer zur Verfügung. Um herauszufinden, wie gut ein Benutzer, der nicht auf die korrekte Zuweisung der Kategorien geschult worden ist, die Kategorisierungsaufgabe löst, wurde ein „Klassifikator“ *Human* getestet, der den Beiträgen die vom Benutzer gewählte Kategorie zuordnet. Dieser Klassifikator kommt auf einen F_1-Wert von 73 %.

Der häufigste Fehler von Human ist das Verwechseln der Kategorien *Beschilderung* und *Radverkehrsführung*: Wie Abbildung 8 entnommen werden kann, werden etwa ein Viertel der Beiträge, die die Kategorie *Beschilderung* erhalten sollten, irrtümlicherweise *Radver-*

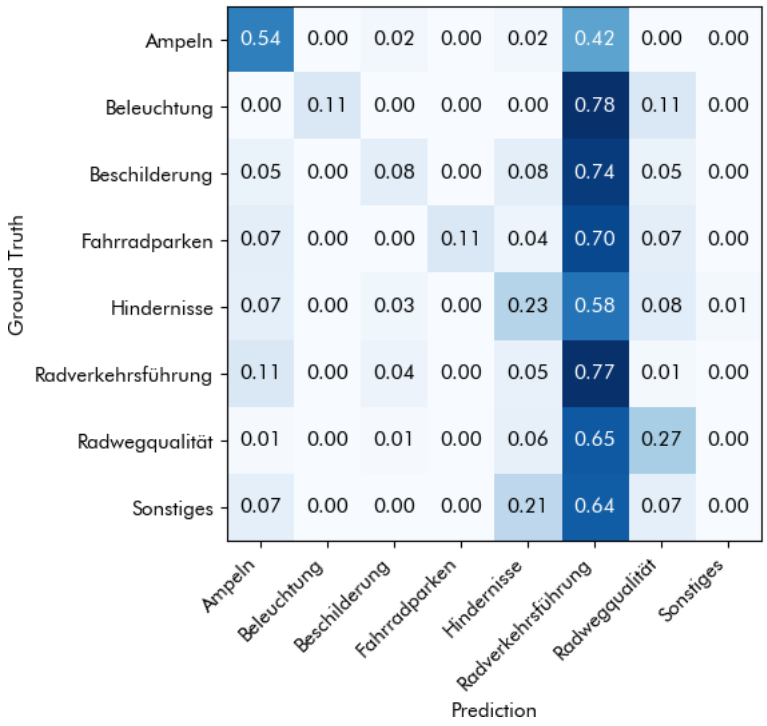


Abbildung 7: Konfusionsmatrix für das k-NN-Baseline-Modell auf den Raddialogen

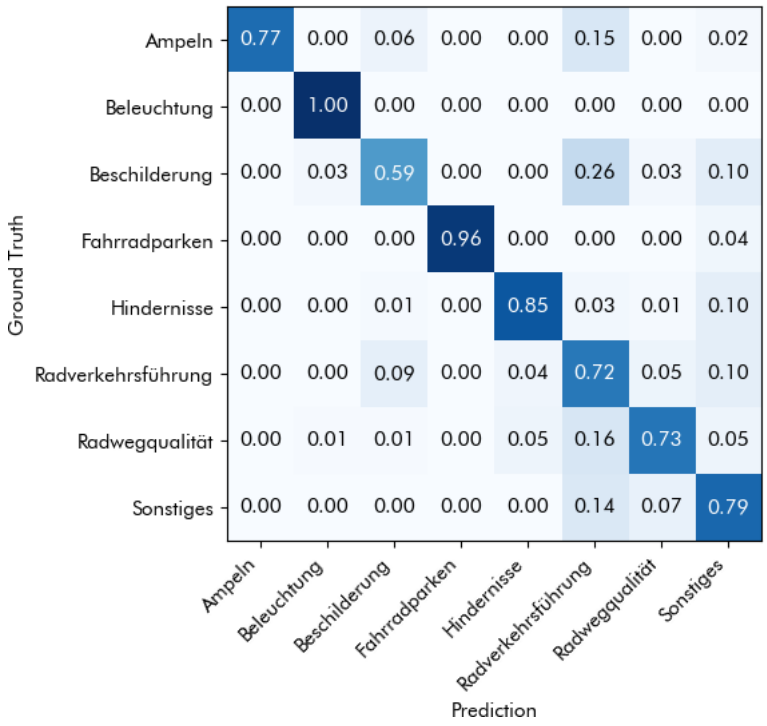


Abbildung 8: Konfusionsmatrix für den *Human*-Klassifikator auf den Raddialogen

Datensatz	k-NN	Human
Raddialoge	31	73
Bonn	28	74
Köln	27	66
Moers	24	69
Bürgerhaushalt		
Bonn	20	
Bonn 2011	21	
Bonn 2015	14	
Köln		
Köln 2012	27	
Köln 2013	51	
Köln 2015	51	
Köln 2016	48	
Mängelmelder Braunschweig	57	
Nahverkehrsplan Ulm	22	
Bad Godesberg	9	

Tabelle 4: Macro-F_1-Scores der Baselines auf den verschiedenen Datensätzen

kehrsführung zugeordnet. Viele der falsch zugeordneten Beiträge schlagen eine verbesserte oder klarere Verkehrsführung durch bessere Beschilderung oder Markierungen vor, was gemäß der in Abschnitt 2.1 beschriebenen Kategorisierungsregeln einer Zuordnung zu *Beschilderung* bedeutet. Beiträge der Kategorie *Beleuchtung* werden immer richtig zugeordnet.

In Tabelle 4 sind die F_1-Werte für alle Datensätze zusammengefasst.

3.2 Verbesserungen des Baseline-Klassifikators

Im Folgenden werden einzelne Verbesserungsmöglichkeiten für das Preprocessing bei der k-NN-Baseline vorgestellt und evaluiert. Dabei wird zunächst dargestellt, welche Auswirkungen die jeweiligen Methoden für sich genommen auf die Klassifikationsperformance haben, um anschließend die Kombination verschiedener Methoden zu evaluieren.

3.2.1 Stoppwortentfernung

Stoppwörter sind Wörter, die in Texten einer gegebenen Sprache häufig vorkommen, aber isoliert keine wirkliche Bedeutung tragen, z. B. im Deutschen „haben“ oder „in“. Das Entfernen von Stoppwörtern verringert die Anzahl der Dimensionen im Featurespace und kann bessere Klassifikationsergebnisse zur Folge haben.

Wenn die deutsche Stoppwortliste von NLTK (Bird et al., 2009) verwendet wird, kann der F_1-Score der k-NN-Baseline um bis zu 21 Prozentpunkte verbessert werden. Nur beim

Bürgerhaushalt Köln 2013 ist der F_1-Score ein Prozentpunkt schlechter, im Durchschnitt liegt die Verbesserung bei ca. 12 Punkten.

Neben dem Zurückgreifen auf fertige Stoppwortlisten gibt es auch die Möglichkeit, eine vom Datensatz abhängige Stoppwortliste zu erstellen, die diejenigen Wörter enthält, die in $p\,\%$ der Beiträge vorkommen. Beispielsweise enthalten knapp 9 % der Beiträge zum Raddialoge Bonn das Wort „Bonn“, was im Allgemeinen kein Stoppwort ist, aber im Zusammenhang mit diesem Raddialog unaussagekräftig ist.

Mit $p = 4$ verbessern sich die F_1-Werte bei allen Datensätzen um bis zu 26 Prozentpunkte, durchschnittlich um 16 Punkte.

Fazit: Insgesamt liefert die datensatzabhängige Stoppwortliste bessere Ergebnisse. Der einzige Datensatz, bei dem dies nicht der Fall ist, ist der Bürgerhaushalt Bonn 2015, bei dem die NLTK-Stoppwortliste einen um vier Prozentpunkte besseren Wert liefert.

3.2.2 Minimale Document Frequency

Neben dem Entfernen häufig auftretender Wörter ist auch das Entfernen selten auftretender Wörter sinnvoll. Einerseits ist es bei seltenen Wörtern unwahrscheinlich, dass sie auch in anderen Beiträgen vorkommen, sodass sie für die Klassifikation eher irrelevant sind, und andererseits ist es wahrscheinlich, dass es sich um Tippfehler handelt, welche ebenfalls eher nicht in anderen Beiträgen vorkommen. Getestet wurden minimale absolute Dokumentfrequenzen von 2 und 3 und relative Frequenzen von 1 % und 2 %.

Fazit: Wenn Wörter, die in weniger als drei Beiträgen im Trainingsdatensatz vorkommen, aus diesem entfernt werden, verbessern sich auf den meisten Datensätzen die Ergebnisse gegenüber der Baseline um bis zu 4 Prozentpunkte. Allerdings werden die F_1-Scores beim Bürgerhaushalt Köln 2013 und beim Raddialog Köln um einen bzw. fünf Prozentpunkte schlechter. Mit einer durchschnittlichen Verbesserung um etwa einen Prozentpunkt ist die Verbesserung durch das Entfernen seltener Wörter eher gering.

3.2.3 tf-idf-Gewichtung

Beim Bag-of-Words-Modell der Baseline gehen alle Wörter eines Beitrags gleichwertig in die Vektorrepräsentation eines Beitrags ein. Sinnvoll wäre es, wenn Wörter, die in vielen Dokumenten vorkommen, einen geringeren Einfluss hätten. Denn oft vorkommende Wörter sind in der Regel so allgemein, dass sie zur Unterscheidung von Kategorien nicht geeignet sind, wohingegen seltenere Wörter eher Kategorie-spezifisch sind. Die tf-idf-Gewichtung setzt diese Idee um.

Fazit: Mit der tf-idf-Gewichtung lassen sich die Baseline-Ergebnisse um einen (Bürgerhaushalt Bonn 2015) bis 26 (Raddialoge Bonn) Prozentpunkte verbessern, im Durchschnitt um etwa 14 Punkte.

3.2.4 Stemming

Bisher werden unterschiedliche Wortformen vom Klassifikator als gänzlich unterschiedliche Wörter behandelt. Allerdings ist es für das Thema eines Textbeitrags in der Regel egal, ob von *Radfahren*, *Radfahrern* oder *radfahren* gesprochen wird, sodass eine Reduzierung auf den Wortstamm (hier *radfahr*), sogenanntes Stemming, sinnvoll ist, um die Dimensionalität des Featureraums zu verringern. Gerade im Deutschen mit seinen vielen Flexionsendungen kann man eine große Dimensionsreduktion erwarten.

Fazit: Tatsächlich verschlechtert Stemming auf den meisten getesteten Datensätzen aber die Performance des k-NN-Klassifikators, im Durchschnitt um knapp einen Prozentpunkt. Nur bei den Bürgerhaushalten Bonn und Köln 2013 gibt es eine leichte Verbesserung um 2 Prozentpunkte, beim Raddialog Köln dagegen eine Verschlechterung um 5 Prozentpunkte.

3.2.5 Lemmatisierung

Unter Lemmatisierung versteht man das Zurückführen eines Wortes auf seine Grundform, also z. B. von *Radfahrern* auf *Radfahrer*. Grundsätzlich wären dieselben Vorteile wie beim Stemming zu erwarten. Für die Lemmatisierung wurde IWNLP (Liebeck und Conrad, 2015) verwendet, welches das deutsche Wiktionary als Grundlage verwendet. Sofern zu einem Wort mehrere Lemmata gefunden werden, wird das erste verwendet.

Fazit: Mit der Lemmatisierung verbessern sich die F_1-Scores im Durchschnitt um knapp einen Prozentpunkt, allerdings sind die Ergebnisse stark vom Datensatz abhängig. Beim Bürgerhaushalt Bonn verbessert sich der Wert um 4,5 Prozentpunkte, beim Raddialog Köln verschlechtert er sich jedoch – wie auch schon beim Stemming – um fast 5 Punkte.

3.2.6 Zerlegung von Komposita

Neben vielen Flexionsendungen gibt es im Deutschen auch viele Komposita wie *Fußgängergruppe* oder *Ampelschaltung*, die aus verschiedenen Wörtern zusammengesetzt sind. Da ein Beitrag, der das Wort *Ampelschaltung* enthält, wahrscheinlich dasselbe Thema hat wie ein Beitrag, der die Wörter *Schaltung* und *Ampel* enthält, kann das Zerlegen von Komposita ein sinnvoller Preprocessingschritt sein. Für die Kompositazerlegung wurde jWordSplitter[11] verwendet.

Fazit: Mit Zerlegung verbessern sich auf allen Datensätzen die Ergebnisse um bis zu 16 Prozentpunkte (Mängelmelder Braunschweig), im Durchschnitt um 6,4 Prozentpunkte.

3.2.7 Rechtschreibkorrektur

In Online-Foren gibt es in den Beiträgen oft Rechtschreibfehler, die die Performance eines Bag-of-Words-Modells verschlechtern können, da „dasselbe" Wort in unterschiedlichen Schreibweisen im Vokabular aufgenommen wird. Die Textbeiträge enthalten zwar nur

[11] `http://danielnaber.de/jwordsplitter/`

selten sprachliche Fehler, aber es gibt teilweise Tippfehler wie **Kennzeichung* oder **Radeweg*. Durch eine Korrektur dieser Fehler könnte die Performance verbessert werden. Für die Rechtschreibkorrektur wurden die Wörter der Beiträge mit Hunspell[12] geprüft und ggf. durch den ersten Korrekturvorschlag ersetzt.

Dieses Vorgehen liefert unterschiedlich gute Ergebnisse. Im Durchschnitt über alle untersuchten Datensätze verbessert sich der F_1-Wert um 0,3 Prozentpunkte. Beim Bürgerhaushalt Köln 2015 liegt die Verbesserung bei mehr als 2,5 Prozentpunkten, beim Raddialog Köln dagegen verschlechtert sich der Wert um mehr als 2,5 Punkte.

Bei Betrachtung der von der automatischen Rechtschreibprüfung gemachten Korrekturen fällt auf, dass zum Teil Wörter auf die falsche Art und Weise korrigiert werden. Beispielsweise wird **Kennzeichung* zu *Kenneichung* korrigiert, jedoch wäre die Korrektur zu *Kennzeichnung* korrekt. Deshalb wurde noch eine Variante der Rechtschreibkorrektur getestet, bei der Korrekturvorschläge bevorzugt werden, die Wörter sind, die bereits im Trainingskorpus vorkommen.

Bei dieser Variante konnte aber gar keine F_1-Verbesserung gegenüber der Baseline festgestellt werden. Eine genauere Betrachtung der Korrekturen zeigt, dass in vielen Fällen Verschlimmbesserungen gemacht werden, z. B. werden die korrekten Wörter *Wiesenweg* und *queren* zu *Wiesen weg* bzw. *Querelen* „korrigiert".

Fazit: Beide dargestellten Varianten für eine automatische Rechtschreibkorrektur haben keinen, höchstens einen geringen Nutzen für die Qualität des Klassifikators, weshalb sie nicht weiter verwendet wurden.

3.2.8 Wort-N-Gramme

Beim Bag-of-Word-Modell gehen die einzelnen Wörter unabhängig von ihrer Reihenfolge ein. Jedoch macht es einen Unterschied, ob in einem Textbeitrag von *weniger Geld ausgeben*, *mehr Geld ausgeben* oder *weniger Geld sparen* die Rede ist, sodass die Erhaltung des Wortkontextes sinnvoll wäre. Bei der Verwendung von Wort-N-Grammen werden für $N > 1$ nicht einzelne Wörter betrachtet, sondern Gruppen von N aufeinanderfolgenden Wörtern.

Fazit: Bei Verwendung einer Kombination von Wort-2- und -3-Grammen werden die F_1-Werte im Durchschnitt um 1,9 Prozentpunkte besser, allerdings auch bei drei Datensätzen schlechter, beim Mängelmelder Braunschweig sogar um 12 Prozentpunkte. Die starke negative Auswirkung beim Mängelmelder könnte damit erklärt werden, dass beim Melden von Mängeln bereits einzelne Wörter reichen, um die Kategorie festlegen zu können. Diese Werte bestätigen die Ergebnisse von Wang und C. Manning (2012), die festgestellt haben, dass Wort-N-Gramme insgesamt gemischte und eingeschränkte Nützlichkeit bei Themenklassifikation haben.

[12] `https://hunspell.github.io/`

3.2.9 Charakter-N-Gramme

Eine Alternative zu Wort-N-Grammen sind Charakter-N-Gramme, bei denen nicht aufeinanderfolgende Wörter, sondern Buchstaben verwendet werden. Ein Vorteil von Charakter-N-Grammen gegenüber Wort-N-Grammen ist, dass sie gewissermaßen automatisch mit verschiedenen Flexionsendungen, Komposita und auch Rechtschreibfehlern umgehen können.

Beispielsweise haben die Wörter *Ampelschaltung* und **Ampleschaltungen* trotz unterschiedlicher Wortformen und Rechtschreibfehler acht Charakter-3-Gramme gemeinsam. Allerdings kann es auch zu „falschen“ Übereinstimmungen kommen, zum Beispiel bei den Wörtern *nach* und *nachts*, die ein gemeinsames 4-Gramm haben, aber inhaltlich nicht verwandt sind.

Fazit: Ein durchschnittlich um ca. 21 Prozentpunkte besserer F_1-Score lässt sich mit einer Kombination von Charakter-5 und -6-Grammen erzielen. Die größte Verbesserung gibt es beim Bürgerhaushalt Bonn 2011 mit 34 Prozentpunkten. Das Verwenden von N-Grammen innerhalb der Wortgrenzen liefert bei den getesteten N-Gramm-Größen in fast allen Fällen mit einem durchschnittlichen Unterschied von drei Prozentpunkten bessere Ergebnisse als die Verwendung von wortübergreifenden N-Grammen.

3.2.10 Einschränkung auf bestimmte Wortarten

Aus anderen Bereichen der Textklassifikation, z. B. dem Opinion Mining (Scholz et al., 2012), ist bekannt, dass Wörter bestimmter Wortarten eine wichtigere Rolle bei der Klassifikation haben als andere, sodass das Beschränken auf bestimmte Wortarten die Klassifikationsergebnisse verbessern kann. Das Herausfiltern bestimmter Wortarten wie z. B. Präpositionen, die häufig in vielen Texten vorkommen, hat außerdem einen ähnlichen Effekt wie das Entfernen von Stoppwörtern. Für das Part-of-Speech-Tagging wurde der POS-Tagger von spaCy[13] verwendet, der eine Variante des Universal Tag Sets (Petrov et al., 2012), die 17 Tags enthält, benutzt.

Fazit: Werden nur Nomen in die Bag-of-Words-Darstellung aufgenommen, so verbessern sich die F_1-Werte auf den Datensätzen durchschnittlich um 11,6 Punkte; nur beim Bürgerhaushalt Köln 2015 gibt es eine Verschlechterung um 1,4 Punkte. Bei der Verwendung von Adjektiven, Nomen, Verben, und Eigennamen kommt es bei allen getesteten Datensätzen zu einer Verbesserung des F_1-Wertes; im Schnitt verbessert sich dieser um 12,5 Prozentpunkte.

3.2.11 Kombination der Verbesserungsmöglichkeiten

Schließlich wurde getestet, wie sich das Kombinieren der oben vorgestellten Verbesserungsmöglichkeiten auf den F_1-Score auswirkt. Wegen der im Vergleich zu Wort-N-Grammen sehr guten Ergebnisse von Charakter-N-Grammen wurde auf die Verwendung von Wort-N-Grammen verzichtet. Ebenfalls wurden Lemmatisierung, Stemming,

[13] `https://spacy.io/`

Zerlegung von Kompositia und Rechtschreibkorrektur nicht verwendet, da davon ausgegangen wird, dass die Probleme, die durch diese Techniken gelöst werden, durch die Verwendung von Charachter-N-Grammen gar nicht auftreten.

Getestet wurden also datensatzabhängige Stoppwortlisten, minimale Document Frequency, tf-idf-Gewichtung, Charakter-N-Gramme und die Einschränkung auf bestimmte Wortarten. Dabei wurden die besten Hyperparameter für diese Verfahren mithilfe von Gridsearch und vierfacher Kreuzvalidierung auf vier großen, thematisch unterschiedlichen Datensätzen ermittelt, nämlich den Raddialogen, dem Bürgerhaushalt Bonn, dem Mängelmelder Braunschweig und Bad Godesberg. Die getesteten Parameterwerte sind im Anhang in Tabelle 35 aufgeführt.

Für die unterschiedlichen Datensätze ergeben sich natürlich unterschiedliche optimale Kombinationen von Parameterwerten. Um eine Parameterwahl zu finden, die für alle Datensätze gut ist, wurden die vier ermittelten Kombinationen noch einmal auf allen Datensätzen getestet. Dadurch ergibt sich folgende Wahl:

- N-Gramm-Größe: 4
- minimale Document Frequency: 3
- maximale Document Frequency: 100 %
- Anzahl der Nachbarn n: 5
- Metrik: Cosinus
- Gewichtung: inverse Distanz
- POS-Filter: Adjektive, Nomen, Verben
- tf-idf-Gewichtung: ja

In Tabelle 5 sind die F_1-Scores dieser optimierten Parameterwahl und die der Baselines gegenüber gestellt. Auf allen Datensätzen sind die Ergebnisse besser geworden, wobei die Verbesserungen bei den Datensätzen mit nur zwei Kategorien (Köln 2013, 2015 und 2016) mit wenigen Prozentpunkten wesentlich kleiner sind als bei den Datensätzen mit mehr Kategorien. Die größte absolute Verbesserung gibt es beim Bürgerhaushalt Bonn 2011 mit knapp 37 Prozentpunkten.

3.3 Evaluation anderer klassischer Machine-Learning-Verfahren

Basierend auf der Featureauswahl und dem Preprocessing des vorherigen Abschnitts wurden andere, klassische Klassifikatoren als k-NN evaluiert. Unter *klassisch* werden hier Klassifikatoren verstanden, die keine neuronalen Netze verwenden; solche Klassifikatoren werden im nächsten Abschnitt betrachtet.

Es wurden folgende Arten von Klassifikatoren evaluiert:

- Multinomial Naive Bayes (C. D. Manning et al., 2008, S. 258 ff.): Laplace-Glättung mit $\alpha = 1$

Datensatz	k-NN	Human	k-NN, optimiert
Raddialoge	31	73	63
Bonn	28	74	64
Köln	27	66	58
Moers	24	69	61
Bürgerhaushalt			
Bonn	20		55
Bonn 2011	21		58
Bonn 2015	14		47
Köln			
Köln 2012	27		63
Köln 2013	51		52
Köln 2015	51		64
Köln 2016	48		54
Mängelmelder Braunschweig	57		73
Nahverkehrsplan Ulm	22		43
Bad Godesberg	9		33

Tabelle 5: F_1-Scores der optimierten Baseline im Vergleich mit den Baselines auf den verschiedenen Datensätzen

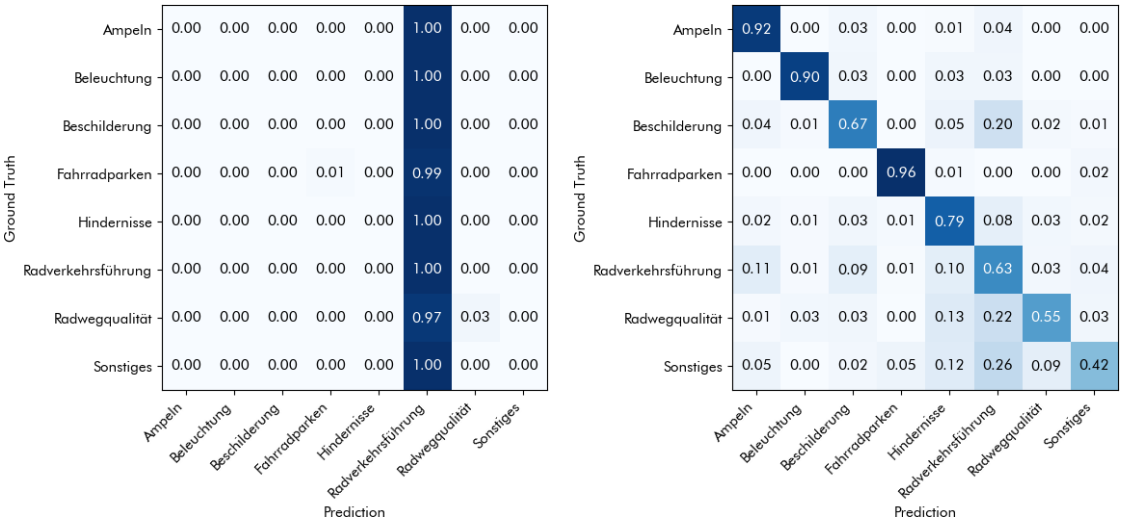
- Logistische Regression (Bishop, 2006, S. 205 f.): L_2-Regularisierung, Regularisierungsstärke $C = 1$, One-vs-Rest
- Random Forest (Breiman, 2001): 10 Bäume, Gini Impurity
- Support Vector Machine (SVM, Cortes und Vapnik, 1995): Regularisierungsstärke $C = 1$, RBF-Kernel, Kernel-Koeffizient $\gamma = 1/\#\text{features}$, One-vs-Rest

Die Parameterwahl entspricht den Standardwerten von scikit-learn (Pedregosa et al., 2011). Im Preprocessing wurden Charakter-N-Gramme der Längen 3 bis 5 erstellt mit einer Document-Frequency von maximal 50 %.[14]

Ersetzt man den k-NN-Klassifikator durch einen der oben genannten Klassifikatoren, so verschlechtern sich in allen Fällen die durchschnittlichen F_1-Scores um 10 bis 40 Prozentpunkte. Ein Blick auf die Konfusionsmatrizen zeigt, dass die Klassifikatoren ein Problem mit den in den Datensätzen vorzufindenen Klassenimbalancen haben; Abbildung 9a zeigt beispielhaft die Konfusionsmatrix für den Naive-Bayes-Klassifikator auf dem Raddialog Bonn.

Eine Möglichkeit mit Klassenimbalancen umzugehen, ist die Verwendung von Random Oversampling (He und Garcia, 2008), bei dem zufällig gewählte Samples der unterrepräsentierten Klassen mehrfach in den Trainingsdatensatz aufgenommen werden, sodass schließlich alle Klassen gleich groß sind. Bei allen getesteten Klassifikatoren hat diese Methode zu einer Verbesserung geführt, was man auch deutlich an der Konfusionsmatrix für Naive Bayes mit Oversampling in Abbildung 9b sehen kann.

[14] Die im vorherigen Abschnitt im Rahmen der Gridsearch bestimmten Parameter wurden an dieser Stelle nicht verwendet, da sie zum Zeitpunkt der Evaluation klassischer Verfahren noch nicht bekannt waren.



(a) Ohne Resampling (b) Mit Oversampling

Abbildung 9: Konfusionsmatrizen für Multinomial Naive Bayes auf dem Raddialog Bonn

Klassifikator	ohne Resampling	mit Oversampling
k-NN	**55**	53
Multinomial Naive Bayes	22	59
Logistiesche Regression	41	**64**
Random Forest	44	51
SVM	15	52

Tabelle 6: Durchschnittliche Macro-F_1-Werte der klassischen Klassifikatoren

Datensatz	k-NN	Human	k-NN, optimiert	Logit, optimiert
Raddialoge	31	73	63	71
Bonn	28	74	64	72
Köln	27	66	58	68
Moers	24	69	61	67
Bürgerhaushalt				
Bonn	20		55	60
Bonn 2011	21		58	62
Bonn 2015	14		47	51
Köln				
Köln 2012	27		63	74
Köln 2013	51		52	59
Köln 2015	51		64	69
Köln 2016	48		54	70
Mängelmelder Braunschweig	57		73	87
Nahverkehrsplan Ulm	22		43	60
Bad Godesberg	9		33	42

Tabelle 7: F_1-Scores der optimierten logistischen Regression im Vergleich mit den Baselines und k-NN auf den verschiedenen Datensätzen

Logistische Regression mit Oversampling ist mit einem durchschnittlichen F_1-Wert von 64 % der beste evaluierte Klassifikator und ist auf jedem Datensatz besser als k-NN, im Durchschnitt um zehn Punkte. Die Werte aller Klassifikatoren können Tabelle 6 entnommen werden.

Für logistische Regression (Logit) wurde wieder – ähnlich wie bei k-NN – eine Gridsearch durchgeführt, um die optimalen Parameter für den Klassifikator und das Preprocessing zu finden. Eine Übersicht der untersuchten Parameter ist in Tabelle 36 im Anhang. Die gefundenen Parameter sind:

- N-Gramm-Größe: 3 und 4
- minimale Document Frequency: 2
- maximale Document Frequency: 50 %
- inverse Regularisierungsstärke C: 1
- Bias für Entscheidungsfunktion: ja
- Bias-Skalierung: 0,1
- POS-Filter: keine
- tf-idf-Gewichtung: ja

In Tabelle 7 sind die F_1-Werte für diese Parameterwahl aufgeführt. Im Vergleich zu k-NN ergibt sich immer eine Verbesserung des F_1-Scores, die durchschnittlich acht Prozentpunkte beträgt. Die F_1-Werte auf den Raddialog-Datensätzen sind bereits sehr nah an

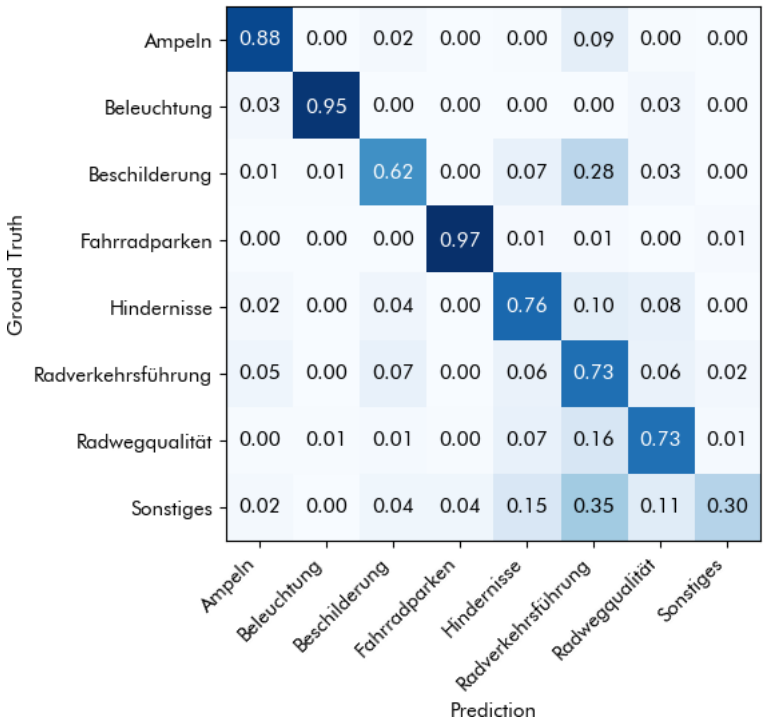


Abbildung 10: Konfusionsmatrix für die optimierte logistische Regression auf den Raddialogen

der Human-Baseline, d. h. der Klassifikator ist in der Klassifikationsaufgabe bereits fast so gut wie ein Mensch, der nicht für die Aufgabe geschult worden ist.

Vergleicht man die Konfusionsmatrizen der logistischen Regression in Abbildung 10 und die des Human-Klassifikators in Abbildung 8, so fällt auf, dass in beiden Fällen die Kategorie *Radverkehrsführung* oft mit *Beschilderung* verwechselt wird. Ein Blick auf die betroffenen Textbeiträge zeigt, dass hier oft eine schlechte Verkehrsführung kritisiert und eine bessere Beschilderung vorgeschlagen wird. Gemäß der Regel 3 für die Kategorievergabe, die in Abschnitt 2.1 beschrieben ist, ist dann *Beschilderung* die korrekte Kategorie. Im Gegensatz zum Menschen ist bei der logistischen Regression aber der Recall bei *Sonstiges* mit 30 % sehr gering.

3.4 Evaluation Graph-basierter Verfahren

Dieser Abschnitt beschäftigt sich mit zwei Graph-basierten Ansätzen zur automatischen Verschlagwortung. Die Verfahren wurden ursprünglich für Klassifikationsaufgaben in anderen Domänen entwickelt und es wurde getestet, wie gut sie zur Kategorisierung von Textbeiträgen geeignet sind.

θ_r	0,01	0,01	0,01	0,13	0,24	0,32	0,24	0,24	0,24	0,24
θ_a	0	0	0	0	0	0	1	2	3	2
Normalisierung	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓
Stemming	✗	✓	✓	✓	✓	✓	✓	✓	✓	✗
F_1	44	35	45	50	51	51	51	52	51	51

Tabelle 8: Durchschnittliche F_1-Scores von BookGraph

3.4.1 BookGraph

Von Pollack (2017) wurde ein Graph-basiertes Verfahren namens BookGraph entwickelt, mit dem Bücher automatisiert kategorisiert und verschlagwortet werden können. Die Knoten des Graphen bestehen aus Kategorien[15], Schlagwörtern, Autoren und Wörtern des Inhaltsverzeichnisses (ohne Stoppwörter).

Durch Kanten werden Knoten verbunden, deren Informationen im selben Buch vorkommen; Ausnahme sind Wort-Knoten, die nur mit Knoten anderer Typen verbunden werden. Die Kanten erhalten ein Gewicht, das der Anzahl der gemeinsamen Vorkommen im Trainingsdatensatz entspricht.

Das Verfahren wurde für die Verwendung zur Kategorisierung von Textbeiträgen leicht abgewandelt. Weil es immer nur einen Autor zu einem Beitrag gibt und außerdem anders als bei Büchern kein direkter Zusammenhang zwischen Autoren und Themen vermutet wird, werden keine Autoren-Knoten verwendet. Der verwendete Algorithmus zur Zuweisung einer Kategorie funktioniert wie folgt:

1. Für jedes Wort im Textbeitrag finde inzidente Kategorien, deren Kantengewichte über einem Schwellwert θ liegen.

2. Für jede gefundene Kategorie bilde den Mittelwert der Kantengewichte, optional normalisiert durch die Klassenverteilung.

3. Wähle jeweils diejenige Kategorie mit dem größten Mittelwert.

Getestet wurden unterschiedliche relative Schwellwerte θ_r und absolute Schwellwerte θ_a sowie die Anwendung von Stemming. Eine Auswahl der getesteten Kombinationen mit den erreichten F_1-Werte ist in Tabelle 8 zu sehen. Ohne Normalisierung gemäß Kategorieverteilung sind die Ergebnisse deutlich schlechter und Stemming verbessert die Ergebnisse.

Der beste erreiche durchschnittliche F_1-Score von 52 % liegt oberhalb der Ergebnisse der k-NN-Baseline (31 %), aber deutlich unterhalb der optimierten logistischen Regression, die auf 64 % kommt. BookGraph funktioniert also für die vorliegende Kategorisierungsaufgabe nicht so gut wie andere Verfahren.

[15] Im ursprünglichen Anwendungskontext heißen diese *Systemstellen*.

Klassen-Normalisierung	✗	✓	✗	✓	✓	✓	✓	✓	✓	✓
Kanten-Normalisierung	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓
Stemming	✗	✗	✗	✗	✓	✓	✓	✓	✓	✓
Schleifen	✗	✗	✗	✗	✗	✓	✓	✓	✓	✓
Mindestfrequenz	1	1	1	1	1	1	2	3	4	5
F_1	35	38	41	43	45	46	47	48	47	46

Tabelle 9: Durchschnittliche F_1-Scores mit Tonalitätsgraph

3.4.2 Tonalitätsgraph

Scholz und Conrad (2013) haben ein Graph-basiertes Verfahren zur Tonalitätsanalyse von Zeitungsartikeln entwickelt. Bei diesem Verfahren werden für jedes vorkommende Adjektiv, Adverb, Nomen, Verb und Negationspartikel, die für die Tonalität relevant sind, Knoten angelegt. Zusammen auftretende Wörter werden mit einer Kante verbunden, wobei das Gewicht der Kante ein Tupel ist, in dem gezählt wird, wie oft die Kombination für die Tonalitäten positiv, neutral und negativ vorkommt. Bei der Klassifikation werden aus den Kanten acht verschiedene Features für eine SVM berechnet.

Im Rahmen dieser Arbeit werden statt Tonalitäten die Kategorien verwendet. Zur Bestimmung der wahrscheinlichsten Kategorie werden alle in einem Textbeitrag vorkommenden Wörter im Graphen nachgeschlagen und die Gewichte der verbindenen Kanten komponentenweise addiert. Die Kategorie mit der größten Summe wird für den Textbeitrag gewählt.

Ein theoretischer Vorteil dieses Verfahrens ist, dass es auch dann gut funktioniert, wenn zwei Texte eine ähnliche BOW-Darstellung haben, weil Kookkurrenzen von Wörtern berücksichtigt werden.

Bei den durchgeführten Experimenten wurden die relevanten Wortarten Adjektiv, Eigenname, Nomen und Verb in den Graphen aufgenommen. Wie schon bei BookGraph wurde auch getestet, wie sich Stemming und eine Normalisierung der gezählten Häufigkeiten auswirkt. Bei der Normalisierung wurden einmal die Zahlen pro Kanten auf 1 normiert oder die Summen der Anzahlen pro Kategorie auf 1 normiert. Weiterhin wurde untersucht, ob sich die Ergebnisse verbessern, wenn Schleifen im Graphen zugelassen werden, d. h. wenn Unigramme gezählt werden. Zuletzt wurde noch ein Schwellwert für die absolute Häufigkeit (Mindestfrequenz) getestet.

Die durchschnittlichen F_1-Werte auf den Datensätzen sind in Tabelle 9 zu sehen. Die einzelnen Normalisierungsmethoden führen jeweils zu einer Verbesserung und deren Kombination zum besten Ergebnis. Auch das Aufnehmen von Schleifen und das Verwenden von Stemming verbessern den F_1-Score. Im Durchschnitt werden mit einer Mindestfrequenz von 3 die besten Ergebnisse mit einem durchschnittlichen F_1-Wert von 48 % erzielt.

Jedoch liegt dieser Wert, wie auch schon bei BookGraph, zwar oberhalb der k-NN-Baseline, aber deutlich unterhalb der logistischen Regression. In der implementierten Variante ist dieses Graph-basierte Verfahren also nicht gut zur Klassifikation der Textbeiträge geeignet.

Titel/Inhalt separat	✗	✗	✓	✓	✓	✓	✓	✓
Oversampling	✗	✓	✗	✓	✗	✗	✓	✓
idf-Gewichtung	✗	✗	✗	✗	✓	✗	✗	✓
Komposita-Behandlung	✗	✗	✗	✗	✗	✓	✓	✓
Raddialoge	47	49	53	55	50	55	**59**	54
Bürgerhaushalt Bonn	48	30	47	42	45	50	**52**	46
Mängelmelder Braunschweig	56	60	59	64	60	66	**68**	66

Tabelle 10: F_1-Scores des Fully-Connected-Netzwerks

3.5 Evaluation künstlicher neuronaler Netze

In diesem Abschnitt werden Tests mit Klassifikatoren beschrieben, die künstliche neuronale Netze benutzen. Da Textbeiträge, die die Eingabe für die Klassifkatoren bilden, unterschiedliche Längen haben, müssen die Architekturen der Netze mit unterschiedlichen Eingabelängen umgehen können. Eine Möglichkeit dafür ist das Überführen der Sätze in eine Darstellung fester Länge wie Continiuous Bag of Words (CBOW, Mikolov et al., 2013), eine andere die Verwendung von Architekturen, die direkt mit unterschiedlichen Eingabelängen umgehen können, wie z. B. Netzwerke mit Long Short-Term Memory (LSTM, Hochreiter und Schmidhuber, 1997).

Im Folgenden werden verschiedene Netzwerkarchitekturen vorgestellt und evaluiert. Wegen der längeren Trainingsdauer und der für ein gutes Training erforderlichen größeren Menge an Trainingsdaten wurden diese Modelle nur auf den Raddialogen, dem Bürgerhaushalt Bonn und dem Mängelmelder Braunschweig getestet, wobei vierfache Kreuzvalidierung verwendet wurde.

3.5.1 Fully Connected Feed Forward Network

Zunächst wurde ein einfaches neuronales Netz evaluiert, das nur aus einem Fully Connected Layer besteht. Als Eingabe erhält das Netz den Textbeitrag in der CBOW-Darstellung, bei der die Word-Embeddings der einzelnen Tokens – optional mit Gewichtung nach inverser Document Frequency (idf) – gemittelt werden. Es wurde ein von Liebeck et al. (2017) auf der deutschen Wikipedia vortrainiertes Word2Vec-Embedding (Mikolov et al., 2013) mit 100 Dimensionen verwendet. Für Komposita, die nicht im Embedding-Vokabular vorhanden sind, wurden optional die gemittelten Embeddings der Wörter verwendet, aus denen das Kompositum besteht, anstatt auf den Nullvektor zurückzugreifen. Ferner wurde eine Variante getestet, bei der die CBOW-Darstellung von Titel und Inhalt separat berechnet und anschließend konkateniert werden.

Es wurde jeweils so lange trainiert, bis sich der kategorische Kreuzentropie-Loss auf den Validierungsdaten über die letzten 10 Epochen nicht verbessert hat, was nach etwa 80 Epochen der Fall war. Der Adam-Optimizer (Kingma und Ba, 2014) mit einer Batch-Größe von 20 wurde verwendet.

Wie Tabelle 10 entnommen werden kann, hat die idf-Gewichtung größtenteils negative Auswirkungen auf den F_1-Score. Oversampling verbessert meistens die Ergebnisse, nur beim Bürgerhaushalt Bonn sind sie in manchen Kombinationen schlechter. Die separate

CBOW-Darstellung von Titel und Inhalt wirkt sich in den meisten Fällen positiv auf den F_1-Wert aus, ebenso das Berechnen von fehlenden Word-Embeddings für Komposita. Auf allen drei Datensätzen schneidet die separate CBOW-Darstellung mit Oversampling und Komposita-Behandlung und ohne idf-Gewichtung am besten ab.

3.5.2 Convolutional Neural Network

Weiterhin wurde ein einfaches Convolutional Neural Network (CNN) getestet, das aus den folgenden Layern besteht (Goldberg, 2016):

- Word-Embedding mit Vektoren mit 100 Dimensionen
- Convolutional-Layer mit 250 Filtern der Größe 5
- Max-Pooling
- Fully Connected Layer (FC1)
- optionales, zweites Fully Connected Layer (FC2) mit 62 Neuronen

Nach den Fully Connected Layern und dem Embedding wurde ein Dropout von 20 % angewendet. Als Aktivierungsfunktion dient ReLU und der Adam-Optimizer mit einer Batch-Größe von 20 wurde verwendet. Die Netze wurden jeweils so lange trainiert, bis sich der kategorische Kreuzentropie-Loss über die letzten 5 Epochen nicht verbessert hat; dies war nach rund 17 Epochen der Fall.

Für das Word-Embedding wurden drei verschiedene Varianten getestet:

- das von Liebeck et al. (2017) auf der deutschen Wikipedia vortrainierte Word2Vec-Embedding, wobei die Vektoren während des Trainings nicht verändert wurden
- dasselbe Embedding, wobei die Embedding-Matrix während des Trainings mittrainiert wurde (Retrofitting)
- ein zufällig initialisiertes Embedding, das alle Wörter des Trainingsdatensatzes enthält

Das Preprocessing besteht aus dem Tokenisieren des Textes, Ersetzen von Groß- durch Kleinbuchstaben und Beschränken bzw. Padding auf eine Textlänge von 400 Tokens.

Das Verwenden von Oversampling hat in fast allen Fällen zu einer Verbesserung der F_1-Werte geführt; nur beim Bürgerhaushalt Bonn gab es bei der Architektur mit nicht-fixiertem Embedding und FC2 eine leichte Verschlechterung. Das Hinzufügen eines zweiten Fully Connected Layers wirkt sich eher negativ auf die Performance aus, was möglicherweise daran liegt, dass es für die erhöhte Anzahl an Parametern zu wenige Trainingsdaten gibt.

Embedding	zufällig				vortrainiert				Retrofitting			
FC2	✗		✓		✗		✓		✗		✓	
Oversampling	✗	✓	✗	✓	✗	✓	✗	✓	✗	✓	✗	✓
Raddialoge	45	53	44	51	52	**59**	48	**57**	**57**	**58**	53	**57**
Bürgerhaushalt Bonn	37	43	34	37	**50**	**50**	47	47	**51**	**51**	**50**	48
M. Braunschweig	68	**78**	62	76	64	69	59	66	71	**76**	70	74

Tabelle 11: F_1-Scores des CNN

Embedding	zufällig		vortrainiert		Retrofitting	
Oversampling	✗	✓	✗	✓	✗	✓
Raddialoge	35	48	17	37	39	**51**
Bürgerhaushalt Bonn	21	27	34	31	**46**	43
Mängelmelder Braunschweig	61	**70**	30	49	62	**68**

Tabelle 12: F_1-Scores des CNN4SC

In allen Fällen führt das Mittrainieren des Embedding-Layers zu besseren Ergebnissen. Bei den Raddialogen und dem Bürgerhaushalt Bonn ist das zufällig initialisierte Embedding deutlich schlechter als die vortrainierten Varianten. Beim Mängelmelder Braunschweig zeigt sich dieser Effekt nicht. Ein möglicher Grund dafür ist, dass das vortrainierte Embedding beim Mängelmelder häufig auftretende Wörter wie *Fahrradwrack*, *Straßenbeleuchtung* und *abgemeldetes* nicht enthält.

Eine Übersicht der F_1-Werte der verschiedenen Architekturen auf den Datensätzen zeigt Tabelle 11. Insgesamt liefert die Architektur mit einem Fully Connected Layer, vortrainierten, aber nicht fixiertem Embedding und Oversampling die besten Ergebnisse.

3.5.3 Convolutional Neural Networks for Sentence Classification

Kim (2014) verwendet eine andere CNN-Architektur mit Filtern unterschiedlicher Größen, die erfolgreich auf verschiedenen Klassifikationsaufgaben eingesetzt wurde. Die in dieser Arbeit verwendete Implementierung CNN4SC besteht aus folgenden Layern:

- Konkatenierte Word-Embeddings des Eingabetexts, beschränkt auf 400 Tokens
- jeweils 100 Convolution-Filter der Größen 3, 4 und 5
- Max-Pooling pro Filter
- Fully Connected Layer

Für das Embedding wurden wieder die drei Varianten aus Abschnitt 3.5.2 getestet. Wie bei Kim (2014) wurde beim letzten Layer ein Dropout von 50 % angewendet. Die Netze wurden bis zu 30 Epochen lang trainiert.

In den meisten Fällen verbessert Oversampling die Ergebnisse. Die Verwendung des fixierten, vortrainierten Embeddings führt gegenüber dem zufällig initialisierten Embedding meistens zu einer Verschlechterung der F_1-Werte, wird es aber mittrainiert, werden

Datensatz	kein Resampling	Oversampling
Raddialoge	10	**29**
Bürgerhaushalt Bonn	**7**	**7**
Mängelmelder Braunschweig	23	**37**

Tabelle 13: F_1-Scores des VDCNN

ähnliche bis bessere Ergebnisse erzielt. Wie in Tabelle 12 zu sehen ist, werden die besten Ergebnisse im Schnitt mit vortrainiertem und mittrainiertem Embedding in Kombination mit Oversampling erreicht.

3.5.4 Very Deep Convolutional Networks

Bei der Klassifikation von Bildern werden tiefe CNNs erfolgreich eingesetzt. Diese Netze können aufgrund ihrer Struktur abbilden, dass Bilder aus einfachen Formen zusammengesetzt sind und diese Formen sich zu immer komplexeren Objekten zusammensetzen, die schließlich erkannt werden können.

Conneau et al. (2017) schlagen vor, tiefe CNNs auch zur Textklassifikation zu verwenden, da sich auch Texte aus kleinen Strukturen, nämlich Zeichen und N-Grammen, zusammensetzen; ihre Architektur nennen sie Very Deep Convolutional Networks (VDCNN). Vorteilhaft an dieser Architektur ist, dass – ähnlich wie bei Charakter-N-Grammen – auf natürliche Art und Weise mit Komposita, Flexionsendungen und unbekannten Wörtern umgegangen werden kann.

Sie erreichen mit einer Architektur aus 29 Layern, die im Wesentlichen aus einfachen Convolutional Layern bestehen, state-of-the-art Ergebnisse auf verschiedenen, großen Datensätzen mit 120.000 bis 3.600.000 Trainingsdatensätzen. Ihre Experimente haben aber auch gezeigt, dass die Architektur auf den kleineren Datensätzen vergleichsweise schlecht abschneidet, sodass auch keine guten Ergebnisse für die im Vergleich sehr kleinen Datensätzen, die in dieser Arbeit behandelt werden, erwartet wurden.

Evaluiert wurde die kleinste Variante von VDCNN, die in Abbildung 11 dargestellt ist. Das erste Layer ist ein Charakter-Embedding, worauf ein Layer mit 64 Convolutions der Größe 3 folgt. Danach folgen vier Convolutional Blocks, die jeweils aus Convolution-Layern der Größe bestehen, deren Anzahl sich mit jedem Block verdoppelt, gefolgt von einem Pooling-Layer. Schließlich sind vorm Output-Layer zwei Fully Connected Layers mit ReLU-Aktivierung.

Wie Tabelle 13 entnommen werden kann, sind die F_1-Werte mit Oversampling besser als ohne, liegen aber mit Werten zwischen 7 und 37 weit unterhalb der Scores, die mit den klassischen Verfahren erreicht werden. Grund dafür ist wahrscheinlich die zu geringe Größe der evaluierten Datensätze.

3.5.5 Hierarchical Attention Networks

Von Yang et al. (2016) wurden Hierarchical Attention Networks (HANs) zur Textklassifikation vorgestellt. Die Charakteristika dieses Modells sind die hierarchische Struktur, die

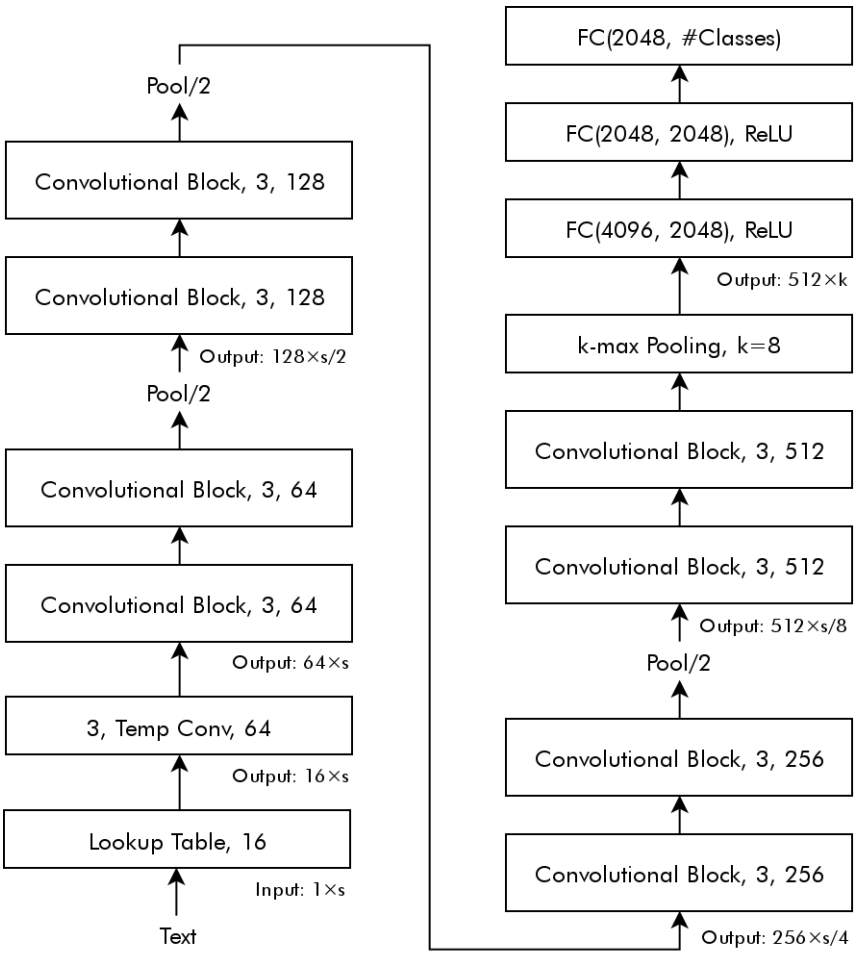


Abbildung 11: VDCNN-Architektur nach Conneau et al. (2017)

Embedding	zufällig		vortrainiert		Retrofitting	
Oversampling	✗	✓	✗	✓	✗	✓
Raddialoge	44	43	**55**	**55**	**53**	47
Bürgerhaushalt Bonn	24	30	**47**	**49**	41	41
Mängelmelder Braunschweig	63	**72**	66	**70**	**71**	69

Tabelle 14: F_1-Scores des HAN

die hierarchische Struktur von Textdokumenten (Dokumente bestehen aus Sätzen, Sätze aus Wörtern) widerspiegeln soll, sowie der Attention-Mechanismus, der die Aufmerksamkeit des Netzes auf wichtige Wörter lenken soll.

Das Netz verwendet Gated Recurrent Units (GRUs), die von Cho et al. (2014) vorgeschlagen wurden. GRUs folgen dem Grundprinzip von LSTM-Zellen, haben aber nur zwei statt drei Gates und keine explizite Memory-Komponente.

Die Wörter des Eingabetextes gehen zunächst satzweise durch einen Wort-Encoder, der die Wörter zunächst mithilfe eines Word-Embeddings in eine Vektordarstellung überführt. Anschließend werden bidirektionale GRUs verwendet, deren versteckte Zustände pro Wort den Informationsgehalt zusammenfassen sollen. Danach wird ein Attention-Mechanismus verwendet, um abzubilden, dass nicht alle Wörter gleich viel zum Inhalt eines Satzes beitragen.

Mit den so erhaltenen Satzrepräsentationen wird analog verfahren, d. h. es wird wieder ein Satz-Encoder mit GRU-Zellen verwendet, der die Satzdarstellungen des vorherigen Schrittes erhält, und Attention auf Satzebene wird angewandt. Der resultierende Dokumentvektor ist dann Eingabe für ein einfaches Fully Conencted Layer mit softmax-Aktivierung.

Für das Embedding wurden wieder das vortrainierte Embedding, Retrofitting und eine zufällige Initialisierung getestet. Ferner wurde jede Variante mit und ohne Oversampling getestet.

Wie Tabelle 14 entnommen werden kann, ist kein deutlicher Trend zu sehen, welche der getesteten Varianten besser funktioniert. Tendenziell funktioniert ein vortrainiertes Embedding – mit oder ohne Retrofitting – besser als ein zufällig initialisiertes Embedding. Oversampling verbessert nicht in allen Fällen die Ergebnisse. Über alle getesteten Datensätze betrachtet ist ein vortrainiertes, fixiertes Embedding mit Oversampling am besten.

3.5.6 FastText

Von Bojanowski et al. (2017) wurde eine Erweiterung des Skipgram-Modells von Mikolov et al. (2013) entwickelt, das auf Charakter-N-Grammen beruht. Das normale Skipgram-Modell hat den Nachteil, dass es keine Teilwortinformationen berücksichtigt, sodass insbesondere für Wörter, die nicht im Trainingsvokabular vorhanden sind, keine Repräsentation angegeben werden kann.

Bei der Erweiterung wird jedes Wort als Bag-of-Character-n-Grams dargestellt, wobei Präfixe und Suffixe speziell markiert werden und außerdem das gesamte Wort selbst

Embedding	zufällig				vortrainiert			
Bigramme	✗	✗	✓	✓	✗	✗	✓	✓
Oversampling	✗	✓	✗	✓	✗	✓	✗	✓
Raddialoge	**56**	**56**	53	52	**56**	**56**	53	55
Bürgerhaushalt Bonn	**42**	41	37	38	**43**	**43**	40	40
Mängelmelder Braunschweig	74	**76**	69	71	**75**	**76**	73	73

Tabelle 15: F_1-Scores von FastText

ebenfalls als n-Gramm mit aufgenommen wird. Ein gesamtes Wort wird dann von der Summe der Vektorrepräsentationen seiner n-Gramme repräsentiert. Dadurch können einerseits Repräsentationen für unbekannte Wörter berechnet werden und andererseits die Repräsentationen für seltene Wörter zuverlässiger gelernt werden.

FastText ist ein von Joulin et al. (2017) entwickelter Klassifikator, der ein auf diese Art erstelltes Embedding als Eingabe erhält. Die Wortrepräsentationen werden zu Satzrepräsentationen gemittelt und anschließend in einen linearen Klassifikator gegeben. FastText erzielt in den Tests der Autoren ähnlich gute Ergebnisse wie VDCNN bei der Sentiment-Erkennung.

Für das Embedding wurde ein zufällig initialisiertes Embedding mit 100 Dimensionen[16], d. h. das Embedding wurde mithilfe der Trainingsdaten trainiert, und ein auf Texten von Common Crawl und Wikipedia vortrainiertes Embedding mit 300 Dimensionen (Grave et al., 2018) getestet. Weiterhin wurden sowohl Wort-Unigramme als auch -Bigramme und die Verwendung von Oversampling evaluiert. In allen Fällen wurde das Modell 1000 Epochen lang trainiert.

Wie Tabelle 15 entnommen werden kann, hat Oversampling keinen großen Einfluss auf den F_1-Score. Die Verwendung von Wort-Bigrammen verschlechtert in allen Fällen die Ergebnisse und ein vortrainiertes Embedding verbessert den F_1-Wert leicht. Am besten ist die Kombination aus vortrainiertem Embedding mit Wort-Unigrammen.

3.5.7 Zusammenfassung und Vergleich mit klassischen Methoden

In Tabelle 16 sind die F_1-Scores der in diesem Abschnitt getesteten Klassifikatoren für die jeweils beste Hyperparameterwahl zusammengefasst. Am besten schneiden die einfachen Modelle ab, die aus Fully Connected Layern (FC) oder Convolutional-Layern (CNN) bestehen.

Alle Modelle liefern im Vergleich zur logistischen Regression Ergebnisse, die mindestens acht Prozentpunkte schlechter sind. Ein möglicher Grund dafür kann sein, dass die geringe Anzahl von wenigen tausend Trainingstexten zu wenig für einen Deep-Learning-Ansatz ist. Weiterhin gäbe es noch weitere Hyperparameter, die bei den Deep-Learning-Klassifikatoren angepasst werden könnten, wie z. B. die Lernrate oder die Größe der Layer, und das Preprocessing könnte verbessert werden.

[16] Es wurde auch ein zufällig initialisiertes Embedding mit 300 Dimensionen mit Unigrammen und ohne Oversampling getestet. Die Ergebnisse waren sehr ähnlich zu denen mit einem Embedding 100 Dimensionen, sodass auf weitere Tests mit 300 Dimensionen verzichtet wurde.

Modell	Raddialoge	Bürgerhaushalt Bonn	M. Braunschweig
FC	**59**	**52**	68
CNN	**58**	**51**	**76**
CNN4SC	51	43	68
VDCNN	29	7	37
HAN	55	49	70
FastText	56	43	**76**
Logit, optimiert	71	60	87

Tabelle 16: F_1-Scores der Klassifikatoren, die ein neuronales Netzwerk benutzen, für die jeweils besten Hyperparameter im Vergleich zur logistischen Regression

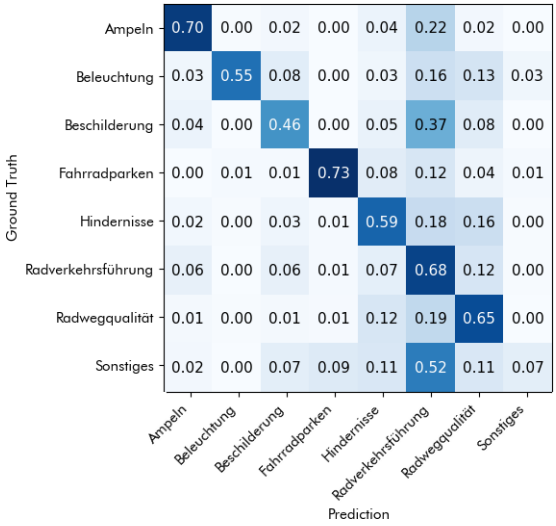
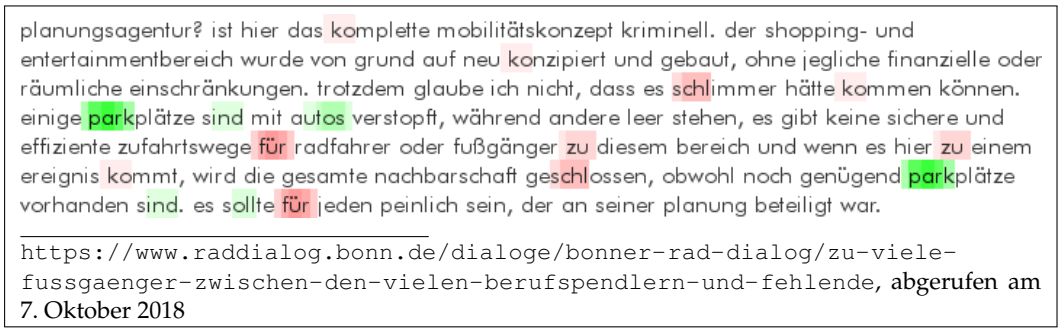


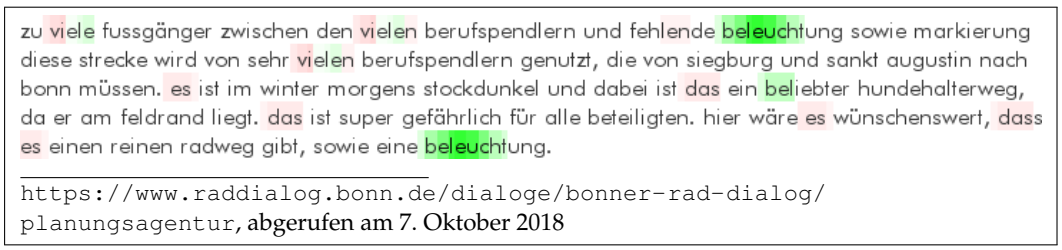
Abbildung 12: Konfusionsmatrix für die FC-Architektur auf den Raddialogen

Da diese Klassifikatoren aber im Vergleich zu den klassischen Verfahren längere Trainingszeiten und – damit die Trainingszeiten nicht noch länger sind – GPUs mit ausreichend viel RAM erfordern, wurde auf eine weitere Optimierung verzichtet. Die klassischen Verfahren haben im praktischen Einsatz den Vorteil, dass ihre Funktionsweise einfacherer nachvollzogen werden kann, Trainingszeiten kürzer sind und handelsübliche Hardware zum schnellen Trainieren ausreichend ist.

Ein Vergleich der Konfusionsmatrizen der gut funktionierenden Architektur FC (Abbildung 12) und logistischer Regression (Abbildung 10) zeigt, dass beide Klassifikatoren – zumindest auf den Raddialogen – ähnliche Schwächen haben: Die Kategorie *Radverkehrsführung* wird oft anstelle von *Beschilderung* gewählt und der Recall bei *Sonstiges* ist gering, wobei FC mit 7 % wesentlich schlechter ist als die logistische Regression mit 30 %. FC hat ferner bei der Kategorie *Beleuchtung* einen um 40 Prozentpunkte geringen Recall und hat im Allgemeinen eine „verwaschenere“ Konfusionsmatrix.



(a) Prediction: *Hindernisse*, Ground Truth: *Sonstiges*, von Benutzer gewählte Kategorie: *Sonstiges*, Schlagwörter: *nicht ortsgebundene Vorschläge*



(b) Prediction: *Beleuchtung*, Ground Truth: *Radverkehrsführung*, von Benutzer gewählte Kategorie: *Sonstiges*, Schlagwörter: *Vorschlag für neuen Radweg*, *Beleuchtung fehlt*

Abbildung 13: Falsch klassifizierte Textbeiträge aus dem Raddialog Bonn mit Hervorhebungen von ELI5 für die vorhergesagte Kategorie

3.6 Analyse falsch kategorisierter Textbeiträge

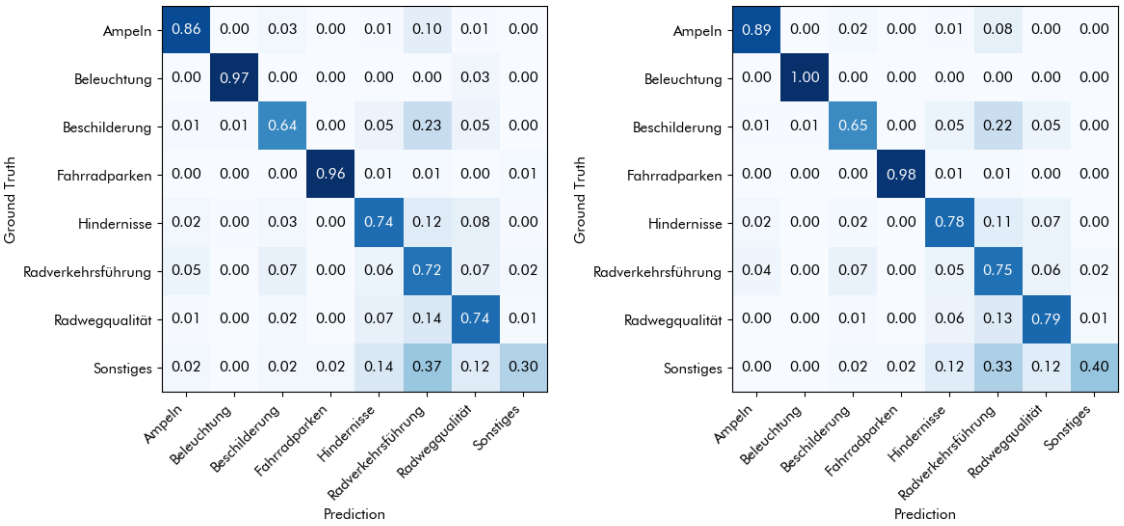
In diesem Abschnitt wird ein Blick auf die falsch kategorisierten Texte pro Datensatz geworfen, um die Schwächen des Klassifikators herauszufinden. Dazu wurden nicht nur die zugewiesenen Kategorien betrachtet, sondern auch ELI5[17], eine Bibliothek zum Debuggen und Untersuchen von Klassifikatoren und deren Predictions, verwendet. Als Klassifikator dient die optimierte logistische Regression, die bisher die besten Ergebnisse erzielt hat.

3.6.1 Raddialoge

Wie schon aus den Konfusionsmatrizen deutlich wird, werden Beiträge der Kategorie *Sonstiges* nur schlecht erkannt. Ein Grund dafür wird am Beispiel in Abbildung 13a deutlich: Es gibt kaum Wörter, die wirklich charakteristisch für Beiträge dieser Kategorie sind, wodurch einzelne Wörter, in diesem Fall *Parkplätze*, den Ausschlag für die vom Klassifikator gewählte Kategorie geben. Daher sollten andere Methoden evaluiert werden, wie die Kategorie *Sonstiges* besser erkannt werden kann.

Weiterhin gibt es einige Textbeiträge, in denen verschiedene Teilaspekte angesprochen werden. Ein Beispiel dafür ist in Abbildung 13b zu sehen. In diesem Beitrag werden

[17] https://github.com/TeamHG-Memex/eli5



(a) nur Ground-Truth-Kategorie als korrekt gewertet

(b) alle Kategorien der Schlagwörter als korrekt gewertet

Abbildung 14: Konfusionsmatrizen auf dem Raddialog Bonn

schon im Titel unterschiedliche Themen angesprochen, darunter „fehlende Beleuchtung“. Im Text wird dann ein neuer Radweg vorgeschlagen, sodass nach Regel 3 in Abschnitt 2.1 von der Moderation die Kategorie *Radverkehrsführung* gesetzt wurde. Der Klassifikator wählt hier jedoch die Kategorie *Beleuchtung*, da die Wörter dieser Kategorie ein größeres Gewicht haben als die Wörter, die auf Radverkehrsführung bzw. neu anzulegende Wege hinweisen.

Beiträge, die mehrere verschiedene Themen unterschiedlicher Kategorien ansprechen, haben in der Regel auch mehrere Schlagwörter, die zu den unterschiedlichen Kategorien gehören. Daher wurde für den Raddialog Bonn getestet, wie sich die Performance verändert, wenn die vom Klassifikator gewählte Kategorie genau dann als richtig gewertet wird, wenn dem Beitrag ein Schlagwort dieser Kategorie zugeordnet worden ist. Beispielweise würde die Prediction *Beleuchtung* für den Beitrag in Abbildung 13b dann als richtig gewertet werden, da der Beitrag das Schlagwort *Beleuchtung fehlt* hat. Dadurch ergibt sich eine Verbesserung des F_1-Scores von 72 % auf 76 %. Wie aus den Konfusionsmatrizen in Abbildung 14 abgelesen werden kann, verbessert sich insbesondere bei *Sonstiges* der Recall um 10 Prozentpunkte.

Darüber hinaus gibt es Fälle, bei denen der Klassifikator einzelne Wörter keiner Kategorie zugehörig ansieht. Beispielsweise ist *ausgewiesener* (bzw. dessen N-Gramme) kein Indikator für eine Kategorie, das ähnliche Wort *gekennzeichneter* aber Indiz für die Kategorie *Beschilderung*. Hier könnte es hilfreich sein, Informationen zu Synonymen, Hyponymen (Unterbegriffen) und Hyperonymen (Oberbegriffen) einzubeziehen.

3.6.2 Bürgerhaushalte

Auch beim Bürgerhaushalt Bonn wird die Kategorie *Sonstiges* mit einem Recall von weniger als 10 % am schlechtesten erkannt, wie in Abbildung 15 zu sehen ist.

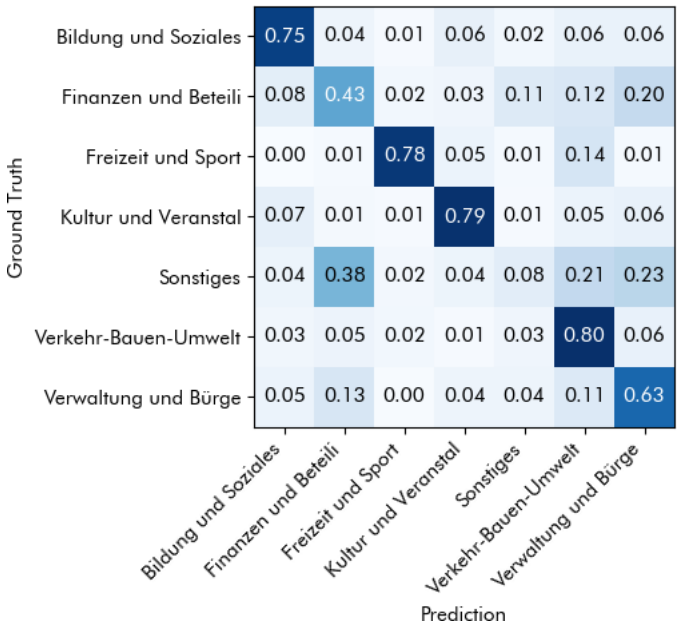


Abbildung 15: Konfusionsmatrix auf dem Bürgerhaushalt Bonn

Außerdem wird die Kategorie *Finanzen und Beteiligung* häufig nicht erkannt und oft mit *Verwaltung und Bürgerservice* verwechselt. Grund dafür ist einerseits geteiltes Vokabular zwischen diesen Kategorien. Beispielweise kommt *Arbeit* sowohl in Beiträgen zu *Finanzen und Beteiligung* als auch bei *Verwaltung und Bürgerservice*, z. B. im Zusammenhang mit Sachbearbeitern, häufig vor. Weiterhin wird oft die Verwaltung aufgefordert, etwas im Finanzbereich zu tun.

Andererseits kommt es zum Teil auch zu „Irreführungen“ durch Namen wie *Beethovenhalle*, sodass fälschlicherweise die Kategorie *Kultur und Veranstaltungen* zugeordnet wird. Hier könnte es zu einer Verbesserung der F_1-Werte kommen, indem Eigennamen, wie z. B. Straßennamen, entfernt werden.

3.6.3 Mängelmelder Braunschweig

Beim Mängelmelder Braunschweig liegt, wie in Abbildung 16 zu sehen, bei fast allen Kategorien der Recall über 80 %, nur bei *Friedhofsunterhaltung* gibt es einen Recall um 60 %. Oft liegt hier eine Verwechslung mit der Kategorie *Straßen-, Radweg- und Gehwegschäden* vor, da z. B. über Wegschäden auf einem Friedhof gesprochen wird.

Die relativ wenigen, anderen missklassifizierten Beiträge kommen durch unterschiedliche Gründe zustande. Es gibt wieder zum Teil Wörter, die „missverstanden“ werden. Beispielsweise ist *Motorradwrack* ein Hinweis auf die Kategorie *abgemeldete Fahrzeuge*, da das Wort aber nicht in den Trainingsdaten vorkam, wird der Wortteil *radwrack* als Hinweis für die Kategorie *Fahrradwracks* gewertet. Ein ähnliches Phänomen gibt es bei einer wilden Müllkippe bestehend aus einer Couch vor einer Spielothek. Da das Wort *Müll* nicht im Beitrag vorkommt und *Couch* nicht in den Trainingsdaten ist, wird fälschlicherweise die Kategorie *Spielplatzunterhaltung* statt *Wilde Müllkippe, Sperrmüllreste* zugewiesen.

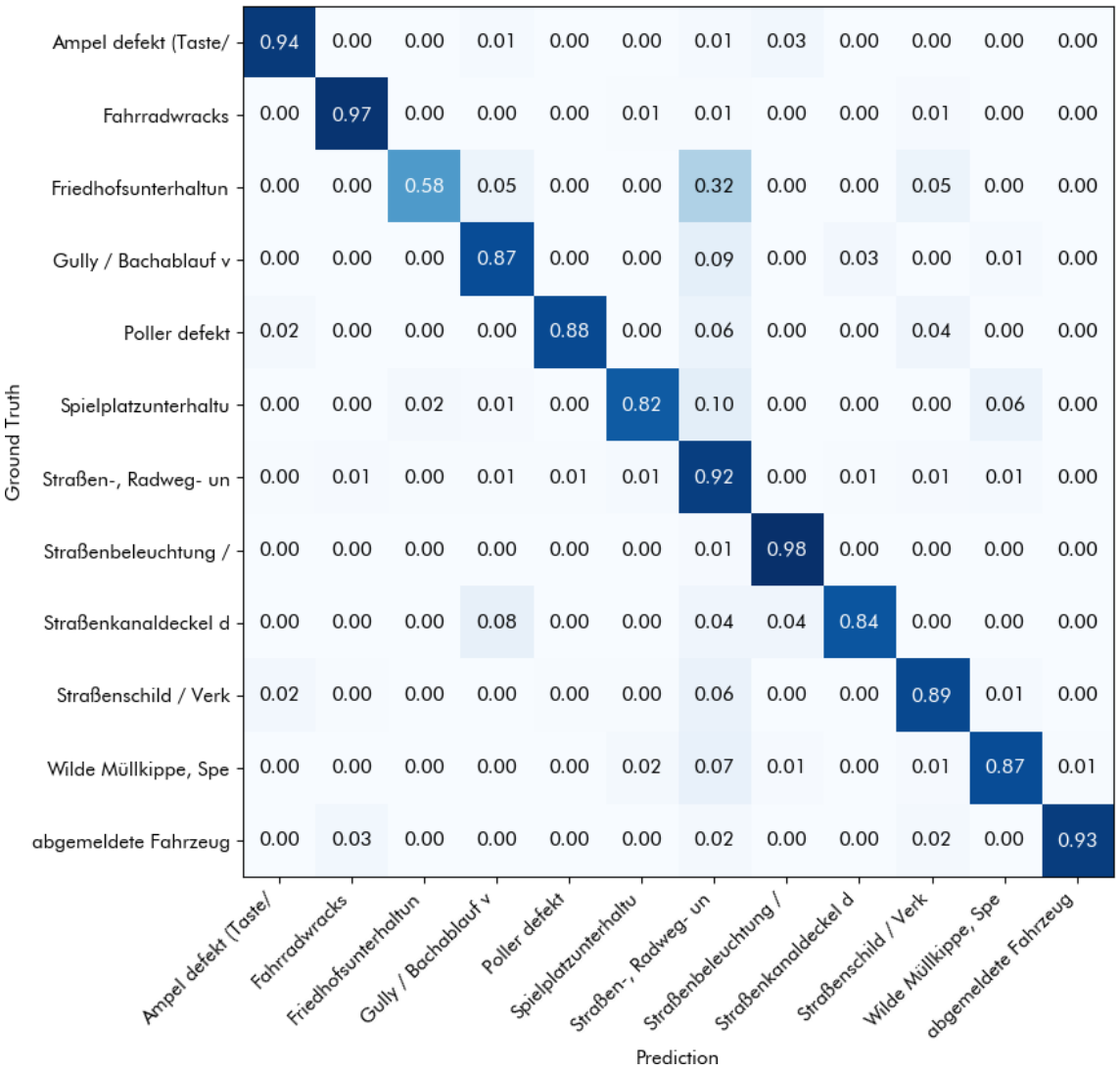


Abbildung 16: Konfusionsmatrix auf dem Mängelmelder Braunschweig

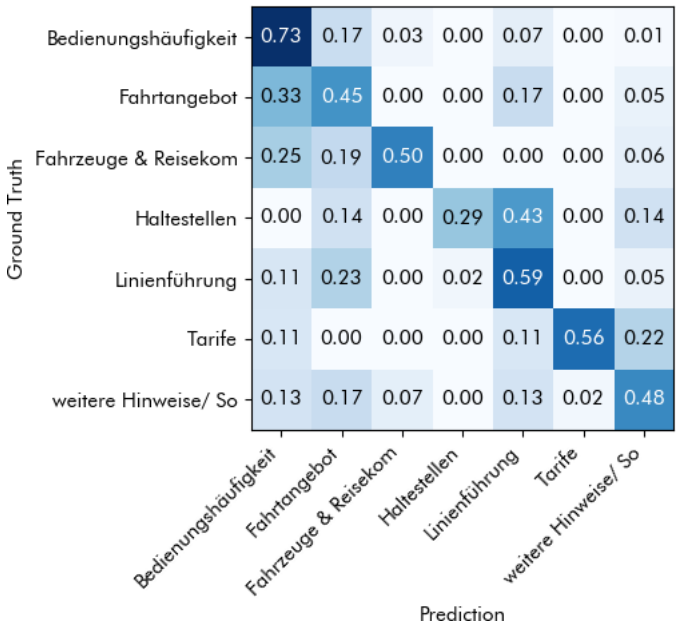


Abbildung 17: Konfusionsmatrix auf dem Nahverkehrsplan Ulm

Des Weiteren hängen auch Kategorien thematisch zusammen. Beispielsweise kann es durch defekte *Straßenkanaldeckel* zu *Straßen-, Radweg- und Gehwegschäden* kommen, sodass hier leicht die Kategorien verwechselt werden, wenn nicht zwischen Mangelursache und -folge unterschieden wird.

Bei einzelnen Beiträgen kommt es auch vor, dass der Hauptinhalt in einem Foto festgehalten ist und nicht direkt im Text angesprochen wird, sodass eine rein textbasierte Klassifikation fast unmöglich ist.

3.6.4 Nahverkehrsplan Ulm

Die Konfusionsmatrix für den Klassifikator auf dem Nahverkehrsplan Ulm ist in Abbildung 17 zu sehen. Es sei angemerkt, dass es für die Kategorien *Tarife*, *Haltestellen* und *Fahrzeuge & Reisekomfort* nur sehr wenige Textbeiträge gibt, weshalb hier wenige falsch klassifizierte Beiträge den F_1-Wert nach unten ziehen.

Auffällig ist, dass die Kategorien *Fahrtangebot* und *Linienführung* oft miteinander verwechselt werden. Das liegt unter anderem daran, dass das Wort *nach* ein starker Indikator für die Kategorie *Fahrtangebot* ist, da hier öfters über Linien, die *nach* X fahren gesprochen wird. Allerdings kommt das Wort natürlich auch oft vor, wenn über Linienführung gesprochen wird, sodass bei Beiträgen, die keine weiteren typischen Wörter dieser Kategorie enthalten, leicht mit *Fahrtangebot* verwechselt werden.

Weiterhin kommt es vor, dass Beiträge verschiedene Themen auf einmal behandeln, z. B. ein größeres Fahrtangebot und eine geänderte Linienführung. Bei diesen Beiträgen wären eigentlich beide Kategorien richtig. Außerdem gibt es Beiträge in der Kategorie *Weitere Hinweise/Sonstiges*, die sich auf einen Beitrag einer anderen Kategorie beziehen, wodurch

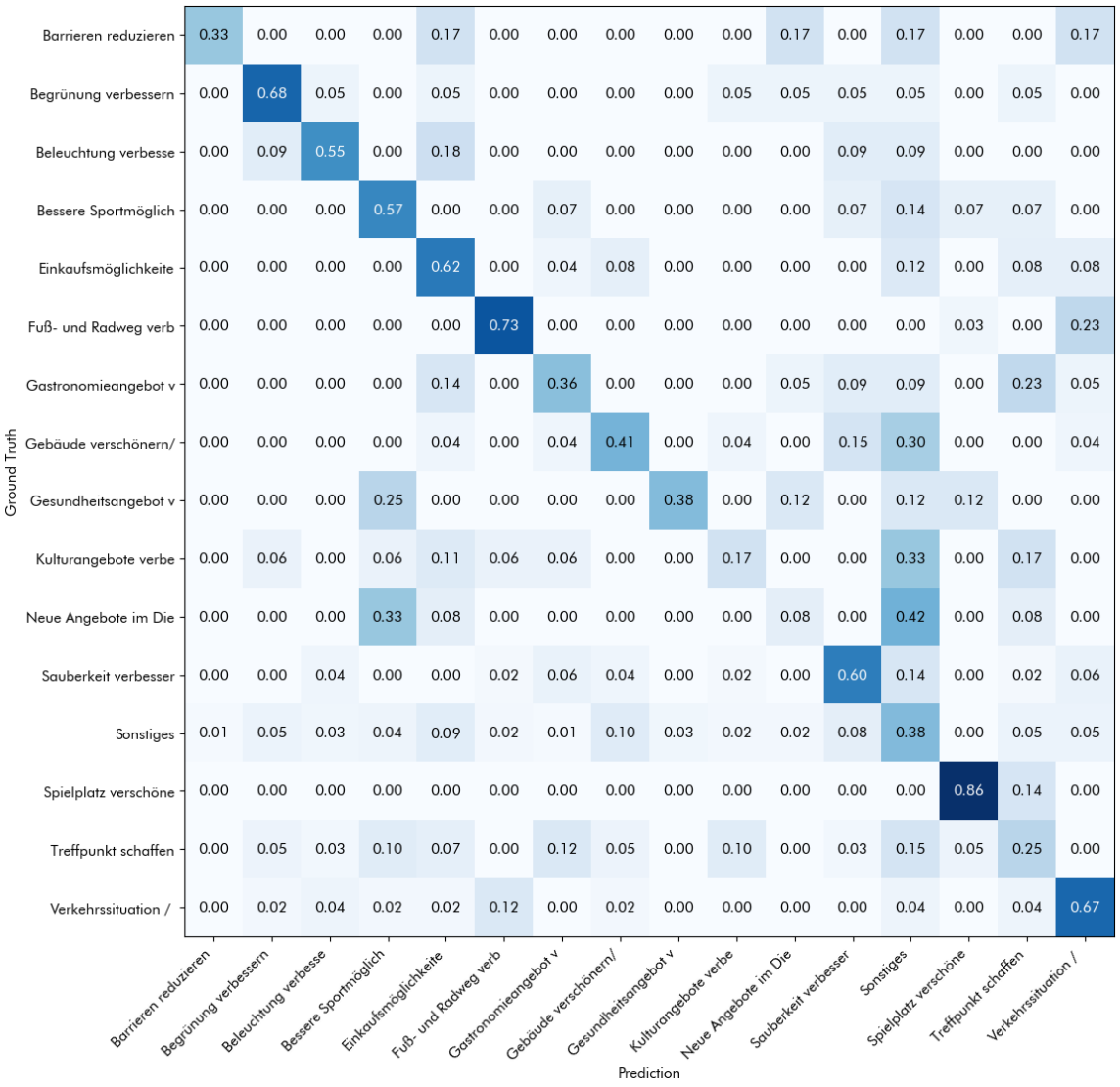


Abbildung 18: Konfusionsmatrix auf dem Leitbild Bad Godesberg

die entsprechenden Schlüsselwörter im Beitrag vorkommen und eine „falsche“ Prediction zur Folge haben.

3.6.5 Leitbild Bad Godesberg

Beim Leitbild Bad Godesberg gibt es mit 42 % den schlechtesten F_1-Score im Vergleich zu den anderen Datensätzen, aber mit 16 Kategorien auch die größte Labelmenge und mit weniger als 600 Samples ist der Datensatz auch eher klein. Es gibt viele Kategorien mit nur wenigen Beiträgen, sodass dort die Predictions entsprechend schlechter ausfallen. Die größte Kategorie ist *Sonstiges*, welche wegen der bereits angesprochenen Gründe schwierig zu erkennen ist, wie in Abbildung 18 gesehen werden kann. Wird die Cross-Validation nur mit den neun Kategorien durchgeführt, die mindestens 20 Beiträge im

Trainingsdatensatz haben, so verbessert sich der F_1-Score um etwa fünf Prozentpunkte auf 47 %.

Es gibt auch Kategorien, die in manchen Themenbereichen sehr nah beieinander sind. Eine Forderung nach mehr Schwimmmöglichkeiten kann je nach Schwerpunktsetzung unter *Bessere Sportmöglichkeiten* oder *Gesundheitsangebot verbessern* fallen. Ähnlich ist es auch bei den Kategorien *Fuß- und Radweg verbessern* und *Verkehrssituation/Anbindung verbessern*, denn eine bessere Anbindung kann beispielsweise durch verbesserte Wege erreicht werden.

Ferner gibt es auch Textbeiträge, deren Ground-Truth-Kategorie tatsächlich nicht die passendste Kategorie ist oder auch gar keinen sinnvollen Inhalt haben. Eine Überprüfung der Kategorien durch Online-Moderatoren hat anscheinend nicht stattgefunden.

3.7 Weitere Optimierungen

Basierend auf den Analysen im vorherigen Abschnitt werden nun weitere Änderungen an der Klassifikationspipeline beschrieben, die zu einer Verbesserung der automatischen Kategorisierung führen sollen. Dabei werden die Featureauswahl und das Preprocessing so angepasst, dass es möglich sein soll, die erkannten Probleme des Klassifikators zu beheben. Als Baseline dient nun die logistische Regression mit Charakter-N-Grammen.

3.7.1 Entfernen von Straßennamen

Es kann angenommen werden, dass in den meisten Zusammenhängen konkrete Straßennamen wie *Mozartstraße* keine Relevanz haben und nur den Feature-Raum unnötig vergrößern und außerdem den Klassifikator mit bestimmten Namen, die auf eine andere, ggf. falsche Kategorie hinweisen, in die Irre führen können. Daher wurde getestet, wie sich das Ersetzen aller Straßennamen durch einen speziellen Token, der nur Zeichen enthält, die in den Textbeiträgen nicht vorkommen, auf die Performance auswirkt.

Als Grundlage dient eine auf OpenStreetMap-Daten[18] basierende Liste aller in Deutschland verkommenen Straßennamen. Um fehlerhafte Einträge wie *W* und auch in normalen Formulierungen vorkommenden Straßennamen wie *Am Markt* nicht zu ersetzen, wird die Liste auf die Straßennamen beschränkt, die mindestens fünf Zeichen lang sind und auf einen der zehn häufigsten Suffixe enden. Weiterhin wurden Groß- und Kleinschreibung ignoriert und die abgekürzten Straßennamen mit *str.* zur Liste hinzugefügt.

Fazit: Im Schnitt gibt es durch das Ersetzen von Straßennamen keine Änderung im F_1-Wert. Es gibt zum Teil Verbesserungen um etwa einen Prozentpunkt, z. B. beim Bürgerhaushalt Bonn 2015, aber auch Verschlechterungen um knapp zwei Prozentpunkte beim Mängelmelder Braunschweig. Dieser Preprocessing-Schritt ist also für die getesteten Datensätze nicht sinnvoll.

[18] `https://www.openstreetmap.org/`

3.7.2 Entfernen von Sonderzeichen

Ein typischer Preprocessing-Schritt ist das Entfernen von Sonderzeichen, um aus dem Feature-Raum wenig benutze Dimensionen zu entfernen und Overfitting zu vermeiden. Getestet wurde das Entfernen von Nicht-ASCII-Zeichen, wobei die im Deutschen häufig verkommenden Zeichen ä, ö, ü (und die entsprechenden Großbuchstaben) und ß durch a, o, u bzw. ss ersetzt worden sind. Diese Ersetzung ist auch dadurch motiviert, dass manche Nutzer statt Umlauten die Grundbuchstaben in ihren Beiträgen verwenden. Ferner wurden alle Ziffern durch Einsen ersetzt, da in der Regel für die Klassifikation nur die Größenordnung einer Zahl relevant ist und nicht ihr genauer Wert. Weiterhin werden so Bezeichnungen von Bundesstraßen vereinheitlicht (*B7* und *B5* würden zu *B1*).

Fazit: Diese Vorverarbeitungsschritte führen zu unterschiedlichen Ergebnissen auf den Datensätzen. Beim Raddialog Moers verschlechtert sich das Ergebnis um knapp 1,8 Prozentpunkte, wohingegen es beim Bürgerhaushalt Köln 2014 eine Verbesserung um knapp 1,5 Prozentpunkte gibt. Insgesamt gibt es im Durchschnitt eine leichte Verschlechterung, weshalb diese Vorverarbeitungsschritte nicht weiter verwendet werden. Auch die einzelne Anwendung von Umlaut- oder Zahlenersetzung hat eine kleine Verschlechterung zur Folge.

3.7.3 Hinzufügen von verwandten Wörtern

Es wurde festgestellt, dass aufgrund der zum Teil wenigen Texte pro Kategorie dem Klassifikator nicht immer alle Synonyme, Hyponyme und Hyperonymen bekannt sind. Das GermaNet (Hamp und Feldweg, 1997) ist ein deutsches Wortnetz, in dem für Nomen, Adjektive und Verben Synonyme, Hyponyme und Hyperonymen hinterlegt sind. Mithilfe des GermaNet wurde eine Variante von Oversampling implementiert, die wie folgt funktioniert:

1. Behalte alle ursprünglichen Textbeiträge.
2. Füge für jeden Textbeitrag mindestens eine Kopie hinzu, bei der alle Wörter durch ein zufälliges Synonym, Hyponym, Hyperonym oder das Wort selbst ersetzt sind, sodass alle Kategorien gleich häufig vorkommen.

Im zweiten Schritt wird auch das Wort selbst in die Menge der Ersetzungskandidaten aufgenommen, um einen Fallback zu haben, falls das Wort nicht im GermaNet vorkommt.

Wenn alle Hyponyme und Hyperonyme für Ersetzungen zugelassen werden, kommt es allerdings häufig vor, dass Wörter gewählt werden, die im Kontext des Partizipationsverfahren sehr wahrscheinlich nicht vorkommen, z. B. *Möbel* (Hyperonym von *Lampe*) oder *Heilpflanze* (Hyponym von *Wurzel*). Deshalb wurde getestet, ob sich Ergebnisse verbessern, wenn die Auswahl der Wörter für die Ersetzungen auf die Wörter beschränkt wird, die im Trainingsdatensatz vorkommen. Außerdem wurde geprüft, ob das Weglassen von Hyponymen oder Hyperonymen zu besseren Ergebnissen führt.

Fazit: Wie Tabelle 17 entnommen werden kann, wurden in allen Kombinationen um etwa vier Prozentpunkte schlechtere Ergebnisse erzielt. Betrachtet man einzelne Datensätze,

beschränkte Wortwahl	✗	✓	✗	✓	✗	✓	✗	✓	Logit-Baseline
Synonyme	✓	✓	✓	✓	✓	✓	✓	✓	
Hyperonyme	✓	✓	✗	✗	✓	✓	✗	✗	
Hyponyme	✓	✓	✓	✓	✗	✗	✗	✗	
F_1-Score	61	61	62	61	61	62	62	62	64

Tabelle 17: Durchschnittliche F_1-Scores mit Germanet-Oversampling

so gibt es nur wenige Fälle mit leichten Verbesserungen. Beispielsweise werden beim Mängelmelder Braunschweig die Ergebnisse bei allen Kombinationen um etwa einen Prozentpunkt besser, bei allen anderen Datensätzen gibt es in den meisten Kombinationen Verschlechterungen. Es ist weiterhin keine eindeutige Tendenz erkennbar, ob die Beschränkung auf Wörter aus dem Trainingsdatensatz oder das Weglassen von Hypero- oder Hyponymen sinnvoll ist. Weil die Ergebnisse im Durchschnitt schlechter sind, wurde das Germanet-Oversampling nicht weiter verwendet.

3.7.4 Korpuserweiterung mithilfe von Suchmaschinen

Von Guzmán et al. (2007) wurde ein semi-supervised Verfahren vorgestellt, bei dem kleine Datensätze mithilfe von Webdokumenten augmentiert werden und das auch dann gut funktioniert, wenn für Kategorien weniger als zehn gelabelte Datensätze vorhanden sind. Das Verfahren besteht aus zwei Schritten: Der Bildung von Websuchanfragen, um zusätzliche thematisch relevante Textdokumente zu erhalten, und dem eigentlichen semi-supverised Learning.

Guzmán et al. (2007) verwenden eine Kombination von Worthäufigkeiten und Information Gain, um relevante Wörter pro Kategorie für Suchanfragen zu finden. Für diese Arbeit wurde stattdessen die von einer logistischen Regression auf BOW-Darstellungen ermittelte Menge der zehn relevantesten Wörter W verwendet. Pro Kategorie wurden dann zehn zufällige Suchanfragen generiert, die fünf zufällige Wörter aus W (gewählt mit Zurücklegen) enthalten. Beispielsweise könnte für die Kategorie *Beleuchtung verbessern* die Anfrage *straßenbeleuchtung licht unterführung straßenbeleuchtung dunkelheit* gemacht werden.

Die URLs der ersten zehn Google-Suchergebnisse wurden besucht, wobei die Seiten der untersuchten Online-Partizipationsverfahren ausgeschlossen wurden. Aus den besuchten Seiten wurden Textpassagen mit mindestens 100 Zeichen herausgeparst und zusammen mit der Kategorie k, die für die Query-Konstruktion verwendet worden ist, im Webkorpus S gespeichert.

Das semi-supervised Learning erfolgt dann ähnlich wie bei Guzmán et al. (2007):

1. Auf dem Trainingsset T wird eine logistische Regression C trainiert.
2. Die Texte in S werden von C klassifiziert.
3. Es wird eine Teilmenge $S_r \subseteq S$ von relevanten Texten gewählt, die folgende Bedingungen erfüllen:

   (a) Die von C bestimmte Kategorie stimmt mit der in S hinterlegten Kategorie k überein.
   (b) Die von C bestimmte Wahrscheinlichkeit für k ist mindestens $\theta \in [0, 1]$.
4. Die logistische Regression wird erneut auf der Menge $T \cup S_r$ trainiert.

Wegen des Crawling-Aufwands wurde dieses Verfahren nur auf drei Datensätzen getestet, nämlich den Raddialogen, dem Bürgerhaushalt Bonn und dem Leitbild Bad Godesberg. Die Tests wurden für $\theta \in \{0{,}2; 0{,}3; 0{,}4; 0{,}5\}$ ausgeführt. Beispielsweise ergab sich für $\theta = 0{,}5$ beim Raddialog Bonn $|S_r| \approx 530$. Außerdem wurde S_r optional so undergesamplet, dass das Verhältnis der Kategorien in S_r dem in T entspricht.

Fazit: Auf allen getesteten Datensätzen wurde nie eine signifikante Veränderung des F_1-Scores festgestellt – weder in positive noch in negative Richtung –, weshalb die F_1-Werte hier nicht noch einmal explizit aufgeführt werden.

Es sei angemerkt, dass es hier durch die einmalige Bildung von W pro Datensatz einen Information-Leak in die Validierungsmenge gibt. Da die Ergebnisse aber sowieso nicht positiv waren, wird dies an dieser Stelle ignoriert.

3.7.5 Bessere Erkennung von Sonstiges

Wie oben festgestellt, ist insbesondere bei der Kategorie *Sonstiges* der Recall gering. Deshalb werden nun unterschiedliche Möglichkeiten untersucht, mit denen der Recall für *Sonstiges* verbessert werden könnte. Dafür werden nur die verschiedenen Raddialog-Datensätze, die Bürgerhaushalte Bonn, der Nahverkehrsplan Ulm und das Leitbild Bad Godesberg betrachtet, weil nur diese Datensätze eine Kategorie *Sonstiges* o. ä. haben. Die anderen Datensätze können bei den Tests ausgeschlossen werden, da a priori bekannt ist, ob es eine *Sonstiges*-Kategorie gibt.

Beiträge der Kategorie *Sonstiges* zeichnen sich – im Gegensatz zu Beiträgen anderer Kategorien – häufig nicht durch eine bestimmte Menge charakteristischer Wörter ab, sondern dadurch, dass sie Wörter enthalten, die in anderen Beiträgen eher nicht vorkommen. Diese Beobachtung motiviert die Hypothese, dass die durchschnittliche Dokumentfrequenz der Wörter in Beiträgen der Kategorie *Sonstiges* geringer ist als in anderen Beiträgen.

Bei einem Blick auf die Raddialoge wird diese Vermutung bestätigt: Die durchschnittliche Dokumentfrequenz in der Kategorie *Sonstiges* liegt bei ca. 15 %, bei allen anderen Kategorien bei mindestens 15,6 %, im Durchschnitt bei 17,4 %. Bei den anderen Datensätzen bestätigt sich die Vermutung allerdings nicht. Beim Bürgerhaushalt Bonn ergibt sich *Sonstiges* eine durchschnittliche Dokumentfrequenz von 17,1 %, bei allen anderen Kategorien ist sie ca. einen Prozentpunkt geringer.

Nichtsdestoweniger wurde untersucht, ob die Dokumentfrequenz als zusätzliches Feature neben den Charakter-N-Grammen nützlich ist. Dazu wurden folgende acht Werte jeweils für die Menge und Liste aller Wörter eines Textbeitrags berechnet:

- durchschnittliche Dokumentfrequenz
- Median der Dokumentfrequenz

- Anzahl der Wörter, die nicht in Trainingsdaten enthalten sind
- Anzahl der Wörter, die weniger als zweimal in den Trainingsdaten sind
- Anzahl der Wörter, die weniger als fünfmal in den Trainingsdaten sind
- Anzahl der Wörter, die weniger als zehnmal in den Trainingsdaten sind
- Anzahl der Wörter, die weniger als #Trainingstexte/10 in den Trainingsdaten sind

Diese Features führen allerdings im Durchschnitt zu einer Verschlechterung der Scores um rund einen Prozentpunkt. Die einzige Verbesserung gibt es beim Raddialog Moers um mehr als drei Punkte, die größte Verschlechterung beim Bürgerhaushalt Bonn 2017 um mehr als fünf Prozentpunkte.

Eine andere getestete Variante ist ein hierarchischer Ansatz. Es wird zunächst ein Klassifikator trainiert, der die einfachere, binäre Frage beantworten soll, ob ein Beitrag zu der Kategorie *Sonstiges* gehört oder nicht. Wenn er nicht zu *Sonstiges* gehört, nimmt ein weiterer Klassifikator, der auf allen Beiträgen, die nicht zu *Sonstiges* gehören, trainiert worden ist, eine Kategorisierung in die übrigen Kategorien vor. Als Klassifikator wurden jeweils logistische Regressionen ohne die Dokumentfrequenz-Features verwendet.

Mit dieser Variante werden die Ergebnisse allerdings nur noch schlechter im Vergleich zur Logit-Baseline, durchschnittlich um fast zwei Prozentpunkte. Die größte Verschlechterung um mehr als sechs Prozentpunkte gibt es beim Nahverkehrsplan Ulm.

Dieses schlechtere Ergebnis motiviert, einen Blick auf die Klassifikationsperformance für das binäre Klassifikationsproblem für die Labels $\{\text{Sonstiges}, \overline{\text{Sonstiges}}\}$ zu werfen. Dort ergeben sich bei den meisten Datensätzen Recalls von unter 20 %, zum Teil auch 0 %, nur beim Nahverkehrsplan Ulm und beim Leitbild Bad Godesberg ergeben sich höhere Recalls von 63 % bzw. 40 %. Bei diesen beiden Datensätzen ist *Sonstiges* im Gegensatz zu den anderen Datensätzen allerdings auch eine größere Kategorie mit mehr als hundert Beiträgen.

Die extrem niedrigen Recalls sind insofern überraschend, als dass bei der Klassifikation mit allen Kategorien höhere Recalls erreicht werden. Das könnte darauf zurückzuführen sein, dass sich – wie schon oben beschrieben – *Sonstiges* vor allem durch Abwesenheit anderer Kategorien auszeichnet, sodass die Labels für andere Kategorien auch dann relevant sind, wenn nur *Sonstiges* klassifiziert werden soll.

Schließlich wurde noch getestet, ob es möglich ist, *Sonstiges* darüber zu erkennen, dass die Wahrscheinlichkeit der wahrscheinlichsten Kategorie unterhalb eines Schwellwertes θ liegt. Dazu wurden der Klassifikator auf der Trainingsmenge ohne die Kategorie *Sonstiges* trainiert und unterschiedliche θ getestet.

Auch dieses Verfahren liefert allerdings keine guten Ergebnisse. Wird θ groß gewählt, lässt sich natürlich der Recall für *Sonstiges* erhöhen, aber gleichzeitig werden die Precision und die Werte der übrigen Kategorien schlechter. Das liegt unter anderem daran, dass die wahrscheinlichsten Kategorien oft, aber nicht immer, nah beieinanderliegende Wahrscheinlichkeiten erhalten, sodass eine Schwellwertbestimmung schwierig ist. Weiterhin ist die geeignete Wahl von θ auch stark datensatzabhängig, sodass das Verfahren in

der Praxis schwierig einsetzbar ist. Gründe dafür sind unterschiedliche Datensatzgrößen und unterschiedliche Anzahlen von Kategorien.

Fazit: Keiner der betrachteten Änderungsvorschläge zur besseren Erkennung von *Sonstiges* konnte verbesserte Klassifikationsergebnisse liefern.

3.7.6 Fokus auf Titel

Eine weitere Schwäche des aktuellen Klassifikators ist, dass er nicht die Regel umsetzt, dass im Titel genannte Themen wichtiger sind als Themen, die im restlichen Beitragstext angesprochen werden. Daher wurden mehrere Möglichkeiten untersucht, mit denen der Klassifikator einen größeren Fokus auf den Titel legen soll.

Eine einfache Möglichkeit, um dem Titel mehr Gewicht zu geben, ist den Titel mehrfach in die BOW-Darstellung aufzunehmen. Beispielweise dient als Eingabe nicht mehr die einfache Konkatenation *Titel Inhalt*, sondern *Titel Titel Inhalt*.

Durch diese Methode konnten die F_1-Scores in den meisten Fällen, durchschnittlich um fast einen Prozentpunkt, verbessert werden. Eine Verbesserung um mehr als zwei Prozentpunkte konnte beim Raddialog Köln und den Bürgerhaushalten Bonn, Bonn 2017 und Köln 2013 festgestellt werden. Überraschenderweise gibt es bei allen einzelnen Raddialogen Verbesserungen, im Gesamtdatensatz aber keine Verbesserung gegenüber der Baseline. Die größte negative Veränderung mit knapp zwei Prozentpunkten Unterschied gibt es beim Leitbild Bad Godesberg.

Außerdem wurde getestet, wie sich die Ergebnisse ändern, wenn der Titel dreimal (anstatt nur zweimal) vor dem Inhalt eingefügt wird. Die Ergebnisse werden dadurch wieder etwas schlechter, aber sind im Durchschnitt weiterhin leicht besser als die Baseline.

Weiterhin wurde geprüft, ob es funktioniert ein Ensemble aus zwei Klassifikatoren C_t und C_c zu bilden. C_c bekommt wie bisher Titel und Inhalt als Eingabe, C_t nur den Titel. Aus den Ausgaben der beiden Klassifikatoren wird dann durch Soft-Voting das endgültige Klassifikationsergebnis gebildet, was im Wesentlichen bedeutet, dass die berechneten Wahrscheinlichkeiten für die einzelnen Kategorien gemittelt werden.

Dieses Verfahren verschlechtert allerdings die Baseline-Ergebnisse um mehr als anderthalb Prozentpunkte, im extremsten Fall um mehr als sieben Prozentpunkte beim Nahverkehrsplan Ulm. Die größte Verbesserung mit mehr als vier Prozentpunkten gibt es beim Bürgerhaushalt Köln 2012. Wegen der zum Teil drastischen negativen Auswirkungen wird dieses Verfahren nicht weiter verwendet.

Als alternativer Ansatz zum Soft-Voting wurde auch Stacking getestet. Die Grundidee des verwendeten Stackings ist die Benutzung der vorhergesagten Wahrscheinlichkeiten von einem oder mehreren Basisklassifikatoren C_B als Meta-Features für einen Meta-Klassifikator C_M. Im Gegensatz zum Voting soll so gelernt werden, wie die Predictions der Basisklassifikatoren miteinander kombiniert werden müssen, um gute Ergebnisse zu erzielen. Stacking mit Cross-Validierung funktioniert wie folgt (Aggarwal, 2015, 498 ff.):

1. Für jeden Cross-Validation-Split T_i, V_i der Trainingsmenge T:
   (a) Trainiere C_B auf T_i.

   (b) Bestimmte mit C_B Klassenwahrscheinlichkeiten für jedes Element von V_i und füge diese in die Trainingsmenge T_M für den Meta-Klassifikor ein.

2. Trainiere C_M auf T_M.

3. Trainiere C_B erneut auf T.

Der so trainierte Klassifikator klassifiziert ein neues Sample, indem zuerst C_B angewendet wird und dessen Ausgaben anschließend in C_M gegeben werden, welcher das endgültige Klassifikationsergebnis ausgibt. Der erste Trainingsschritt mit Cross-Validation ist sinnvoll, da das Weglassen potentiell zu Overfitting führen kann, wenn die Trainingsmengen für C_B und die von C_B benutzen Samples für die Inputvorbereitung für C_M dieselben sind.

Bei den durchgeführten Tests wurden dieselben Basisklassifikatoren wie beim Voting, logistische Regression als Metaklassifikator und zweifache Kreuzvalidierung verwendet.

Die Ergebnisse mit Stacking sind bei allen Datensätzen deutlich schlechter als die der Baseline. Im Durchschnitt werden die F_1-Scores um 18 Prozentpunkte schlechter.

Fazit: Die doppelte Aufnahme des Titels verbessert die Ergebnisse leicht und wird daher für den finalen Klassifikator übernommen. Soft-Voting und Stacking werden nicht weiterverwendet.

3.7.7 Ensemble-Techniken

Ensemble-Methoden kombinieren die Predictions von Basisklassifikatoren mit dem Ziel, die Generalisierung und die Robustheit des Klassifikators zu verbessern.

Bagging (Breiman, 1996) ist eine Methode, um die Varianz eines Klassifikators durch Hinzufügen von Randomisierung zu verringern. Dies wird dadurch erreicht, dass mehrere Kopien des Klassifikators auf zufällig mit Zurücklegen gebildeten Teilmengen der Trainingsmenge trainiert werden und das Gesamtergebnis der Klassifikation durch Voting bestimmt wird.

Das Bagging wurde mit zehn logistischen Regressionen und randomisierten Trainingsmengen, die jeweils genauso viele Elemente wie die ursprüngliche Trainingsmenge haben, durchgeführt. Die Ergebnisse wurden durch Bagging im Durchschnitt weder besser noch schlechter. Beim Bürgerhaushalt Bonn und dem Nahverkehrsplan Ulm verschlechterten sich die F_1-Scores um knapp einen Prozentpunkt, beim Bürgerhaushalt Köln 2013 wurden sie um mehr als 2 Prozentpunkte besser.

Eine Variante des Baggings sind Random Patches (Louppe und Geurts, 2012), bei denen nicht nur die Trainingsmengen zufällig zusammengestellt werden, sondern auch in jeder erstellten Trainingsmenge nur ein zufällig gewählter Teil der Features benutzt wird, d. h. es werden zufällig Dimensionen des Feature-Raums entfernt.

Die Verwendung von Random Patches führt zu ähnlichen Ergebnissen wie Bagging, bringt also im Durchschnitt keine relevanten Verbesserungen.

Eine andere getestete Ensemble-Technik ist Boosting, wobei hier AdaBoost-SAMME (Hastie et al., 2009) verwendet worden ist. Die Grundidee hinter Boosting ist, dass der

Klassifikator zunächst auf der Trainingsmenge gefittet wird, und danach weitere Klassifikatoren auf derselben Menge gefittet werden. Dabei werden jedoch die Samples, die von den vorherigen Klassifikatoren missklassifiziert worden sind, höher gewichtet.

Durch die Verwendung von AdaBoost mit 50 Klassifikatoren ergibt sich ebenfalls im Durchschnitt keine Verbesserung. Auffällig ist jedoch eine starke Verschlechterung des F_1-Werts beim Nahverkehrsplan Ulm um rund 16 Prozentpunkte. Bei diesem Datensatz ist vor allem die Precision stark gesunken, nämlich um fast 20 Prozentpunkte auf 42 %.

Weiterhin wurde Voting getestet, was bereits in Abschnitt 3.7.6 erklärt worden ist. Dabei wurden als Basisklassifikatoren die optimierten Machine-Learning-Pipelines mit k-NN und logistischer Regression verwendet.

Mit Voting gibt es außer beim Bürgerhaushalt Bonn 2015 nur Verschlechterungen, die durchschnittlich bei ungefähr vier Prozentpunkten liegen.

Außerdem wurde Voting mit zwei anderen Basisklassifikatoren getestet, nämlich mit der bisherigen logistischen Regression mit Charakter-N-Grammen und einer logistischen Regression, die tf-idf-gewichtete Wort-Unigramme erhält. Diese Variante ist deutlich besser als die vorherige, liefert im Durchschnitt aber weiterhin Ergebnisse, die einen Prozentpunkt schlechter sind als die der Baseline.

Fazit: Weder Bagging, Boosting, Random Patches noch Voting haben die F_1-Werte verbessert.

3.7.8 Embeddings als zusätzliches Feature

Die im Abschnitt 3.5 verwendeten Word-Embeddings können auch als Eingabe für ein klassisches Machine-Learning-Verfahren verwendet werden. Da ähnliche Begriffe ähnliche Vektorrepräsentationen haben, können so prinzipiell Synonyme, Hyperonyme etc. auch dann erkannt werden, wenn der Klassifikator die entsprechenden Begriffe nicht aus den Trainingsdaten kennt.

Für die Tests wurde das oben beschriebene Common-Crawl-Embedding mit 300 Dimensionen benutzt. Es wurde jeweils das durchschnittliche Embedding der Wörter eines Satzes mit den aus den Charakter-N-Grammen erstellen Features konkateniert.

Fazit: Mit dem zusätzlichen Embedding-Feature ändern sich die F_1-Werte im Durchschnitt fast gar nicht. Beim Nahverkehrsplan Ulm gibt es eine Verschlechterung um knapp drei Prozentpunkte, beim Raddialog Köln eine Verbesserung um knapp 3 Punkte. Daher wird dieses Feature nicht weiterverwendet.

3.7.9 Dimensionsreduktion

Mithilfe von Dimensionsreduktionsverfahren lassen sich Overfitting reduzieren und auch latente Konzepte in Textdaten finden. Es wurde Dimensionsreduktion über Singulärwertzelergung, wie sie bei LSA verwendet wird (vgl. Abschnitt 5.1.2), benutzt. Getestet wurde sowohl eine Ergänzung der N-Gramm-Features mit den Features aus der Dimensionsreduktion, als auch die alleinige Verwendung letzterer.

Bei einer Reduktion auf 1000 Dimensionen[19] erhält man mit den kombinierten Features durchschnittlich etwa einen halben Prozentpunkt bessere Ergebnisse. Dabei schwanken die Änderungen zwischen mehr als drei Prozentpunkte Verschlechterung bei den Raddialogen und knapp vier Prozentpunkte Verbesserung beim Raddialog Köln. Werden nur die LSA-Features benutzt, sind die Ergebnisse leicht schlechter, aber immer noch leicht besser als die Baseline, jedoch auch nicht in einem signifikanten Bereich.

Verwendet man eine Reduktion auf 500 Dimensionen, werden die Ergebnisse leicht schlechter. Werden statt Charakter-N-Grammen Wort-Unigramme für die Dimensionsreduktion genutzt, so sind die Ergebnisse um mehr als ein Prozentpunkt schlechter als bei der Baseline.

Fazit: Auch die hier untersuchte Dimensionsreduktion eignet sich also nicht zur Verbesserung des Klassifikators.

3.7.10 Zusammenfassung

Die allermeisten der hier diskutierten Verbesserungsvorschläge konnten keine Verbesserung der Cross-Validation-Scores erzielen. Dazu zählen das Entfernen von Straßennamen und Sonderzeichen, Hinzufügen verwandter Wörter, Augmentierung mit Suchmaschinenergebnissen, besondere Maßnahmen zur Erkennung von *Sonstiges*, Ensemble-Techniken, Nutzung von Embeddings sowie Dimensionsreduktionsverfahren. Nur durch die Dopplung des Titels konnten die Ergebnisse leicht verbessert werden, sodass dieser Vorverarbeitungsschritt für die finale Evaluation des Klassifikators verwendet wird.

3.8 Finale Evaluation

Um beurteilen zu können, wie gut die entwickelte Klassifikationspipeline auf unbekannten Daten funktioniert, werden nun die Ergebnisse auf dem bisher nicht verwendeten Test-Datensatz angeschaut.

Wie Tabelle 18 entnommen werden kann, sind die Ergebnisse auf dem Testdatensatz im Durchschnitt vergleichbar mit denen der Cross-Validation der optimierten Logit-Pipeline (Tabelle 7). An der Accuracy (Acc) ist ablesbar, dass im Durchschnitt über alle Datensätze mehr als 72 % der Textbeiträge die richtige Kategorie zugeordnet bekommen haben.

Der um mehr als zehn Prozentpunkte schlechtere F_1-Score bei dem Raddialog Moers im Vergleich zum Cross-Validation-Score ist darauf zurückzuführen, dass im Testdatensatz die kleinste Kategorie *Fahrradparken* vollständig fehlt und mit einem Wert von 0 in den Score eingeht. Der Klassifikator liegt bei den anderen Raddialogen nur wenige Prozentpunkte unter der menschlichen Baseline.

Um dieses Problem bei der Auswertung abzuschwächen, wurden die Metriken noch einmal auf leicht reduzierten Datensätzen getestet, bei denen die Kategorien eine Mindestanzahl von 5 Beiträgen benötigen, um in Trainings- und Testmenge aufgenommen zu werden. Dadurch erhält man beim Raddialog Moers, beim Bürgerhaushalt Bonn 2015 und dem Leitbild Bad Godesberg wesentlich bessere F_1-Werte.

[19] Zur Orientierung: Bei den Raddialogen hat der N-Gramm-Featureraum ca. 15.000 Dimensionen.

Mindestklassengröße	1 Beitrag				5 Beiträge			
Klassifikator	Logit		Human		Logit		Human	
Datensatz	F_1	Acc	F_1	Acc	F_1	Acc	F_1	Acc
Raddialoge	68	74	74	77	68	74	74	77
Bonn	69	73	74	76	69	73	74	76
Köln	66	71	68	78	71	72	78	80
Moers	57	75	82	81	71	81	86	81
Bürgerhaushalt								
Bonn	64	70			64	70		
Bonn 2011	66	72			66	72		
Bonn 2015	39	51			47	53		
Köln								
Köln 2012	76	77			76	77		
Köln 2013	65	83			66	83		
Köln 2015	71	81			71	81		
Köln 2016	66	87			66	87		
Mängelmelder Braunschweig	85	92			89	92		
Nahverkehrsplan Ulm	59	63			59	63		
Bad Godesberg	46	49			52	52		

Tabelle 18: Ergebnisse des finalen Logit-Klassifikators auf den unterschiedlichen Test-Datensätzen

Grundsätzlich lässt sich beobachten, dass die F_1-Scores für die kleineren Datensätze (Bonn 2015 und Bad Godesberg) schlechter sind als die der größeren Datensätze, was somit bedeutet, dass eine gewisse Mindestmenge an Trainingsdaten vorhanden sein muss, damit der Klassifikator gut funktioniert. Außerdem funktioniert die Klassifikation besser, je weiter die Themen der verschiedenen Kategorien auseinanderliegen. Beim Nahverkehrsplan Ulm sind sich viele Kategorien sehr ähnlich, wie z. B. *Bedienungshäufigkeit* und *Fahrtangebot*, sodass es eine große Überlappung bei den verwendeten Wörtern gibt. Dadurch sind BOW-Ansätze nicht ausreichend, um die Kategorie sicher bestimmen zu können.

Wie auch schon in den bereits gezeigten Konfusionsmatrizen abgelesen werden kann, haben Kategorien mit charakteristischem Vokabular, das sie ziemlich eindeutig von anderen Kategorien eines Datensatzes abgrenzt, einen höheren Recall. Darunter fallen z. B. die Kategorien *Beleuchtung* und *Fahrradparken* der Raddialoge mit Recalls über 90 % oder auch *Friedhofsunterhaltung* beim Mängelmelder. Dagegen ist, wie bereits in Abschnitt 3.7.5 beschrieben, die Erkennung von Sonstiges schwierig, weil es hier kein typisches „Sonstiges-Vokabular“ gibt.

3.9 Praxisrelevante Experimente

Abschließend werden noch Experimente beschrieben, die für den Praxiseinsatz relevante Erkenntnisse liefern.

Trainingsdatensätze	Testdatensatz	F_1-Score	Test-F_1-Score	Human F_1
Rd. Bonn	Rd. Köln	64	66	61
Rd. Bonn	Rd. Moers	70	57	77
Rd. Moers	Rd. Bonn	62	69	74
Rd. Moers, Rd. Köln	Rd. Bonn	65	69	74
Bonn 2011	Bonn 2015	55	39	
Bonn 2011, 2015	Bonn 2017	43	n/a[20]	
Köln 2013	Köln 2015	61	71	
alle außer Testdatensatz	Rd. Bonn	65	69	74
alle außer Testdatensatz	Rd. Köln	71	66	61
alle außer Testdatensatz	Rd. Moers	72	57	77

Tabelle 19: Ergebnisse bei Training auf einem anderen Datensatz im Vergleich zum Test-F_1-Score und Human

3.9.1 Training auf altem Datensatz

Bei wiederkehrenden Online-Partizipationsverfahren mit gleichen Schlagwörtern bietet es sich an, einen Klassifikator auf einem alten Datensatz zu trainieren. Dann kann dieser Klassifikator benutzt werden, um von Anfang an den Benutzern der Online-Plattform eine oder mehrere passende Kategorien vorzuschlagen oder sie auf eine möglicherweise falsch gewählte Kategorie hinzuweisen. Dafür wurden verschiedene Kombinationen von Trainings- und Test-Datensätzen verschiedener Verfahren mit gleichen Kategorien getestet.

Außerdem wurde getestet, wie gut die Klassifikationsergebnisse sind, wenn als Trainingsgrundlage alle Datensätze außer dem Testdatensatz dienen. Dieses Experiment ist für den Praxiseinsatz insofern relevant, als dass dann nur ein großer Klassifikator trainiert werden muss und nicht für jede Kategorienmenge ein eigener.

In allen Fällen wurden die Predictions des Klassifikators dann so angepasst, dass nur für diejenigen Kategorien Predictions gemacht werden, die auch im Testdatensatz vorkommen. Da dazu in der Trainingsmenge bereits die Kategorien vorkommen müssen, die im Testdatensatz verwendet werden, sind hier mit den verfügbaren Datensätzen nur wenige Kombinationen verfügbar.

Die getesteten Kombinationen sind in Tabelle 19 aufgeführt, wobei jeweils auch die F_1-Scores auf dem Testset des jeweiligen Testdatensatzes aus Abschnitt 3.8 und die menschliche Baseline zum Vergleich angegeben sind. Zu beachten ist, dass die Scores auf dem Testset nur auf selbigen und nicht den gesamten Datensatz berechnet worden sind, also nur eine grobe Orientierung liefern können, ob die Ergebnisse durch Verwendung einer anderen Trainingsgrundlage besser oder schlechter werden.

Grundsätzlich sind die Ergebnisse beim Training auf anderen Datensätzen mit denselben Labels ähnlich zu denen der Cross-Validation und durchschnittlich auf einem Niveau, das vergleichbar mit denen der Benutzer ist. Die Falschklassifizierungen weisen dieselben Muster auf, die bereits diskutiert worden sind; beispielsweise werden wieder häufig

[20] wegen der geringen Anzahl von Beiträgen wurde der Datensatz Bonn 2017 nicht auf sich selbst evaluiert.

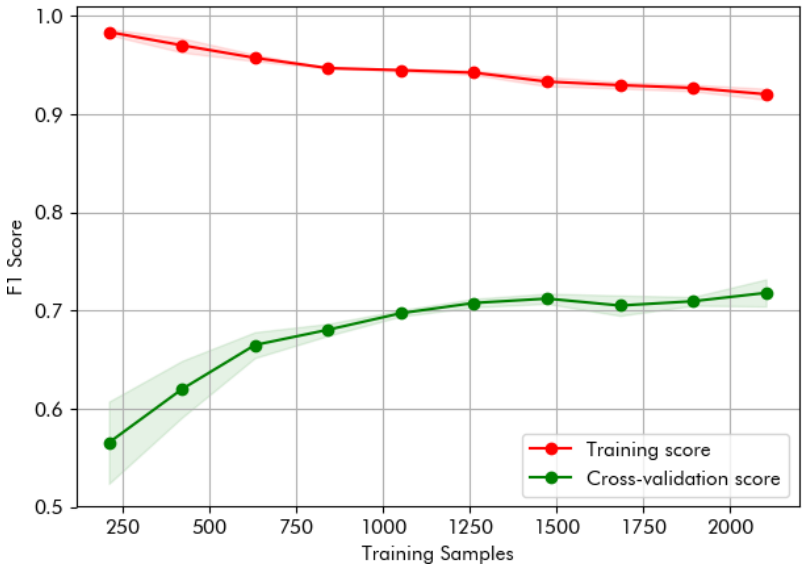


Abbildung 19: Lernkurve für die Raddialoge

Beschilderung und *Radverkehrsführung* verwechselt und *Sonstiges* hat einen schlechten Recall.

Werden in der Trainingsmenge zusätzlich auch Datensätze mit für den Testdatensatz irrelevanten Kategorien aufgenommen, so hat dies keine negativen Auswirkungen bei den getesteten Kombinationen und wirkt sich zum Teil sogar leicht positiv auf den Recall bei der Kategorie *Sonstiges* aus. Es wäre im Praxiseinsatz also potentiell möglich, einen einzelnen großen Klassifikator für verschiedene Partizipationsverfahren zu verwenden.

3.9.2 Betrachtung der Lernkurve

Es ist für den Praxiseinsatz interessant zu wissen, wie viele Textbeiträge mindestens in der Trainingsmenge sein müssen, um gute Ergebnisse erzielen zu können. Anhand einer Lernkurve kann festgestellt werden, welche Scores mit wie vielen Trainingsdaten erzielt werden können.

Für die großen Datensätze Raddialoge, Bürgerhaushalt Bonn, Mängelmelder Braunschweig und Nahverkehrsplan Ulm wurden die Lernkurven mit dreifach Cross-Validation berechnet. Die Lernkurve für die Raddialoge ist in Abbildung 19 zu sehen, die übrigen sind im Anhang in den Abbildungen 24, 25 bzw. 26.

Die F_1-Werte der Cross-Validation aller Lernkurven liegen unter den Werten auf dem Trainingsdatensatz. Dies kann ein Zeichen für Overfitting sein. Das „Schließen“ der Lücke bei flacher Cross-Validation-Kurve könnte nur mit sehr viel mehr Trainingsdaten erfolgen.

Bei den Raddialogen ist ab etwa 1250 Beiträgen im Trainingsdatensatz keine Verbesserung des Cross-Validation-Wertes mehr zu erkennen, d. h. wenige weitere Beiträge werden wahrscheinlich die Klassifikationsergebnisse mit der gewählten Klassifikationspipeline nicht verbessern. Beim Bürgerhaushalt Bonn tritt dieser Effekt bei etwa 500 Beiträgen

Datensatz	p	$i = 0$	1	2	3	Test-F_1
Raddialoge	50 %	**68**	67	67	67	68
	70 %	64	**67**	**67**	66	68
	90 %	59	62	**63**	**63**	68
Bürgerhaushalt Bonn	50 %	58	**60**	59	59	64
	70 %	53	**57**	55	**57**	64
	90 %	41	**49**	47	47	64
Mängelmelder Braunschweig	50 %	**85**	84	**85**	**85**	85
	70 %	**82**	**82**	81	81	85
	90 %	73	**76**	**76**	75	85
Nahverkehrsplan Ulm	50 %	55	57	**58**	**58**	59
	70 %	51	**57**	55	56	59
	90 %	**29**	26	26	28	59

Tabelle 20: F_1-Scores bei Verwendung von semi-supervised Learning nach Xuan et al. (2017) im Vergleich zu den F_1-Werten auf der Testmenge

ein. Sowohl beim Mängelmelder Braunschweig als auch beim Nahverkehrsplan Ulm ist keine so eindeutige Sättigung zu sehen, sodass hier mehr Trainingsdaten zu einer Verbesserung der Ergebnisse führen können.

3.9.3 Semi-supervised Learning

Wenn im Nachhinein manuell Textbeiträge mit Kategorien versehen werden, gibt es die Situation, dass man eine nur teilweise gelabelte Menge von Texten hat. Wie in Abschnitt 3.9.2 gezeigt, kann man bereits mit einer Teilmenge an gelabelten Daten gute Ergebnisse erzielen, aber in der beschriebenen Situation könnte man auch die noch nicht kategorisierten Texte nutzen, um das Klassifikationsergebnis zu verbessern.

Xuan et al. (2017) haben einen semi-supervised Ansatz vorgeschlagen, um Klassifikationsergebnisse zu verbessern, wenn nur eine Teilmenge T_l der möglichen Trainingsdaten gelabelt ist. Der Ansatz funktioniert wie folgt:

1. Auf der gelabelten Trainingsmenge T_l wird ein Klassifikator C trainiert.
2. Wiederhole solange sich C verbessert:
   (a) Wende C auf die ungelabelte Trainingsmenge T_u an, um die gelabelte Menge T'_u zu erhalten.
   (b) Trainiere C auf $T_l \cup T'_u$.

Dieses Verfahren wurde auf den Raddialogen, dem Bürgerhaushalt Bonn, dem Mängelmelder Braunschweig und dem Nahverkehrsplan Ulm getestet. Dabei wurden $i \in \{0; 1; 2; 3\}$ Iterationen durchgeführt und $p \in \{50\,\%; 70\,\%; 90\,\%\}$ des Trainingsdatensatzes ungelabelt gelassen.

Datensatz	p	$\theta = 10\,\%$	$20\,\%$	$30\,\%$	$40\,\%$	$50\,\%$	$\theta = 0$	Test-F_1
Raddialoge	$50\,\%$	**68**	67	67	67	**68**	67	68
	$70\,\%$	64	**67**	**67**	66	64	**67**	68
	$90\,\%$	59	62	**63**	**63**	59	62	68
Bh. Bonn	$50\,\%$	58	**60**	59	59	58	**60**	64
	$70\,\%$	53	**57**	55	**57**	53	**57**	64
	$90\,\%$	41	**49**	47	47	41	**49**	64
Mm. Braunschweig	$50\,\%$	**85**	84	**85**	**85**	**85**	84	85
	$70\,\%$	**82**	**82**	81	81	**82**	**82**	85
	$90\,\%$	73	**76**	**76**	75	73	**76**	85
Np. Ulm	$50\,\%$	55	57	**58**	**58**	55	57	59
	$70\,\%$	51	**57**	55	56	51	**57**	59
	$90\,\%$	**29**	26	26	28	**29**	26	59

Tabelle 21: F_1-Scores bei Verwendung von semi-supervised Learning mit Wahrscheinlichkeits-Schwellwert für $i = 1$ im Vergleich zum F_1-Score auf dem Testset

In Tabelle 20 sind die Ergebnisse mit semi-supervised Learning im Vergleich zu den F_1-Scores auf der Testmenge dargestellt. Wie zu erwarten sind die F_1-Scores ohne supervised-Learning ($i = 0$) geringer als die Test-Scores, da nur ein Teil des Datensatzes benutzt wird. Dies entspricht den Erkenntnissen, die aus der Betrachtung der Lernkurven gezogen werden konnten.

Wenn 90 % des Datensatzes ungelabelt sind, verbessert das semi-supervised Verfahren bei den ersten drei getesteten Datensätzen die F_1-Scores um drei bis acht Prozentpunkte nach einer oder zwei Iterationen. Nur beim Nahverkehrsplan Ulm zeigt sich ein anderes Bild, denn dort werden die Ergebnisse nur für eine größere gelabelte Datenmenge besser, bei nur 10 % gelabelter Daten schlechter.

Mit dem Ziel, diese Ergebnisse zu verbessern, wurde eine Abwandlung des Algorithmus getestet, bei der C nicht auf $T_l \cup T'_u$ trainiert wird, sondern nur diejenige Teilmenge von T'_u zu T_l hinzugefügt, bei der Wahrscheinlichkeiten für die Predictions des Klassifikators oberhalb eines Schwellwertes $\theta \in [0, 1]$ liegen. Damit soll verhindert werden, dass der Klassifikator auf falsch klassifizierten Textbeiträgen lernt.

Wie Tabelle 21 entnommen werden kann, hat diese Änderung keine wesentlichen Verbesserungen im Vergleich zur Verwendung keines Schwellwertes ($\theta = 0$) zur Folge. Einzig beim Nahverkehrsplan Ulm werden die Ergebnisse um rund drei Prozentpunkte bei 90 % ungelabelten Daten leicht besser. Allerdings sind bei diesem Datensatz auch die F_1-Scores mit so viel ungelabelten Daten sehr weit von denen des supervised Learning entfernt, sodass diese Methode nicht sinnvoll anwendbar ist.

Zusammenfassend ist es mit semi-supervised Verfahren bei Anwendung einer Iteration oft, wenn auch nicht immer möglich, die Ergebnisse zu verbessern, wenn nur etwa ein Zehntel der Daten gelabelt ist.

3.10 Zusammenfassung

In diesem Kapitel wurde das Kategorisieren von Textbeiträgen als überwachtes Multiclass-Problem betrachtet. Von den verschiedenen getesteten Klassifikatoren hat logistische Regression die besten Ergebnisse im Vergleich zu anderen klassischen Verfahren, neuronalen Netzen und Graph-basierten Methoden erzielt.

Auf den getesteten Datensätzen wurden verschiedene Vorverarbeitungsschritte getestet. Filterung nach minimaler und maximaler Dokumentfrequenz, td-idf-Gewichtung, Zerlegung von Komposita, Beschränkung auf Adjektive, Nomen, Verben und Eigennamen sowie die Bildung von Charakter-N-Grammen haben die Ergebnisse verbessert. Das Verwenden von Wort-N-Grammen hatte stark unterschiedliche Ergebnisse auf den verschiedenen Datensätzen erzielt.

Eine vorgeschaltete Rechtschreibkorrektur, Normalisieren von Straßennamen, Ziffern und Sonderzeichen, Augmentieren mit Web-Daten, verschiedene Ensemble-Methoden und das Verwenden von Word-Embeddings hatten keinen wesentlichen Einfluss auf die Performance des Klassifikators. Negative Auswirkungen wurden bei Stemming, Lemmatisierung und dem Anreichern mit Synonymen, Ober- und Unterbegriffen festgestellt.

Die finale Preprocessing-Pipeline für die logistische Regression besteht aus doppelter Gewichtung des Titels, Random Oversampling, Bildung von Charakter-3- und -4-Grammen mit minimaler Dokumentfrequenz von 2 und maximaler Dokumentfrequenz von 50 % und tf-idf-Gewichtung.

Insgesamt konnte ein Klassifikator entwickelt werden, der gute Ergebnisse liefert, die vergleichbar mit denen eines ungeschulten Menschen sind. Im Praxiseinsatz ist es außerdem möglich, einen Klassifikator zu benutzen, der auf einer Vielzahl von Datensätzen mit unterschiedlichen Labels trainiert worden ist, die eine Obermenge der relevanten Kategorien darstellen. Grundsätzlich muss aber eine datensatzabhänge Mindestmenge an Samples pro Kategorie vorhanden sein, um gute Klassifikationsergebnisse zu erzielen.

Weiterhin ist es sinnvoll, im Klassifikator das Regelwerk zur Klassifizierung der Textbeiträge, das in der finalen Auswertung des Partizipationsverfahrens benutzt wird, widerzuspiegeln. Beispielsweise führte eine doppelte Gewichtung des Titels zu besseren Ergebnissen. Weiterführend könnte hier auch untersucht werden, ob das Erkennen von Vorschlägen und Zustandsbeschreibungen in den Textbeiträgen möglich ist, um auch ein höheres Gewicht auf Vorschläge legen zu können, wie es im Regelwerk für die Kategorisierung der Raddialog-Beiträge vorgesehen ist.

Darüber hinaus gibt es noch bei der Erkennung der Kategorie *Sonstiges*, die nur schwer mit BOW-Features erfasst werden kann, sowie bei der Unterscheidung von Kategorien mit ähnlichem Vokabular Verbesserungsbedarf. Letzteres erfordert wahrscheinlich komplexere Modelle, die besser den Sinn eines Satzes begreifen können. Geeignete neuronale Netze könnten hier hilfreich sein. Obwohl die ersten Experimente mit neuronalen Netzen in dieser Arbeit eher negative Ergebnisse gezeigt haben, könnten mit größeren Datensätzen und weiterer Hyperparameteranpassung eventuell gute Ergebnisse erzielt werden. Insbesondere könnte mit Word-Embeddings experimentiert werden, die für jedes Partizipationsverfahren speziell vortrainiert worden sind.

Sofern eine Methode auf ein spezielles Online-Partizipationsverfahren angepasst werden soll, könnten noch weitere verfügbare Daten neben dem Text des eigentlichen Beitrags verwendet werden. Darunter fielen beispielsweise die Einbeziehung von Kommentaren, Bildern, Koordinaten und das verwendete Unterforum. Dabei muss beachtet werden, dass z. B. Moderationskommentare, in der über eine geänderte Kategorie geschrieben wird, nicht verwendet werden.

4 Multi-Label-Klassifikation

In diesem Abschnitt geht es um das Zuweisen von Schlagwörtern zu Textbeiträgen, wobei jeder Beitrag ein oder mehrere Schlagwörter erhalten kann. Zum Testen der automatischen Verschlagwortung mittels supervised Learning stehen die Raddialog-Datensätze, der Bürgerhaushalt Bonn 2011 sowie das Bürgerbudget Wuppertal zur Verfügung. Bei den Raddialogen sind die Schlagwörter zusätzlich zusammen mit den Kategorien Teil einer zweischichtigen Hierarchie (vgl. Tabelle 33 im Anhang), sodass hier die Möglichkeit besteht, die Verschlagwortung mit Verfahren vorzunehmen, die die Hierarchieinformation ausnutzen.

Zunächst werden im Folgenden die verwendeten Evaluationsmaße für Mehrfachverschlagwortung definiert. Anschließend werden verschiedene Multi-Label-Verfahren vorgestellt und evaluiert und danach Hierarchie-basierte Verfahren betrachtet.

4.1 Verwendete Evaluationsmaße

Anders als bei der Single-Label-Klassifkation können bei der Multi-Label-Klassifikation Predictions nicht nur richtig oder falsch sein, sondern auch teilweise richtig oder unvollständig. Daher werden in diesem Kapitel andere Evaluationsmetriken verwendet. Die verwendeten Definitionen basieren auf einer Arbeit von Sorower (2010).

Im Folgenden seien L die Menge aller Schlagwörter, $k = |L|$, n die Anzahl der Testsamples, $Y \subseteq \{0,1\}^{n \times k}$ die Ground Truths, $Z \subseteq \{0,1\}^{n \times k}$ die Predictions des Klassifikators und I die Indikatorfunktion.

Eine einfache Möglichkeit, mit teilweise richtigen Predictions umzugehen, ist das Behandeln von teilweise richtigen Predictions als falsche Predictions. Damit lässt sich ein scharfes Maß definieren, welches Exact Match Ratio (MR) genannt wird und eine Erweiterung der Accuracy auf den Multi-Label-Fall darstellt.

$$MR = \frac{1}{n} \sum_{i=1}^{n} I(Y_i = Z_i)$$

Auch Macro-Precision P, -Recall R und F_1-Maß können intuitiv auf den Multi-Label-Fall erweitert werden:

$$P = \frac{1}{k} \sum_{\lambda \in L} \frac{\sum_{i=1}^{n} Y_i^\lambda Z_i^\lambda}{\sum_{i=1}^{n} Z_i^\lambda}$$

$$R = \frac{1}{k} \sum_{\lambda \in L} \frac{\sum_{i=1}^{n} Y_i^\lambda Z_i^\lambda}{\sum_{i=1}^{n} Y_i^\lambda}$$

$$F_1 = \frac{1}{k} \sum_{\lambda \in L} \frac{2 \sum_{i=1}^{n} Y_i^\lambda Z_i^\lambda}{\sum_{i=1}^{n} Z_i^\lambda + \sum_{i=1}^{n} Y_i^\lambda}$$

mit

$$Y_i^\lambda = \begin{cases} 1 & \text{wenn } i\text{-tes Sample Schlagwort } \lambda \text{ hat} \\ 0 & \text{sonst} \end{cases}$$
$$Z_i^\lambda = \begin{cases} 1 & \text{wenn } i\text{-tes Sample Prediction } \lambda \text{ hat} \\ 0 & \text{sonst} \end{cases}$$

Wie auch beim F_1-Maß für den Single-Label-Fall wird das so definierte Macro-F_1-Maß mehr von der Performance auf kleineren Klassen beeinflusst.

Eine alternative Definition von Precision und Recall, die z. B. auch von Godbole und Sarawagi (2004) verwendet wird, ist die folgende:

$$P' = \frac{1}{n} \sum_{i=1}^{n} \frac{|Y_i \cap Z_i|}{|Z_i|}$$
$$R' = \frac{1}{n} \sum_{i=1}^{n} \frac{|Y_i \cap Z_i|}{|Y_i|}$$

Im Gegensatz zur ersten Definition ist diese Definition nicht Label-basiert, sondern Sample-basiert. Da aber wie in Kapitel 3 alle Labels gleich gewichtet werden sollen, wird die erste Definition verwendet.

Bei der Evaluation der verschiedenen Verfahren wurde vierfache Kreuzvalidierung verwendet.

4.2 Evaluation von Multi-Label-Verfahren

Zunächst werden Multi-Label-Verfahren betrachtet, die keine Hierarchie-Informationen einbeziehen. Dazu werden als erstes eine einfache Baseline-Methode getestet, danach unterschiedliche Multi-Label-Verfahren vorgestellt und evaluiert. Zuletzt werden die Ergebnisse zusammengefasst, die Verfahren auf dem Testset getestet und die von den Algorithmen gemachten Fehler untersucht.

4.2.1 Single-Tag-Baseline

Als Baseline wird zunächst ein Klassifikator betrachtet, der immer nur genau ein Schlagwort vorhersagt. Für das Training werden dazu für jeden Textbeitrag T mit Schlagwörtern $s_1, \ldots, s_n$ die Trainingsdaten $(T, s_1), \ldots, (T, s_n)$ verwendet, d. h. ein Textbeitrag mit n Schlagwörtern kommt n-mal mit n verschiedenen Schlagwörtern im Trainingsdatensatz vor. Als Klassifikator kommt die Klassifikationspipeline mit logistischer Regression ohne Oversampling aus Kapitel 3 zum Einsatz. Es wurde außerdem getestet, wie sich das Gewichten der Trainingsbeispiele gemäß Klassengröße, also das stärkere Gewichten von Samples kleinerer Klassen, auswirkt.

Gewichtung	✗		✓	
Datensatz	MR	F_1	MR	F_1
Raddialoge	46	31	**48**	**46**
Raddialog Bonn	44	28	**47**	**44**
Raddialog Köln	39	21	**44**	**36**
Raddialog Moers	41	17	**47**	**36**
Bürgerhaushalt Bonn 2011	**6**	4	**6**	33
Bürgerbudget Wuppertal	11	13	**14**	**25**

Tabelle 22: Performance der Single-Tag-Baseline

In Tabelle 22 sind die Exact Match Ratios und der Macro-F_1-Score für diese Baseline zu sehen. Mit Gewichtung der Samples werden immer bessere Ergebnisse erzielt, sodass diese Ergebnisse im Folgenden als Baseline dienen. Auffällig sind die vergleichsweise hohen MRs auf den Raddialog-Datensätzen. Dies ist darauf zurückzuführen, dass bei den Raddialogen die meisten Beiträge genau ein Schlagwort haben (vgl. Abbildung 2 in Abschnitt 2.1). Beim Bürgerhaushalt Bonn 2011 und beim Bürgerbudget Wuppertal liegt der maximal mögliche MR für Predictions mit genau einem Schlagwort bei knapp 6 % bzw. knapp 19 %.

4.2.2 Problemtransformationsverfahren

Problemtransformationsverfahren führen das Multi-Label-Problem auf ein Multi-Class-Problem zurück und trainieren Basisklassifikatoren, die diese Multi-Class-Probleme lösen sollen. Als Basisklassifikator dient hier zunächst immer die logistische Regression aus Kapitel 3.

Beim Label-Powerset-Verfahren (LP) wird jede im Trainingsdatensatz vorkommende Kombination von Schlagwörtern als eine eigene Klasse angesehen. Hat also ein Textbeitrag die Schlagwörter *a* und *b*, dann hat dieser nach der Transformation die Klasse *a_b*. Es gibt somit bis zu 2^k Klassen, wobei die Anzahl der Klassen in der Praxis deutlich kleiner sein dürfte, da nicht alle möglichen Kombinationen von Schlagwörtern vorkommen.

Der Binary-Relevance-Ansatz (BR) trainiert k verschiedene Basisklassifikatoren, die für jedes Schlagwort angeben sollen, ob das Schlagwort bei einem Textbeitrag gesetzt ist oder nicht. Diese Methode kann also Korrelationen zwischen Schlagwörtern nicht berücksichtigen.

Zuletzt wurden Classifier Chains (CCs, Read et al., 2009) betrachtet. Diese funktionieren ähnlich wie BR und trainieren ebenfalls k Klassifikatoren in einer festgelegten Reihenfolge. Diese Klassifikatoren erhalten im Gegensatz zu BR jedoch zusätzlich die Predictions der vorhergehenden Klassifikatoren, sodass Korrelationen zwischen Schlagwörtern berücksichtigt werden können. Beim Training werden nicht die tatsächlichen Predictions eingesetzt, sondern die Ground Truths.

Für alle drei Verfahren wurden die Implementierungen von Szymański und Kajdanowicz (2017) verwendet.

Verfahren	Baseline		Label Powerset				Binary Relevance				Classifier Chain			
Gewichtung	✓		✗		✓		✗		✓		✗		✓	
Datensatz	MR	F_1	MR	F_1	MR	F_1	MR	F_1	MR	F_1	MR	F_1	MR	F_1
Raddialoge	**48**	**46**	44	28	22	29	13	12	27	**45**	24	17	34	42
Rd. Bonn	**47**	**44**	42	27	23	28	10	9	26	42	18	14	33	39
Rd. Köln	**44**	**36**	33	16	25	**37**	5	4	28	**36**	9	6	31	30
Rd. Moers	**47**	**36**	39	15	29	31	8	4	35	33	13	5	36	28
Bonn 2011	6	33	**15**	16	8	32	6	5	14	**40**	7	5	9	31
Wuppertal	14	25	**16**	29	11	**47**	6	23	9	**48**	7	19	7	43

Tabelle 23: Performance der Problemtransformationsverfahren im Vergleich zur Baseline

Wie in Tabelle 23 zu sehen, sind die Baseline-Ergebnisse bei allen Raddialog-Datensätzen am besten. Grund dafür dürfte die Tatsache sein, dass hier keine wirkliche Multi-Label-Verschlagwortung vorliegt, sondern mehr als 83 % der Beiträge genau ein Schlagwort besitzen.

Bei BR und CC hat eine Gewichtung der Samples eine positive Auswirkung auf beide betrachteten Metriken, bei LP verschlechtern sich durch die Gewichtung die MR-Werte, die F_1-Scores werden jedoch besser. Dass sich LP hier anders verhält als die anderen beiden Verfahren liegt wahrscheinlich daran, dass LP das einzige Verfahren ist, das deutlich mehr als k Klassifikatoren verwendet.

Im Allgemeinen lässt sich die Tendenz feststellen, dass die MR-Werte von LP besser sind als die der CC und diese wiederum besser als die der BR-Methode, wenn keine Gewichtung verwendet wird. Jedoch sind auch bei der LP-Methode mit etwa 15 % die MR-Werte nicht sehr hoch, was aber auf die scharfe Metrik zurückzuführen ist. Mit Gewichtung ist es abhängig vom Datensatz, welche Methode besser funktioniert.

Beim F_1-Wert ist LP am besten, wenn keine Gewichtung verwendet wird. Mit Gewichtung liefert BR die besten Ergebnisse.

Sofern kein wirkliches Multi-Label-Problem vorliegt, sollte also direkt auf ein Single-Label-Verfahren zurückgegriffen werden. Dies wurde bei den Raddialogen mit durchschnittlich weniger als 1,2 Schlagwörtern pro Beitrag deutlich. Bei Partizipationsverfahren mit „richtiger“ Mehrfachverschlagwortung ist BR das beste der getesteten Problemtransformationsverfahren.

4.2.3 Adaptierte Verfahren

Nun werden verschiedene Verfahren betrachtet, die speziell an den Multi-Label-Fall angepasst worden sind.

Von Eleftherios Spyromitros (2008) wurde BRkNN vorgestellt, was eine Adaptierung von k-NN für den Multi-Label-Fall ist. Konzeptionell entspricht BRkNN dem Binary-Relevance-Verfahren mit k-NN, vermeidet aber unnötige Wiederholungen der Nachbarnsuche. BRkNN gibt es in zwei Varianten: BRkNN-a ordnet ein Label zu, wenn mindestens die Hälfte der Nachbarn dieses Label hat. Bei BRkNN-b werden zunächst die durch-

Verfahren	Baseline		BRkNN-a		BRkNN-b		ML-k-NN	
Datensatz	MR	F_1	MR	F_1	MR	F_1	MR	F_1
Raddialoge	**48**	**46**	30	29	5	1	32	34
Raddialog Bonn	**47**	**44**	30	29	6	1	31	32
Raddialog Köln	**44**	**36**	22	22	5	1	24	20
Raddialog Moers	**47**	**36**	29	22	3	1	30	25
Bürgerhaushalt Bonn 2011	6	**33**	**15**	27	0	0	**15**	30
Bürgerbudget Wuppertal	**14**	25	8	**39**	0	23	7	**40**

Tabelle 24: Performance der adaptierten Verfahren im Vergleich zur Baseline

schnittliche Anzahl s der Labels der k nächsten Nachbarn bestimmt und danach die s häufigsten Label dieser Nachbarn zugeordnet.

Ein anderer Multi-Label-Algorithmus ist ML-k-NN (Zhang und Zhou, 2007), der ebenfalls grundsätzlich wie k-NN funktioniert, aber die Labels mit Hilfe der Maximum-a-posteriori-Methode zuweist.

Zum Testen dieser Verfahren wurde das für den Single-Label-k-NN als optimal bestimmte Preprocessing aus Abschnitt 3.2.11 verwendet.

Tabelle 24 kann entnommen werden, dass BRkNN-b auf allen Datensätzen sehr schlechte Ergebnisse liefert. Bei den Raddialog-Datensätzen ist wieder die Baseline-Methode am besten und ML-k-NN funktioniert etwas besser als BRkNN-a. Für die letzten beiden Datensätze ist ML-k-NN ebenfalls ein klein wenig besser, sodass dieses Verfahren für echte Mehrfachverschlagwortung insgesamt hier am besten funktioniert.

4.2.4 Ensemble-Methoden

Von Tsoumakas et al. (2011) wurde RAkEL (Random k Labelsets) vorgestellt. Dieser Algorithmus basiert auf dem Grundprinzip der LP-Methode, schwächt aber das Problem ab, dass die Anzahl der Labels sehr groß wird und es oft nur wenige Trainingsbeispiele pro Label gibt. Dazu wird die Menge der Labels in n Labelsets der Größe k zufällig eingeteilt, auf die jeweils die LP-Methode angewendet wird.

Werden die Labelsets überlappend gebildet, wird die Methode RAkEL$_o$ genannt, andernfalls RAkEL$_d$. Bei RAkEL$_o$ werden die Predictions von m Basisklassifikatoren mit Voting kombiniert, um die finale Prediction zu erhalten. Im Gegensatz zur normalen LP-Methode kann RAkEL Schlagwortkombinationen vorhersagen, die im Trainingsdatensatz nicht vorkommen.

Als Basisklassifikator kommt wieder logistische Regression zum Einsatz. Anders als bei der LP-Methode verbessern sich die Ergebnisse, wenn die Samples abhängig von der Klassengröße gewichtet werden, was wahrscheinlich auf die kleinere Anzahl an Klassen gegenüber LP zurückzuführen ist. Für RAkEL$_d$ wurde $k \in \{3, 4, 5, 7\}$ getestet, wobei 5 die besten Ergebnisse geliefert hat. Für RAkEL$_o$ wurde $k = 5$ mit $m \in \{30, 60\}$ getestet. Dabei wurden in beiden Fällen sehr ähnliche Ergebnisse erzielt.

Verfahren	Baseline		BR		BR		RAkEL$_o$		RAkEL$_d$	
Gewichtung			✗		✓		✓		✓	
Datensatz	MR	F_1	MR	F_1	MR	F_1	MR	F_1	MR	F_1
Raddialoge	**48**	**46**	13	12	27	**45**	34	38	33	37
Raddialog Bonn	**47**	**44**	10	9	26	42	30	33	29	33
Raddialog Köln	**44**	**36**	5	4	28	**36**	21	23	22	23
Raddialog Moers	**47**	**36**	8	4	25	33	29	19	24	19
Bürgerhaushalt Bonn 2011	6	**33**	6	5	**14**	40	13	23	**14**	23
Bürgerbudget Wuppertal	**14**	**25**	6	23	9	48	11	42	10	44

Tabelle 25: Performance der RAkEL-Klassifikatoren im Vergleich zur Baseline und BR

θ	Baseline		0,05		0,1		0,15		0,2	
Datensatz	MR	F_1	MR	F_1	MR	F_1	MR	F_1	MR	F_1
Raddialoge	**48**	**46**	0	28	13	22	21	14	9	7
Raddialog Bonn	**47**	**44**	0	27	12	22	19	14	8	7
Raddialog Köln	**44**	**36**	0	26	18	24	21	17	10	10
Raddialog Moers	**47**	**36**	0	20	11	15	26	14	21	12
Bürgerhaushalt Bonn 2011	**6**	**33**	4	17	**7**	6	5	2	2	1
Bürgerbudget Wuppertal	**14**	**25**	1	40	9	21	6	7	6	4

Tabelle 26: Performance der Verschlagwortung mit BookGraph im Vergleich zur Baseline

Die Ergebnisse für diese Parameterwahlen sind in Tabelle 25 zusammen mit den Ergebnissen der BR-Methode, die in Abschnitt 4.2.2 die besten Ergebnisse geliefert hat, aufgeführt. Gegenüber der Baseline und BR gibt es keine wesentlichen Verbesserungen.

4.2.5 Graph-basierte Methode

Als Graph-basierte Methode wurde eine Variante von BookGraph evaluiert. Dabei wurden Schwellwerte $\theta \in \{0{,}05; 0{,}1; 0{,}15; 0{,}20\}$ getestet. Wie in Tabelle 26 dargestellt ist, ist diese Methode in keinem Fall besser als die Baseline.

4.2.6 Zusammenfassung

In Tabelle 27 sind die Performancewerte der besten Multi-Label-Verfahren, die in den vorherigen Abschnitten betrachtet worden sind, zusammengefasst. Bei den Raddialogen, bei denen im Wesentlichen eine Einfachverschlagwortung vorliegt, funktioniert das Baseline-Verfahren am besten, das einfach immer nur genau ein Schlagwort zuweist. Bei den beiden anderen Datensätzen ist das BR-Verfahren tendenziell besser.

Weil aber nur zwei Datensätze mit „echter“ Mehrfachverschlagwortung zum Testen zur Verfügung gestanden haben, kann an dieser Stelle keine wirkliche Empfehlung von Multi-Label-Verfahren für Online-Partizipationsverfahren gegeben werden.

	Baseline		BR		ML-k-NN	
Datensatz	MR	F_1	MR	F_1	MR	F_1
Raddialoge	**48**	**46**	27	**45**	32	34
Raddialog Bonn	**47**	**44**	26	42	31	32
Raddialog Köln	**44**	**36**	28	**36**	24	20
Raddialog Moers	**47**	**36**	35	33	30	25
Bürgerhaushalt Bonn 2011	6	33	**14**	**40**	**15**	30
Bürgerbudget Wuppertal	**14**	25	9	**48**	7	40

Tabelle 27: Vergleich der Performance der besten betrachteten Multi-Label-Verfahren

	Baseline		BR		ML-k-NN	
Datensatz	MR	F_1	MR	F_1	MR	F_1
Raddialoge	**46**	30	16	14	32	**36**
Raddialog Bonn	**44**	29	11	10	29	**33**
Raddialog Köln	**41**	**35**	7	6	22	27
Raddialog Moers	**43**	14	13	6	29	**17**
Bürgerhaushalt Bonn 2011	5	4	6	6	**13**	**36**
Bürgerbudget Wuppertal	**8**	14	0	23	4	**40**

Tabelle 28: Performance auf den Testsets für die besten betrachteten Multi-Label-Verfahren

4.2.7 Finale Evaluation und Fehlerbetrachtung

Schließlich wurden die Baseline-Methode und das BR-Verfahren noch auf dem Testdatensatz evaluiert und die von den Klassifikatoren gemachten Fehler untersucht.

In Tabelle 28 sind die MR- und F_1-Werte auf dem Testdatensatz dargestellt. Überraschend ist an dieser Stelle das schlechtere Abschneiden von BR auf den letzten beiden Datensätzen. Dies könnte daran liegen, dass die Datensätze relativ klein sind und es daher leicht zu Schwankungen bei den Metriken durch unterschiedlich gewählten Testsets kommt.

Für die Baseline und ML-k-NN werden nun die bei der Verschlagwortung der Testdatensätze gemachten Fehler untersucht.

Bei den Raddialogen werden 56 % der Beiträge von der Baseline mindestens ein richtiges Schlagwort zugewiesen. Bei falsch zugewiesen Schlagwörtern kommen zum Teil auch sinnvolle Schlagwörter vor, die von den Moderatoren nicht ausgewählt worden sind. Es gibt z. B. einen Beitrag, der einen schlechten Fahrbahnzustand und zugeparkte Radwege thematisiert, aber nur das Ground-Truth-Schlagwort *Radweg permanent zugeparkt* hat, aber nicht das Schlagwort *Unebenheit Brüche oder Risse*, das der Klassifikator gewählt hat. In den meisten Fällen sind die Vorschläge des Klassifikators sinnvoll.

68 % der beim Bürgerhaushalt Bonn 2011 vom Klassifikator vergebenen Schlagwörter sind korrekt. In den anderen Fällen ist der Vorschlag des Klassifikators teilweise auch sinnvoll. Beispielsweise gibt es Sparvorschläge, die nicht explizit als solche verschlagwortet sind. Allerdings schlägt der Klassifikator für die meisten Beiträge *Sparvorschlag*

vor, was zwar korrekt ist, wobei es aber oft konkrete Schlagwörter gibt, deren Zuweisung „erwünschter“ wäre.

Knapp drei Viertel der Beiträge zum Bürgerbudget Wuppertal haben ein richtiges Schlagwort vom Baseline-Verfahren zugeordnet bekommen. Bei falsch vergebenen Schlagwörtern ist es oft der Fall, dass z. B. *Freizeit und Kultur* und *Gemeinschaft* verwechselt wurden. Dem Klassifikator kann hier zugutegehalten werden, dass Freizeit-Aktivitäten oft die Gemeinschaft fördern und Vorschläge für mehr Gemeinschaft oft auch Freizeit zugeordnet werden könnten. Zum Teil gibt es auch Beiträge, bei denen das vom Klassifikator gewählte Schlagwort sinnvoll ist, aber nicht Teil der Ground Truth ist. Beispielsweise gibt es einen Beitrag, der über die Verbesserung der Zufriedenheit mit der Gesundheitsversorgung spricht und nur das Schlagwort *Gesundheit* besitzt, aber nicht das vom Klassifikator gewählte Schlagwort *Zufriedenheit*.

ML-k-NN hat bei den Raddialogen bei knapp ein Drittel der Beiträge gar kein Schlagwort zugeordnet. Bei etwa 30 % der Beiträge war die bestimmte Menge an Schlagwörtern keine Teilmenge der richtigen Schlagwörter. Dieser Fall kommt u. a. dann vor, wenn Nebenaspekte eines Beitrags vom Klassifikator für die Schlagwortzuweisung benutzt werden. Beispielsweise gibt es einen Textbeitrag, der grundsätzlich einen neuen Radweg vorschlägt, aber als mögliche Alternative eine verbesserte Beschilderung fordert, sodass der Klassifikator das Schlagwort *Radwegweisung fehlt oder schlecht sichtbar* anstatt *Vorschlag für neuen Radweg* wählt.

Beim Bürgerhaushalt Bonn 2011 schlägt dieses Verfahren bei knapp der Hälfte der Beiträge richtige Schlagwörter, d. h. genau die richtige Menge oder eine echte Teilmenge der richtigen Schlagwörter, vor. Es gibt einige „schwierige“ Schlagwörter, die der Klassifikator nicht gut erkennt. Beispielsweise kann eine *Einnahmeerhöhung* thematisiert sein, ohne dass Einnahmen direkt mit bestimmten Begriffen angesprochen werden. Außerdem sind nah verwandte Schlagwörter wie *Theater*, *Oper* und *Musik* basierend auf oft zum Teil kurzen Beiträgen und wenig Trainingsdaten nur schwer zu unterscheiden. Insgesamt sind die Vorschläge des Klassifikators meist sinnvoll, um sie als Vorschlag für einen Benutzer anzuzeigen.

43 % der Beiträge zum Bürgerbudget Wuppertal bekommen nur richtige Schlagwörter zugewiesen. Auch in den anderen Fällen ergeben viele der gesetzten Schlagwörter Sinn und es werden oft zu viele Schlagwörter vorgeschlagen. Zum Teil haben Benutzer auch Schlagwörter gewählt, die nicht wirklich zum Inhalt passen, z. B. bei Meldung einer Straße im schlechten Zustand das Schlagwort *Gesundheit*.

Weil bei den Raddialogen die verwendeten Schlagwörter eine Untergliederung der Kategorien sind, kann hier noch betrachtet werden, in wie vielen Fällen die Klassifikatoren ein Schlagwort gewählt haben, was mindestens eine Kategorie mit einem Ground-Truth-Schlagwort gemeinsam hat. Es werden also nur solche Schlagwörter als falsch gezählt, die zu einer Kategorie gehören, die nicht durch ein Schlagwort der Ground Truth abgedeckt ist.

Damit ergibt sich beim Raddialog-Datensatz für das Baseline-Verfahren ein MR-Wert von 64 % und ein F_1-Wert von 63 % statt 46 % bzw. 30 %. Insgesamt bekommen 77 % der Beiträge ein Schlagwort zugeordnet, das von der Kategorie her passend ist, d. h. nur in wenigen Fällen wählt der Klassifikator ein gar nicht passendes Schlagwort.

4.3 Evaluation von Hierarchie-basierten Verfahren

Wenn die zuzuweisenden Schlagwörter einer Hierarchie unterliegen, können diese Hierarchie-Informationen potentiell dazu beitragen, die automatische Verschlagwortung zu verbessern. Bei den Raddialogen kann beispielsweise das Schlagwort *Radweg häufig blockiert* nur dann vorkommen, wenn der Beitrag (auch) das Thema *Hindernisse* hat.[21]

Die in diesem Abschnitt verwendeten Begriffe orientieren sich an der Nomenklatur von Silla und Freitas (2011). Für die betrachteten Datensätze irrelevante Aspekte von hierarchischer Klassifikation, wie z. B. das Zuweisen von Labels, die keine Blätter sind, wird hier nicht eingegangen.

4.3.1 Hierarchische Verschlagwortung

Wenn eine hierarchische Anordnung von Labeln in einem Baum gegeben ist, gibt es verschiedene Möglichkeiten, Klassifikatoren für diese Hierarchie zu konstruieren, von denen hier zwei betrachtet werden.

Zunächst kann für jeden inneren Knoten ein Klassifikator trainiert werden, der ein Multi-Class- bzw. Multi-Label-Problem löst (*Local Classifier per Parent Node Approach*, LCPN). Ein unbekanntes Sample wird dann zunächst vom Klassifikator an der Wurzel einer Klasse k (bzw. mehreren Klassen) in der obersten Hierarchie zugeordnet. Anschließend bekommt der Klassifikator zum Knoten der Klasse k die Eingabe usw., bis eine Klasse eines Blattknotens erreicht ist.

Alternativ kann für jeden Knoten ein binärer Klassifikator gefittet werden, der entscheiden soll, ob ein Sample das zugehörige Label bekommt oder nicht (*Local Classifier per Node* oder *Top-Down Approach*, LCN). In der Testphase berechnen dann zuerst alle Klassifikatoren der obersten Ebene Wahrscheinlichkeiten für das jeweilige Label und das Label mit der größten Wahrscheinlichkeit (Single-Label-Fall) bzw. die Labels mit einer Wahrscheinlichkeit über einem Schwellwert θ (Multi-Label-Fall) werden gewählt. Die entsprechenden Klassifikatoren der Kindknoten werden dann nacheinander verwendet, bis schließlich ein Label bzw. mehrere Labels aus Blattknoten zugewiesen werden.

Es gibt verschiedene Strategien, die Trainingsmengen für die lokalen Klassifikatoren für ein Schlagwort k bei LCN zu bilden. Bei der Methode *inclusive* (inc) sind im Trainingsset für ein Schlagwort k alle Samples, die k oder ein Unterschlagwort zugeordnet haben, positive Beispiele und alle anderen negative. Bei der Variante *siblings* (sib) werden dagegen bei den negativen Beispielen nur Samples aufgenommen, die das Schlagwort eines Geschwisterknotens haben; beispielsweise wäre *Radweg häufig blockiert* dann weder negatives noch positives Beispiel für *Beleuchtung fehlt*, da diese zu unterschiedlichen Kategorien gehören.

Da bei den Raddialogen nur Blätter als Schlagwörter vergeben werden und keine inneren Knoten, fallen einige andere Methoden zur Konstruktion der Trainingsmengen, die in der Literatur gefunden werden können, weg. Als Basisklassifikator kommt wieder logistische Regression zum Einsatz.

[21] Wegen der Mehrfachverschlagwortung muss die Kategorie aber nicht zwangsläufig *Hindernisse* sein.

Verfahren	Baseline	LCPN				LCN, sib				LCN, inc			
Oversampling	✓	✗	✓	✗	✓	✗	✓	✗	✓	✗	✓	✗	✓
Gewichtung		✗	✗	✓	✓	✗	✗	✓	✓	✗	✗	✓	✓
Raddialoge	**52**	31	50	48	50	31	50	50	**51**	32	49	48	**51**
Raddialog Bonn	**50**	29	46	46	47	29	47	48	47	30	47	47	47
Raddialog Köln	**52**	17	37	39	46	17	40	**51**	44	17	39	50	46
Raddialog Moers	**56**	14	46	44	50	14	46	51	49	17	49	52	49

Tabelle 29: F_1-Scores für Hierarchie-basierte Verschlagwortung im Vergleich zur Baseline

Weil zum Testen von hierarchischer Verschlagwortung nur der Raddialog-Datensatz zur Verfügung steht und es in diesem kaum Mehrfachverschlagwortung gibt, wird hier nur einfache, hierarchische Verschlagwortung betrachtet. Dafür werden nur diejenigen Beiträge der Raddialoge verwendet, die genau ein Schlagwort zugewiesen bekommen haben, was etwa 83 % der Textbeiträge sind. Als Baseline dient hier die logistische Regression aus Kapitel 3.

Getestet wurden unterschiedliche Arten, mit Klassenimbalancen umzugehen: Das Anwenden von Oversampling vor dem hierarchischen Klassifikator, Sample-Gewichtung in der logistischen Regression innerhalb des hierarchischen Klassifikators und die Kombination beider Maßnahmen.

In Tabelle 29 sind die F_1-Scores der getesteten Varianten abgebildet. Grundsätzlich sind die Ergebnisse schlechter, wenn mit den Klassenimbalancen gar nicht umgegangen wird. Die Auswirkungen von Oversampling und Gewichtung sind unterschiedlich, aber die Ergebnisse liegen sehr nah beieinander. Insgesamt schneidet LCN mit der siblings-Methode in Kombination mit Gewichtung knapp am besten ab.

Allerdings sind alle Verfahren nicht so gut wie die Baseline, die keine Hierarchie-Information mit einbezieht. Auf den getesteten Datensätzen lohnen sich die komplexeren Verfahren LCPN und LCN in den getesteten Varianten also nicht.

4.3.2 Verschlagwortung mit gegebener Oberkategorie

Bei den Raddialogen war es im Praxiseinsatz so, dass die Kategorie der Beiträge zum Zeitpunkt der Verschlagwortung bereits bekannt waren. Somit ist es sinnvoll zu evaluieren, wie gut der Klassifikator für Einfachverschlagwortung ist, wenn die korrekte Kategorie bereits bekannt ist. Konkret wurden hier pro Kategorie Multi-Class-Klassifikatoren mit logistischer Regression trainiert und es wurden Varianten mit Oversampling und ohne Oversampling getestet.

Wie basierend auf den vorherigen Single-Label-Experimenten zu erwarten war, sind die Ergebnisse, die in Tabelle 30 dargestellt sind, durchgehend besser, wenn Oversampling verwendet wird. Ebenfalls wenig überraschend ist, dass durchgehend die Ergebnisse der Baseline übertroffen werden, wenn die Kategorie der Beiträge als bekannt vorausgesetzt wird. Denn in diesem Fall müssen die einzelnen Basisklassifikatoren nur zwischen jeweils zwei bis elf verschiedenen Schlagwörtern unterscheiden können und nicht alle gleichzeitig zwischen allen 33 Schlagwörtern.

Verfahren	Baseline	Kategorie gegeben	
Oversampling	✓	✗	✓
Raddialoge	52	45	**65**
Raddialog Bonn	50	42	**64**
Raddialog Köln	52	59	**69**
Raddialog Moers	56	45	**68**

Tabelle 30: F_1-Scores für Hierarchie-basierte Verschlagwortung bei gegebener Kategorie im Vergleich zur Baseline

Im Anhang sind in den Abbildungen 27 und 28 die Konfusionsmatrizen für die Verschlagwortung mit logistischer Regression (Baseline) bzw. bei gegebener Kategorie dargestellt. Bei letzterer Methode sieht man deutlich, dass Verwechslungen größtenteils nur innerhalb der Schlagwörter derselben Kategorie auftreten, wodurch der Recall bei den einzelnen Schlagwörtern steigt.

4.4 Zusammenfassung

Für eine umfassende Evaluation von Multi-Label-Verfahren und hierarchischer Verschlagwortung für Online-Partizipationsverfahren standen zu wenige verschiedene Datensätze zur Verfügung. Es wurde festgestellt, dass sich spezialisierte Multi-Label-Verfahren nicht lohnen, wenn der Großteil der Beiträge nur ein Schlagwort hat.

Als bestes Verfahren zur Mehrfachverschlagwortung hat sich tendenziell ML-k-NN gezeigt, gefolgt von der Binary-Relevance-Methode. Sofern vor einer hierarchischen Verschlagwortung den Beiträgen schon Oberkategorien zugordnet worden sind, sollten diese bei der Zuweisung von Schlagwörtern mit berücksichtigt werden. Die getesteten Verfahren liefern in vielen Fällen sinnvolle Vorschläge, die für ein automatisches Vorschlagen von Schlagwörtern genutzt werden könnten.

Weiterführend kann versucht werden, auch für die Verschlagwortung Augmentierungsverfahren, z. B. mithilfe von Suchmaschinen (vgl. Abschnitt 3.7.4), anzuwenden. Beispielsweise hat sich beim Bürgerhaushalt Bonn 2011 als problematisch erwiesen, dass es viele Schlagwörter mit wenigen Trainingsbeispielen gab. Außerdem kann noch untersucht werden, inwiefern gegenüber dem Single-Label-Fall geändertes Preprocessing und angepasste Hyperparamter die automatische Verschlagwortung verbessern.

Für hierarchische Verschlagwortung mit ggf. mehr als nur zwei Schichten kann sich außerdem ein Blick auf Algorithmen lohnen, die Fehler korrigieren können, die früh im Laufe der Klassifikation gemacht worden sind. Unter anderem wurden von Cheng et al. (2001) solche Verfahren vorgestellt.

5 Kategorienbildung durch automatisierte Themenextraktion

In diesem Kapitel werden unsupervised Verfahren betrachtet, die eine Menge von Textbeiträgen thematisch clustern sollen. Ein Anwendungsfall sind Partizipationsverfahren, bei denen im Vorfeld keine Kategorien festgelegt werden und dann in der Auswertung herausgefunden werden soll, über welche Themen diskutiert worden ist, wie es z. B. beim Nahverkehrsplan Ulm der Fall war. Bei den Raddialogen gab es einen ähnlichen Anwendungsfall, denn es gab im Vorfeld vorgegebene Kategorien, aber im Nachhinein wurden basierend auf den Diskussionsthemen Unterkategorien hinzugefügt.

Topic-Modeling-Algorithmen analysieren mithilfe statistischer Methoden die Wörter in Texten, um mögliche Themen zu extrahieren. Als Ausgabe erhält man eine Liste von Themen und für jeden Text die Information, zu wie viel Prozent er zu einem Thema gehört. Die Themen bestehen dabei aus Listen von charakteristischen Wörtern (Top-Wörter), beispielsweise *lampe licht dunkelheit gefährlich*. Es gibt also noch keine fertigen Themenbezeichnungen wie *Beleuchtung*; diese müssen, wenn gewünscht, manuell hinzugefügt werden.

Die Anwendung von Topic-Modeling erfolgt auf den Datensätzen unter folgenden Fragestellungen:

- Inwiefern stimmen die vom Modell gefundenen Themen mit den vom Menschen vorgegebenen/gewählten (Unter-)Themen überein?
- Sind die vom Modell gefundenen Themen sinnvoll?

Zunächst werden die getesteten Verfahren vorgestellt. Anschließend werden die Ergebnisse beschrieben und zusammenfassend beurteilt.

5.1 Verwendete Topic-Modeling-Algorithmen

In den folgenden Abschnitten werden die in dieser Arbeit für das Topic-Modeling verwendeten Algorithmen vorgestellt.

5.1.1 Latent Dirichlet Allocation

Bei der von D. M. Blei et al. (2003) entwickelten Latent Dirichlet Allocation (LDA) wird davon ausgegangen, dass es in den Dokumenten einer Dokumentenkollektion k unterschiedliche Themen gibt. Jedes Thema ist dabei eine Verteilung von Wörtern. Jedes Dokument der Kollektion wird von einer Themenverteilung generiert, aus der jeweils gemäß der Wortverteilung Wörter ausgewählt werden. Das Ziel von LDA ist es, in einem Inferenzschritt diese latenten Verteilungen zu bestimmen.

5.1.2 Latent Semantic Analysis

Die Grundidee von Latent Semantic Analysis (LSA, Deerwester et al., 1990), im Information Retrieval auch als Latent Semantic Indexing (LSI) bekannt, ist die Zerlegung einer

Dokument-Term-Matrix $A \in \mathbb{R}^{m \times n}$ in eine Dokument-Thema-Matrix $U_t \in \mathbb{R}^{m \times t}$ und eine Term-Thema-Matrix $V_t \in \mathbb{R}^{n \times t}$. Um dieses Ziel zu erreichen, wird eine Dimensionsreduktion mithilfe einer Singulärwertzerlegung durchgeführt. Durch die Singulärwertzerlegung erhält man eine Zerlegung $A = U\Sigma V^\intercal$ mit $U \in \mathbb{R}^{m \times r}$, $V \in \mathbb{R}^{n \times r}$ und $\Sigma \in \mathbb{R}^{r \times r}$, wobei Σ die Singulärwerte von A auf der Diagonalen hat.[22]

Durch Beschränkung von Σ auf die t größten Singulärwerte erhält man eine Zerlegung $A \approx U_t \Sigma_t V_t^\intercal$. Die Spalten von U und V entsprechen jetzt t Themen; die Zeilen sind eine Repräsentation von Texten bzw. Termen in Form von Themen.

In dieser Darstellung kann nun einfach die Ähnlichkeit von Dokumenten festgestellt werden und diese z. B. geclustert werden. Ein großer Nachteil gegenüber LDA ist, dass das Modell nicht einfach interpretierbar ist. Es ist nicht klar, was die t Themen sind. Die Dokumentvektoren der Länge t können als Projektion auf t Themen angesehen werden, wobei der Betrag jeder Vektorkomponente die „Stärke“ jedes Themas beschreibt.

5.1.3 Biterm Topic Model

Das von Yan et al. (2013) entwickelte Biterm Topic Model (BTM) ist ein spezielles Topic-Model für kurze Texte. Traditionelle Modelle wie LDA erfassen implizit Wortkookkurrenzen in Texten (Boyd-Graber und D. Blei, 2008), was bei kurzen Texten wegen der geringeren Anzahl von Wörtern schwieriger ist. Das BTM modelliert explizit Kookkurrenzen von Wörtern. BTM geht davon aus, dass ein Dokumentkorpus aus einer Mischung von Themen besteht, bei der jeder Biterm unabhängig aus einem bestimmten Thema gewählt wird. Ein Biterm ist eine Menge von zwei Wörtern, die gemeinsam in einem Text vorkommen. Für kurze Texte ist das Inferieren von Themen über den gesamten Korpus einfacher als das Inferieren von Themen von einzelnen kurzen Texten, sodass BTM potentiell bei kurzen Textbeiträgen besser funktioniert als LDA.

5.2 Evaluation der Modelle

Zunächst wird erklärt, auf welche Arten die Güte der Topic-Modeling-Verfahren untersucht wird. Anschließend werden die oben beschriebenen Verfahren anhand der beschriebenen Methoden getestet. Zur besseren Übersichtlichkeit befinden sich viele Tabellen und Abbildungen im Anhang.

5.2.1 Evaluationsverfahren

Wie Chang et al. (2009) festgestellt haben, beurteilen objektive Maße für Topic-Models, wie z. B. Perplexity, die Kohärenz und Relevanz von gefundenen Themen zum Teil sehr anders als Menschen. Daher wurde im Rahmen dieser Arbeit auf objektive Maße verzichtet und stattdessen die Qualität der Modelle subjektiv in Bezug auf deren Praxistauglichkeit beurteilt.

[22] In alternativen Formulierungen der Singulärwertzerlegung sind $U \in \mathbb{R}^{m \times m}$, $V \in \mathbb{R}^{n \times n}$ und $\Sigma \in \mathbb{R}^{m \times n}$. Diese äquivalente Darstellung kann durch Ausmultiplizieren der Nullzeilen von Σ in die oben genannte Darstellung überführt werden.

Da für alle Datensätze bereits von Menschen festgelegte Kategorien vorliegen, wurde betrachtet, inwiefern das von einem Topic-Model festgestellte dominante Thema eines Textbeitrags mit seiner Kategorie übereinstimmt. Dazu wurden die Textbeiträge ihrem dominanten Thema zugeordnet und für jedes Thema gezählt, wie oft welche Kategorie vorkommt. Wenn das Topic-Model eine ähnliche Kategorisierung wie der Mensch erstellt hat, sollten im besten Fall pro Thema nur Beiträge einer Kategorie vorkommen und jedes Thema gehört nur zu einer Kategorie. Weil die meisten Verfahren eine vorherige Festlegung der Themenanzahl k erfordern, wurden Werte für k im Bereich von 4 bis 20 getestet.

Es ist aber durchaus möglich, dass ein Topic-Model andere, sinnvolle Themen erkennt, als diejenigen, die menschlich bestimmt worden sind. Um dies zu berücksichtigen, wurden die Top-Wörter pro Thema und die Textbeiträge, die das jeweilige Thema laut Modell am meisten enthalten, beurteilt.

Zuletzt wurde getestet, ob die in einem Textbeitrag erkannte Themenverteilung sich als Feature für einen supervised Klassifikator eignet. Wenn dies der Fall ist, hat das Topic-Model es geschafft, die menschlich gesetzten Kategorien ggf. durch eine bestimmte Mischung verschiedener Themen gut abzubilden. Als Klassifikator wurde logistische Regression nach Oversampling mit den in Abschnitt 3 bestimmten Parametern verwendet. Beim Training wurden also immer ein Topic-Model auf den Trainingsdaten unsupervised trainiert und danach die Textbeiträge in eine Vektordarstellung gebracht, die der gefundenen Themenverteilung entspricht. Anschließend wurde anhand der bekannten Ground-Truth-Kategorien Oversampling angewandt und eine logistische Regression trainiert.

Wegen des hohen manuellen Aufwands bei diesen Beurteilungsverfahren wurden nur die Datensätze zu den Raddialogen, dem Mängelmelder Braunschweig und dem Nahverkehrsplan Ulm für die Evaluation verwendet. Bei den Raddialogen wurde zusätzlich ein Teildatensatz getestet, der nur Beiträge der größten Kategorie *Radverkehrsführung* mit jeweils genau einem von elf Schlagwörtern enthält, das als Ground-Truth-Label dient.

Für die ersten beiden Evaluationsverfahren wurden jeweils die kompletten Datensätze verwendet, für das letzte der Trainingsdatensatz für eine vierfache Kreuzvalidierung.

Die Textbeiträge wurden wie folgt vorverarbeitet: Alle Wörter, die keine Adjektive, Nomen oder Verben sind, wurden entfernt, um die Menge der Wörter auf die im Abschnitt 3.2.10 festgestellten bedeutungstragenden Wörter zu beschränken. Anschließend wurden die Wörter mithilfe von IWNLP lemmatisiert und Großbuchstaben in Kleinbuchstaben umgewandelt. Zuletzt wurden Wörter mit einer relativen Dokumentfrequenz von mehr als 20 % entfernt, da diese wahrscheinlich eher das allgemeine Thema des Datensatzes beschreiben und nicht relevant für konkrete Unterthemen sind.

5.2.2 LDA

Die von LDA gefundenen Themen sind in Tabelle 31 zu sehen. Bei den Raddialogen sind in Thema 1 vor allem Beiträge der Kategorie *Ampeln* enthalten, was auch aus der Menge der fünf relevantesten Wörter in diesem Thema deutlich wird. Die Beiträge in Thema 4 handeln hauptsächlich von schlecht erhaltenen oder schlecht markierten Radwegen, die

Thema	Top-Wörter
1	ampel, autofahrer, grün, kreuzung, abbiegen
2	schmal, gefährlich, kommen, müssen, häufig
3	kommen, weg, fahrbahn, autofahrer, gefährlich
4	weg, geben, hoch, kreuzung, radverkehr
5	stelle, fußgänger, gehweg, autofahrer, fahrbahn
6	gefährlich, weg, beleuchtung, fußgänger, gut
7	rad, fahrrad, geben, stadt, fahrradständer
8	weg, schlecht, zustand, schlagloch, muss
9	kommen, geben, kreuzung, muss, fußgänger

(a) Raddialoge

Thema	Top-Wörter
1	muss, kommen, geben, gehweg, weg
2	fußgänger, kreuzung, weg, gefährlich, autofahrer
3	geben, kreuzung, gefährlich, ampel, seite
4	gefährlich, stelle, freigeben, muss, fahrradfahrer
5	kommen, kreuzung, geben, fußgänger, gefährlich
6	fußgänger, weg, kommen, fahrradfahrer, autofahrer
7	autofahrer, kommen, müssen, gefährlich, geben
8	geben, autofahrer, kommen, weg, muss

(b) Raddialoge, Kategorie Radverkehrsführung

Thema	Top-Wörter
1	straßenbeleuchtung, ausfallen, tag, beleuchtung, haus
2	fahrradwrack, schlagloch, öffentlich, radständer, straße
3	straße, beschädigen, spielplatz, rüningen, fallen
4	müll, liegen, gehweg, radweg, weg
5	gehwegplatte, los, hoch, radweg, straße
6	herr, dame, geehrte, spielplatz, grundstück
7	straße, verstopfen, eck, woche, weg
8	stehen, schild, verkehrsschild, eck, wilder
9	auto, fahrzeug, gehweg, stehen, fahren
10	stehen, fahrradwrack, fahrradständer, monat, rad

(c) Mängelmelder Braunschweig

Thema	Top-Wörter
1	uni, haltestelle, gut, uhr, süd
2	wiblingen, gut, lehr, uni, haltestelle
3	haltestelle, schnell, stadt, fahrgast, richtung
4	finden, gut, haltestelle, kommen, stadt
5	stadt, haltestelle, minute, kommen, gut
6	takt, haltestelle, wiblingen, uhr, anbindung
7	uhr, verbindung, schüler, fahrgast, müssen
8	haltestelle, straßenbahn, neues, ehinger, minute

(d) Nahverkehrsplan Ulm

Tabelle 31: Von LDA gefundene Themen mit den zugehörigen Top-Wörtern

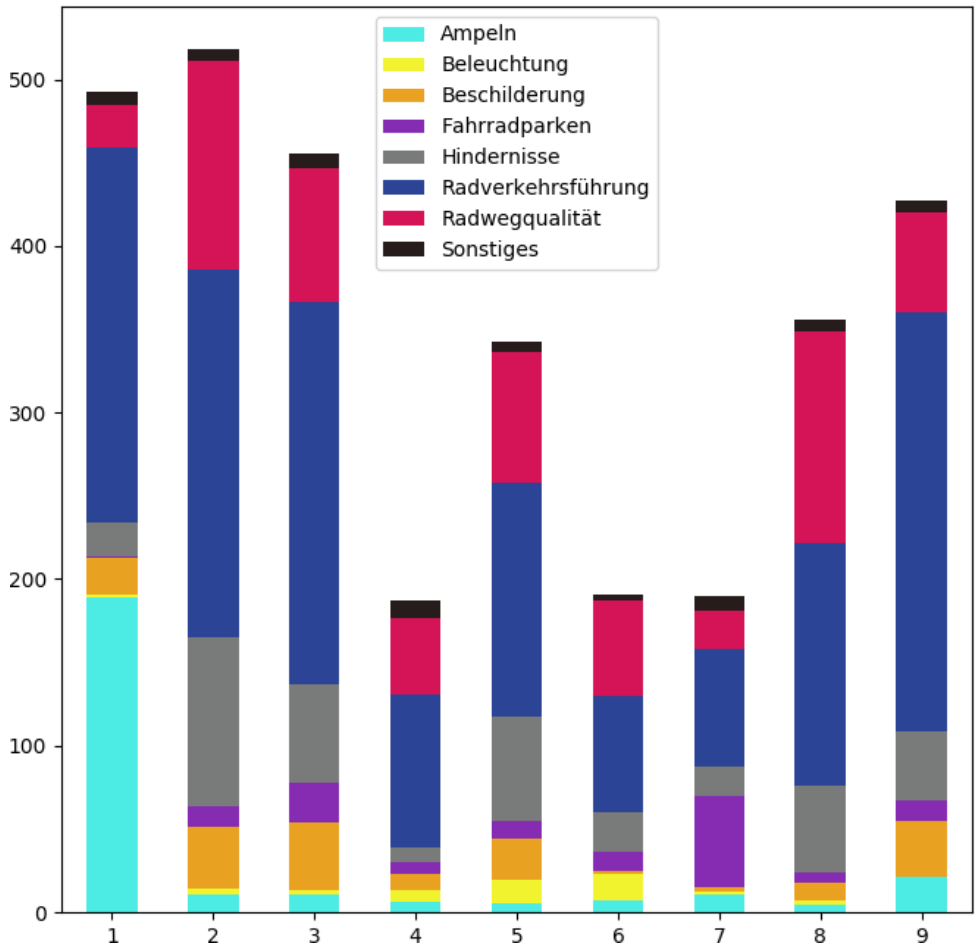


Abbildung 20: Verteilung der Kategorien auf die LDA-Themen bei den Raddialogen

Top-Wörter weisen darauf allerdings nicht hin. Die meisten Beiträge der Kategorie *Fahrradparken* werden, wie man in Abbildung 20 sehen kann, Thema 7 zugeordnet, welches auch das Wort *Fahrradständer* häufig enthält. Bis auf die zwei genannten Kategorien gibt es keinen deutlichen Zusammenhang zwischen den gefundenen Themen und den Kategorien. Die Kategorien *Ampeln* und *Fahrradparken* sind auch diejenigen, die bei der Klassifikation in Abschnitt 3 am sichersten klassifiziert werden konnten, was darauf hinweist, dass diese Kategorien (wenige) charakteristische Wörter enthalten, die diese Themen beschreiben.

Die für die Beiträge der Kategorie *Radverkehrsführung* gefundenen Themen scheinen keine wirklichen sinnvollen Themen zu bilden. Die in der manuellen Auswertung gewählten Unterkategorien sind auf die LDA-Themen keinem Muster folgend verteilt, wie man in Abbildung 29 im Anhang sehen kann.

Beim Mängelmelder Braunschweig passen Kategorien und Themen besser zueinander. Die Themen 1 und 5 bestehen fast nur aus Beiträgen der Kategorien *Straßenbeleuchtung/Laterne defekt* bzw. *Straßen-, Radweg- und Gehwegschäden*. Beiträge der Kategorie *Fahrradwracks* verteilen sich auf die Themen 1 und 10, die beide das Top-Wort *Fahrradwrack* enthalten. In Thema 6 fallen Beiträge, die die Wendung *Sehr geehrte Damen und Herren* enthalten; dieses Thema ist daher unaussagekräftig.

Beim Nahverkehrsplan Ulm sind Themen und Kategorien bunt durchmischt. Fast alle Themen beinhalten das Wort *Haltestelle*, was im Kontext des Dialogs keine Aussagekraft hat. Das Wort bzw. seine Formen kommen in knapp 17 % der Textbeiträge vor, sodass hier evtl. eine Verringerung der maximalen Dokumentfrequenz eine bessere Themenfindung begünstigt.

Verwendet man die von LDA bestimmte Themenverteilung pro Beitrag als Feature für die logistische Regression, so erhält man bei 100 Themen die F_1-Scores 32, 46 bzw. 23 auf den getesteten Datensätzen. Bei Verwendung der logistischen Regression mit Charakter-N-Grammen erhielt man im Vergleich bei der finalen Evaluation Werte von 68, 85 bzw. 59.

5.2.3 LSA

Das „stimmigste“ Bild ergibt sich bei den Raddialogen mit 14-LSA-Dimensionen. Wie in Abbildung 30 und Tabelle 37 im Anhang gesehen werden kann, dominiert Thema 1 stark und enthält Wörter im Zusammenhang mit Ampeln und Kreuzungen – aber auch allgemeine Wörter zum Thema Radfahren, wodurch eine Häufung der Beiträge in diesem Thema erklärbar ist. Das zweite Thema ist anhand der dominanten Wörter eindeutig Ampeln zuzuordnen, was auch dadurch bestätigt wird, dass in diesem Thema fast ausschließlich Beiträge der Kategorie *Ampel* zu finden sind. In Thema 5 dominieren Beiträge der Kategorie *Fahrradparken*, was aber aus den Wörtern des Themas nicht hervorgeht. Das letzte Thema enthält vor allem Beiträge der Kategorie *Radwegqualität*. Die relevanten Wörter deuten darauf hin, dass es hauptsächlich um Unterführungen und schmale Wege geht.

Die Themen, die bei den Beiträgen zur Radverkehrsführung gefunden werden, scheinen nicht sinnvoll zu sein. Es geht um „Autofahrer“, „Fußgänger“ und „Gefahren“, was keinen Erkenntnisgewinn bringt.

Beim Mängelmelder Braunschweig gibt es ein besseres Bild. Das Thema 2 enthält fast nur Beiträge der Kategorie *Fahrradwracks* und entsprechend ist bei diesem Thema *Fahrradwrack* ein relevantes Wort; allerdings taucht das Wort auch bei einigen anderen Themen auf. Bei Thema 5 wird wieder die Formulierung *Sehr geehrte Damen und Herren* zu einem Thema, überraschenderweise werden aber diesem Thema fast ausschließlich Beiträge der Kategorie *Straßenbeleuchtung/Laterne defekt* zugeordnet. Im vorletzten Thema befinden sich größtenteils Beiträge, die Wegschäden beschreiben.

Beim Nahverkehrsplan Ulm kommen keine guten Themen zustande. Die extrahierten wichtigen Wörter *Uhr*, *Haltestelle* usw. sind zu allgemein.

Die Verwendung der LSA-transformierten Dokumente als Eingabe für logistische Regression ergibt F_1-Werte von 49, 64 bzw. 38.

5.2.4 BTM

Bei 11 generierten Themen entsprechen die BTM-Themen am ehesten den Kategorien der Raddialoge, was in Abbildung 31 im Anhang zu sehen ist. Thema 2 besteht zum größten Teil aus Beiträgen zur *Radverkehrsführung*, wobei hier mehrere Beiträge, die nur den Inhalt *Freilaufende Rechtsabbieger zurückbauen* haben, zugeordnet werden. Die meisten Beiträge der Kategorie *Ampeln* sind in Thema 4 enthalten, entsprechend finden sich bei diesem Thema die relevanten Wörter *Ampel*, *Kreuzung*, *grün* usw., wie Tabelle 38 entnommen werden kann. In Thema 9 kommen vor allem Wörter in Zusammenhang mit *Radwegqualität* vor, z. B. *schlecht*, *Zustand* und *Schlagloch*. Thema 8 ist das dominanteste Thema, in dem viele verschiedene Kategorien vertreten sind. Dieses Thema enthält viele allgemeine Wörter wie *kommen* und *Autofahrer*.

Für die Beiträge der Kategorie *Radverkehrsführung* gibt es keine Themenanzahl, sodass die gefundenen Themen mit den Kategorien zumindest teilweise übereinstimmen – die Kategorien sind scheinbar immer gleichverteilt auf die Themen. Auch die Inhalte der Beiträge desselben Themas scheinen keinen inhaltlichen Zusammenhang zu haben.

Beim Nahverkehrsplan Ulm ergibt sich ein ähnliches Bild. Die Beiträge der verschiedenen Kategorien sind gleichmäßig auf die Themen verteilt und Beiträge eines Themas scheinen in keinem sinnvollen Zusammenhang zu stehen.

Beim Mängelmelder Braunschweig hingegen ergibt sich bei 6 Themen eine sinnvolle Verteilung. Fast alle Beiträge der Kategorie *Fahrradwracks* sind Thema 6 zugeordnet, das auch entsprechende Top-Wörter enthält. In Thema 4 dominieren Wörter wie *Gehwegplatte* und *Schlagloch*; in diesem Thema befinden sich vor allem Beiträge der Kategorie *Straßen-, Radweg- und Gehwegschäden*. Das Thema 2 beinhaltet hauptsächlich Beiträge zur Kategorie *Straßenbeleuchtung/Laterne defekt*. In Thema 1 befinden sich wieder Beiträge, die *Sehr geehrte Damen und Herren* enthalten.

Topic-Model	Logit	LDA	LSA	BTM
Raddialoge	71	32	**49**	45
Mängelmelder Braunschweig	87	46	**64**	58
Nahverkehrsplan Ulm	60	23	**38**	25

Tabelle 32: F_1-Scores für die Klassifikation mithilfe der Themenzugehörigkeiten im Vergleich zur logistischen Regression mit Charakter-N-Grammen

Werden die von BTM bestimmten Themenzugehörigkeiten für 100 Themen als Features für logistische Regression benutzt, erhält man auf den drei getesteten Datensätzen F_1-Werte von 45, 58 bzw. 25.

5.3 Zusammenfassende Beurteilung

Keines der existierenden Topic-Models war in der Lage, Themen zu finden, die größtenteils mit den menschlich festgelegten Kategorien übereinstimmen. Nur in wenigen Fällen konnten Kategorien, für die es charakteristische Wörter wie *Fahrradwrack* gibt, wiedergefunden werden. Oft haben die automatisch erkannten Themen für den Menschen keinen Sinn ergeben.

Eine Ursache dafür kann – ähnlich wie schon in Kapitel 3 – sein, dass es einige Themen gibt, für deren Beschreibung viele unterschiedliche Wörter verwendet werden, aber durch die kleine Größe der Datensätze die Modelle nicht in der Lage sind, Synonyme, ähnliche Wörter und Konzepte zu erkennen. Bei den getesteten Datensätzen wurden eher Themen zu Wörtern erstellt, die häufig, aber nicht überall vorkommen, wie z. B. *Auto* oder *Weg*, was aber in den meisten Fällen keinen Erkenntnisgewinn bringt.

Die Themen-Verteilung als Feature zu verwenden, funktioniert mit den getesteten Topic-Models unterschiedlich gut, wie in Tabelle 32 zusammengefasst ist. Am besten haben die Themen-Verteilungen von LSA funktioniert. Als alleiniges Feature zur Klassifikation sind sie nicht gut geeignet und auch in Kombination mit BOW-Features haben diese, wie in Abschnitt 3.7.9 beschrieben, keine positive Auswirkung.

Bei einer weiterführenden Arbeit zur automatisierten Themenextraktion müsste ein Verfahren gefunden oder entwickelt werden, das mit kleinen Datensätzen, Vokabularüberlappungen und thematisch ähnlichen Textbeiträgen zurecht kommt. Gerade durch die eingeschränkte Menge an Themen bei Online-Partizipationsverfahren könnte es schwierig sein, auf vortrainierte Topic-Models zurückzugreifen.

6 Zusammenfassung und Future Work

Ein Großteil der Ziele dieser Arbeit konnte umgesetzt werden. In Kapitel 3 wurde ein Verfahren zur automatisierten Zuweisung von Kategorien entwickelt, das auf Charakter-N-Grammen und logistischer Regression beruht. Es arbeitet vergleichbar gut wie ein Mensch und ordnet rund drei Viertel der Textbeiträge der korrekten Kategorie zu. Die Güte der Klassifikation ist allerdings auch von der Anzahl der Trainingsdatensätze pro Kategorie, der Anzahl der Kategorien und der thematischen Verwandtschaft dieser abhängig.

Eine automatische Verschlagwortung bereits kategorisierter Textbeiträge hat in den durchgeführten Experimenten gut funktioniert, allerdings standen für eine ausführliche Evaluation der verwendeten Methoden zu wenige Datensätze zur Verfügung. Wenn die meisten Testbeiträge eines Datensatzes nur einfach verschlagwortet sind, sollte auf Single-Label-Verfahren zurückgegriffen werden. Bei richtiger Mehrfachverschlagwortung haben sich das Binary-Relevance-Verfahren und ML-k-NN als gut herausgestellt.

Die automatische Extraktion von Themen eines Online-Partizipationsverfahrens hat mit den untersuchten Methoden nicht funktioniert. Die bekannten Topic-Modeling-Verfahren scheitern u. a. daran, dass die Themen der Beiträge sehr nah zusammenliegen und es nur wenige Beiträge gibt, sodass allein durch Vokabularüberschneidungen keine Themen extrahiert werden können.

Weiterführend wäre die Betrachtung größerer Datensätze interessant, um insbesondere zu untersuchen, ob neuronale Netze auf größeren Datenmengen bessere Ergebnisse liefern als klassische Verfahren, vor allem bei der Unterscheidung von Kategorien, bei denen BOW-Features nicht ausreichend sind. Weiterhin kann die Performance bei der Erkennung der Kategorie *Sonstiges* verbessert werden. Dort müssten neue Features entwickelt werden, da hier BOW-basierte Features zur Klassifikation nicht gut geeignet sind.

Ferner sollten Verschlagwortung und hierarchische Verschlagwortung anhand weiterer Datensätze untersucht werden, da die verfügbare Datengrundlage nicht ausreichend für eine umfangreiche Beurteilung war. Bei der automatischen Themenextraktion müssten andere Verfahren gefunden oder entwickelt werden, die mit thematisch nah beieinanderliegenden Beiträgen sinnvolle Unterthemen finden können.

Sofern Partizipationsverfahren mit ähnlichem Aufbau wiederholt auftreten, kann auch eine Spezialisierung des entwickelten, allgemeinen Klassifikators sinnvoll sein. Dort könnten dann z. B. Features aus Kommentaren, Kostenschätzungen oder Bildern extrahiert werden.

Für den Einsatz in der Praxis muss die erstellte Klassifikationspipeline noch über eine API bereitgestellt werden. Dafür würde sich das Bereitstellen eines fertigen Docker-Containers anbieten.

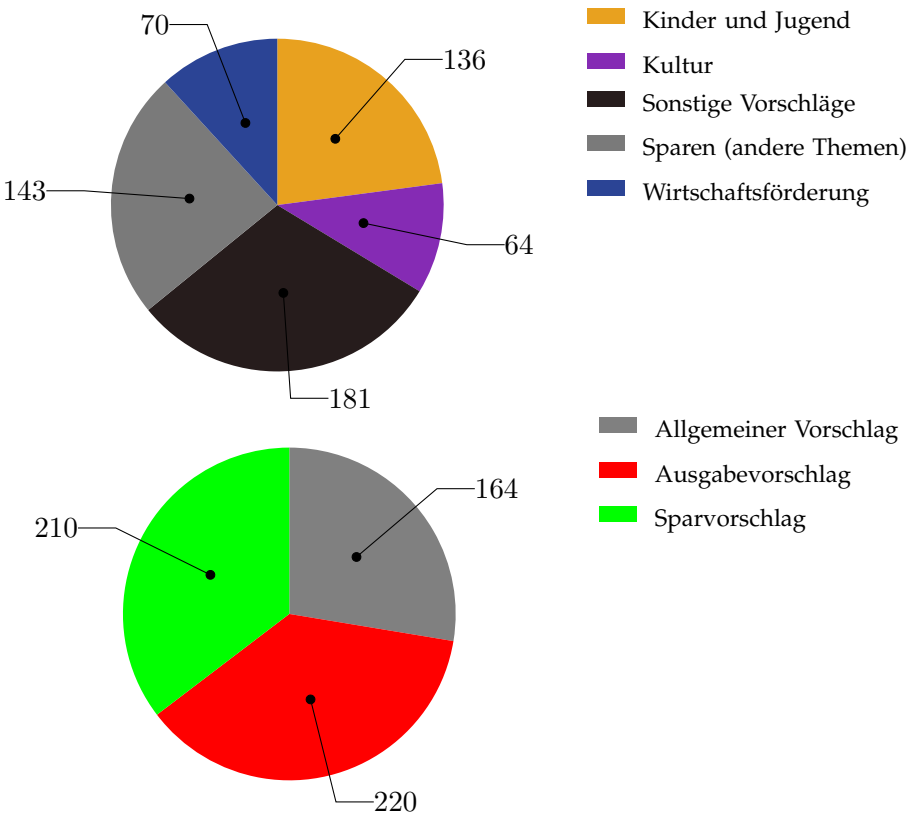
A Anhang

A.1 Datensatzeigenschaften

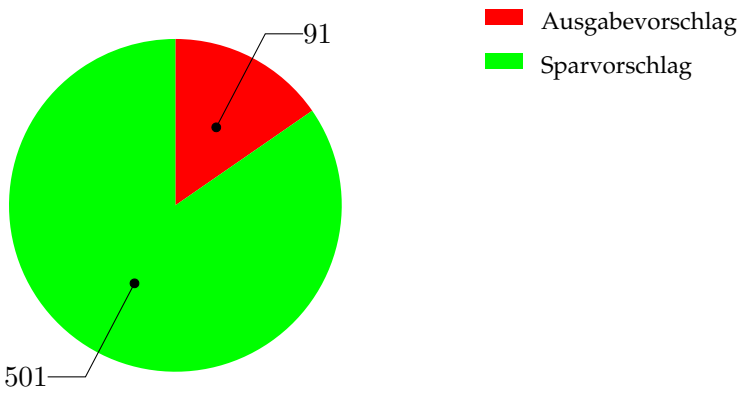
In den folgenden Tabellen und Abbildungen sind Eigenschaften der in Kapitel 2 beschriebenen Datensätze dargestellt.

Schlagwort	Bonn	Köln	Moers	$\sum$
Ampeln				
Ampel entfernen	12	6	0	18
Ampel(ergänzung) vorschlagen	68	18	6	92
Ampelschaltung ungünstig	123	17	46	186
Beleuchtung				
Beleuchtung fehlt	42	2	12	56
falsche Beleuchtung	5	0	3	8
Beschilderung				
Fahrbahnmarkierung Radweg fehlt oder schlecht sichtbar	112	20	15	147
Radwegweisung fehlt oder schlecht sichtbar	71	0	13	84
Fahrradparken				
keine oder zu wenig Abstellmöglichkeiten	96	26	10	132
ungeeignete Abstellanlagen	20	0	0	20
unsichere Abstellanlagen	1	0	0	1
Hindernisse				
Behinderung durch feste Gegenstände	109	8	15	132
Radweg häufig blockiert	63	0	5	68
Radweg permanent zugeparkt	206	37	14	257
Radverkehrsführung				
Auffahrt auf Radweg nur mit Umweg möglich	24	0	0	24
Einbahnstraße für Radverkehr öffnen	37	32	5	74
Fahrradstraße einrichten	74	15	4	93
Geschwindigkeitsbegrenzung	23	15	0	38
Radweg beidseitig befahren	48	0	12	60
Radwegebenutzungspflicht überprüfen	31	7	7	45
Vorschlag für neuen Radweg	398	87	121	606
mangelnde Sichtbeziehungen	82	9	16	107
regelwidriges Verhalten	74	4	9	87
sichere Straßenquerung fehlt	140	43	28	211
unklare Verkehrsführung für Radfahrende	209	11	37	257
Radwegqualität				
Übergänge mit zu großen Höhenunterschieden	49	10	13	72
Unebenheit Brüche oder Risse	223	32	77	332
wiederholt Schmutz oder Wasser auf Radweg	48	5	11	64
zu geringe Breite	245	29	38	312
Sonstiges				
Mängelmeldung	11	0	4	15
Verwarnung	14	4	2	20
nicht ortsgebundene Vorschläge	60	23	22	105
sonstige Hinweise	23	4	5	32

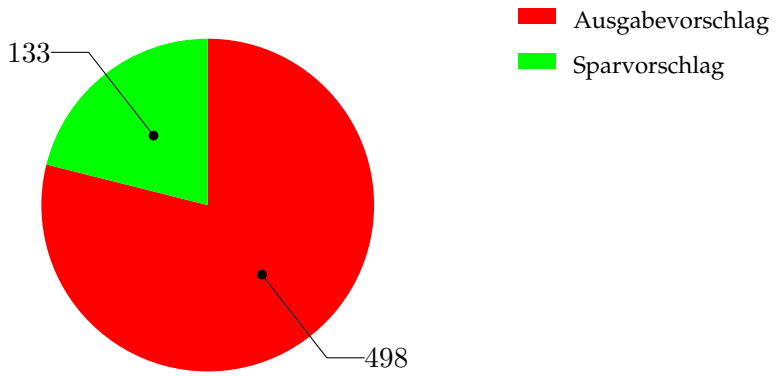
Tabelle 33: Häufigkeit der bei den Raddialogen vergebenen Schlagwörter



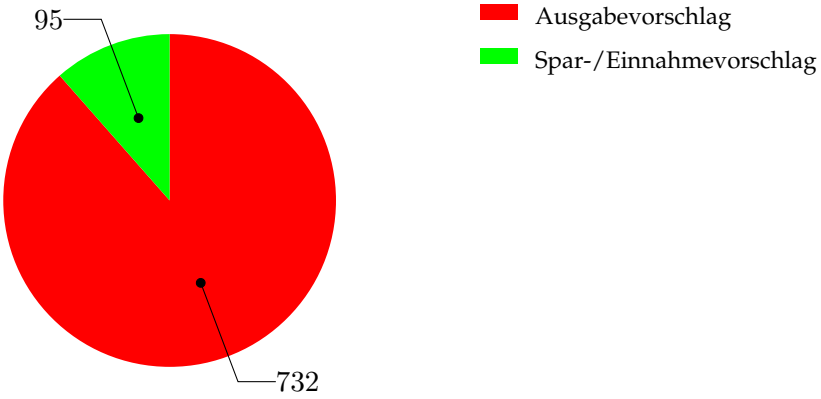
(a) Bürgerhaushalt Köln 2012, insgesamt 594 Beiträge



(b) Bürgerhaushalt Köln 2013, insgesamt 592 Beiträge



(c) Bürgerhaushalt Köln 2015, insgesamt 631 Beiträge



(d) Bürgerhaushalt Köln 2016, insgesamt 827 Beiträge

Abbildung 21: Verteilung der Beiträge der Kölner Bürgerhaushalte auf die Kategorien

Schlagwort	#	Schlagwort	#
Sparvorschlag	566	Energie sparen	18
Einnahmeerhöhung	243	Wirtschaftsförderung	17
Verwaltung	164	Wohnen	15
Straßen	95	Bürgerbeteiligung	15
Gebühren	90	Rheinaue	15
Veranstaltungen & Feste	67	Beethovenhalle	14
Musik	61	Klimaschutz	13
Zuschüsse	60	Vereine	12
Schwimmbad	58	Städtische Grundstücke	11
Städtische Gebäude	57	Erneuerbare Energien	10
Theater	56	Karneval	10
Stadtgestaltung	54	Musikschule	10
Oper	49	Winterdienst	9
Verkehrsführung	45	Porto	9
ÖPNV	44	Spielplatz	9
Stadtplanung	43	OGS	9
Schule	43	VHS	9
Stadtentwicklung	41	Literatur	8
Steuern	39	Patenschaft	7
Sponsoring	34	Werbung	7
Natur und Grünflächen	33	Geschwindigkeitskontrollen	7
Beleuchtung	33	Fahrrad	6
Ampelanlagen und Kreisel	32	Hardtbergbahn	6
Müllabfuhr und Sauberkeit	31	Integration	5
Politik	29	Senioren	4
Parkplätze	29	Toiletten	3
Bibliothek	27	Gehälter	3
WCCB	24	Feuerwehr	3
Museum	23	Ehrenamt	2
Jugendarbeit	23	Laubgebläse	2
Beratung	23	Weihnachten	1
Kindergarten	20	Flugverkehr	1
Fuhrpark	20	Ausbildung	1
Naherholung	20	Kraftfahrzeuge	1

Tabelle 34: Häufigkeiten der beim Bürgerhaushalt Bonn 2011 vergebenen Schlagwörter

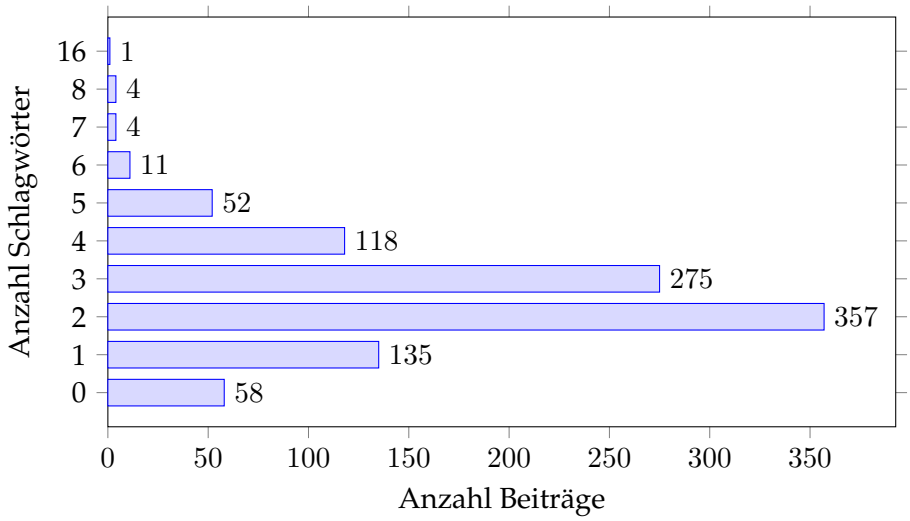


Abbildung 22: Übersicht über die Anzahl der Schlagwörter, die pro Beitrag zum Bürgerhaushalt Bonn 2011 vergeben wurden

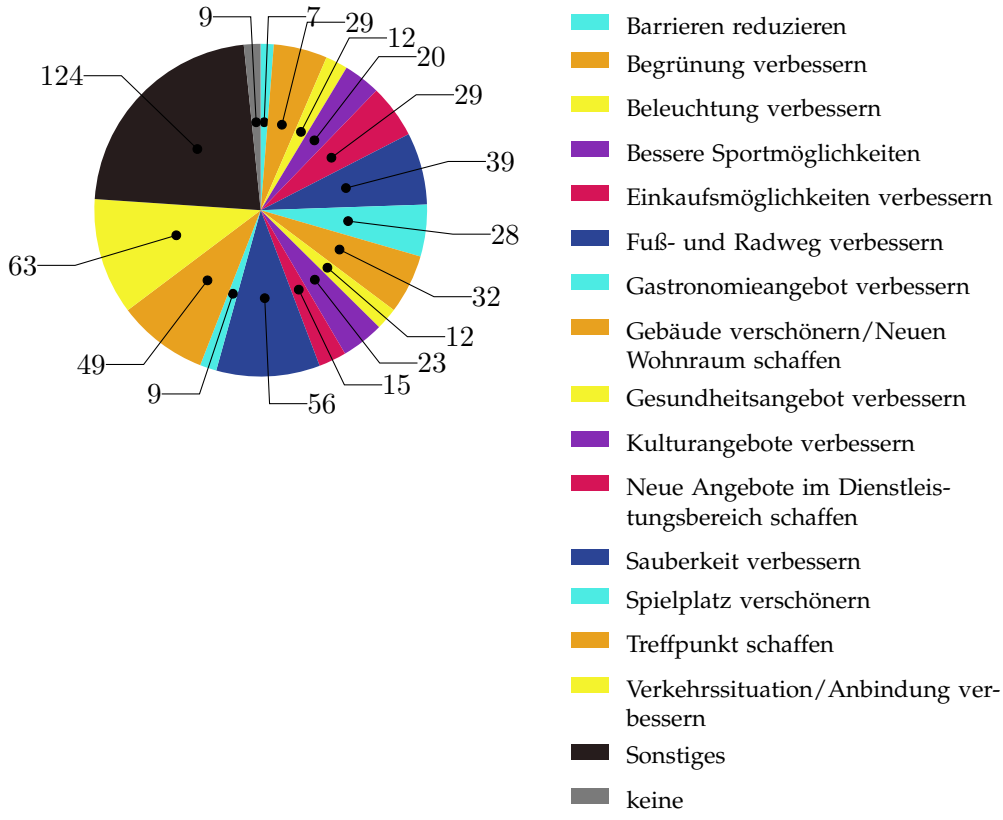


Abbildung 23: Verteilung der Beiträge zum Leitbild Bad Godesberg auf die Kategorien

A.2 Gridsearch-Parameter

Die folgenden Tabellen geben Auskunft über die Gridsearch-Parameter, die in Kapitel 3 benutzt wurden.

Parameter	Werte
N-Gramm-Größen	3, 3+4, 3+4+5, 4, 4+5, 5+5
minimale Document Frequency	1, 2, 3, 4
maximale Document Frequency	3 %, 4 %, 5 %, 10 %, 20 %, 50 %, 100 %
Anzahl der Nachbarn n	4, 5, 6, 7, 8
Metrik	Cosinus, Minkowski
p der Minkowski-Metrik	1, 2, 3
Gewichtung	keine, inverse Distanz
POS-Filter	keiner; Adjektive, Nomen, Verben; Adjektive, Eigennamen, Nomen, Verben
tf-idf-Gewichtung	ja, nein

Tabelle 35: Gridsearch-Parameter für k-NN

Parameter	Werte
N-Gramm-Größen	3, 3+4, 3+4+5, 4, 4+5, 5+5
minimale Document Frequency	1, 2, 3
maximale Document Frequency	20 %, 50 %, 100 %
inverse Regularisierungsstärke C	0,1, 1, 10
Bias für Entscheidungsfunktion	ja, nein
Bias-Skalierung	0,1, 1, 10
POS-Filter	keiner; Adjektive, Nomen, Verben; Adjektive, Eigennamen, Nomen, Verben
tf-idf-Gewichtung	ja, nein

Tabelle 36: Gridsearch-Parameter für logistische Regression

A.3 Lernkurven

In den folgenden Grafiken sind die Lernkurven für die übrigen in Abschnitt 3.9.2 gewählten Datensätzen dargestellt.

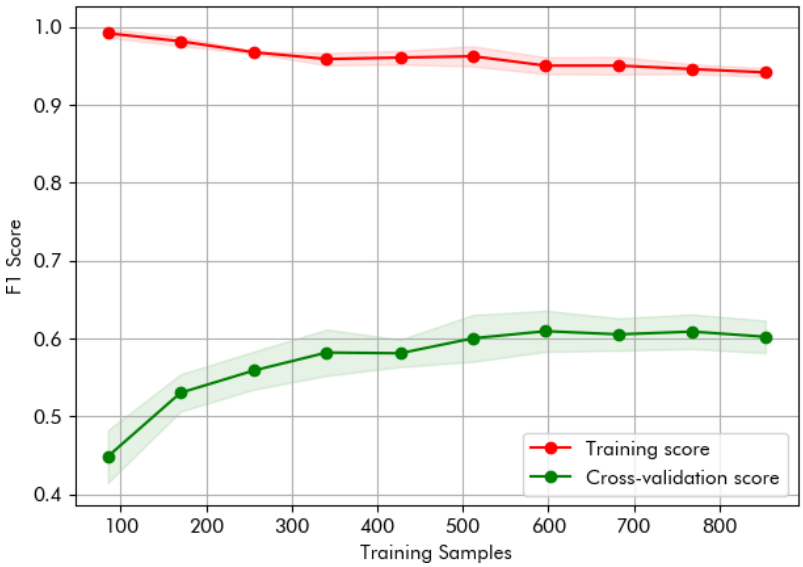


Abbildung 24: Lernkurve für den Bürgerhaushalt Bonn

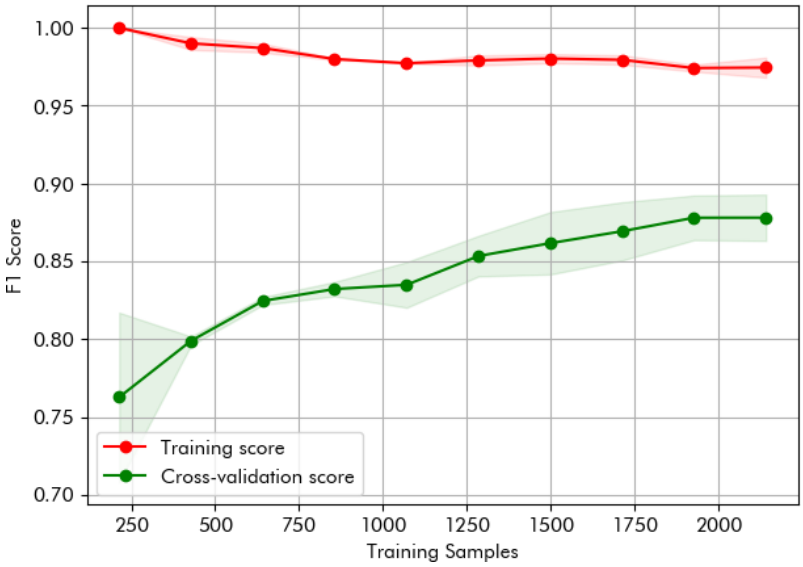


Abbildung 25: Lernkurve für den Mängelmelder Braunschweig

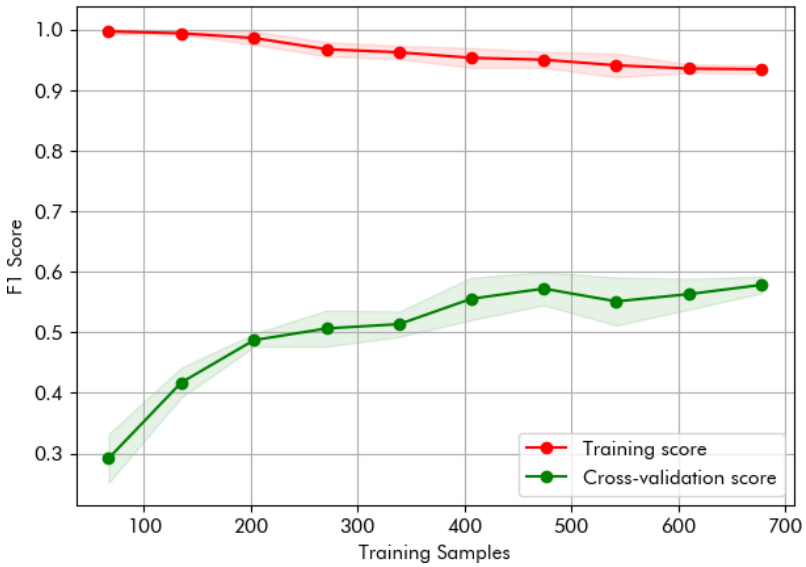


Abbildung 26: Lernkurve für den Nahverkehrsplan Ulm

A.4 Konfusionsmatrizen für hierarchische Einfachverschlagwortung

Die folgenden Konfusionsmatrizen sind Ergebnis der Hierarchie-basierten Einfachverschlagwortung mit gegebener Kategorie, wie sie in Abschnitt 4.3.2 beschrieben wird.

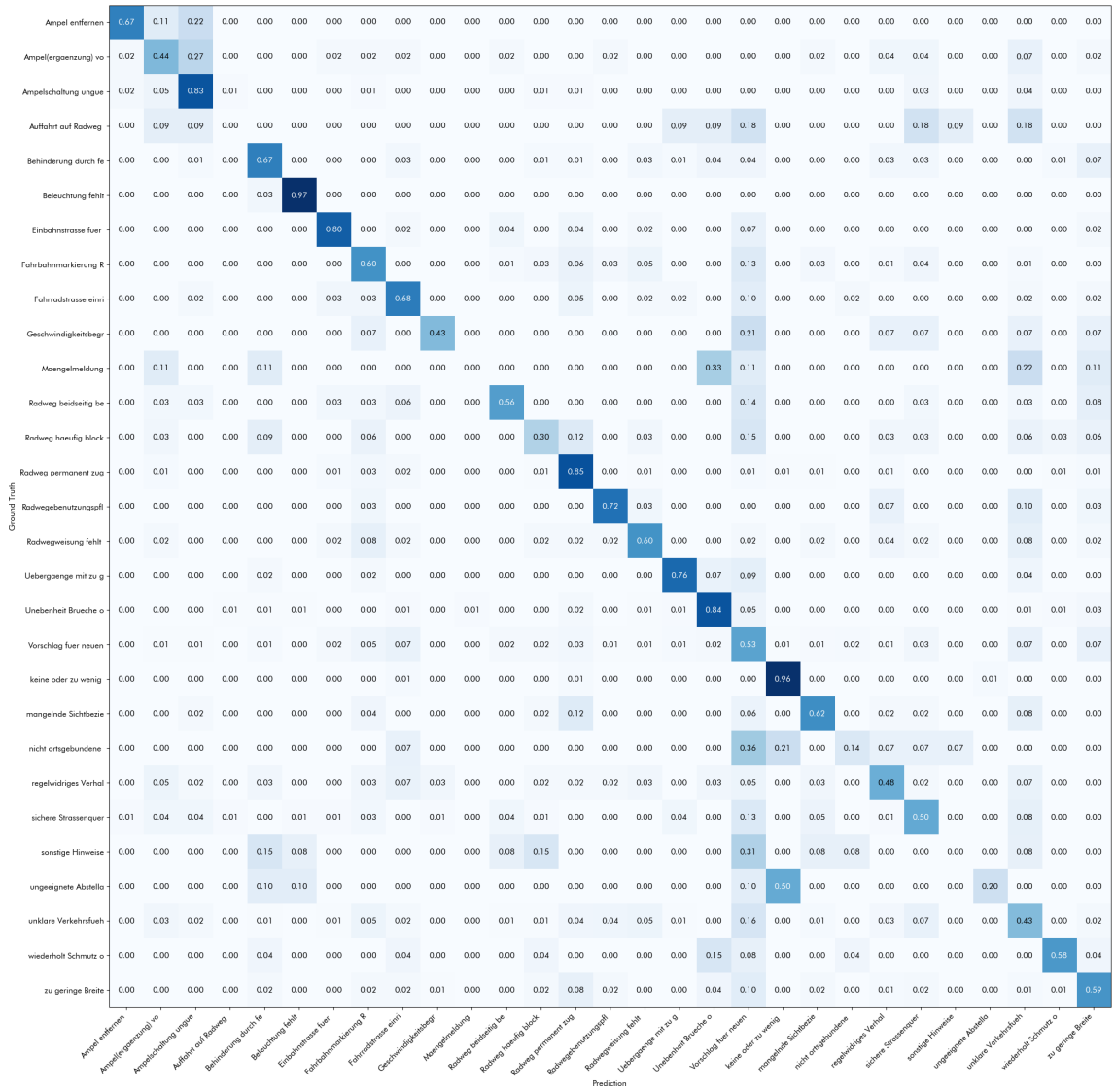


Abbildung 27: Konfusionsmatrix für hierarchische Einfachverschlagwortung mit der Logit-Baseline

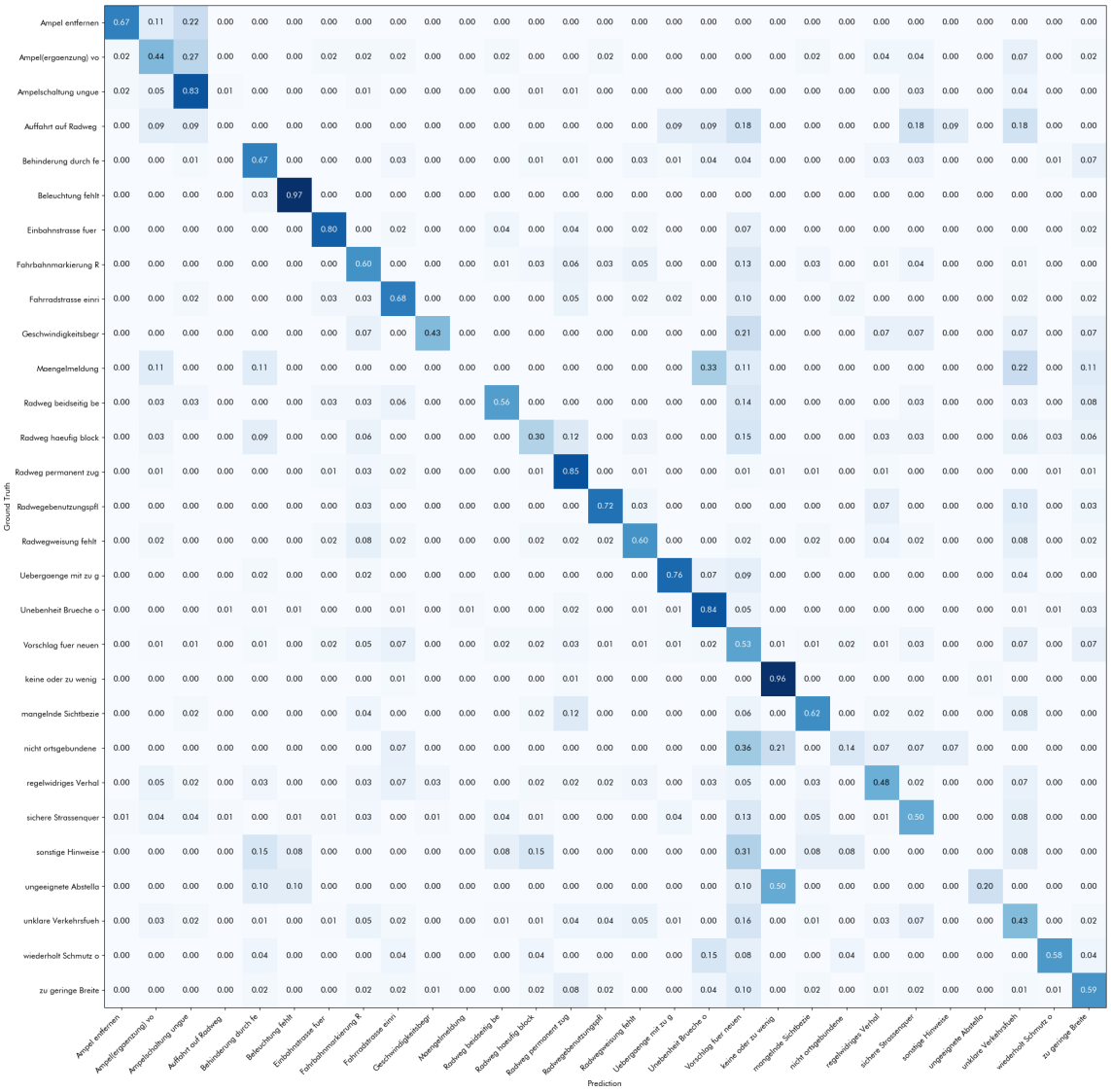
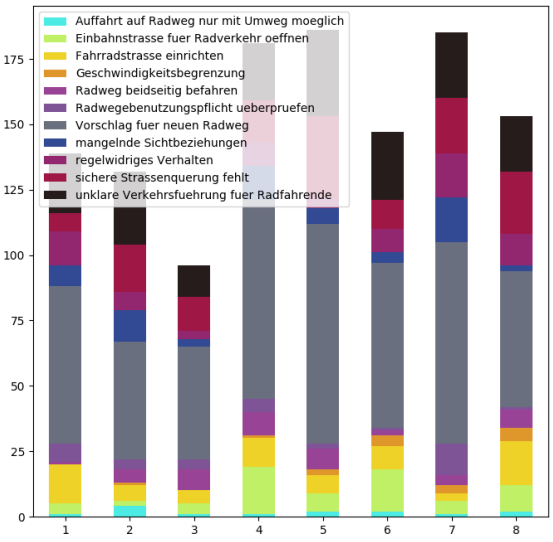


Abbildung 28: Konfusionsmatrix für hierarchische Einfachverschlagwortung mit gegebener Oberkategorie; zu beachten ist, dass die Schlagwörter alphabetisch und nicht nach Kategorien sortiert sind

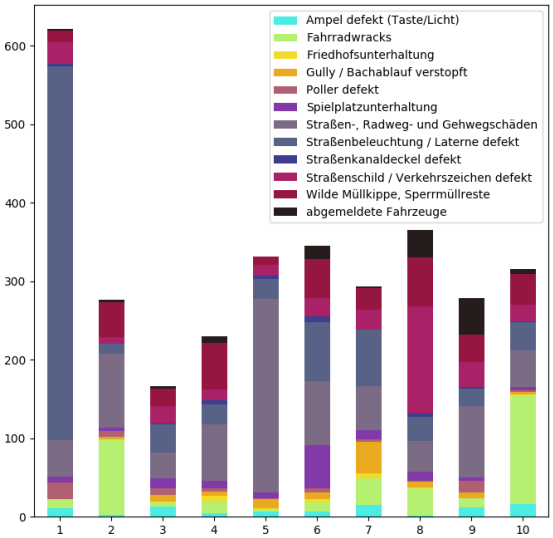
A.5 Evaluation der Topic-Modeling-Algorithmen

Im Folgenden sind die übrigen Ergebnisse der Evaluation der in Abschnitt 5.2 beschriebenen Topic-Modeling-Verfahren abgebildet.

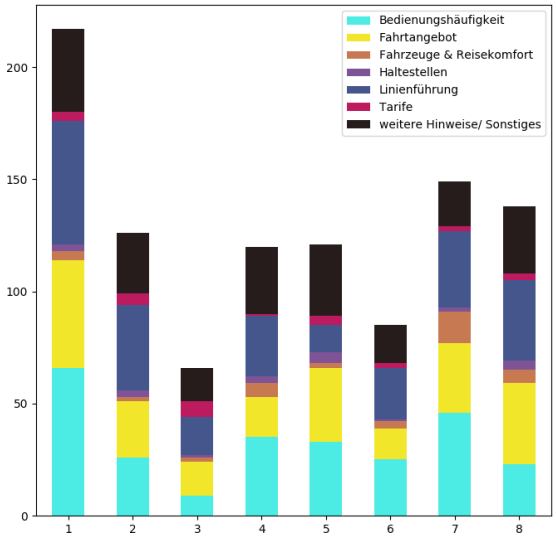
A.5.1 LDA



(a) Raddialoge, Kategorie Radverkehrsführung



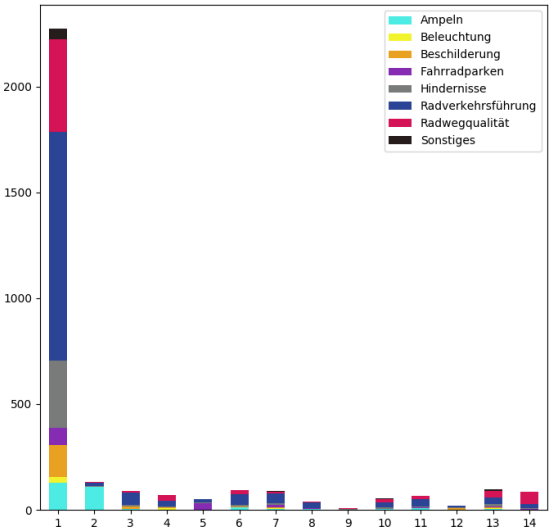
(b) Mängelmelder Braunschweig



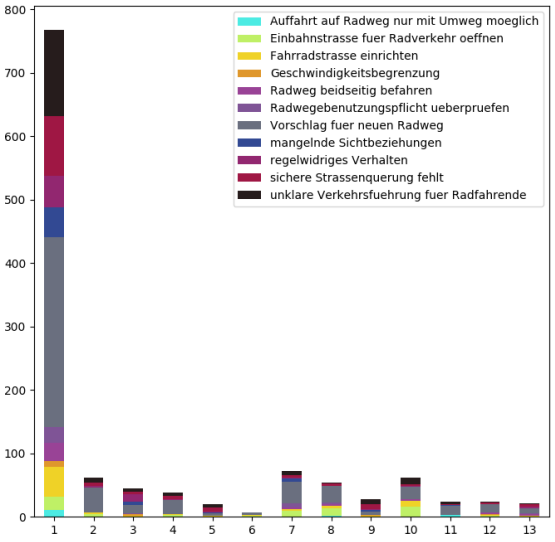
(c) Nahverkehrsplan Ulm

Abbildung 29: Verteilung der Kategorien auf die LDA-Themen

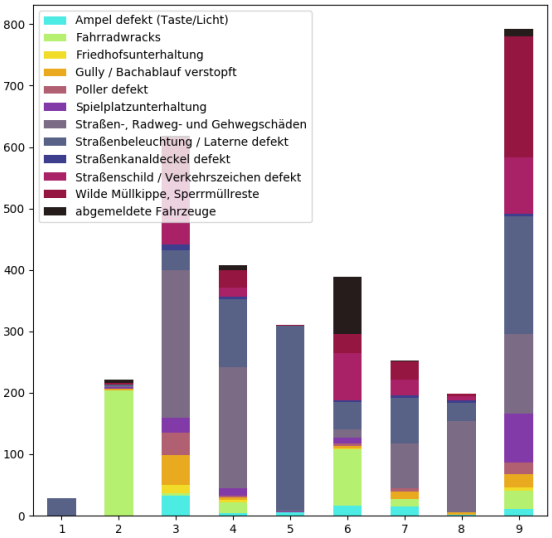
A.5.2 LSA



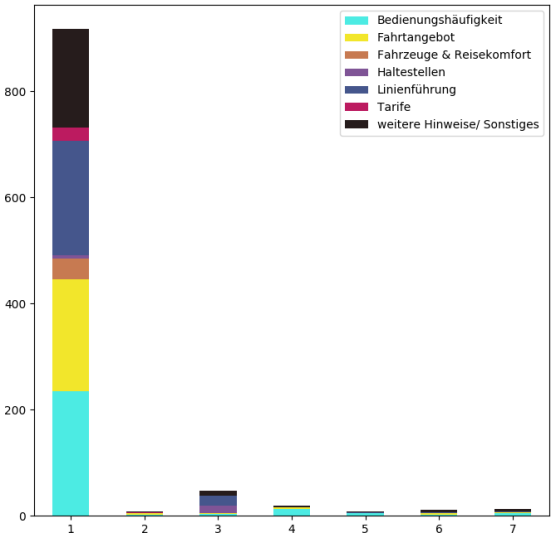
(a) Raddialoge



(b) Raddialoge, Kategorie Radverkehrsführung



(c) Mängelmelder Braunschweig



(d) Nahverkehrsplan Ulm

Abbildung 30: Verteilung der Kategorien auf die LSA-Themen

Thema	Top-Wörter
1	kommen, autofahrer, ampel, fußgänger, geben
2	ampel, weg, grün, kreuzung, geben
3	autofahrer, weg, ampel, fußgänger, gefährlich
4	fußgänger, autofahrer, weg, kreuzung, gefährlich
5	weg, kreuzung, geben, kommen, muss
6	fußgänger, kreuzung, autofahrer, geben, kommen
7	kommen, geben, fußgänger, gefährlich, weg
8	kommen, gefährlich, kreuzung, ampel, fahrbahn
9	muss, geben, gefährlich, ampel, kreuzung
10	gefährlich, geben, str, muss, müssen
11	fahrradfahrer, str, stelle, geben, kommen
12	str, fahrradfahrer, stelle, kreuzung, gefährlich
13	stelle, schmal, fahrbahn, str, schutzstreifen
14	müssen, stelle, geben, unterführung, schmal

(a) Raddialoge

Thema	Top-Wörter
1	kommen, autofahrer, weg, geben, fußgänger
2	weg, autofahrer, kreuzung, geben, hoch
3	autofahrer, kreuzung, str, geben, weg
4	fußgänger, geben, muss, fahrradfahrer, weg
5	kreuzung, weg, kommen, fußgänger, autofahrer
6	kommen, fußgänger, weg, geben, fahrradfahrer
7	gefährlich, fahrradfahrer, radverkehr, stelle, autofahrer
8	str, gefährlich, geben, fahrradfahrer, kreuzung
9	muss, geben, strass, müssen, fahrbahn
10	fahrradfahrer, strass, fußgänger, geben, autofahrer
11	str, fahrradfahrer, geben, kreuzung, unterführung
12	gefährlich, muss, unterführung, schild, fahrbahn
13	stelle, radverkehr, gefährlich, ampel, muss

(b) Raddialoge, Kategorie Radverkehrsführung

Thema	Top-Wörter
1	stehen, fahrradwrack, straße, radweg, monat
2	straße, fahrradwrack, stehen, straßenbeleuchtung, radweg
3	straßenbeleuchtung, ausfallen, radweg, fahrradwrack, gehwegplatte
4	straße, gehwegplatte, fahrradwrack, herr, dame
5	herr, dame, geehrte, fahrradwrack, gehwegplatte
6	gehwegplatte, stehen, fahrradwrack, los, öffentlich
7	straße, radweg, weg, stehen, gehwegplatte
8	schlagloch, gehweg, radweg, spielplatz, höhe
9	schlagloch, radweg, gehweg, befinden, straße

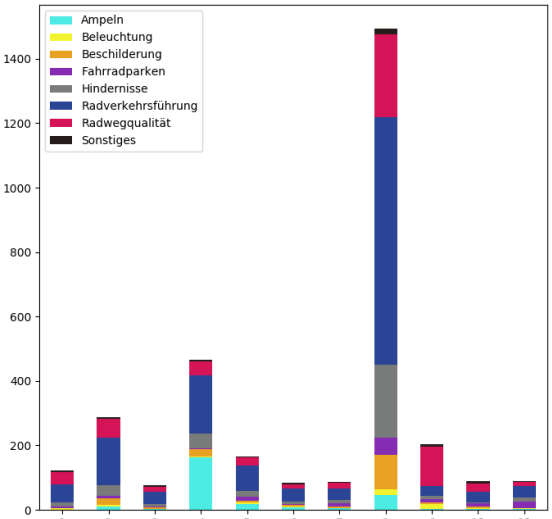
(c) Mängelmelder Braunschweig

Thema	Top-Wörter
1	haltestelle, uni, stadt, uhr, minute
2	uni, süd, lehr, uhr, wiblingen
3	haltestelle, uhr, ehinger, tor, stadt
4	uhr, haltestelle, minute, lehr, uni
5	minute, stadt, uni, haltestelle, ehinger
6	takt, uhr, gut, lehr, straßenbahn
7	wiblingen, minute, ehinger, tor, haltestelle

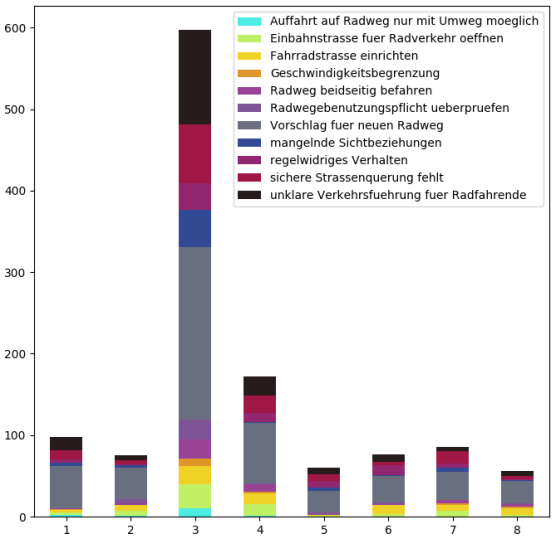
(d) Nahverkehrsplan Ulm

Tabelle 37: Von LSA gefundene Themen mit den zugehörigen Top-Wörtern

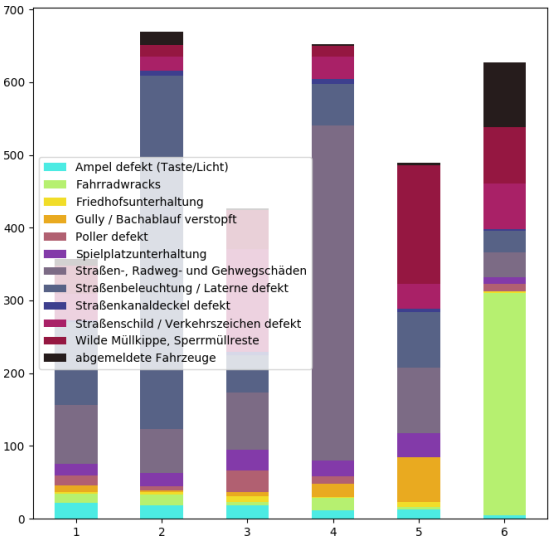
A.5.3 BTM



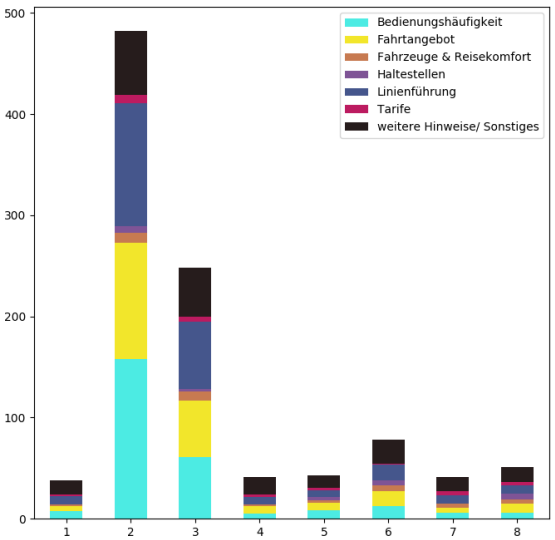
(a) Raddialoge



(b) Raddialoge, Kategorie Radverkehrsführung



(c) Mängelmelder Braunschweig



(d) Nahverkehrsplan Ulm

Abbildung 31: Verteilung der Kategorien auf die BTM-Themen

Thema	Top-Wörter
1	unterführung, verbindung, bonner, gesamt, weg
2	fahrbahn, überholen, stelle, breite, markieren
3	geschwindigkeit, unmittelbar, überholen, kurve, abschüssig
4	ampel, kreuzung, grün, rot, muss
5	stelle, möglich, grün, kreuzung, fahrbahn
6	königswinterer, fahrrad, spät, radverkehr, verbot
7	verbindung, breite, gering, unfall, berücksichtigen
8	kommen, geben, gefährlich, autofahrer, muss
9	schlecht, zustand, schlagloch, wurzel, regen
10	bahn, befahrbar, sinnvoll, möglich, schlagloch
11	bonner, fahrrad, wichtig, gering, bewohner

(a) Raddialoge

Thema	Top-Wörter
1	enden, führen, sehen, kreuzung, mögen
2	verkehr, radverkehr, mögen, fußgängerampel, soll
3	kommen, gefährlich, muss, geben, autofahrer
4	geben, kommen, müssen, platz, muss
5	https, dürfen, anbringen, entstehen, entfernen
6	überholen, zeit, frei, kurzer, benutzen
7	kurve, hoch, breit, kind, radverkehr
8	vorschlag, gruß, verkehr, freundlich, verbund

(b) Raddialoge, Kategorie Radverkehrsführung

Thema	Top-Wörter
1	herr, dame, geehrte, straße, können
2	straßenbeleuchtung, straße, ausfallen, beleuchtung, woche
3	richtung, sehen, können, straße, radfahrer
4	straße, radweg, befinden, gehwegplatte, hoch
5	straße, müll, weg, wasser, liegen
6	stehen, fahrradwrack, monat, jahr, rad

(c) Mängelmelder Braunschweig

Thema	Top-Wörter
1	erreichbar, tatsächlich, diskutieren, abstimmen, eselsberger
2	gut, haltestelle, müssen, stadt, kommen
3	auto, anbindung, uni, lang, stehen
4	alt, blautalcenter, zahl, herr, beitrag
5	problem, hart, busfahrer, wohngebiet, grund
6	weg, böfingen, pendler, mähringen, alt
7	kinderwagen, gemeinsam, effizient, planen, laut
8	sicherheit, stehen, fahrzeit, blick, ziel

(d) Nahverkehrsplan Ulm

Tabelle 38: Von BTM gefundene Themen mit den zugehörigen Top-Wörtern

Literaturverzeichnis

Charu C. Aggarwal, Hrsg. (2015). *Data Classification: Algorithms and Applications*. CRC Press.

Steven Bird, Ewan Klein und Edward Loper (2009). *Natural Language Processing with Python*. O'Reilly Media, Inc.

Christopher M. Bishop (2006). *Pattern Recognition and Machine Learning*. Springer.

David M. Blei, Andrew Y. Ng und Michael I. Jordan (2003). „Latent Dirichlet Allocation". In: *Journal of Machine Learning Research* 3.Jan, S. 993–1022.

Piotr Bojanowski, Edouard Grave, Armand Joulin und Tomas Mikolov (2017). „Enriching Word Vectors with Subword Information". In: *Transactions of the Association for Computational Linguistics* 5, S. 135–146.

Jordan Boyd-Graber und David Blei (2008). „Syntactic Topic Models". In: *Proceedings of the 21st International Conference on Neural Information Processing Systems*. Curran Associates Inc., S. 185–192.

Leo Breiman (1996). „Bagging Predictors". In: *Machine Learning* 24.2, S. 123–140.

Leo Breiman (2001). „Random Forests". In: *Machine Learning* 45.1, S. 5–32.

Jonathan Chang, Sean Gerrish, Chong Wang, Jordan L Boyd-Graber und David M Blei (2009). „Reading Tea Leaves: How Humans Interpret Topic Models". In: *Advances in neural information processing systems*, S. 288–296.

Chun-hung Cheng, Jian Tang, Ada Wai-chee Fu und Irwin King (2001). „Hierarchical Classification of Documents with Error Control". In: *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, S. 433–443.

Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau und Yoshua Bengio (2014). „On the Properties of Neural Machine Translation: Encoder–Decoder Approaches". In: *Syntax, Semantics and Structure in Statistical Translation*, S. 103.

Alexis Conneau, Holger Schwenk, Loïc Barrault und Yann Lecun (2017). „Very Deep Convolutional Networks for Text Classification". In: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*. Association for Computational Linguistics, S. 1107–1116.

Corinna Cortes und Vladimir Vapnik (Sep. 1995). „Support-Vector Networks". In: *Machine Learning* 20.3, S. 273–297.

Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer und Richard Harshman (1990). „Indexing by Latent Semantic Analysis". In: *Journal of the American Society for Information Science* 41.6, S. 391–407.

Ioannis Vlahavas Eleftherios Spyromitros Grigorios Tsoumakas (2008). „An Empirical Study of Lazy Multilabel Classification Algorithms". In: *Proc. 5th Hellenic Conference on Artificial Intelligence (SETN 2008)*.

Shantanu Godbole und Sunita Sarawagi (2004). „Discriminative Methods for Multi-labeled Classification". In: *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, S. 22–30.

Yoav Goldberg (2016). „A Primer on Neural Network Models for Natural Language Processing". In: *Journal of Artificial Intelligence Research* 57, S. 345–420.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin und Tomas Mikolov (2018). „Learning Word Vectors for 157 Languages". In: *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

Rafael Guzmán, Manuel Montes, Paolo Rosso und Luis Villaseñor (2007). „Improving Text Classification by Web Corpora“. In: *Advances in Intelligent Web Mastering*. Hrsg. von Katarzyna M. Wegrzyn-Wolska und Piotr S. Szczepaniak. Springer Berlin Heidelberg, S. 154–159.

Birgit Hamp und Helmut Feldweg (1997). „GermaNet - a Lexical-Semantic Net for German“. In: *Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications*.

Trevor Hastie, Saharon Rosset, Ji Zhu und Hui Zou (2009). „Multi-class AdaBoost“. In: *Statistics and its Interface* 2.3, S. 349–360.

Haibo He und Edwardo A Garcia (2008). „Learning from Imbalanced Data“. In: *IEEE Transactions on Knowledge & Data Engineering* 9, S. 1263–1284.

Sepp Hochreiter und Jürgen Schmidhuber (Nov. 1997). „Long Short-Term Memory“. In: *Neural Comput.* 9.8, S. 1735–1780.

Armand Joulin, Edouard Grave, Piotr Bojanowski und Tomas Mikolov (2017). „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*. Association for Computational Linguistics, S. 427–431.

Yoon Kim (2014). „Convolutional Neural Networks for Sentence Classification“. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, S. 1746–1751.

Diederik P Kingma und Jimmy Ba (2014). „Adam: A Method for Stochastic Optimization“. In: *arXiv preprint arXiv:1412.6980*.

Matthias Liebeck und Stefan Conrad (2015). „IWNLP: Inverse Wiktionary for Natural Language Processing“. In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. Association for Computational Linguistics, S. 414–418.

Matthias Liebeck, Katharina Esau und Stefan Conrad (2017). „Text Mining für Online-Partizipationsverfahren: Die Notwendigkeit einer maschinell unterstützten Auswertung“. In: *HMD Praxis der Wirtschaftsinformatik* 54.4, S. 544–562.

Gilles Louppe und Pierre Geurts (2012). „Ensembles on Random Patches“. In: *Machine Learning and Knowledge Discovery in Databases*, S. 346–361.

Christopher D Manning, Prabhakar Raghavan und Hinrich Schütze (2008). *Introduction to Information Retrieval*. Bd. 39. Cambridge University Press.

Tomas Mikolov, Kai Chen, Greg Corrado und Jeffrey Dean (2013). „Efficient Estimation of Word Representations in Vector Space“. In: *arXiv preprint arXiv:1301.3781*.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot und E. Duchesnay (2011). „Scikit-learn: Machine Learning in Python“. In: *Journal of Machine Learning Research* 12, S. 2825–2830.

Slav Petrov, Dipanjan Das und Ryan McDonald (2012). „A Universal Part-of-Speech Tagset“. In: *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*. European Language Resources Association (ELRA).

Philipp Pollack (2017). „Automatisierte Verschlagwortung und Zuordnung von Büchern in eine Systematik“. Masterarb. Heinrich-Heine-Universität Düsseldorf.

Jesse Read, Bernhard Pfahringer, Geoff Holmes und Eibe Frank (2009). „Classifier chains for multi-label classification“. In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, S. 254–269.

Thomas Scholz und Stefan Conrad (2013). „Opinion Mining in Newspaper Articles by Entropy-based Word Connections“. In: *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, S. 1828–1839.

Thomas Scholz, Stefan Conrad und Isabel Wolters (2012). „Comparing Different Methods for Opinion Mining in Newspaper Articles“. In: *International Conference on Application of Natural Language to Information Systems*. Springer, S. 259–264.

Carlos N Silla und Alex A Freitas (2011). „A survey of hierarchical classification across different application domains“. In: *Data Mining and Knowledge Discovery* 22.1-2, S. 31–72.

Mohammad S Sorower (2010). „A Literature Survey on Algorithms for Multi-label Learning“. In: *Oregon State University, Corvallis* 18.

P. Szymański und T. Kajdanowicz (Feb. 2017). „A scikit-based Python environment for performing multi-label classification“. In: *arXiv preprint arXiv:1702.01460*.

G. Tsoumakas, I. Katakis und I. Vlahavas (Juli 2011). „Random k-Labelsets for Multilabel Classification“. In: *IEEE Transactions on Knowledge and Data Engineering* 23.7, S. 1079–1089.

Sida Wang und Christopher Manning (2012). „Baselines and Bigrams: Simple, Good Sentiment and Topic Classification“. In: *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, S. 90–94.

Jifeng Xuan, He Jiang, Zhilei Ren, Jun Yan und Zhongxuan Luo (2017). „Automatic Bug Triage using Semi-Supervised Text Classification“. In: *arXiv preprint arXiv:1704.04769*.

Xiaohui Yan, Jiafeng Guo, Yanyan Lan und Xueqi Cheng (2013). „A Biterm Topic Model for Short Texts“. In: *Proceedings of the 22nd international conference on World Wide Web*. ACM, S. 1445–1456.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola und Eduard Hovy (2016). „Hierarchical Attention Networks for Document Classification“. In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, S. 1480–1489.

Min-Ling Zhang und Zhi-Hua Zhou (2007). „ML-KNN: A Lazy Learning Approach to Multi-Label Learning“. In: *Pattern Recognition* 40.7, S. 2038–2048.

Abbildungsverzeichnis

1 Verteilung der Beiträge der Raddialoge auf die vorgegebenen Kategorien 7

2 Übersicht über die Anzahl der Schlagwörter, die pro Raddialog-Beitrag vergeben wurden 7

3 Beispiele für Beiträge zum Raddialog Bonn 8

4 Verteilung der Beiträge des Mängelmelder Braunschweig auf die Kategorien 11

5 Verteilung der Beiträge zum Nahverkehrsplan Ulm auf die Kategorien . . 11

6 Übersicht über die Anzahl der Schlagwörter, die pro Beitrag zum Bürgerbudget Wuppertal vergeben wurden 13

7 Konfusionsmatrix für das k-NN-Baseline-Modell auf den Raddialogen . . 16

8 Konfusionsmatrix für den *Human*-Klassifikator auf den Raddialogen . . . 16

9 Konfusionsmatrizen für Multinomial Naive Bayes auf dem Raddialog Bonn 24

10 Konfusionsmatrix für die optimierte logistische Regression auf den Raddialogen 26

11 VDCNN-Architektur nach Conneau et al. (2017) 33

12 Konfusionsmatrix für die FC-Architektur auf den Raddialogen 36

13 Falsch klassifizierte Textbeiträge aus dem Raddialog Bonn mit Hervorhebungen von ELI5 für die vorhergesagte Kategorie 37

14 Konfusionsmatrizen auf dem Raddialog Bonn 38

15 Konfusionsmatrix auf dem Bürgerhaushalt Bonn 39

16 Konfusionsmatrix auf dem Mängelmelder Braunschweig 40

17 Konfusionsmatrix auf dem Nahverkehrsplan Ulm 41

18 Konfusionsmatrix auf dem Leitbild Bad Godesberg 42

19 Lernkurve für die Raddialoge 54

20 Verteilung der Kategorien auf die LDA-Themen bei den Raddialogen . . . 75

21 Verteilung der Beiträge der Kölner Bürgerhaushalte auf die Kategorien . . 83

22 Übersicht über die Anzahl der Schlagwörter, die pro Beitrag zum Bürgerhaushalt Bonn 2011 vergeben wurden 85

23 Verteilung der Beiträge zum Leitbild Bad Godesberg auf die Kategorien . 85

24 Lernkurve für den Bürgerhaushalt Bonn 87

25 Lernkurve für den Mängelmelder Braunschweig 87

26 Lernkurve für den Nahverkehrsplan Ulm 88

27 Konfusionsmatrix für hierarchische Einfachverschlagwortung mit der Logit-Baseline 89

28 Konfusionsmatrix für hierarchische Einfachverschlagwortung mit gegebener Oberkategorie 90
29 Verteilung der Kategorien auf die LDA-Themen 91
30 Verteilung der Kategorien auf die LSA-Themen 92
31 Verteilung der Kategorien auf die BTM-Themen 94

Tabellenverzeichnis

1 Zusammenfassung der Charakteristika der Datensätze 5
2 Verteilung der Beiträge der Bonner Bürgerhaushalte auf die Kategorien . . 10
3 Häufigkeiten der beim Bürgerbudget Wuppertal vergebenen Schlagwörter 13
4 Macro-F_1-Scores der Baselines auf den verschiedenen Datensätzen 17
5 F_1-Scores der optimierten Baseline im Vergleich mit den Baselines auf den verschiedenen Datensätzen 23
6 Durchschnittliche Macro-F_1-Werte der klassischen Klassifikatoren 24
7 F_1-Scores der optimierten logistischen Regression im Vergleich mit den Baselines und k-NN auf den verschiedenen Datensätzen 25
8 Durchschnittliche F_1-Scores von BookGraph 27
9 Durchschnittliche F_1-Scores mit Tonalitätsgraph 28
10 F_1-Scores des Fully-Connected-Netzwerks 29
11 F_1-Scores des CNN 31
12 F_1-Scores des CNN4SC 31
13 F_1-Scores des VDCNN 32
14 F_1-Scores des HAN 34
15 F_1-Scores von FastText 35
16 F_1-Scores der Klassifikatoren, die ein neuronales Netzwerk benutzen, für die jeweils besten Hyperparameter im Vergleich zur logistischen Regression 36
17 Durchschnittliche F_1-Scores mit Germanet-Oversampling 45
18 Ergebnisse des finalen Logit-Klassifikators auf den unterschiedlichen Test-Datensätzen 52
19 Ergebnisse bei Training auf einem anderen Datensatz im Vergleich zum Test-F_1-Score und Human 53
20 F_1-Scores bei Verwendung von semi-supervised Learning nach Xuan et al. (2017) im Vergleich zu den F_1-Werten auf der Testmenge 55
21 F_1-Scores bei Verwendung von semi-supervised Learning mit Wahrscheinlichkeits-Schwellwert für $i = 1$ im Vergleich zum F_1-Score auf dem Testset 56
22 Performance der Single-Tag-Baseline 61
23 Performance der Problemtransformationsverfahren im Vergleich zur Baseline 62
24 Performance der adaptierten Verfahren im Vergleich zur Baseline 63
25 Performance der RAkEL-Klassifikatoren im Vergleich zur Baseline und BR 64

26 Performance der Verschlagwortung mit BookGraph im Vergleich zur Baseline 64
27 Vergleich der Performance der besten betrachteten Multi-Label-Verfahren 65
28 Performance auf den Testsets für die besten betrachteten Multi-Label-Verfahren 65
29 F_1-Scores für Hierarchie-basierte Verschlagwortung im Vergleich zur Baseline 68
30 F_1-Scores für Hierarchie-basierte Verschlagwortung bei gegebener Kategorie im Vergleich zur Baseline 69
31 Von LDA gefundene Themen mit den zugehörigen Top-Wörtern 74
32 F_1-Scores für die Klassifikation mithilfe der Themenzugehörigkeiten im Vergleich zur logistischen Regression mit Charakter-N-Grammen 78
33 Häufigkeit der bei den Raddialogen vergebenen Schlagwörter 81
34 Häufigkeiten der beim Bürgerhaushalt Bonn 2011 vergebenen Schlagwörter 84
35 Gridsearch-Parameter für k-NN 86
36 Gridsearch-Parameter für logistische Regression 86
37 Von LSA gefundene Themen mit den zugehörigen Top-Wörtern 93
38 Von BTM gefundene Themen mit den zugehörigen Top-Wörtern 95