Projektdokumentation im Modul Semantic Web

Vergleich der Bewertungen von Romanen und ihren Verfilmungen

Florens Rohde

31.07.2016

Recherchefragestellung: Werden literarischen Vorlagen im Schnitt besser oder schlechter als ihre Filmadaptionen bewertet? Welche Romane wurden besonders häufig verfilmt?

1 Inhaltliche Interpretation der Fragestellung

Eine bekannte Literaturadaption der letzten Jahre ist die Serie „Game of Thrones“ von David Benioff und D. B. Weiss nach den Büchern von George R. R. Martin. Die Serie wird allgemein hochgelobt und ist sicherlich ein positives Beispiel. Der Blick auf die durchwachsene Qualität der zahlreichen Comic-Verfilmungen der letzten Jahre zeichnet ein anderes Bild.
In dieser Recherche soll es darum gehen, die Qualität von Filmadaptionen und ihrer Vorlagen anhand von Bewertungen in populären Online-Plattformen zu vergleichen. Die Untersuchung wird sich allerdings auf die Umsetzung von einzelnen Romanen in Filmen beschränken. Wenn – wie in den obigen Beispielen – entweder die Verfilmung (hier die Serie) aus vielen Episoden oder die Vorlage aus vielen Heften besteht, ist eine klare Zuordnung von Bewertungen schwierig und es müsste auf Durchschnittsberechnungen mit der Gewichtungs-Problematik zurückgegriffen werden.

Mit den zusammengetragenen Daten lassen sich weiterhin, unter Einbeziehung der Erscheinungsdaten, die Bewertungen im Zeitverlauf vergleichen und Aussagen zu häufig genutzten Vorlagen treffen.

2 Relevante Datenquellen

Es sind drei wesentliche Informationen zu beschaffen: Buch-Film-Zuordnungen, sowie Buch- und Filmbewertungen.

2.1 Wikipedia

Die bekannte Online-Enzyklopädie enthält u.a. Artikel zu zahlreichen Filmen und Büchern. Es existiert eine Kategorie, die Filme auflistet, welche auf einem Roman basieren. Über die API des zugrunde liegenden MediaWikis können einzelne Abschnitte von Artikeln in Wikitext-Syntax bezogen werden. Insbesondere die Infobox ist hierbei von Interesse, denn dort gibt es bei vielen Film-Seiten einen Verweis auf die Romanvorlage.

Link	`http://en.wikipedia.org/`
Datenformat	Wikitext, JSON, XML
Schnittstelle	Rest-API
Lizenz	CC BY-SA 3.0 + GFDL
Open Data	⋆⋆⋆

2.2 DBpedia

Die DBpedia hat sich die Extraktion von Informationen aus der Wikipedia und ihre Bereitstellung in maschinenlesbarer, strukturierter Form zur Aufgabe gemacht. Über SPARQL-Endpunkte können diese Daten direkt im RDF-Format abgerufen werden.

Link	`http://dbpedia.org/`
Datenformat	RDF
Schnittstelle	SPARQL
Lizenz	CC BY-SA 3.0 + GFDL
Open Data	⋆⋆⋆⋆⋆

2.3 OMDb

Die Open Movie Database ermöglicht den Abruf von diversen Film-Metadaten in strukturiertem Format. Die Suche kann über die IMDb-ID oder über den Titel (optional mit Erscheinungsjahr) erfolgen. Die Suche zeigt sich tolerant gegenüber alternativer Schreibweisen. Neben der IMDb-Bewertung werden auch die Rating-Scores von RottenTomatoes bereitgestellt.

Link	`http://www.omdbapi.com/`
Datenformat	JSON, XML
Schnittstelle	Rest-API
Lizenz	CC BY-NC 4.0
Open Data	⋆⋆⋆

2.4 Goodreads

Die zu Amazon gehörende Website Goodreads bietet Zugang zu einer großen Datenbank an Metadaten, Bewertungen und Rezensionen zu Büchern an. Die Suche mittels Titel und Autor lieferte eine Liste an möglichen Treffer. Neben einer durchschnittlichen Bewertung ist für viele Bücher auch das Datum der ursprünglichen Publikation auslesbar. Eine API kann nach kostenloser Registrierung mit einem Schlüssel genutzt werden. Der Zugriff muss allerdings auf einen Abruf pro Sekunde begrenzt werden.

Link	`https://www.goodreads.com/`
Datenformat	JSON, XML
Schnittstelle	Rest-API
Lizenz	siehe `https://www.goodreads.com/api/terms`
Open Data	⋆⋆⋆

3 Extraktion relevanter Daten und Import in einen Triplestore

Die Extraktion erfolgt mithilfe von Python-Skripten und Paketen, wie urllib3, json und lxml, zur Abfrage der APIs und zur Verarbeitung der Ergebnisse. Die erzeugten RDF-Triple werden mit der Bibliothek rdflib in einem internen Graph gespeichert, sowie für die einzelnen Quellen jeweils lokal auf der Festplatte im Turtle-Format gesichert.

3.1 Extraktion der Paare aus Wikipedia und DBpedia

In der Kategorie „Films based on Novels “werden tausende Werke gelistet, die auf Romanen basieren. Von diesen Filmen sind allerdings nur einige Dutzend über das Prädikat „dbo:basedOn“ bzw. „dbp:basedOn“ mit dem Buch verknüpft. Die Live-DBpedia liefert zwar mehr, aber auch vergleichsweise wenige Ergebnisse über folgende Abfrage:

Listing 1: SPARQL-Anfrage nach Film-Buch-Paaren anhand von dbo:basedOn

```
CONSTRUCT { ?film dbo:basedOn ?novel .
    ?film dbp:name ?filmtitle .
    ?film dbp:released ?filmyear .
    ?novel dbo:author ?author .
    ?novel dbp:name ?noveltitle .
    ?author dbp:name ?authorname . }
WHERE {
    ?film rdf:type dbo:Film .
    ?novel rdf:type dbo:Work .
    ?class skos:broader* dbc:Films_based_on_novels .
    ?film dct:subject ?class .
    ?film dbo:basedOn ?novel .
    ?film dbp:name ?filmtitle .
    ?film dbp:released ?filmyear .
    ?novel dbo:author ?author .
    ?novel dbp:name ?noveltitle .
    ?author dbp:name ?authorname
} GROUP BY ?film
```

Auf der Website der Wikipedia finden sich in den Infoboxen auf der rechten Seite jedoch deutlich häufiger Links auf das Buch unter dem Label „Based on“.
Daher werden die Informationen in einem mehrstufigen Prozess extrahiert:
Zunächst werden über den SPARQL-Endpunkt der Live-DBpedia die URIs aller Filme in der beschriebenen Kategorie abgerufen, die das Template „dbt:BasedOn“ nutzen.

Listing 2: SPARQL-Anfrage nach Filmen in einer bestimmten Wikipedia-Kategorie

```
SELECT ?film
WHERE {
?film rdf:type dbo:Film .
?class skos:broader* dbc:Films_based_on_novels .
?film dct:subject ?class .
?film dbp:wikiPageUsesTemplate dbt:Based_on
} GROUP BY ?film
```

Filmtitel und, falls eingetragen, auch das Erscheinungsdatum werden mit der URI aus der normalen DBpedia geholt, da diese weniger falsche Angaben aufweist. Falls es kein

eingetragenes Release-Datum gibt, wird die URL auf eine Jahreszahl geprüft, wie sie häufig bei Mehrfachverfilmungen nach dem Schema „... (1922 film)“ vorkommt. Anschließend wird über die Wikipedia API der Wikitext der oberen Seitenabschnitte der Filmartikel extrahiert. Mithilfe des Parsers mwparserfromhell wird der BasedOn-Eintrag der Infobox entnommen. Enthält dieser einen Verweis auf einen Buchartikel, so werden durch weitere Anfragen an die DBpedia, Autor und Buchtitel ermittelt. Im folgenden Listing wird die Buch-URI in die Anfrage als `<book>` eingesetzt.

Listing 3: SPARQL-Anfrage an die DBpedia nach dem Buch

```
SELECT ?noveltitle ?author ?authorname
WHERE {
<<book> rdf:type dbo:Work .
<book> dbo:author ?author .
<book> dbp:name ?noveltitle .
?author dbp:name ?authorname
} LIMIT 1
```

Die Namespaces der DBpedia werden beim Import in den Triplestore beibehalten.

Listing 4: Beispiel-Triple, die je Buch-Film-Paar erzeugt werden

```
dbr:Gone_Girl_(film) dbo:basedOn dbr:Gone_Girl_(novel) ;
dbp:name "Gone Girl" .
dbr:Gone_Girl_(novel) dbo:author dbr:Gillian_Flynn ;
dbp:name "Gone Girl" .
dbr:Gillian_Flynn dbp:name "Gillian Flynn" .
```

3.2 Extraktion OMDb

Mit Filmtitel und, falls vorhanden, dem Jahr wird die OMDb-API angefragt und liefert den besten Treffer im JSON-Format.

Listing 5: API-Anfrage an OMDb

```
http://www.omdbapi.com/?t=TITEL&y=JAHR&r=json&tomatoes=true
```

Die Informationen werden mit geeigneten Datentypen im Graphen eingetragen, wie das folgende Listing zeigt. Der Score wird als xsd:decimal, der Titel und das Land als xsd:string und das Jahr als xsd:dateTime angelegt. Diverse Funktionen, die später in der SPARQL-Anfrage genutzt werden, verlangen diesen Datentyp. Ausgewertet wird nur

die Jahreszahl. Die restlichen Zeitinformationen sind nur wegen des Datentyps generiert worden. Der Typ gYear ist in der rdflib bislang nicht implementiert.

Listing 6: Beispiel-Triple, die je Film erzeugt werden

```
<http://www.frohde.de/ontology/omdb/omdbID#tt0042189> omdb:country "USA, France, UK" ;
omdb:imdbRating 6.8 ;
omdb:title "Alice in Wonderland" ;
omdb:year "1949-01-01T00:00:00"^^xsd:dateTime .
```

3.3 Extraktion Goodreads

Die API von Goodreads wird mit den aus der DBpedia abgerufenen Parametern Titel und Autorenname angefragt.

Listing 7: API-Anfrage an Goodreads

```
https://www.goodreads.com/book/title.xml?author=AUTOR&title=TITEL&key=KEY
```

Die Antwort ist stärker untergliedert als das JSON von OMDb, daher wird an dieser Stelle XML verwendet, um die interessierenden Tags per XPath leichter adressieren zu können.
Die API ist empfindlich gegenüber Vertauschungen von Vor- und Nachnamen bei der Anfrage. Da in der DBpedia häufig mehrere Namen für Autoren hinterlegt sind (in verschiedenen Schreibweisen) und nur eine davon extrahiert wird, werden bei fehlerhaften Abrufversuchen verschiedene Namensvarianten durchprobiert, zB nur der erste Vorname und der Nachname bzw. beim letzten Versuch nur der Nachname.

Listing 8: Beispiel-Triple, die je Buch erzeugt werden

```
<http://www.frohde.de/ontology/goodreads/bookID#22034> gr:rating 4.35 ;
gr:title "The Godfather" ;
gr:year "1969-01-01T00:00:00"^^xsd:dateTime .
```

4 Verlinkung von Ressourcen

Der Abfrage der Film- bzw. Bücherbewertung geht jeweils eine SPARQL-Anfrage an den Graphen mit den Wikipedia-Daten voraus, um eine Liste der zu suchenden Medien zu erhalten. Daher können die Film- und Buchdaten leicht zugeordnet werden. Die

Verlinkung erfolgt über die OMDb-ID bzw. die Goodreads-ID (siehe Abb. 1). Die drei Graphen werden mit der rdflib in einen zusammenfassenden Datensatz importiert.

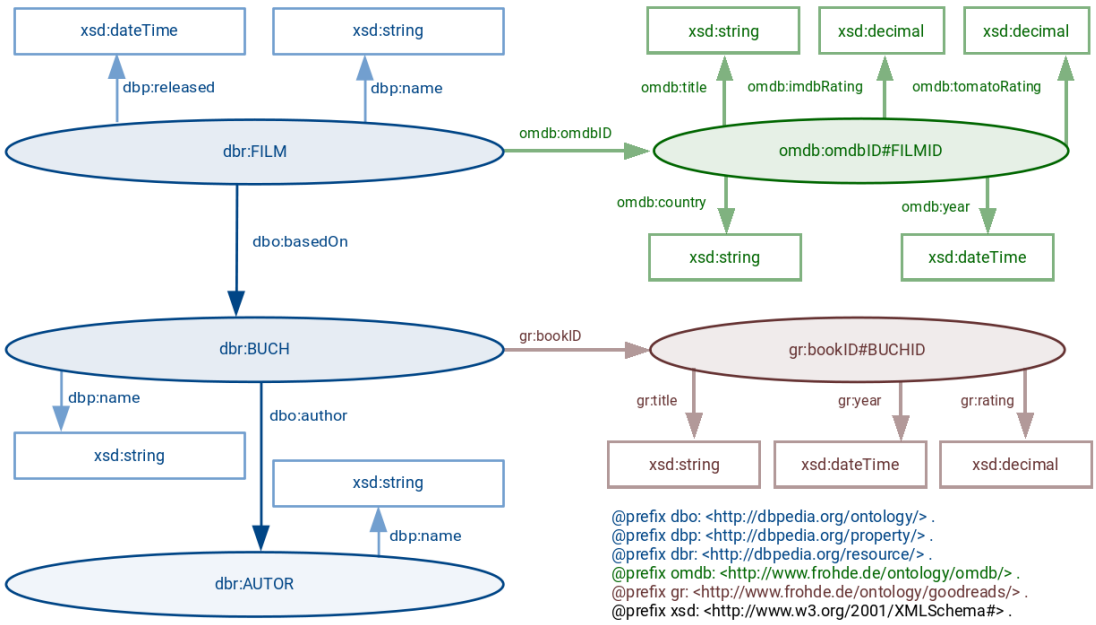

Abbildung 1: Ontologie

Der Triplestore enthält ca. 1200 Buch-Roman-Paare mit Bewertungen und weiteren Metadaten.

5 Anfragen an die Forschungswissensbasis

Es werden Anfragen an den Triplestore von rdflib nach der Verhältnis der Bewertungen von Buch und Film gestellt, sowie mögliche zeitliche Verschiebungen in diesem Verhältnis untersucht. Abschließend wird ermittelt, welche Bücher am häufigsten verfilmt wurden. Allen Anfragen werden die folgenden Prefixe vorangestellt:

Listing 9: Prefixe in den SPARQL-Anfragen

```
@prefix dbo: <http://dbpedia.org/ontology/> .
@prefix dbp: <http://dbpedia.org/property/> .
@prefix dbr: <http://dbpedia.org/resource/> .
@prefix gr: <http://www.frohde.de/ontology/goodreads/> .
@prefix omdb: <http://www.frohde.de/ontology/omdb/> .
```

In mehreren Anfragen werden die Zeilen im folgenden Listing genutzt, um die Bewertungsskalen von IMDb (0-10) und Goodreads (0-5) zu normieren. Außerdem wird die Jahreszahl mit der Funktion `year` aus dem dateTime-Typ bestimmt.

Listing 10: SPARQL-Abschnitt mit Normierung und Jahresauswahl

```
BIND (year(?filmDate) as ?filmYear)
BIND ((?novelRating/5) as ?novelNormed)
BIND ((?imdbRating/10) as ?imdbNormed)
```

5.1 Filme, die besser als ihre Vorlage bewertet werden

Die folgende Anfrage liefert das Verhältnis der Anzahl der Paare, bei denen das Buch auf der normierten Skala besser bewertet wurde, zur Gesamtzahl an Film-Buch-Paaren.

$$r = \frac{Anzahl\ Paare\ mit\ IMDb - Score > Goodreads - Score}{Gesamtanzahl\ Paare}$$

Die Anzahl der Paare, die die Bedingung erfüllen, wird über eine Kombination von IF- und SUM-Funktion bestimmt. Wenn die Bedingung erfüllt ist, wird eine 1 in die Summe gegeben, ansonsten eine 0. Durch einen Filter werden Romane, die keine Bewertung erhalten haben, vernachlässigt.
Das Ergebnis lautet `0.155`. Folglich wurden lediglich 15% der Verfilmungen besser bewertet als ihre Buchvorlagen.

Listing 11: SPARQL-Anfrage zum Bewertungsverhältnis

```
SELECT (SUM(IF(?imdbNormed > ?novelNormed, 1, 0))/COUNT(?film) as ?ratio)
WHERE {
    ?film dbo:basedOn ?novel .
    ?film omdb:omdbID ?omdbFilm .
    ?omdbFilm omdb:imdbRating ?imdbRating .
    ?novel gr:bookID ?grBook .
    ?grBook gr:rating ?novelRating
    BIND ((?novelRating/5) as ?novelNormed)
    BIND ((?imdbRating/10) as ?imdbNormed)
    FILTER (?novelRating != "0.00")
}
```

5.2 Bewertungen in Abhängigkeit von Erscheinungsdaten

Möglicherweise sind Verfilmungen im Lauf der Zeit besser oder schlechter geworden. Mit folgender Anfrage wird die (normierte) Bewertungsdifferenz nach den Erscheinungs-

jahren der Filme ausgegeben. Die Ausgabe wird im Python-Skript in eine CSV-Datei geschrieben, die anschließend in Libreoffice als Datengrundlage für das Diagramm in Abb. 2 dient. Die Bezeichnung „Durchschnittliche Bewertungsdifferenz“ steht für die Differenz von normierter Filmbewertung zu normierter Buchbewertung im Durchschnitt über alle Filme, die eine bestimmte Eigenschaft haben. Dies kann, wie im folgenden Beispiel, das Erscheinungsjahr sein oder, wie in der zweiten Anfrage dieses Abschnitts, die Zeitspanne zwischen den Erscheinungsdaten von Buch und Film. Eine positive Differenz bedeutet, dass der Film besser als das Buch bewertet wurde, eine negative Differenz entsprechend das Gegenteil.

Listing 12: SPARQL-Anfrage zur durchschnittlichen Bewertungsdifferenz nach Jahren

```
SELECT ?filmYear (AVG(?imdbNormed-?novelNormed) as ?diffRating)
WHERE {
    ?film dbo:basedOn ?novel .
    ?film omdb:omdbID ?omdbFilm .
    ?omdbFilm omdb:imdbRating ?imdbRating .
    ?omdbFilm omdb:year ?filmDate .
    ?novel gr:bookID ?grBook .
    ?grBook gr:rating ?novelRating
    BIND (year(?filmDate) as ?filmYear)
    BIND ((?novelRating/5) as ?novelNormed)
    BIND ((?imdbRating/10) as ?imdbNormed)
    FILTER (?novelRating != "0.00")
} GROUP BY ?filmYear
ORDER BY ?filmYear
```

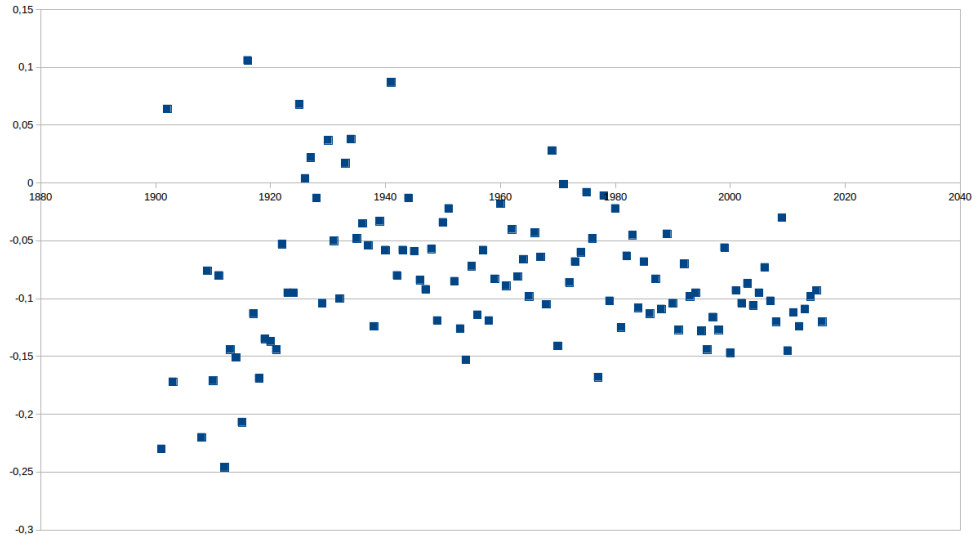

Abbildung 2: Durchschnittliche Bewertungsdifferenz im Zeitverlauf

Eine ähnliche Anfrage soll herausfinden, ob die durchschnittlichen Bewertungsdifferenz davon abhängig ist, wie lange nach dem Buch der Film erschienen ist.

Listing 13: SPARQL-Anfrage zur Bewertungsdifferenz nach Erscheinungszeitspanne

```
SELECT ?yearDiff (AVG(?imdbNormed-?novelNormed) as ?diffRating)
WHERE {
    ?film dbo:basedOn ?novel .
    ?film omdb:omdbID ?omdbFilm .
    ?omdbFilm omdb:imdbRating ?imdbRating .
    ?omdbFilm omdb:year ?filmDate .
    ?novel gr:bookID ?grBook .
    ?grBook gr:year ?novelDate .
    ?grBook gr:rating ?novelRating
    BIND ((year(?filmDate) - year(?novelDate)) as ?yearDiff)
    BIND ((?novelRating/5) as ?novelNormed)
    BIND ((?imdbRating/10) as ?imdbNormed)
    FILTER (?novelRating != "0.00")
} GROUP BY ?yearDiff
ORDER BY ?yearDiff
```

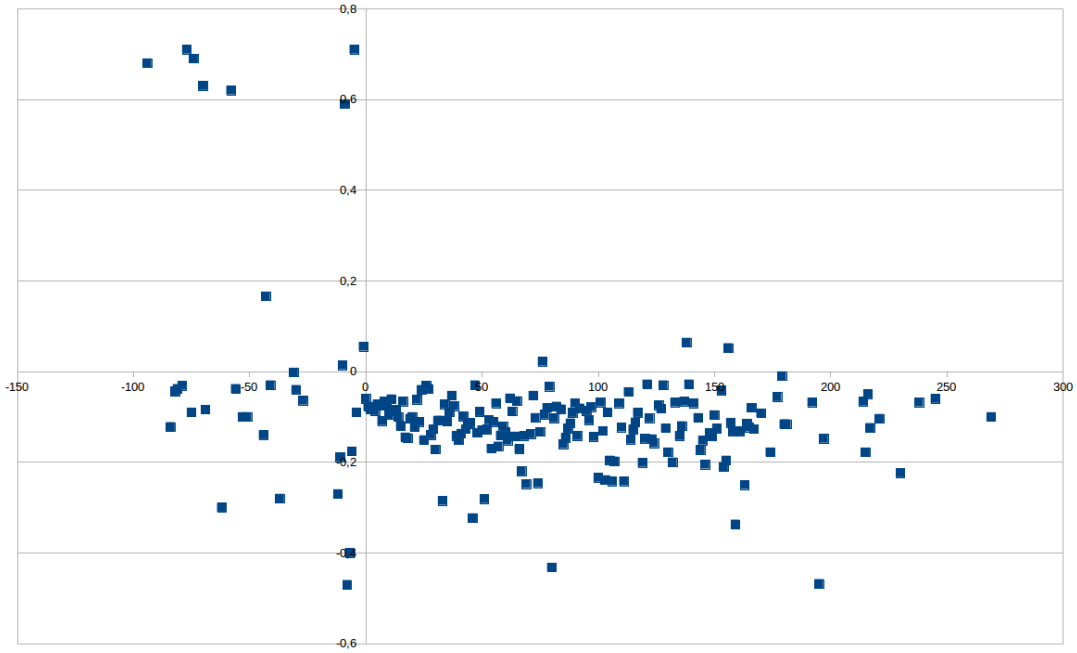

Abbildung 3: Durchschnittliche Bewertungsdifferenz in Abhängigkeit von der Erscheinungszeitspanne

5.3 Häufig adaptierte Bücher

Abschließend wird in einer Anfrage eine Liste derjenigen Bücher zusammengestellt, die am häufigsten adaptiert wurden. Die Tabelle zeigt nur die oberen Einträge der Ergebnisliste. Diese enthält alle Bücher, die mehr als einmal verfilmt wurden.

Listing 14: SPARQL-Anfrage zur durchschnittlichen Bewertungsdifferenz nach Jahren

```
SELECT (COUNT(?film) as ?adaptationsNum) ?novelTitle ?novelAuthor
WHERE {
    ?film dbo:basedOn ?novel .
    ?novel gr:bookID ?grBook .
    ?novel dbo:author ?author .
    ?grBook gr:title ?novelTitle .
    ?author dbp:name ?novelAuthor
} GROUP BY ?novel
HAVING (COUNT(?film) > 1)
ORDER BY DESC(?adaptationsNum)
```

Tabelle 1: Liste der am häufigsten verfilmten Bücher

Anzahl Verfilmungen	**Titel**	**Autor**
18	A Christmas Carol	Charles Dickens
15	Alice in Wonderland	Lewis Carroll
13	Oliver Twist	Charles Dickens
12	Frankenstein	Mary Shelley
12	Dracula	Bram Stoker
8	The Works of H. Rider Haggard	Haggard, Henry Rider
7	Pinocchio	Carlo Collodi
6	The Four Feathers	A. E. W. Mason
6	Tarzan of the Apes	Edgar Rice Burroughs
6	Great Expectations	Charles Dickens

6 Interpretation und Zusammenfassung

Die Buchvorlagen werden bei 85% der Verfilmungen als besser bewertet. Dieser Wert erscheint überraschend hoch. Eine Erklärung liegt möglicherweise in den Datenquellen. Es haben nicht die gleichen Personengruppen beide Medien bewertet und daher sind die durchschnittlichen Scores nur bedingt vergleichbar. Einen Einfluss hat vermutlich auch die unterschiedliche Skala. Eine 7 von 10 ist subjektiv besser als eine 3,5 von 5, auch wenn es eigentlich äquivalent ist. Daher tendieren Nutzer eventuell bei der kleineren Skala zu verhältnismäßig höheren Einstufungen.

Der Zeitverlauf in Abb. 2 zeigt, dass die Filme aus den ersten beiden Jahrzehnten des 20. Jahrhundert verhältnismäßig schlechter bewertet werden, als in den späteren Jahren. Alternativ könnte man auch vermuten, dass zu dieser Zeit besonders gute Bücher verfilmt wurden und die Filme deshalb schlechter abschneiden.

In Abb. 3 fällt auch, dass es Einträge auf der negativen Zeitachse gibt. Dies bedeutet, dass bei einigen Datensätzen der Film **vor** der Buchvorlage erschienen sein soll. Dies ist natürlich nicht möglich. Der Grund liegt vermutlich in den Daten von Goodreads, bei denen im Feld `original_publication_year` bei manchen Werken nicht die Erstveröffentlichung des Werkes, sondern der Publikation eingetragen ist. Dies kann bei Neufassungen der Bücher auch nach der Filmveröffentlichung gewesen sein.
Abgesehen von dieser Merkwürdigkeit werden die Filme nicht je nach Zeitspanne zum Buch unterschiedlich bewertet.

Charles Dickens ist mit seinen Werken „A Christmas Carol“, „Oliver Twist“ und „Great Expectations“ der Autor mit den meisten Verfilmungen.