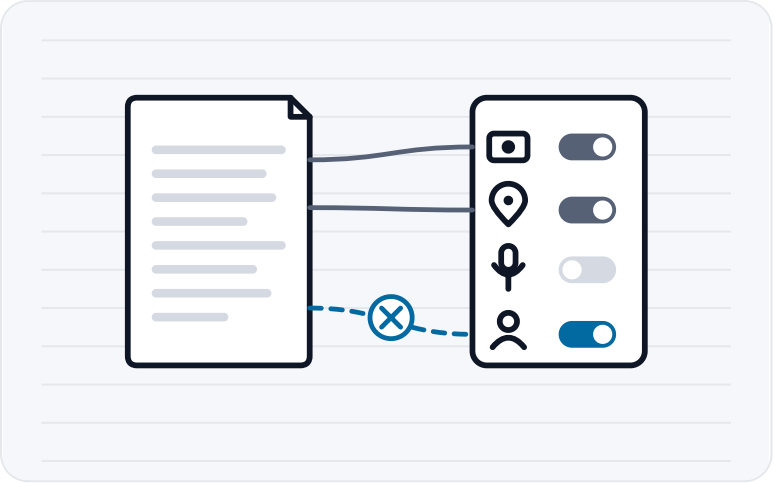
A-SIT Technology
E-Government Innovationszentrum

15. Februar 2023 Analyse · Mobile Sicherheit · Privatsphäre & Krypto · 22 Min. Lesezeit

Erkennen von Inkonsistenzen zwischen Android-App-Beschreibungen und Berechtigungen

Emina Ahmetovic



Autorinnen und Autoren	Emina Ahmetovic
Stand	2023-02-03
Quelle	https://technology2.a-sit.at/veroeffentlichungen/inkonsistenzen-zwischen-app-beschreibungen-und-berechtigungen/
Hinweis	Maschinell übersetzt.

Kernaussagen

- Der Datensatz umfasst 46 431 App-Metadaten aus 49 Kategorien; ursprünglich waren es 80 000, nicht englische oder zu kurze Beschreibungen wurden entfernt.
- Das auf BERT basierende Transformer-Modell sagt Berechtigungsgruppen aus der App-Beschreibung vorher: Micro-Precision 70 %, Recall 67 %, F1 69 %.
- Die besten F1-Werte erreicht das Modell für Location und Storage (je 0,78), den niedrigsten für Calendar (0,33).
- Die Ergebnisse sind mit früheren Arbeiten auf Basis von RNN und CNN vergleichbar.

- Als Einschränkung nennt der Bericht die Datensatzgröße; mehr Daten würden das Transformer-Modell verbessern.

Android-Benutzern steht eine große Anzahl von Apps zur Verfügung, die eine Vielzahl von Funktionen und Hilfestellungen für das tägliche Leben bieten. Während die Funktionalität der Anwendungen einen starken Einfluss auf die Privatsphäre des Nutzers haben kann, werden Berechtigungen als Mechanismus eingeführt, der das Vermögen des Nutzers schützt, indem beim Zugriff auf datenschutzrelevante Daten die ausdrückliche Zustimmung eingeholt wird. Dennoch fällt es den Nutzern oft schwer, eine Verbindung zwischen den angeforderten Berechtigungen und der Beschreibung der Anwendung herzustellen.

Zuverlässig zu erkennen, ob die Notwendigkeit einer Genehmigung gerechtfertigt ist, ist eine anspruchsvolle Aufgabe, die wir in diesem Projekt angehen wollen. Wir schlagen einen neuartigen Ansatz des maschinellen Lernens vor, der das Verhalten von Apps auf der Grundlage der von den Entwicklern bereitgestellten Informationen vorhersagt. Wir erstellen einen Datensatz mit 46 000+ App-Beschreibungen und -Erlaubnissen. Darüber hinaus entwickeln wir ein Modell, das mit Hilfe eines hochmodernen Transformers erkennt, ob und in welchem Umfang die Notwendigkeit einer Erlaubnis in der Beschreibung der App dargelegt wird.

1. Einleitung

Mit 3,55 Millionen Android-Apps - einer Zahl, die voraussichtlich weiter wächst - ist Google Play der führende App-Markt mit über 2,8 Milliarden aktiven Nutzern und einem weltweiten Marktanteil von 75 Prozent[1]. Ohne Zweifel sind Apps im Alltag allgegenwärtig. Die Erfolgsgeschichte macht Android-Apps jedoch zu einem häufigen Ziel von Schadanwendungen, die von den privaten Daten der Nutzer profitieren. Die Beliebtheit mobiler Geräte im Alltag führt über Berechtigungsmodelle häufig zu Datenschutzvorfällen auf mobilen Geräten. Während Apps tendenziell mehr Funktionen anbieten, haben sie auch mehr Möglichkeiten, in die Privatsphäre der Nutzer einzudringen. Sehr oft haben Nutzer die anspruchsvolle Aufgabe, Berechtigungsanfragen ihrer Apps abzulehnen oder zuzulassen. Die Auswirkungen von Apps auf die Privatsphäre stehen im Fokus, vor allem seit den Datenschutzverletzungen, die durch Apps verursacht wurden. Wie frühere Studien warnen, kann das Gewähren bestimmter Berechtigungsanfragen zu Datenschutzverletzungen führen, die den Nutzern nicht bewusst sind (Felt A. P., 2011) (Felt A. P., 2012) (Li, 2021) (Almuhimedi, 2015). Während ein großer Teil der Nutzer die Auswirkungen von Berechtigungen auf die Privatsphäre versteht, fällt es ihnen schwer, richtig einzuschätzen, ob eine Berechtigung für die jeweilige App notwendig ist. In vielen Fällen finden Nutzer keine Erklärung für die Verwendung von Berechtigungen, die ihre Daten stark beeinträchtigen können (Liu B. a., 2016), (Shen, 2021).

In diesem Projekt wollen wir beurteilen, in welchem Umfang die beanspruchten Berechtigungen den Nutzern tatsächlich vermittelt werden. Berechtigungen sind einer der wichtigsten Sicherheitsmechanismen eines mobilen Betriebssystems und spielen eine wichtige Rolle beim Schutz der Nutzerdaten. Wenn Nutzer bestimmte Berechtigungen erteilen, erhalten Apps Zugriff auf die privaten Daten der Nutzer. Aus diesem Grund vertreten wir die Ansicht, dass Apps den Zugriff auf private Ressourcen erklären sollten und dass Nutzer gefährliche Berechtigungen gut verstehen sollten. Daher untersuchen wir die App-Beschreibungen,

um die deklarierten Berechtigungen vorherzusagen. Diese Untersuchung könnte auch mit anderen Arten von Metadaten durchgeführt werden. Wir konzentrieren uns jedoch auf die App-Beschreibung als informellen, nutzerfreundlichen Kanal, der Nutzern helfen kann, die Funktionen der App für Privatsphäre und Sicherheit besser zu verstehen. Wir vertreten die Ansicht, dass Beschreibungen das geeignete Mittel sind, um Nutzern die Verwendung gefährlicher Berechtigungen zu erläutern.

1.1 Beitrag

In diesem Projekt führen wir eine Analyse von Anwendungsbeschreibungen durch, um Inkonsistenzen zwischen Beschreibung und Berechtigungen in Android-Apps zu finden. Unsere Studie unterscheidet sich von früheren, weil wir nach unserem besten Wissen die Ersten sind, die ein hochmodernes neuronales Transformer-Modell für die Prüfung der Übereinstimmung zwischen Berechtigung und Beschreibung verwenden. Zusammenfassend besteht unser Beitrag aus Folgendem:

- Wir haben einen Crawler implementiert, um einen Datensatz mit 46 000+ Anwendungsbeschreibungen und Berechtigungen zu erstellen. Unsere Suche basierte auf den beliebtesten Anwendungen aus 49 Kategorien.
- Wir entwerfen und implementieren das tiefe neuronale Netz, das das Multiklassifikationsproblem löst und die Berechtigung anhand des Textdokuments vorhersagt. Dafür verwendeten wir die Transformer-Architektur mit dem vortrainierten Modell BERT.
- Schließlich evaluieren wir unser Modell an realen Apps anhand der sieben Berechtigungsgruppen, die der gefährlichen Stufe zugeordnet sind.

Der Rest dieser Studie ist wie folgt aufgebaut: In Abschnitt 2 erklären wir den Mechanismus der Berechtigungen und welche Arten von Berechtigungen wir in diesem Bericht behandeln. In Abschnitt 3 geben wir einen kurzen Überblick über die Grundlagen des tiefen neuronalen Netzes vom Typ Transformer. In Abschnitt 4 erklären wir, wie wir einen Datensatz erstellt haben. In Abschnitt 5 stellen wir die Ergebnisse unserer Evaluierung dar. In Abschnitt 6 diskutieren wir die Einschränkungen und die zukünftige Arbeit. In Abschnitt 7 geben wir einen Überblick über verwandte Arbeiten.

2. Hintergrund

Android-Anwendungen werden zentral verteilt. Das bedeutet, dass Entwickler zusätzlich zu den Apps auch Metadaten wie Beschreibungen, Bewertungen und Ähnliches veröffentlichen. Diese Metadaten wurden zusammen mit NLP- und Machine-Learning-Techniken in der Analyse von Anwendungsmetadaten verwendet. Die neuartige Analyse hat im Gegensatz zu anderen Analysearten wie der statischen oder dynamischen Analyse Erfolge bei der Prüfung der Description-to-Permission-Fidelity gezeigt. Das ist ein Begriff aus der Literatur dafür, wie gut das beschriebene Verhalten einer App mit ihrem tatsächlichen Verhalten übereinstimmt. Viele Studien haben solche Analysen durchgeführt. Die Idee hinter der Studie bezieht sich auf Beschreibungen als wichtigen Kommunikationskanal zwischen Entwicklern und Endnutzern.

In den folgenden Abschnitten stellen wir die für unsere Studie relevanten Hintergrundtechniken vor.

2.1 Berechtigungen unter Android

Das Berechtigungsmodell wurde als Mechanismus eingeführt, der Nutzern die Kontrolle über ihre Daten gibt und damit ihre Privatsphäre schützt. Eine App muss die ausdrückliche Erlaubnis eines Nutzers einholen, um auf eine bestimmte Ressource zuzugreifen oder sie zu verwenden. Je nach Umfang der eingeschränkten Aktionen und des Zugriffs auf die eingeschränkten Ressourcen definiert Android verschiedene Berechtigungstypen: gefährliche Berechtigungen oder Laufzeitberechtigungen und normale Berechtigungen oder Installationszeit-Berechtigungen. Installationszeit-Berechtigungen [2] führen Aktionen mit minimalen Auswirkungen auf andere Apps oder das System aus. Daher werden sie vom System automatisch bei der Installation gewährt (zu dem Zeitpunkt, an dem ein Nutzer die App installiert). Die Installationszeit-Berechtigungen werden den Nutzern standardmäßig nicht angezeigt, sondern sind in den Details im Play Store zu finden. Die normalen Berechtigungen und die Signaturberechtigungen gehören zu den Installationszeit-Berechtigungen.

Laufzeitberechtigungen [3] wie SMS, CONTACTS, CALENDAR, LOCATION und ähnliche sind hingegen der gefährlichen Schutzstufe zugeordnet, da sie Apps den Zugriff auf eingeschränkte Daten und die Ausführung eingeschränkter Operationen ermöglichen. Sie werden zur Laufzeit angefordert (unmittelbar vor der Ausführung der eingeschränkten Aktion) und erfordern die ausdrückliche Zustimmung des Nutzers. Im Gegensatz zu den Installationszeit-Berechtigungen können Nutzer Laufzeitberechtigungen jederzeit widerrufen.

3. Attention und Transformer-Modell

Rekurrente neuronale Netze (Graves, 2012) wie Long Short-Term Memory und Gated Recurrent Neural Networks waren viele Jahre lang der Stand der Technik bei Sequenzmodellierungsaufgaben wie Sprachmodellierung und maschineller Übersetzung. Sie sind jedoch bekannt für ihre inhärenten Einschränkungen bei der Verarbeitung von Sequenzen. Das ist einer der Gründe, warum sie durch den Transformer ersetzt werden. Der Transformer (Vaswani, 2017) ist ein neuronales Netzmodell, das den Self-Attention-Mechanismus verwendet, der im Gegensatz zu RNNs und CNNs eine globale Attention zwischen Eingabe und Ausgabe definiert. Das bedeutet, dass bei Self-Attention jedes Wort im Satz auf jedes andere Wort im Satz achtet. Wie rekurrente neuronale Netze verarbeiten auch Transformer die Eingabedaten sequenziell. Im Gegensatz zu RNNs verarbeiten Transformer jedoch die gesamten Eingabedaten auf einmal, was in Matrixform hochgradig parallelisierbar ist und die Trainingszeit verkürzt. Der Transformer basiert auf einer Encoder-Decoder-Architektur, wie in Abbildung 1 dargestellt.

Der Encoder besteht aus zwei Schichten: Multi-Head-Self-Attention und einem vollständig verbundenen Feed-Forward-Netz. Zu Beginn erhält er als Eingabeinformation die Positionskodierung, die die Reihenfolge der Sequenz enthält, und das Input-Embedding. In jeder Schicht wird aus dem vorherigen Encoder eine Eingabekodierung erzeugt und dem Self-Attention-Mechanismus zugeführt, der deren Relevanz gewichtet und die Ausgabekodierung erzeugt. Die Ausgabekodierung wird an das Feed-Forward-Netz weitergegeben, das die Ausgabekodierung einzeln berechnet. Während der Encoder eine Kodierung erzeugt und an den nächsten Encoder weitergibt, empfängt der Decoder die Kodierung und erzeugt die 2 https://developer.android.com/guide/topics/permissions/overview#install-time 3 https://developer.android.com/guide/topics/permissions/overview#runtime Ausgabesequenz. Der Decoder besteht aus drei Schichten: Multi-Head-Self-Attention, vollständig verbundenem Feed-Forward-Netz und

Encoder-Decoder-Self-Attention. Die Architektur des Decoders ähnelt der des Encoders. Er hat jedoch eine zusätzliche Schicht, die Multi-Head-Attention über die von den Encodern erzeugten Kodierungen durchführt.

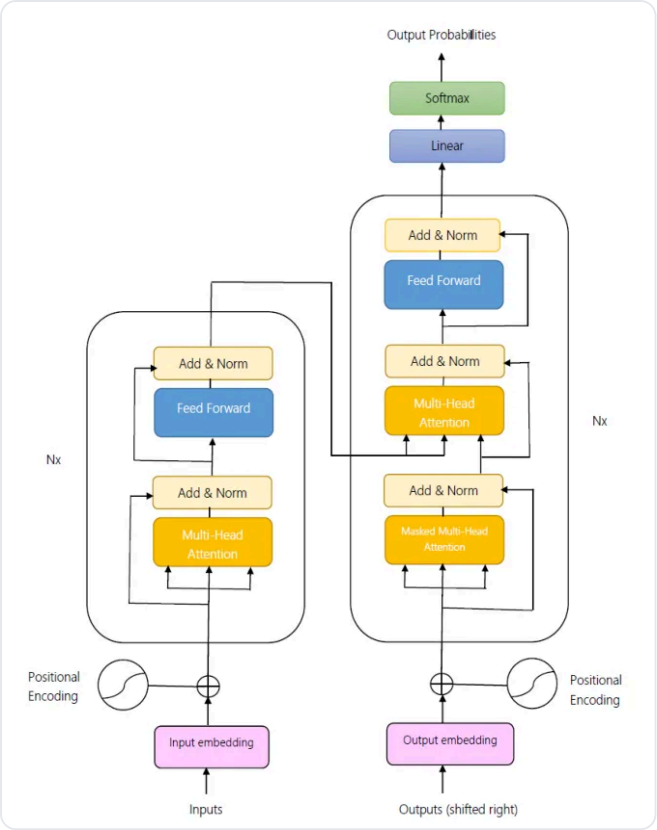


Abbildung 1. Die Encoder-Decoder-Architektur des Transformers.

3.1 Attention-Mechanismus

Das traditionelle Seq-to-Seq-Modell, das aus einem Encoder und einem Decoder auf Basis von GRU- oder LSTM-Einheiten besteht, zielte allgemein gesprochen darauf ab, die Eingabesequenz beliebiger Länge in eine Ausgabesequenz ebenfalls beliebiger Länge zu überführen (Cho, 2014). Der Nachteil dieses Ansatzes war jedoch die Unfähigkeit, sich lange Sätze zu merken. Bei der Textgenerierung zum Beispiel wird der Eingabeteil vergessen, wenn der Abstand zwischen Ausgabe und Eingabe zu groß ist. Der Attention-Mechanismus entstand als effizienter Mechanismus, um das Problem großer Abstände zwischen Quelle und Ziel zu lösen. Attention bedeutet grob gesagt, auf die verschiedenen Wörter in einem Satz zu achten. Im Gegensatz zu den Gated RNNs, bei denen ein Zustandsvektor eine Repräsentation der Daten vor dem aktuellen Token enthält, kann die Attention-Schicht auf alle vorherigen Zustände zugreifen und sie nach Relevanz gewichten. Dank der Abkürzungen (Bahdanau, 2014), (Luong, 2015), die zwischen dem Kontextvektor und der gesamten Quelleingabe geschaffen werden, können deren Gewichte für jedes Ausgabeelement einzeln berechnet werden. Während ein Attention-Mechanismus bereits zusammen mit der RNN-Architektur verwendet wurde, stützt sich der Transformer, der die rekurrenten Schichten ersetzt, ausschließlich auf den Attention-Mechanismus und erzielt bei vielen NLP-Aufgaben wie der Textübersetzung hochmoderne Ergebnisse.

4. Datensatz

Um einen Datensatz der Beschreibungen zu erstellen, haben wir einen Crawler implementiert, der die Metadaten der beliebtesten Anwendungen abruft. Neben den für unsere Studie notwendigen Beschreibungen und Berechtigungen haben wir auch zusätzliche Informationen wie Anwendungs-ID, Bewertung, Datensicherheitsabschnitt, Ranking, Kategorie und Ähnliches sowie Datenschutzerklärung erfasst. Ein JSON-Objekt unten zeigt ein Beispiel der abgerufenen App-Metadaten.


```
{ "genreID": "SHOPPING", "score": "0.0", "ratings": "0", "permissions": [
"Location" ], "appID": "com.fressnapf.mobileapp", "description": "
'Maxizoo/Fressnapf is the partner for you and your pet - right in your pocket. The
most important features in the app: - Save 5% on every purchase in the store and
online with the Friends discount*.
- Save easily with exclusive coupons in our stores - Shop online and find and order
your favorite products Create an individual profile for your pet - Exciting content
and articles about your pet - Find your favorite store near you and have all the
offers from your store at your fingertips Maxizoo/Fressnapf has been the contact
for all pet-related questions since 1990. Customers and their pets trust us because
we also love pets and are committed to their well-being. In our stores, you&#39;ll
find everything you need for pets, whether it&#39;s pet food or pet accessories.
With the exclusive coupons, you can also save money easily and conveniently. In our
online store, you can access our extensive product range at any time from anywhere.
Whether in the store or online: your pet is always our priority. *See terms and
conditions per country: Germany: https://www.fressnapf.de/friends/   - Austria:
https://www.fressnapf.at/friends/   - France: https://www.maxizoo.fr/friends/ -
Poland: https://www.maxizoo.pl/friends/ For suggestions for improvement, or if you
have problems with our app, you can contact us at app-android.Team@fressnapf.com.
Thank you and happy shopping! Your Maxizoo/Fressnapf App Team',\n ", "title":
"Maxizoo / Fressnapf", "privacy policy url":
"https://www.fressnapf.de/app/datenschutzhinweise/" }
```


Der ursprüngliche Datensatz enthielt 80 000 Anwendungsmetadaten. Nach dem Entfernen der nicht englischsprachigen oder zu kurzen Beschreibungen verwendeten wir jedoch 46 431 App-Metadaten aus insgesamt 49 Kategorien. Die Verteilung der Anwendungen auf die Kategorien ist in der folgenden Tabelle dargestellt.

SHOPPING	**1310**	**AUTO_AND_VEHICLES**	**1168**
ENTERTAINMENT	1062	NEWS_AND_MAGAZINES	1142
SPORTS	1138	MEDICAL	1143
VIDEO_PLAYERS	837	HOUSE_AND_HOME	1056
SOCIAL	873	COMICS	401
PHOTOGRAPHY	841	GAME_PUZZLE	886
ART_AND_DESIGN	895	GAME_BOARD	854
FINANCE	1190	GAME_EDUCATIONAL	804
COMMUNICATION	909	PARENTING	683
BOOKS_AND_REFERENCE	948	GAME_ARCADE	849
MAPS_AND_NAVIGATION	1187	GAME_CASUAL	863
HEALTH_AND_FITNESS	1150	GAME_SIMULATION	864
TOOLS	1097	LIBRARIES_AND_DEMO	443
WEATHER	1001	GAME_WORD	827
FOOD_AND_DRINK	1445	GAME_TRIVIA	856
BUSINESS	1296	GAME_RACING	828
EVENTS	611	GAME_ADVENTURE	853
MUSIC_AND_AUDIO	932	GAME_SPORTS	890
LIFESTYLE	1175	GAME_ACTION	811
TRAVEL_AND_LOCAL	1257	GAME_MUSIC	853
EDUCATION	1037		

Tabelle 1. Anwendungskategorien und die Anzahl der Apps, die zur jeweiligen Gruppe gehören.

4.1 Berechtigungen im Datensatz

Ziel unserer Studie ist es, Inkonsistenzen zwischen den von der App deklarierten Berechtigungen und der Beschreibung der Anwendung zu finden. Wir berücksichtigen nur Berechtigungen, die der gefährlichen Stufe zugeordnet sind, wegen ihrer möglichen Auswirkungen auf die Privatsphäre. Der Vorteil, Berechtigungsgruppen statt einzelner Berechtigungen zu betrachten, liegt darin, dass Berechtigungen auf Gruppenebene gewährt werden. Benötigt eine App READ_EXTERNAL_STORAGE, muss der Nutzer also die gesamte Gruppe Storage gewähren, da diese von der App angefordert wird. Wir evaluieren unsere Lösung anhand von sieben Berechtigungsgruppen, die in Tabelle 2 dargestellt sind.

calendar	READ_CALENDAR WRITE_CALENDAR	1672
camera	CAMERA	14472
contacts	READ_CONTACTS WRITE_CONTACTS GET_ACCOUNTS	6729
location	ACCESS_FINE_LOCATION ACCESS_COARSE_LOCATION ACCESS_BACKGROUND_LOCATION ACCESS_MEDIA_LOCATION	16432
microphone	RECORD_AUDIO	7407
phone	READ_PHONE_NUMBERS READ_PHONE_STATE CALL_PHONE READ_CALL_LOG WRITE_CALL_LOG ADD_VOICEMAIL USE_SIP PROCESS_OUTGOING_CALLS ANSWER_PHONE_CALLS ACCEPT_HANDOVER	10668
storage	READ_EXTERNAL_STORAGE WRITE_EXTERNAL_STORAGE	28907

Tabelle 2. Gefährliche Berechtigungsgruppen und die einzelnen Berechtigungen, die zur Gruppe gehören. Die dritte Spalte gibt die Anzahl der Berechtigungsanfragen im Datensatz an.

Die Verteilung der Berechtigungen über den gesamten Datensatz zeigt, dass die häufigste Berechtigungsanfrage Storage betrifft, gefolgt von Location und Camera. Die seltenste Berechtigungsanfrage betrifft Calendar. Abbildung 2 zeigt die Verteilung der Berechtigungen in unserem Datensatz. Anfangs haben wir zwei zusätzliche Berechtigungen einbezogen, nämlich SMS und Sensors. Wegen ihrer sehr seltenen Verwendung haben wir uns jedoch entschieden, sie aus unserer Studie auszuschließen.

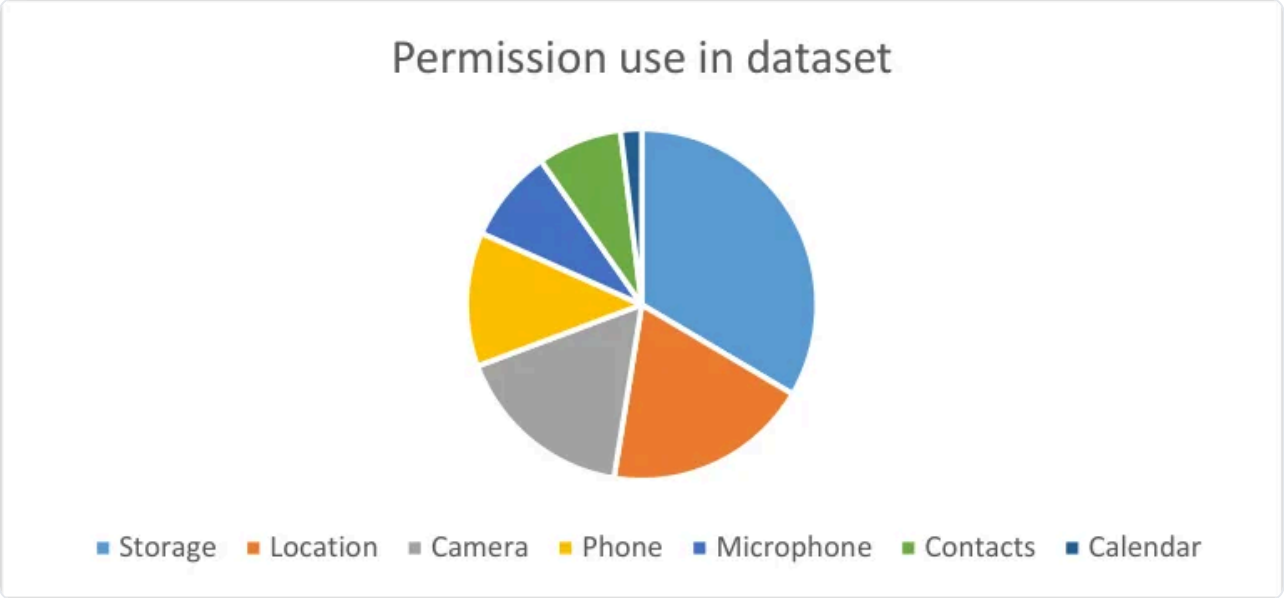


Abbildung 2. Die Verwendung der Berechtigungen im Datensatz.

5. Evaluierung

Wir formulieren unser Problem als Multi-Label-Klassifikationsproblem, bei dem die Eingabe ein Anwendungsbeschreibungstext und die Ausgabe ein One-Hot-kodierter Vektor ist. Wir kodieren den Vektor One-Hot und setzen Einsen an den Stellen, an denen die Anwendung eine Berechtigung anfordert, und 0, wenn die App keine Berechtigung anfordert. Ein Beispiel für Ein- und Ausgaben:

Eingabetext: “Product Features Amazon Shopping offers apponly benefits to help make shopping on Amazon faster and easier than shopping on your desktop Never miss a delivery Get realtime tracking and delivery notifications so you know where your package is and when it arrives Know exactly what youre purchasing Full 360 product view lets you see items from every angle View in you room makes sure it fits by using your phones camera and VR so you can see it in your space Well notify you when items go on sale Just tap the heart icon to save items to Your Lists and well alert you of price drops so you dont miss a deal Never forget your password Save time by staying securely signed in If you prefer to sign out use facial or fingerprint identification to sign back in Connect with us when it works best for you Live chat support is open 24 hours 7 days a week Once youve started a chat it stays that way for 24 hours so you dont have to start your support session from the beginning Well find that item for you Not sure of an items brand or where to but it Just tap the scan icon in the search bar take a picture of the item or its barcode and well find it for you Product Description Browse search view product details read reviews and purchase millions of products We deliver to 100 countries in as quickly as 35 days Whether youre buying gifts reading reviews tracking orders scanning products or just shopping Amazon Shopping app offers more benefits than shopping on Amazon via your desktop Important Note Regarding Permissions Please note that the Amazon Shopping app requires access to the following services to operate properly Contacts Allows you to send Amazon gift cards to your contacts or invitation to install the Amazon app Camera Allows the Amazon app to access your camera on the device You can use your camera to find products by scanning the cover or its barcode to add gift cards and credit cards or to add pictures in the product reviews Flashlight Allows the Amazon app to turn on the flashlight You can use the flashlight to find products with the camera feature even in lowlight or dark conditions Microphone Allows the Amazon app to access your microphone to use your voice to search and interact with your Assistant Location Allows the Amazon app to access your location to help you discover local offers and select addresses fast Account Allows you to share products on Amazon with your friends and families through Facebook or other social networks Phone Allows the Amazon app to prepopulate the Amazon Customer Service number on your phones keypad Storage Allows the Amazon app to store your preferences so that some features can load and run faster on the device WiFi This permissions is used when setting up either a Dash Button or Dash Wand using the Amazon Shopping app The Amazon App for Tablets is available on Google Play Search for quotAmazon Tabletquot to install the app and begin shopping For customers located within the European Union United Kingdom Brazil or Turkey By using this app you agree to Amazons Conditions of Use applicable for your country Please also see the applicable Privacy Notice Cookies Notice and InterestBased Ads Notice for your country Links to these terms and notices can be found in the footer of your local Amazon homepage For all other customers By using this app you agree to the applicable Amazon Conditions of Use eg wwwamazoncomconditionsofuse and Privacy Notice eg wwwamazoncomprivacy for your country Links to these terms and notices can be found in the footer of your local Amazon homepage”

One-Hot-Kodierung: [0, 1, 1, 1, 1, 1, 1], wobei die Labels jeweils [‘Calendar’, ‘Camera’, ‘Contacts’, ‘Location’, ‘Microphone’, ‘Phone’, ‘Storage’] sind.

Bevor wir die Daten verarbeiten, führen wir eine Vorverarbeitung der Anwendungsbeschreibung durch. Die Vorverarbeitung besteht darin, Text zu entfernen, der HTML-Tags und andere Sonderzeichen wie Emojis enthält. Außerdem teilen wir die Daten in Trainings-, Validierungs- und Testsätze auf, die jeweils 80-10-10 Prozent der ursprünglichen Daten enthalten. Wir wenden den Transfer-Learning-Ansatz auf das BERT-Modell[2] mit den folgenden Hyperparametern an:

Hyperparameter	Wert	Beschreibung
num_train_epochs	10	Anzahl der Epochen, über die unser Modell trainiert wird; eine Epoche ist ein Durchlauf über den gesamten Trainingsdatensatz
gradient_accumulation_steps	2	Anzahl der Schritte, über die Gradienten akkumuliert werden; wir akkumulieren die Gradienten über 2 Schritte und wenden sie auf unsere Lernparameter an
manual_seed	43	Der manuelle Seed zum Zweck der Reproduzierbarkeit
max_seq_length	512	Die maximale Anzahl an Wörtern in der Beschreibung
use_early_stopping	True	Um Überanpassung zu verhindern, verwenden wir Early Stopping
early_stopping_delta	0.01	Wir prüfen, ob sich unsere Metriken auf den Validierungsdaten um mindestens 0.01 von unserer Metrik auf den Trainingsdaten unterscheiden
early_stopping_metric	mcc	Der Matthews-Korrelationskoeffizient ist die Metrik, die für Early Stopping verwendet wird
early_stopping_metric_minimize	False	Wir wollen die Metrik maximieren
early_stopping_patience	3	Wird Überanpassung erkannt, warten wir 3 Schritte, bis das Training gestoppt wird
save_model_every_epoch	False	Zur Speicheroptimierung speichern wir das Modell nicht in jeder Epoche

Tabelle 3. Die Tabelle zeigt die Hyperparameter, die wir in unserem Modell verwendet haben.

Die vorherigen Hyperparameter sind spezifisch für die Simple-Transformers-Bibliothek[3], die für das Modelltraining verwendet wird. Wir erhalten folgende Ergebnisse auf dem Testsatz:

	precision	recall	F1-score	support
Calendar	0.32	0.35	0.33	135
Camera	0.72	0.76	0.74	1428
Contacts	0.47	0.55	0.51	660
Location	0.80	0.75	0.78	1640
Microphone	0.55	0.61	0.58	761
Phone	0.49	0.55	0.52	1080
Storage	0.92	0.68	0.78	2880
Micro avg	0.70	0.67	0.69	8548
Macro avg	0.61	0.61	0.61	8548

Tabelle 4. Die Tabelle zeigt die erzielten Ergebnisse.

Da unser Datensatz nicht ausgewogen ist, sind Genauigkeitswerte (Accuracy) möglicherweise keine gute Metrik und würden zu einer irreführenden Interpretation führen. Um die erzielten Ergebnisse richtig zu interpretieren, verwenden wir stattdessen die folgenden Metriken:

- Precision misst, wie viele der gefundenen True Positives korrekt sind (True Positives / (True Positives + False Positives))
- Recall oder Sensitivität misst die Anzahl der True Positives im Verhältnis zu allen Positiven in den Daten (True Positives / (True Positives + False Negatives))
- F1-Score als harmonisches Mittel aus Precision und Recall
- Support als Anzahl aller True Positives, die das Modell identifiziert hat
- Micro Average berechnet die durchschnittlichen Metriken über die Ergebnisse aller Klassen
- Macro Average berechnet die Metriken unabhängig für alle Klassen und mittelt die Ergebnisse

Aus den obigen Ergebnissen können wir schließen, dass unsere Precision zwischen 61 % und 70 % liegt. Wir stellen außerdem fest, dass wir die besten Ergebnisse für Storage und Location erzielen, die mit Abstand die am häufigsten verwendeten Berechtigungen sind. Den niedrigsten Wert haben wir hingegen für Calendar, wegen der seltenen Verwendung. Der Macro-Wert für F1 ist der niedrigste, was darauf hindeutet, dass der Datensatz bis zu einem gewissen Grad unausgewogen ist.

5.1 Vergleich mit anderen Modellen

Die früheren Arbeiten, die darauf abzielen, die Description-to-Permission-Fidelity zu berechnen, wurden hauptsächlich mit verschiedenen Deep-Learning-Architekturen auf Basis von rekurrenten oder faltenden neuronalen Netzen durchgeführt. Der Vergleich anderer Modelle mit unserem ist aus mehreren Gründen

schwierig: Die Autoren machen ihren Datensatz nicht öffentlich, der vorhandene Datensatz ist veraltet oder die Anzahl der Berechtigungen unterscheidet sich. In Tabelle 3 stellen wir jedoch drei frühere Frameworks dar, die unterschiedliche Modelle verwendeten, sowie ihre jeweiligen Ergebnisse.

Col1	Apps	Berechtigungen	Modell	Ergebnisse	Col6	Col7	Col8
AC-Net: Assessing the Consistency of Description and Permission in Android Apps (Feng, 2019)	1 415	16	RNN GRU	ROC-AUC	ROC-AUC	0.974	0.974
AC-Net: Assessing the Consistency of Description and Permission in Android Apps (Feng, 2019)	1 415	16	RNN GRU	PR-AUC	PR-AUC	0.669	0.669
FCDP: Fidelity Calculation for Description-to-Permissions in Android Apps (Wu, 2020)	64 265	16	RNN LSTM	ROC-AUC	ROC-AUC	0.9679	0.9679
FCDP: Fidelity Calculation for Description-to-Permissions in Android Apps (Wu, 2020)	64 265	16	RNN LSTM	PR-AUC	PR-AUC	0.5992	0.5992
Understanding Privacy Awareness in Android App Descriptions Using Deep Learning (Feichtner, 2020)	77 758	9	CNN		Precision	Recall	F1 Score
Understanding Privacy Awareness in Android App Descriptions Using Deep Learning (Feichtner, 2020)	77 758	9	CNN	Micro	81%	65%	77%
Understanding Privacy Awareness in Android App Descriptions Using Deep Learning (Feichtner, 2020)	77 758	9	CNN	Macro	77%	56%	70%

Col1	Apps	Berechtigungen	Modell	Ergebnisse	Col6	Col7	Col8
Unser Modell	46 000+	7	Transformer		Precision	Recall	F1 Score
Unser Modell	46 000+	7	Transformer	Micro	70%	67%	69%
Unser Modell	46 000+	7	Transformer	Macro	61%	61%	61%

Tabelle 5. Vergleich der Ergebnisse früherer Arbeiten.

Wir sehen, dass unsere Ergebnisse mit den bestehenden vergleichbar sind. Der Transformer bietet im Gegensatz zu CNN oder RNN zusätzliche Vorteile, etwa eine Vielzahl vortrainierter Modelle mit hochmodernen Ergebnissen, die wiederverwendet werden können, sowie eine einfache Implementierung.

6. Einschränkungen und zukünftige Arbeit

Eine der Einschränkungen des aktuellen Modells ist die Größe des Datensatzes. Mehr Daten im Datensatz würden die Ergebnisse verbessern, da das Transformer-Modell mit einer großen Datenmenge besser funktioniert. Außerdem sollten zukünftige Arbeiten verschiedene vortrainierte Modelle vergleichen. Da wir das vortrainierte Modell BERT verwendet haben, wäre es sinnvoll, die Leistung anderer vortrainierter Modelle wie XLNet[4] zu vergleichen. Bei den Hyperparametern gehört zur zukünftigen Arbeit auch, die maximale Sequenzlänge abzustimmen, um einen längeren Kontext zu erfassen, da beobachtet wurde, dass einige Beschreibungen sehr lang sein können.

7. Verwandte Arbeiten

Eine der ersten Forschungsarbeiten im Bereich der Analyse von Anwendungsbeschreibungen stammt von (Pandita, 2013). Die Autoren haben ein Framework namens Whyper vorgeschlagen, das prüft, ob die Notwendigkeit gefährlicher Berechtigungen, die von der App verwendet werden, in der Beschreibung erläutert wird. Es ist eine der ersten Studien, die Natural Language Processing (NLP) einsetzen, um semantische Bedeutung aus Beschreibungen zu extrahieren und Text zu erkennen, der sich auf Berechtigungen bezieht. Sie haben die Einschränkungen des schlüsselwortbasierten Ansatzes wie verwechselbare Bedeutung und semantische Schlussfolgerung adressiert und das semantische Modell durch manuelle Analyse von Android-API-Dokumenten aufgebaut. Ihr Datensatz umfasst 581 Apps, und Whyper wird anhand von drei Berechtigungen evaluiert: READ CALENDAR, READ CONTACTS und RECORD AUDIO. Der Begriff Description-to-Permission-Fidelity wurde von Qu et al. (Qu, 2014) eingeführt. Die Autoren argumentieren, dass Nutzer durch das Lesen der Beschreibungen eine intuitive Vorstellung von der Sicherheits- und Datenschutzfunktionalität der Anwendung gewinnen sollten und dass Beschreibungen eine Vorstellung von den angeforderten Berechtigungen vermitteln sollten. Sie schlagen ein vollautomatisches Framework namens AutoCog vor, in dem sie NLP-Techniken verwenden, um die Verwendung von Berechtigungen aus App-Beschreibungen abzuleiten. Ihre Schlüsselkomponente für die semantische Extraktion ist jedoch die Explicit Semantic Analysis (ESA), die eine umfangreiche Wissensbasis (z. B. Wikipedia) nutzt, im Gegensatz zu einem wörterbuchbasierten Korpus wie WordNet, das in Whyper

verwendet wird. In einer Studie von Gorla et al. (Gorla, 2014) schlagen die Autoren das Framework CHABADA vor, das die Beschreibung und die aufgerufenen APIs verwendet, um Anomalien zu erkennen und das Verhalten von Anwendungen zu prüfen. Chabada verwendet auf den Beschreibungen die Latent Dirichlet Allocation (LDA), ein Werkzeug und eine Technik zur Themenmodellierung, um das Hauptthema jeder Anwendung zu ermitteln und die Anwendungen weiter nach verwandten Themen zu gruppieren. In jedem Cluster extrahieren sie sensible APIs und verwenden einen unüberwachten Clustering-Algorithmus, um Ausreißer in Bezug auf die API-Nutzung zu finden. Ihr Datensatz mit 22500+ Anwendungen aus dem Google Play Store ist zu Reproduktionszwecken verfügbar. Um zu beantworten, warum App-Beschreibungen die Verwendung datenschutzrelevanter Ressourcen nicht erwähnen, führen Watanabe et al. (Watanabe, 2015) das Framework ACODE ein, das einen schlüsselwortbasierten Ansatz nutzt. ACODE verwendet einen zweistufigen Filter, der statische Code-Analyse und schlüsselwortbasierte Textanalyse kombiniert. Sie evaluieren ACODE an einem Datensatz mit 200 000 Apps aus dem offiziellen Markt sowie aus Märkten von Drittanbietern. Die Autoren berichten vergleichbare Ergebnisse. Anders als andere Studien beziehen sie jedoch sowohl die englische als auch die chinesische Sprache ein.

Referenzen

Almuhimedi, H. a. (2015). Your location has been shared 5,398 times! A field study on mobile app privacy nudging. *Proceedings of the 33rd annual ACM conference on human factors in computing systems*, (pp. 787–796).

Bahdanau, D. K. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* .

Cho, K. V. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint* .

Feichtner, J. a. (2020). Understanding privacy awareness in android app descriptions using deep learning. *Proceedings of the tenth ACM conference on data and application security and privacy*, 203–214.

Felt, A. P. (2011). Android permissions demystified. *Proceedings of the 18th ACM conference on Computer and communications security*, (pp. 627–638).

Felt, A. P. (2012). Android permissions: User attention, comprehension, and behavior. *Proceedings of the eighth symposium on usable privacy and security*, (pp. 1–14).

Feng, Y. a. (2019). AC-Net: Assessing the consistency of description and permission in Android apps. *IEEE Access (7)*, 57829–57842.

Gorla, A. a. (2014). Checking app behavior against app descriptions. *Proceedings of the 36th international conference on software engineering*, 1025–1035.

Graves, A. (2012). Long short-term memory. *Supervised sequence labelling with recurrent neural networks*, 37–45.

Li, R. a. (2021). Android Custom Permissions Demystified: From Privilege Escalation to Design Shortcomings. *2021 IEEE Symposium on Security and Privacy (SP)* (pp. 70–86). IEEE.

Liu, B. a. (2016). Follow my recommendations: A personalized privacy assistant for mobile app permissions. *Twelfth symposium on usable privacy and security (SOUPS 2016)*, (pp. 27–41).

Liu, Y. M. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* .

Luong, M.-T. H. (2015). Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025* .

Pandita, R. X. (2013). {WHYPER}: Towards automating risk assessment of mobile applications. *USENIX Security Symposium (USENIX Security 13)*, 527–542.

Qu, Z. a. (2014). Autocog: Measuring the description-to-permission fidelity in android applications. *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, 1354–1365.

Shen, B. a. (2021). Can Systems Explain Permissions Better? Understanding Users' Misperceptions under Smartphone Runtime Permission Model. *30th USENIX Security Symposium (USENIX Security 21)*, (pp. 751–768).

Vaswani, A. N. (2017). Attention is all you need. *Advances in neural information processing systems 30* .

Watanabe, T. a. (2015). Understanding the inconsistencies between text descriptions and the use of privacy-sensitive resources of mobile apps. *Eleventh Symposium On Usable Privacy and Security (SOUPS 2015)*, 241–255.

Wu, Z. a.-J. (2020). FCDP: Fidelity calculation for description-to-permissions in Android apps. *IEEE Access (9)*, 1062-1075.

---

1. https://www.statista.com/statistics/276623/number-of-apps-available-in-leading-app-stores/ ↩
2. https://huggingface.co/docs/transformers/model_doc/bert ↩
3. https://github.com/ThilinaRajapakse/simpletransformers ↩
4. https://huggingface.co/docs/transformers/model_doc/xlnet ↩