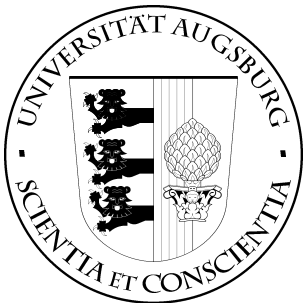
UNIVERSITÄT AUGSBURG
INSTITUT FÜR INFORMATIK



Bachelorarbeit im Studiengang Informatik und Multimedia

Analyzing article histories and authors interactions on Wikipedia using Apache Spark

vorgelegt von
Arutiun Mokoian

Gutachter und Betreuer: Prof. Dr. Peter Fischer
Zweitgutachter: Priv.-Doz. Dr. habil. Markus Endres
Datum: 16. Oktober 2018

Inhaltsverzeichnis

1 Einleitung

Wikipedia ist eine umfangreiche online Enzyklopädie, die in 2001 gegründet war. Auf Wikipedia werden verschiedene Artikel auf verschiedene Sprachen über allerlei Themen erstellt und bearbeitet. Daher ist besonders wichtig die Qualität von Wikipedia-Artikeln. Obwohl sich die faktische Richtigkeit der Artikel mit den Jahren deutlich verbessert hat, findet man immer noch fehleranfällige Artikel, in denen mit der Absicht falsche Fakten oder Unwahrheiten verbreitet wurden. Im Besonderen sind umfasst dies die Themenfelder Politik, Wirtschaft und Religion. Dies hat zur Folge, dass Autoren regelmäßig Artikel Anderer überarbeiten beziehungsweise korrigieren müssen. Ein Blick in die Versionsgeschichte eines Wikipedia-Artikels, in der alle Eingriffe protokolliert werden, zeigt, die Änderungshistorie des jeweiligen Artikels.

Die öffentliche Bearbeitbarkeit von Wikipedia erleichtert es Vandalen ebenso wie Nutzern mit klaren Interessenskonflikten oder politischer Agenda, absichtlich Artikelinhalte zu ändern und damit die Glaubwürdigkeit der Enzyklopädie zu gefährden. Böswillige Änderungen umfassen beispielsweise das Fälschen von Fakten, das Verbreiten von Falschinformationen oder das Entfernen missliebiger Passagen in Artikeln. Betroffen waren unter anderem Einträge über Abgeordnete des Bundestages, die vor Bundestagswahlen in vielen Fällen regelmäßig geschönt werden. Wie groß die inhaltlichen Mängel oder die Tendenziösität von Wikipedia-Artikeln unter Umständen sein können, zeigen Fälle wie der Konflikt um die Angliederung der Halbinsel Krim an Russland. In der ukrainischen Wikipedia steht, die Krim sei Teil der Ukraine, befinde sich im Süden des Landes und sei seit Ende Februar 2014 temporär von Russland okkupiert. In der russischen Wikipedia hingegen wird die Krim als „Halbinsel im Schwarzen Meer“ bezeichnet, die als ein Objekt territorialer Streitigkeiten sowohl von Russland als auch der Ukraine beansprucht wird, jedoch faktisch nur von Russland kontrolliert wird.

Bei vielen kontroversen Themen kommt es regelmäßig zu sogenannten *„edit wars“*. Ein Edit War ist einen digitalen Krieg zwischen zwei Gruppen von Autoren, die Inhalte eines Artikels ohne offensichtlichen Grund immer wieder überschreiben. [1] Je mehr sich zwei Benutzer gegenseitig ändern, desto größer ist der Konflikt

zwischen ihnen. Falls sich mehrere Autoren an diesem Konflikt bewusst oder unbewusst beteiligen, deutet dies auf gemeinsame Positionen hin.

Um die Glaubwürdigkeit von Wikipedia dennoch aufrecht zu erhalten, durchsuchen und analysieren tausende von Wikipedia-Bots Artikel und führen so alltägliche und sich wiederholende Aufgaben wie Löschen von Vandalismus, Erzwingen von Sperren sowie inzwischen auch das Korrigieren von Rechtschreibfehlern. Jedes Bot ist für eine bestimmte Aufgabe zuständig. Manchmal erledigte Aufgaben von einem Bot, führen wieder zu einem Konflikt mit einem anderen Bot woraus auch ein sogenannter *„Bot War“* entstehen kann.

1.1. Motivation

Das Ziel dieser Bachelorarbeit ist die Einführung und Entwicklung eines Programms, das die Versionsgeschichte der Wikipedia-Artikel sowie Interaktionen zwischen Autoren durchführen kann. Mit Hilfe von Apache Spark wird die Versionsgeschichte für Wikipedia-Artikel von einem bestimmten Zeitpunkt in die Datenbank hochgeladen, um die Daten zu verstehen und zu analysieren. Zu diesem Zweck muss untersucht werden, welche Informationen in Wikipedia-Artikel existieren und wie diese strukturiert sind sowie wie diese Informationen aus einem Artikel extrahiert werden können. Sobald die relevanten Informationen extrahiert worden sind, soll der analysierte Datensatz in der Apache Spark Datenbank angelegt werden, so dass die extrahierten Informationen in einer klaren und strukturierten Form gespeichert sind und für eine weitergehende Analyse genutzt werden können.

Im Zuge dieser Arbeit werden mehrere Methoden zur Untersuchung des Revisionsgeschichte zu Wikipedia-Artikeln implementiert. Es wird untersucht, welche Artikel am meisten bearbeitet wurden, ebenso wie die Lebensdauer und das Alter eines Artikels. Daneben sollen Hotspot-Aktivitäten für Artikel bestimmt wurden. Weiter versucht die Arbeit festzustellen, welche gemeinsamen Autoren Wikipedia-Artikel haben und wie groß das Verhältnis des Bearbeitungsanteils von Wikipedia-Bots und menschlichen Benutzern ist. Die so ermittelten Statistiken werden in entsprechenden Histogrammen visuell dargestellt.

Es wird die gesamte Historie der Wikipedia seit der Gründung der Wikipedia bis 20. Mai 2018 analysiert. Insbesondere wird sowohl die Anzahl von Bearbeitungen Autoren als auch die Anzahl von

Revisionen betrachtet. Es werden Möglichkeiten zur Identifikation von interessanten Abschnitten untersucht und diese anschließend bewertet. Außerdem werden mit Hilfe Min-Hashing Verfahren die Ähnlichkeiten zwischen Artikeln über Autoren bestimmt.

2 Wikipedia Datenmodell

Da Wikipedia eine freie zugängliche Online-Enzyklopädien ist, stellt sie die Daten zur Verfügung als Data Dumps. Für die maschinelle Analyse von Wikipedia Artikeln wird zuerst der Wikipedia Datensatz aus der Seite *https://dumps.wikimedia.org/*, wo sich alle Datendumps zu allen Artikeln aus Wikipedia finden, heruntergeladen. Die Datenbankdumps werden einmal pro Monat für jede Sprachversion von Wikipedia erstellt. Zu jeder Revisionshistorie des Artikels wurden Titel, der Name und die ID des Autors, Bearbeitungszeitpunkt, der Bearbeitungskommentar sowie auch alle anderen, für diese Arbeit weniger relevanten Daten gespeichert, wie die ID des Artikels, Weiterleitung auf andere Artikel und Bearbeitungsnummer.

Im Rahmen dieser Arbeit wird der Artikelsatz der englischen Wikipedia heruntergeladen und analysiert. Wikipedia Data Dump wird in XML-Format dargestellt und entsteht aus allen Seiten aller Namensräume. In der Abbildung 1 wird die physische Struktur eines Datensatzes im XML-Format dargestellt. Als Beispiel wird die englische Version des Wikipedia-Artikel zu „Accessible Computing“ genommen. Die Syntax der XML-Datei sieht wie folgt aus:

```
<mediawiki version="0.10" xml:lang="en">
  <siteinfo>
    <sitename>Wikipedia</sitename>
    <dbname>enwiki</dbname>
    <base>https://en.wikipedia.org/wiki/Main_Page</base>
    <generator>MediaWiki 1.32.0-wmf.4</generator>
    <case>first-letter</case>
    <namespaces>
      <namespace key="-2" case="first-letter">Media</namespace>
    </namespaces>
  </siteinfo>
  <page>
      <title>AccessibleComputing</title>
      <ns>0</ns>
      <id>10</id>
      <redirect title="Computer accessibility" />
      <revision>
        <id>233192</id>
        <timestamp>2001-01-21T02:12:21Z</timestamp>
        <contributor>
          <username>RoseParks</username>
          <id>99</id>
        </contributor>
        <comment>*</comment>
        <model>wikitext</model>
        <format>text/x-wiki</format>
        <text id="233192" bytes="124" />
        <sha1>8kul9tlwjm9oxgvqzbwuegt9b2830vw</sha1>
      </revision>
      <revision>
        <id>862220</id>
        <parentid>233192</parentid>
        <timestamp>2002-02-25T15:43:11Z</timestamp>
        <contributor>
          <username>Conversion script</username>
          <id>0</id>
        </contributor>
        <minor/>
        <comment>Automated conversion</comment>
        <model>wikitext</model>
        <format>text/x-wiki</format>
        <text id="862220" bytes="35" />
        <sha1>i8pwco22fwt12yp12x29wc065ded2bh</sha1>
      </revision>
      ........
  </page>
```

Abbildung 1: *Ausschnitt vom XML-Dokument des Artikels „Accessible Computing“, eingeleitet durch den Auszeichner <page>. Jede Revision dieses Artikels abgetrennt durch den Auszeichner <revision>.*

Der erste Teil der XML-Datei besteht aus einem <siteinfo>-Tag, der allgemeine Informationen über den gesamten Wikipedia-Dump beinhaltet, nämlich:
<*sitename*> – der Name des Wikis
<*dbname*> – die Sprache für das Wiki
<*base*> – der absolute Pfad zur Hauptseite
<*generator*> – API-Versionsinformationen
<*case*> – der erste Buchstabe in einem Titel „case-insensitive“
<*namespaces*> – stellt eine Liste aller Namensräume bereit

Der zweite Teil der XML-Datei besteht aus <page>-Elementen, die jede Seite von jeder anderen abgrenzt. Diese Elemente unterteilen sich wiederum in mehrere Kindelemente, wobei hier die <revision>-Elemente hervorzuheben sind, welche für jede Revision der entsprechenden Seite, neben ID, Zeitstempel und Autor folgende Informationen enthalten:
<*title*> – Name des Artikels
<*revision*> – enthält Information für einzelne Bearbeitungen
<*timestamp*> – enthält Bearbeitungszeitpunkt
<*contributor*> – enthält Benutzername des Bearbeiters
<*comment*> – gibt den Kommentar des Benutzers für die Bearbeitung.

Für die kommende Bewertung und Datenanalyse wird ein Wikipedia-Datensatz vom 20. Mai 2018 gewählt. Der Wikipedia Data Dumps ist mit Gzip komprimiert und ungefähr 1.86 Gigabyte groß. Der verwendete Datensatz enthält 22,315 eindeutige Wikipedia-Artikel und über 100,000 Revisionen insgesamt.

3 Apache Spark

Apache Spark ist eine Datenbank, die für die Arbeit mit sehr großen Datenmengen ausgelegt ist. Mit Hilfe von Spark werden Datenabfragen auf große Datensätze in hoher Geschwindigkeit und guter Performance ausgeführt. Im Besonderen werden sehr komplexe mathematische Analysen mit Spark implementiert. Apache Spark ist zurzeit sehr beliebt bei großem Unternehmen, die Daten im Tera- und Petabereich analysieren müssen.

Da Spark für große Datenmengen geeignet ist, wird der verwendete Wikipedia Datensatz mit Hilfe von Spark analysiert. Die Daten werden dynamisch im Arbeitsspeicher des Server-Clusters hochgeladen und direkt dort verarbeitet.

Apache Spark besteht aus einem Spark-Core und mehreren Worker Nodes (siehe Abb. 2). Der Spark Core ist eine grundlegende Engine für die parallele und verteilte Datenverarbeitung. Der Spark Core ist für Speicherverwaltung und Wiederherstellung nach aufgetretenen Fehlern verantwortlich. Außerdem wird die Cluster-Planung und Interaktion mit Speichersystemen mit Hilfe von Spark Core realisiert. [3] Die Datenstruktur im Spark Core basiert auf sogenannten RDDs. RDD ist eine unveränderbare fehlertolerante verteilte Sammlung von Objekten, die parallel verarbeitet werden können. RDD kann Objekte beleibeige Typen enthalten und mit einem externen Datensatz erstellt werden. Mit Hilfe von RDDs kann man die großen Dateien in hunderte von Gigabyte oder sogar Terabyte parallelisieren.

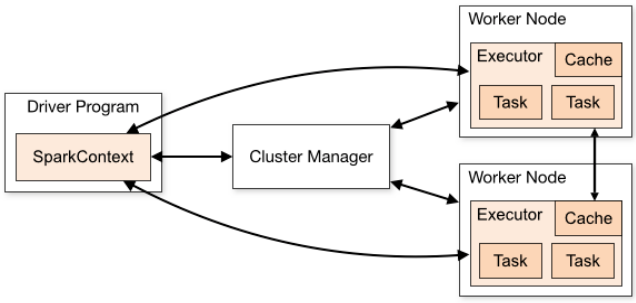


Abbildung 2: *Spark Architektur [2]*

Apache Spark bietet verschieden Funktionen zur Datenverarbeitung mit umfangreichen Bibliotheken. Apache Spark stellt zur Verfügung zahlreiche Bibliotheken wie beispielweise Maschine Learning (MLlib), Spark SQL, GraphX, Spark Streaming usw. Im Folgenden werden die relevante APIs kurz vorgestellt.

3.1. Spark SQL

Spark SQL ermöglicht strukturierte Daten effizient zu verarbeiten und Daten aus verschieden Datenquellen direkt zu laden, wie zum Beispiel JSON, CSV-Datei und XML. Spark SQL gehört zu den am häufigsten eingesetzten Komponenten. Diese Bibliothek unterstützt nicht nur verschieden Datenquellen, sondern ermöglicht auch die Kombination von SQL-Abfragen mit Transformation Funktionen. Um die Datenverarbeitung zu ermöglichen, abstrahiert Spark SQL in drei Arten von Datenabstraktionen – RDDs, DataFrames und Datasets. RDD ist die grundlegende Datenstruktur von Spark. Sie ermöglicht speicherinterne Berechnungen in großen Clustern fehlertolerant durchzuführen, so dass die Ausführungsgeschwindigkeit steigen kann. Die DataFrame-Abstraktion im Gegensatz zu einer RDD organisiert wie eine Tabelle in einer relationalen Datenbank. DataFrame in Spark kann einer verteilten Sammlung von Daten eine Struktur zuweisen, die eine Abstraktion auf höherer Ebene ermöglicht. [4]

3.2. MLlib

Apache Spark unter anderem bietet eine Bibliothek, die allgemeine Machine Learning Funktionalität enthält, was MLlib genannt wird. MLlib bietet mehrere Arten von maschinellen Lernalgorithmen an, wie zum Beispiel Klassifizierung, Regression, Clustering sowie Locality Sensitive Hashing (LSH) Algorithmen. LSH ist ein randomisierter Algorithmus und eine Hashing-Technik, die häufig bei großen maschinellen Lernaufgaben verwendet werden. Zum anderen bietet MLlib den MinHash-Algorithmus an, der zu der LSH-Familie gehört. Der MinHash-Algorithmus wird angewendet, um eine große Anzahl von Dokumenten effizient miteinander zu vergleichen. [4]

3.3. Spark Python API (PySpark)

PySpark ist eine Python API für Spark. Sie wird auf Sparks Java API aufgebaut. PySpark Shell ist dafür verantwortlich, die Python-API mit dem Spark-Core zu verknüpfen und den *SparkContext* zu initialisieren. Daten werden in Python verarbeitet und in der JVM zwischengespeichert.

3.4. Visualisierung

Für die Visualisierung der Daten in Spark wird eine Plot-Bibliothek benutzt, hier wird auf *PySpark-Dist-Explore* zurückgegriffen. Diese Bibliothek ermöglicht es einen schnellen Einblick in Daten aus Spark DataFrames durch Histogramme und Plots zu erhalten. *PySpark-Dist-Explore* ermöglicht Matplotlib-Graphen und Histogramme einfach zu erstellen. Das Erstellen eines Histogramms wird mit folgender Funktion gemacht:

- **hist (ax, dataframe, **kwargs)** – wobei
  - **ax** ein Matplotlib Axes Objekt ist
  - **dataframe** ist ein PySpark DataFrame mit einer einzelnen Spalte, einer Liste einspaltiger DataFrames oder einem DataFrame mit mehreren Spalten
  - **kwargs** - alle Keyword Argumente, die in der Matplotlib-Hist-Funktion verwendet wurden

4 Evaluation

Im folgenden Kapitel werden die Bearbeitungen von Wikipedia-Artikeln sowie die Interaktionen zwischen Autoren bewertet und diskutiert.

4.1 Eingabedaten

Um auf den Datenbestand der Wikipedia zugreifen zu können, wird zuerst die Frage gestellt, wie dieser von der Enzyklopädie organisiert wird. Zuerst wird der aktuelle Datenbank-Dump vom 20. Mai 2018 heruntergeladen. Es wird die englische Wikipedia analysiert. Es wurde die englische Version gewählt, da sie deutlich mehr Anzahl von Artikeln und Nutzer im Vergleich zu deutscher Wikipedia hat. [5] Um die heruntergeladene Wikipedia-Dump-Datei in Apache Spark zu verarbeiten, bestehen zwei Möglichkeiten: entweder kann direkt eine XML-Datei in Spark als DataFrame hochgeladen werden oder die XML-Datei wird vor dem Hochladen mit Hilfe eines Python-Skripts in das JSON-Format umgewandelt. Obwohl Spark eine Spark-XML-Bibliothek zum Parsen und Analysieren von XML-Daten anbietet, schlug der Versuch eine verschachtelte XML-Datei in Spark hochzuladen fehl. Das Problem lag daran, dass das Lesen aus verschachtelten XML-Tags nicht korrekt war. Es wurde entschieden, eine Umwandlung von XML in JSON mit Hilfe eines Python-Skripts durchzuführen, weil Spark SQL eine integrierte Unterstützung für JSON bietet und man deswegen schnell zu Ergebnissen kommt. Das Python-Skript *„xmlparse.py“* enthält die folgende Methode:

- **parse_xml(filename):** Die Methode wird alle Informationen zwischen öffnendes XML-Tag gespeichert. Handelt es sich dabei um ein page-Tag, wurde ein kompletter Artikel-Datensatz eingelesen (title, id, revision) und in eine JSON-Datei gespeichert. Als Übergabeparameter wird der heruntergeladene Wikipedia-Dump benutzt.

Nachdem die XML-Datei mit Hilfe der *parse_xml()* Methode in das JSON-Format konvertiert wurde, sieht die JSON-Datei wie folgt aus:

Abbildung 3: *JSON-Struktur des Wikipedia-Artikels*

Während der Umwandlung von XML in JSON wurden nicht relevante für unsere Analyse Daten wie beispielwiese Kommentar, Text, Format usw. weggelassen. Die umgewandelte JSON-Datei beinhaltet mehreren JSON-Objekten. Diese JSON-Objekte entsprechen jeweils einem Wikipedia-Artikel, der die Attribute des Wikipedia-Artikels (title, id, revision) beinhaltet. Die „Revision"-Attribute bestehen aus einem Array von JSON-Objekten, die entsprechende Felder für einzelne Bearbeitungen enthalten. Da die JSON-Datei ein verschachteltes Format hat, wird sie zuerst in ein flaches Format überführt, in dem die Artikel-Historie ausgerollt wird. Zwar kann Spark die verschachtelteten Daten sehen, aber für weitere Bearbeitungen braucht man die Entschachtelung. Von Anfang an wurde herausgestellt, dass die relevanten Operationen am besten an dem entschachtelten und normalisierten Daten funktionieren.

Nachdem wir den Datensatz vorbereitet und in ein geeignetes Format überführt haben, werden die Daten in Apache Spark hochgeladen. Als nächstes werden die Daten mit Hilfe von Spark SQL Bibliothek entschachtelt. In der Abbildung 4 wird der entschachtelte Wikipedia Datensatz in Apache Spark dargestellt.

id	title	author	authorID	editTime
15943	John Newton	Conversion script	0	2002-02-25 16:51:15
15943	John Newton	Amillar	206	2002-03-19 00:53:27
15943	John Newton	null	null	2003-01-19 18:45:03
15943	John Newton	null	null	2003-01-19 18:45:54
15943	John Newton	null	null	2003-01-19 18:46:13
15943	John Newton	null	null	2003-01-19 18:51:34
15943	John Newton	null	null	2003-01-19 18:58:37
15943	John Newton	Ams80	7543	2003-02-09 12:51:47
15943	John Newton	null	null	2003-02-09 12:54:47
15943	John Newton	Hephaestos	3628	2003-06-10 13:39:59
15943	John Newton	Charles Matthews	12978	2003-10-03 12:01:04
15943	John Newton	Francs2000	22841	2004-02-08 13:41:58
15943	John Newton	Francs2000	22841	2004-02-08 13:42:34
15943	John Newton	Nikai	9759	2004-03-09 04:03:16
15943	John Newton	Smjg	38198	2004-03-24 19:07:16
15943	John Newton	Deb	1219	2004-04-09 17:18:37
15943	John Newton	Deb	1219	2004-04-11 21:21:50
15943	John Newton	Astronautics~enwiki	26195	2004-06-12 22:26:07
15943	John Newton	Avnative	84356	2004-08-08 07:47:36
15943	John Newton	Guanabot	82928	2004-09-19 16:42:36

Abbildung 4: *Darstellung Wikipedia-Datensatz in Apache Spark.*

In der Tabelle befindet sich die Liste aller Artikel der beteiligten Autoren sowie der Name des Editors, die Autor-ID und der Bearbeitungszeitpunkt. Diese Informationen werden benötigt um weitere Statistiken zwischen Artikeln und um die Autoren eines Artikels herauszufinden.

4.2 Grundlegende Eigenschaften

Es wird am Anfang die grundlegenden Daten für Wikipedia-Artikel herausgefunden. Es wird zu Beginn folgende Statistiken berechnet:

- Das Erstellungsdatum eines Artikels
- Das Alter von Wikipedia-Artikel
- Die Anzahl von Bearbeitungen pro Artikel
- Die Aktualität des Artikels
- Zeitraum seit letzter Änderung

Dieser Abschnitt setzt sich mit der Frage auseinander, wie sich die Bearbeitungsaktivitäten von Wikipedia-Artikeln seit dem Start des Projekts im Jahr 2001 entwickelt haben. Man kann behaupten, dass die meisten Wikipedia-Artikel kurz nach der Gründung von Wikipedia erstellt wurden. Da die Mehrheit von Artikeln in den ersten Jahren eingelegt wurde, reduziert sich stark die Anzahl von Artikeln. In der folgenden Untersuchung wird anhand eines Histogramms der Verlauf des Erstellungsdatum eines Artikels dargestellt (siehe Abb. 5).

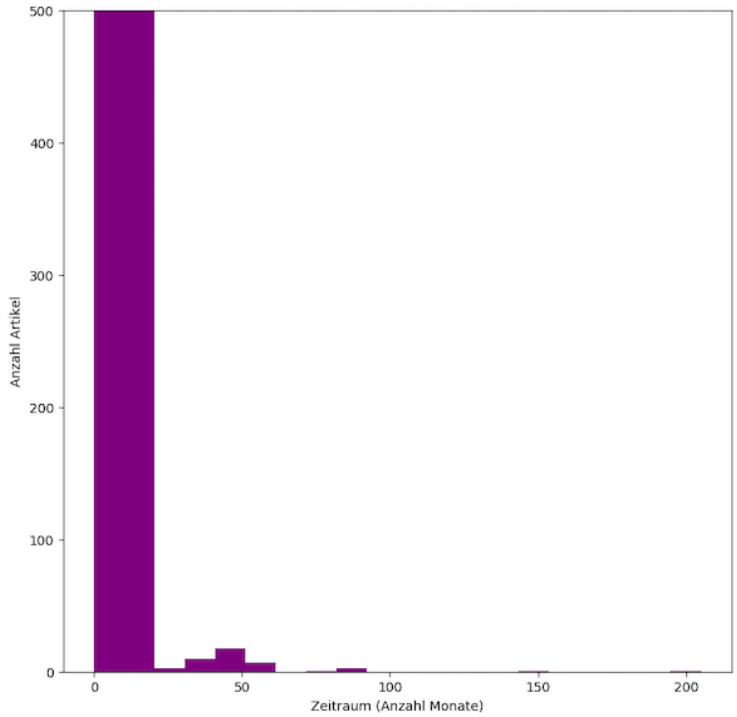


Abbildung 5: *Zeitraum seit der Erstellung Wikipedia-Artikel nach der Gründung Wikipedia*

Um das Erstellungsdatum eines Artikels zu bestimmen, wird das erste Element aus dem Array von Revisionen extrahiert und die Differenz zwischen Datum der Gründung Wikipedia berechnet.

`def creation_date_of_article(df)` – gibt ein DataFrame mit dem Erstellungsdatum für alle Artikel zurück

Anhand der Ergebnisse (Abb. 5) lässt sich sagen, dass die meisten Wikipedia-Artikel innerhalb von 2 Jahren nach der Gründung der Wikipedia erstellt wurden. Obwohl Artikelwachstum nicht mehr so stark ansteigt, ist kein Ende abzusehen. Jedes Mal wird es immer schwieriger, noch etwas zu finden, über das noch kein Wikipedia-Artikel angelegt wurde.

Als nächstes wird die Aktualität eines Artikels bestimmt. Da die meisten Wikipedia-Artikel ganz am Anfang nach der Gründung angelegt wurden, kann man vermuten, dass ein großer Teil der Wikipedia-Artikel inzwischen nicht mehr „aktiv“ ist, also bereits lange nicht mehr bearbeitet wurde. Um die Lebensdauer für Wikipedia-Artikel zu bestimmen, muss sowohl das Datum der ersten Bearbeitung als auch das Datum der letzten Bearbeitung eines Artikels festgestellt

werden. Sowohl das Erstellungsdatum als auch das Datum der letzten Änderungen aller Artikel werden direkt aus dem Wikipedia-Daten-Dump übernommen. Da alle Bearbeitungen für Wikipedia-Artikel in der chronologischen Reihenfolge gespeichert sind, wurde das erste Element beziehungsweise das letzte Element aus dem Array von Revisionen extrahiert. Mit Hilfe der folgenden beiden Funktionen werden die oben genannten Daten bestimmt:

- `def creation_date_of_article(df)` – gibt ein DataFrame mit dem Erstellungsdatum für alle Artikel zurück
- `def last_edit_date(df)` – gibt ein DataFrame mit der letzten Bearbeitung für alle Artikel zurück

Nachdem beide Daten bestimmt wurden, wird die Lebensdauer eines Artikels als Subtraktion zwischen den beiden Zeitpunkten berechnet. Die Differenz zwischen den zwei Datumswerten wird als Anzahl ganzer Monate ausgegeben.

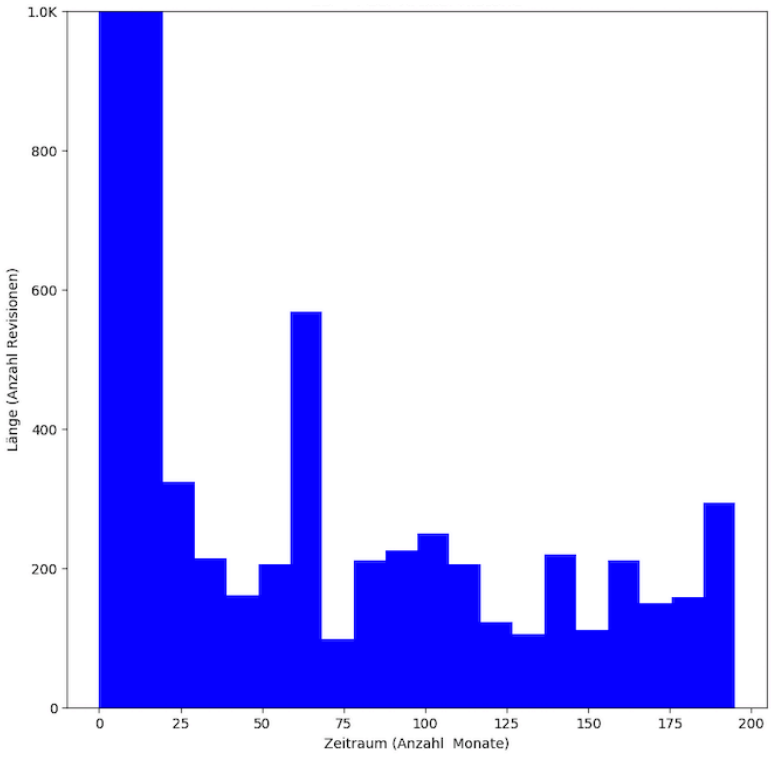


Abbildung 6: *Zeitraum seit letzter Änderung*

Aus der Abbildung 6 wird ersichtlich, dass die meisten Bearbeitungen auf Wikipedia in die ersten 2-3 Jahren nach der Gründung Wikipedia waren. Danach ist die Anzahl der Bearbeitungen deutlich gesunken. Daraus lässt sich schließen, dass mittlerweile nicht mehr einfach einen Artikel zu verbessern. So sind Rechtschreib- oder Grammatikfehler bei den meisten Artikeln inzwischen selten geworden und relevante Fakten in der Regel bereits eingearbeitet. Dies ist insbesondere bei „abgeschlossenen“ Themen der Fall, wie etwa zu mathematischen Sätzen oder Pflanzenarten.

Abbildung 7 zeigt die Anzahl von Revisionen (y-Achse) in Abhängigkeit der aktiven Zeit (x-Achse) auf. Das Histogramm veranschaulicht, dass die absolute Mehrheit an Artikeln eine Lebenszeit von mehr als 180 Monaten hat. Dies sagt aus, dass die Mehrheit aller Artikel über einen Zeitraum von mehreren Jahren bearbeitet wird. Der anfängliche Anstieg kann damit erklärt werden, dass Artikel mit noch wenig Bearbeitungen vermutlich auch nur wenig Inhalt besitzen und dadurch nur weniger Fläche für Beanstandungen bieten.

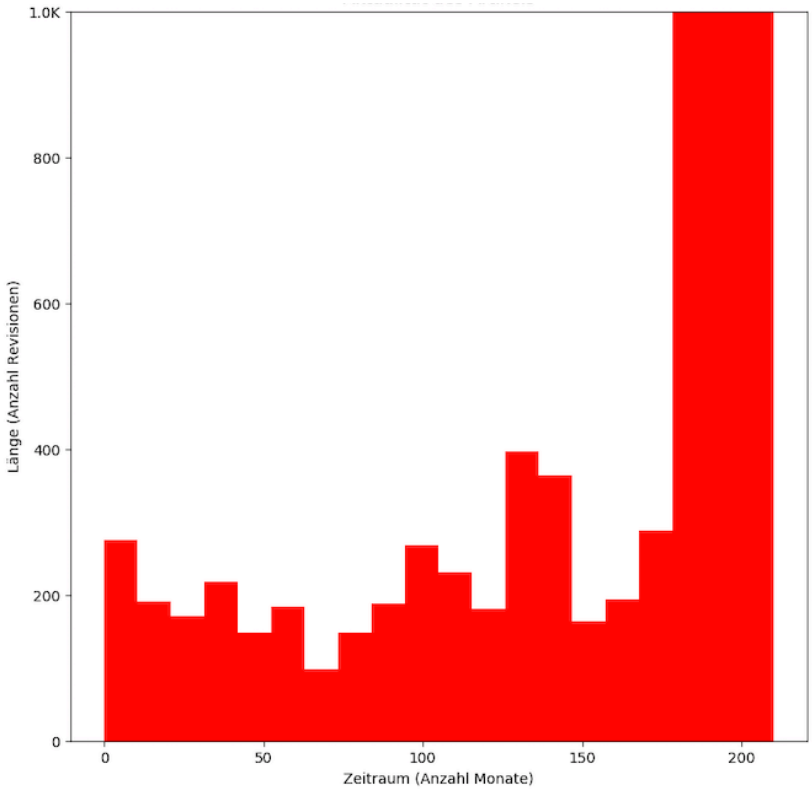


Abbildung 7: *Länge (Zeit) der Artikel-Historie*

Um die oben genannten Statistiken zu bestimmen, nämlich das Alter eines Artikels, die Artikel-Aktivität und die Länge der Artikel-Historie, braucht man insgesamt ca. 2 Minuten.
Ein gutes Indiz für die Popularität eines Artikels bietet Anzahl Revisionen oder Anzahl Autoren. Umso mehr Autoren an einem Artikel mitgearbeitet haben, desto populärer ist dieser. Um interessanten Artikeln mit der höchsten Bearbeitungszahl zu identifizieren, wurden als erstes alle Artikel anhand des Artikelnamen gruppiert und mit Hilfe von der Aggregationsfunktion von Spark die Länge der Revisionshistorie bestimmt. Die Abbildung 8 zeigt die gesamte Anzahl von Revisionen (x-Achse) in Abhängigkeit der Anzahl der Artikel (y-Achse).

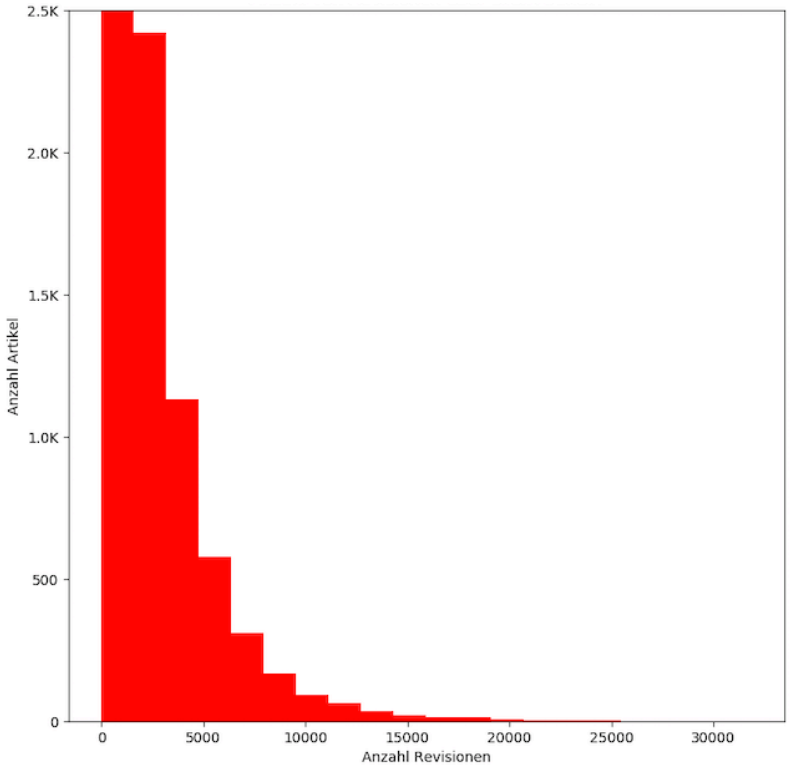


Abbildung 8: *Anzahl von Revisionen über allen Artikeln*

Aus den oben beschriebenen Statistiken kann man schließen, dass die Zahl der Bearbeitungsaktivitäten, die ein Artikel aufweist, auch eine Aussage über dessen Relevanz ist. Gleiches gilt für die Anzahl verschiedener Autoren: Aus einer höheren Autorenvielfalt lässt sich eine größere Fehlerrate oder ein höheres allgemeines Interesse ableiten. Manche Artikel haben eine große Anzahl von Autoren und relativ niedrige Anzahl von Revisionen. Daraus kann man schließen, dass solche Artikel sich in der Regel mit relativ unbekannten Themen beschäftigen.

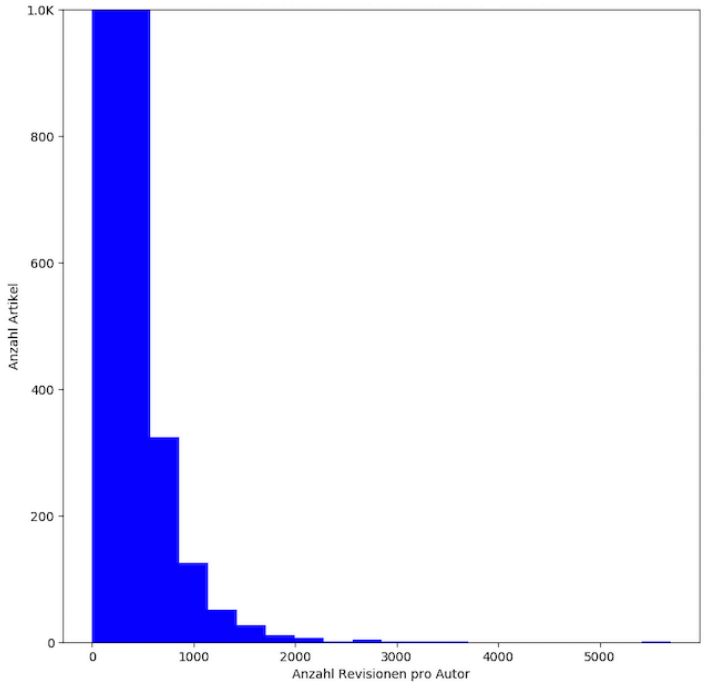


Abbildung 9: *Anzahl von Revisionen über Autoren pro Artikel*

Die Laufzeit für die Bestimmung der gesamten Anzahl von Revisionen über allen Artikeln und die Bestimmung der Anzahl von Revisionen von Autoren pro Artikel beträgt auf dem Cluster Node weniger als 2 Minuten.

4.3 Identifizieren interessanter Abschnitte für Wikipedia-Artikeln

In diesem Abschnitt wird auf interessante Abschnitte von Wikipedia-Artikeln eingegangen. Wie schon erwähnt wurde, spielt der Zeitraum, in dem bestimmte Artikel am häufigsten bearbeitet wurden, eine große Rolle. Wenn man etwa den Konflikt zwischen der Ukraine und Russland als Beispiel nimmt, sollte dies bedeuten, dass die Anzahl von Bearbeitungen von für dieses Thema relevanten Wikipedia-Artikeln ab einem bestimmten Zeitpunkt (Anfang 2014) stark angestiegen ist.

Am Anfang wird mit Hilfe der Ergebnisse aus Abschnitt 4.1 ein DataFrame erstellt mit der monatlichen Anzahl von Bearbeitungen für jeden Artikel. Als nächstes wird das Datum der ersten Bearbeitung (*min_date*) und Datum der letzten Bearbeitung (*max_date*) bestimmt

und das DataFrame mit allen Daten zwischen „*min_date*“ und „*max_date*“ generiert. Als Zeitintervall wird ein Monat genommen, da in dieser Arbeit die Anzahl der Bearbeitungen pro Monat untersucht wird. Anschließend wird das kartesische Produkt zwischen allen Zeitstempel und alle Artikelnamen erstellt, um Artikel mit fehlenden Zeitstempeln zu erfüllen.

Außer Aggregationsfunktionen und benutzerdefinierte Funktionen bietet Apache Spark sogenannte Window-Funktionen, die eine große Anzahl von Operationen ermöglichen. Mit Hilfe von Window-Funktion kann man mit einer Gruppe von Zeilen arbeiten und immer noch ein einzelner Wert für jede Eingabezeile zurückliefern.
Am Ende wird das Sliding-Window-Verfahren von Spark angewendet, um die durchschnittliche Anzahl von Bearbeitungen für einen Artikel zu berechnen. Mit Hilfe der Window-Funktion kann man mit einer Gruppe von Zeilen arbeiten. Es wird ein Rückgabewert für jede Eingabezeile einer Tabelle berechnet, basierend auf einer Gruppe von Zeilen, welche Frame genannt wird. [6] Zuerst wird die Window-Spezifikation über alle Artikel durch einen Partitionierungsausdruck definiert. Als Partitionierungsausdruck wird der Artikel-Titlel angegeben und dann über Zeitstempel geordnet. Zusätzlich zum Ordnen und Partitionieren muss die Anfangsgrenze des Frames sowie die Endgrenze des Frames definiert werden. Die folgende Abbildung 10 zeigt ein Row-Frame, das die durchschnittliche Anzahl von Revisionen zwischen dem vorhergehenden Monat und dem folgenden Monat berechnet.

```
window = Window.partitionBy("title").orderBy('yearmonth').rows
Between(-1, 1)
df_avg = df_allts.select('title', 'yearmonth', 'count', f.roun
d(f.avg('count').over(window), 2).alias('avg')).na.fill(0).ord
erBy(desc('count'))
```

Abbildung 10: *Berechnung der durchschnittlichen Anzahl von Revisionen pro Monat mit Hilfe von Sliding-Window*

Die folgende Abbildung 11 zeigt die durchschnittliche Anzahl von Revisionen pro Monat.

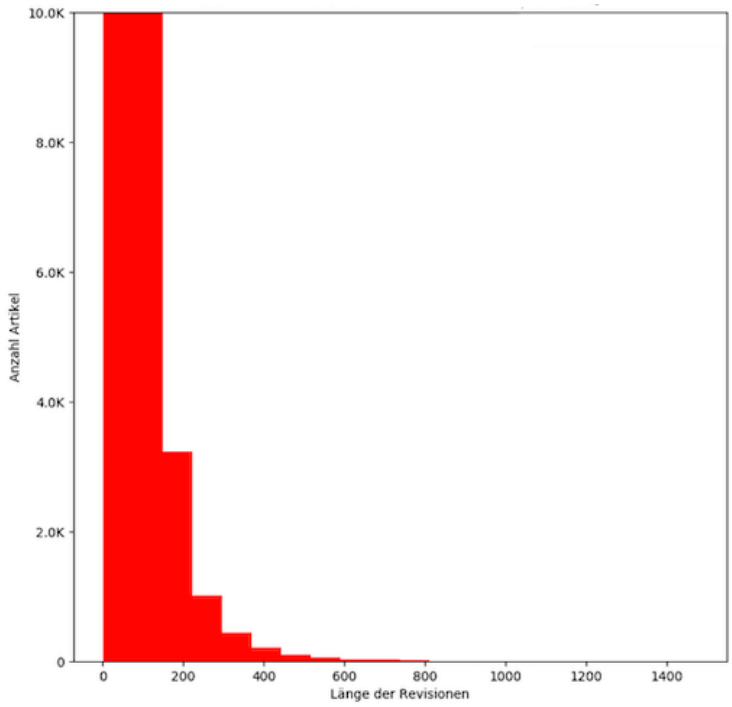


Abbildung 11: *Durchschnittliche Anzahl von Revisionen pro Monat*

Um die durchschnittliche Anzahl von Revisionen pro Monat zu bestimmen, braucht der Cluster Node zwischen 8 und 9 Minuten. Da für diese Statistik mehrere Joins vor allem ein kartesisches Produkt (als „CrossJoin“ genannt) zwischen DataFrames benutzt wird, wird die Berechnung relativ aufwendig.
Bei der Varianz handelt es sich um eine Maßzahl, die angibt wie weit die einzelnen Werte im Durchschnitt von Mittelwert entfernt liegen.

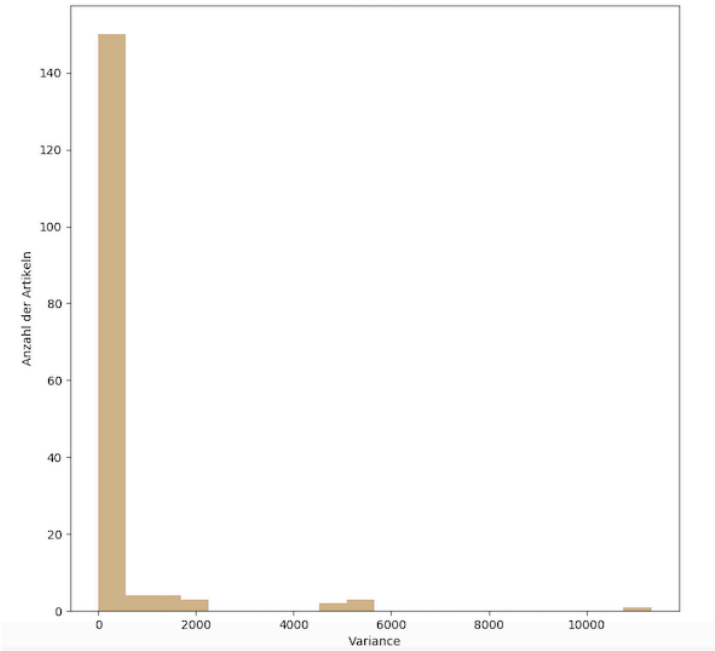


Abbildung 12: *Varianz von Artikel-Aktivtäten über alle Revisionen*

Die Laufzeit für die Identifizierung interessanter Abschnitte für Wikipedia-Artikeln beträgt auf dem Cluster Node ungefähr 2 Minuten.

4.4 Identifizieren Wikipedia-Bots

In diesem Abschnitt wird analysiert wie groß der Anteil von Botbearbeitungen in Wikipedia ist. Ebenfalls wird untersucht wie häufig Artikel von Wikipedia-Bots bearbeitet wurden. In Wikipedia ist es relativ einfach Bots zu identifizieren, speziell im Vergleich zu anderen sozialen Netzwerken wie Twitter. Dort kann sich ein Bot als normaler Nutzer tarnen und selbständig gefertigte Inhalte posten, auf Hashtags und Keywords reagieren oder sich mit anderen Nutzern anfreunden.

Da alle Wikipedia-Bots zwingend das Wort „Bot“ im Namen enthalten, kann einfach nach „Bot“ als Bestandteil des Benutzernamens gesucht werden um Bots zu identifizieren. Im Vergleich von Abbildung 13 ist zu erkennen, dass Bots nur einen verschwinden geringen Anteil aller aktiven Benutzer ausmachen, aber mit über 10% einen erheblichen Anteil an der Anzahl der insgesamt getätigten Bearbeitungen haben.

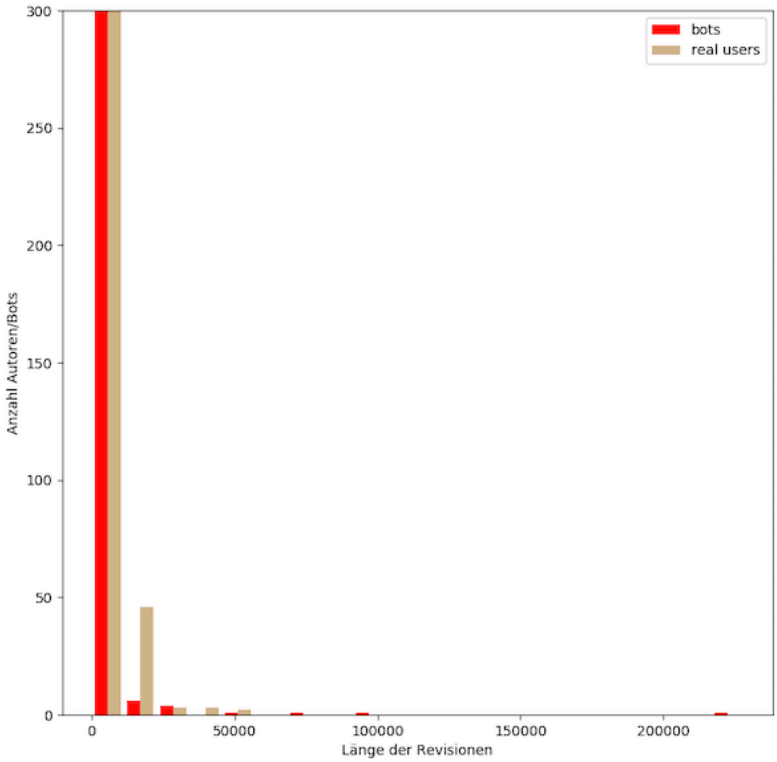


Abbildung 13: *Anzahl Benutzerbearbeitungen (brauen) im Vergleich mit der Anzahl von Botsbearbeitungen (rot)*

Die Laufzeit für oben beschriebene Statistik beträgt auf dem Cluster Node ca. 80 Sekunden.

4.5 Ähnlichkeiten zwischen Artikeln über Autoren

Autoren mit relativ großer Anzahl an Bearbeitungen können auch an den gleichen Wikipedia-Artikeln arbeiten. Um diese Artikel zu bestimmen, wird eine Methode, die nicht die genauen Ergebnisse liefert, sondern eine Approximation des Ergebnisses, angewendet. Alle solche Artikel werden mittels MinHashing Methode bestimmt. Im Folgenden wurde diese Gruppe des Artikels explorativ analysiert.

4.5.1 Jaccardsimilarität

Hier wird ein Algorithmus implementiert, welcher die Aktivität bestimmter Autoren auf bestimmten Wikipedia-Artikeln anhand der semantischen Ähnlichkeit der Titel der bearbeitenden Artikel vorhersagt. Dieser wird verwendet, um den Einfluss des semantischen Aspekts auf das Autorenverhalten zu untersuchen. Um die Ähnlichkeit zwischen Artikeln über Autoren zu bestimmen, wird die sogenannte Jaccardsimilarität Funktion angewendet. Für die Ähnlichkeit wird der Jaccard-Index festgestellt. Um Jaccard-Index zu berechnen, teilt man die Anzahl der Schnittmenge durch die Größe Vereinigungsmenge. [7]

$$J\,(A, B) = |A \cap B| \,/\, |A \cup B|$$

Je höher der Wert J, desto ähnlicher sind sich die beide Mengen.

4.5.2 Vektorisierungs-Model

Für alle Artikel wird mit dem Vektorisierungs-Model aus Autorenanzahl abgebildet. Da die Anzahl der Vergleiche geometrisch mit der Größe des Artikels wächst, ist die Durchführung paarweiser Vergleiche zeitaufwendig. An Stelle von allen Artikeln miteinander zu vergleichen, werden Wikipedia-Artikel in einem metrischen Vektor umgewandelt. Um ein Vektor zu erstellen, wird ein Index aller Autoren (außer Wikipedia-Bots) zugeordnet und abgespeichert. Wikipedia-Bots werden herausgenommen, da manche Artikel einen relativ großen Anteil von Botbearbeitungen haben können, was eine Maßnahme zur Bekämpfung von Vandalismus sein kann. Die Erstellung des fortlaufenden Index wird mit Hilfe der Spark Windows-Funktion implementiert, bei dem jedem Autor eine eindeutige Identifikationsnummer von 1 bis N zugeordnet ist, um die Beziehungen zwischen Autoren und Titel der Artikel zu ermitteln (siehe Abb. 14).

```
windowSpec = W.orderBy("author")
df_authors = df_real_users.withColumn("Id", f.row_number()
.over(windowSpec))
df_authors.show()
```

Abbildung 14: *Zuordnung des Indexwert zu allen Autoren*

Als nächstes wird ein DataFrame erstellt mit allen Artikeln und Autoren, die jeweils einen Artikel bearbeitet haben. Für jeden Artikel wird ein n-dimensionaler Vektor erstellt. Die Generierung der Vektoren wird durch Feature-Hashing abgebildet (siehe Abb. 15). Zwischenwerte

werden entsprechend des Vorkommens der Autoren in den Artikel anhand ihrer Hashwerte berechnet. Für jeder Autor, der einen Artikel bearbeitet hat, wird ein Wert mit 1 definiert und für alle andere mit 0 initialisiert.

```
mh = MinHashLSH(inputCol="features", outputCol="hashes", numHa
shTables=5)
model = mh.fit(df_res)
model.transform(df_res).show()
```

Abbildung 15: *Generierung Feature-Hashing Vektoren*

Min-Hashing Verfahren ermöglicht die Kosten der Berechnung linear anstatt exponentiell wachsen. Es werden ähnliche Artikel gefunden und zusammengefasst (siehe Abb. 16). [8]

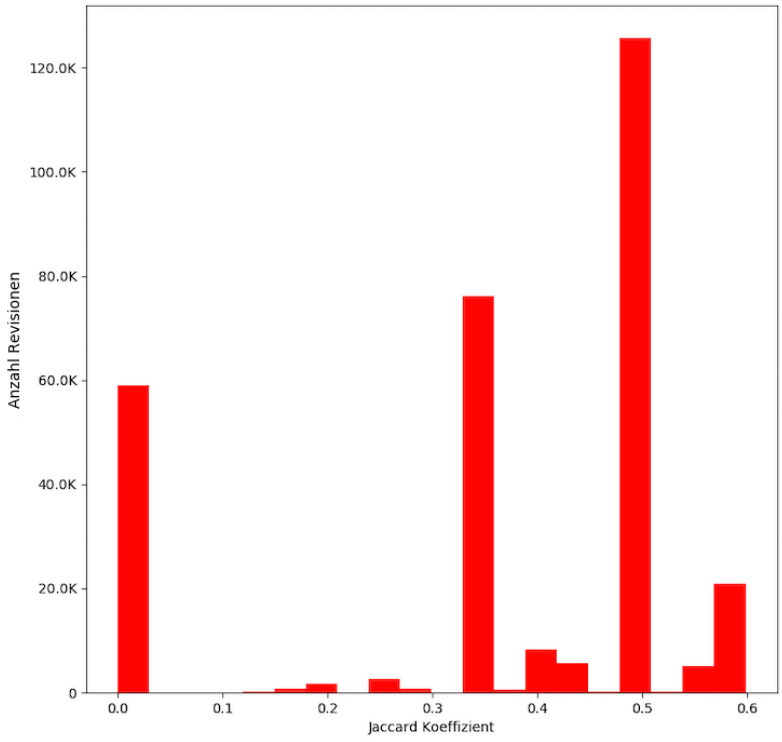


Abbildung 16: *Ähnlichkeit zwischen Artikeln über Autoren*

5 Zusammenfassung

In der Arbeit wurde gezeigt, wie sich die Artikel-Historie für Wikipedia-Artikel sowie die Benutzeraktivitäten verhaltet. Mit der Anpassung der zugrundeliegenden Bibliotheken von Apache Spark konnte die genannten Untersuchungen effektiv anwenden. Apache Spark lässt sich mit sehr großen Datenmengen umgehen und die Rechenzeit erheblich reduzieren. Die genutzten Methoden lassen sich daher auf große Wikipedia-Dumps anwenden, um die Artikel-Historie zu analysieren und Daten zu visualisieren. Es wurde einige Methoden umgesetzt, welche die Revisionsgeschichten der englischen Wikipedia analysieren. Es wurde die Aktivität eines Nutzers sowie Wikipedia-Bots auf Wikipedia-Artikel gezeigt. Außerdem kann die Ähnlichkeit von Artikel anhand der Nutzeraktivität bestimmt werden. Ein entsprechendes Min-Hashing Verfahren wurde implementiert. Mithilfe dieser Implementierungen kann auch zukünftig sowohl *„edit wars“* als auch *„bots wars“* zwischen Wikipedia Nutzer beziehungsweise zwischen Wikipedia-Bots vorhergesagt werden.

6 Literatur

1. Wikipedia: Edit-War
   Link: https://de.wikipedia.org/wiki/Wikipedia:Edit-War
   Zugriff am 12.09.2018
2. Cluster Mode Overview. Link:
   http://spark.apache.org/docs/2.3.2/cluster-overview.html
   Zugriff am 16.09.2018
3. Wills, Josh; Owen, Sean; Laserson, Uri; Ryza, Sandy:
   "Advanced Analytics with Spark", First Edition, O'Reilly Media, Inc., 2015
4. Karau, Holden; Konwinski, Andy; Wendell, Patrick; Zaharia, Matei: "Learning Spark: Lightning-Fast Big Data Analysis". First Edition, O'Reilly Media, Inc., 2015
5. Wikipedia: Number of user accounts. Link:
   https://en.wikipedia.org/w/index.php?title=List_of_Wikipedias&oldid=478588948
   Zugriff am 16.09.2018
6. Introducing Window Functions in Spark SQL
   Link: https://databricks.com/blog/2015/07/15/introducing-window-functions-in-spark-sql.html
   Zugriff am 10.10.2018
7. Wikipedia: Jaccard-Koeffizient
   Link: https://de.wikipedia.org/wiki/Jaccard-Koeffizient
   Zugriff am 8.10.2018
8. Leskovec, Jure; Rajaraman, Anand; Ullman, Jeff. Finding Similar Sets. In "Mining of Massive Datasets" (Chapter 3) Cambridge: Cambridge University Press, 2011