„Würden Sie dieses Spiel anderen Spielern empfehlen?“

Topic Modeling mit R:
Rezensionen zum Spiel
‚Hogwarts Legacy‘

Fragestellung

Das Videospiel ‚Hogwarts Legacy' zählt zu den erfolgreichsten der neueren Zeit. In den ersten zwei Wochen nach Erscheinen wurden bereits über 12 Millionen Exemplare verkauft.[8] Viele Spieler geben ihre Bewertung auf Plattformen wie Steam ab, die wiederum Unentschlossenen als Orientierung dienen.

In diesem Projekt soll der Frage nachgegangen werden, welche Aspekte in den Bewertungen besonders genannt werden.

Dafür wird Topic Modeling mit R von 46405 englischsprachigen Bewertungen auf Steam durchgeführt.

Was ist Topic Modeling?

Topic Modeling (TM) ist ein unüberwachtes Verfahren in der natürlichen Sprachverarbeitung, um abstrakte Themen in einer Auswahl von Dokumenten zu identifizieren. Die TM-Algorithmen erzeugen statistische Modelle (Topics) zur Darstellung häufiger gemeinsamer Vorkommnisse von Wörtern.[2]

Was ist LDA?

Der Latent Dirichlet Allocation-Algorithmus nimmt jedes Dokument als eine Mischung aus allen Topics des Gesamtkorpus. Dabei hat jedes Wort eine spezielle Wahrscheinlichkeit zu einem Topic zu gehören. Die Topics hingegen bestimmen das gemeinsame auftreten von Wörtern in einem Text.

Der LDA gehört zu den am weitest verbreiteten Methoden, um ein TM-Model zu entwickeln.[7]

Vorgehen

0. Packages (installieren und) laden
1. Daten laden
2. Preprocessing
3. Model Building
4. Visualisierung

I Daten laden

Nachdem installieren und Laden aller nötigen Packages werden die Daten geladen.

Der ursprüngliche Dataframe enthält neben dem zu modellierenden Text auch noch weitere Informationen, die für ein erstes Model jedoch nicht benötigt werden.

Daher werden nur die „Review“-Spalte sowie eine ID in einen neuen Dataframe gespeichert.

	ï..	Playtime	Feedback	Review
1	0	16	Positive	Greattt Game!
2	1	26	Positive	9/10Fantastic experience. A true Wizarding World experienc...
3	2	29	Positive	worth it
4	3	24	Positive	I've been waiting 84 YEARSSSSSSSS.The game is everything ...
5	4	7	Positive	very fun game (it is not transphobic at all)
6	5	8	Positive	Better than expected! But bad optimization.
7	6	0	Negative	Garbage
8	7	3	Positive	Full of surprises!
9	8	2	Positive	Fun Harry Potter Game! My sister and I Enjoyed it.
10	9	15	Positive	One of the best games I have bought in my recent memory

2 Preprocessing

Bevor das Model angewandt werden kann, muss weitere irrelevante Information aus dem Text entfernt werden.

Dazu wird die „Review“-Spalte tokenisiert, bereinigt und in einem „tokens“-Dataframe gespeichert.

Die Schritte zur Textbereinigung beinhalten:

- Entfernen von Emojis, Zahlen und Zeichensetzung
- Herausfiltern von Kurz- und Stoppwörtern

Ergebnis Preprocessing

	ID	Review
1	1	greattt game
2	10	games bought recent memory
3	100	contender goty addictive nastalgic worth buy
4	1000	game peak
5	10000	game explore addicted day played
6	10001	playing dream true finally received hogwarts letter explorin...
7	10002	highly recommend game people enjoy free roam adventure...
8	10003	utiful
9	10004	recomend
10	10005	fun

3 Model Building

Zunächst wird der Text in eine Document-Term-Matrix (DTM) formatiert. Diese beinhaltet als Sparse-Matrix die Dokumente als Zeilen, die Wörter als Spalten.

Um den später auftretenden Fehler beim bauen des Models (Error: „empty rows“) zu vermeiden, werden zusätzlich leeren Zeilen gelöscht.

```
> as.matrix(dtm)[1:10,1:5]
    Terms
Docs game greattt bought games memory
  1    1       1      0     0      0
  2    0       0      1     1      1
  3    0       0      0     0      0
  4    1       0      0     0      0
  5    1       0      0     0      0
  6    1       0      0     0      0
  7    1       0      0     1      0
  8    0       0      0     0      0
  9    0       0      0     0      0
  10   0       0      0     0      0
```

LDA-Model

Verwendet wird die Gibbs-Methode.

- Sie ist vergleichsweise einfach zu implementieren.[9]
- Nach einigen Iterationen liefert sie stabile Ergebnisse.

Der wichtigste Parameter, um das geeignete Model zu finden, ist die Zahl der Topics k.

Zur Optimierung von k werden zwei Methoden verwendet:

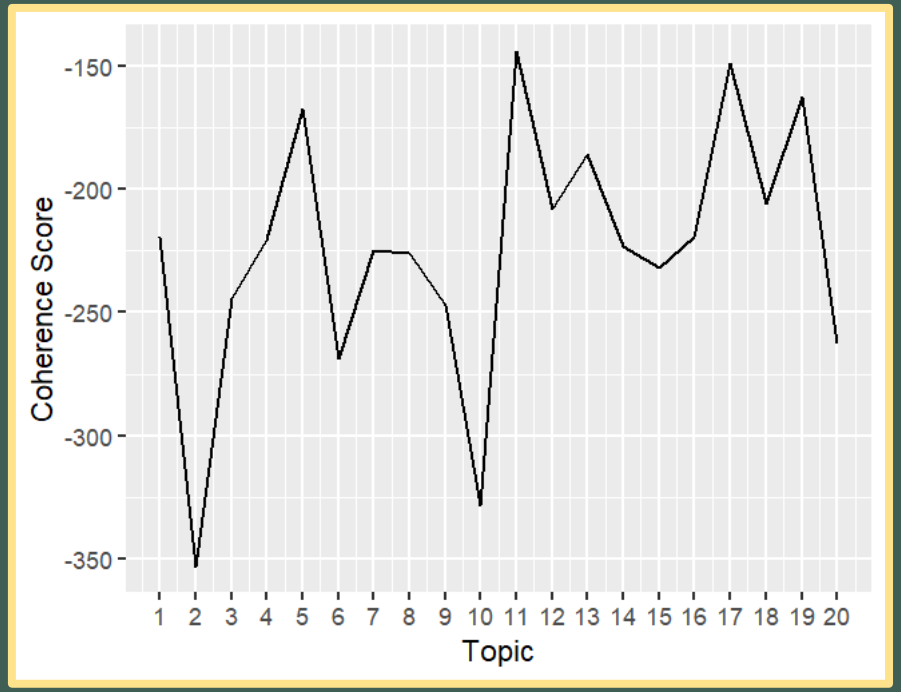
A: Berechnung der Kohärenz mit `topic_coherence()`

B: Kreuzvalidierung mit `FindTopicsNumber()`

Modeloptimierung A

Das Ziel der Kohärenzmessung mit `topic_coherence()` besteht darin, die semantische Kohärenz zwischen den Wörtern in einem Thema zu messen, indem sie bewertet, wie gut die Wörter sich zu einem Thema zuordnen lassen.

Je höher der coherence score für eine bestimmte Anzahl k, desto besser passen die Wörter je Topic zusammen.[1]



→ In dem gewählten Rahmen von 20 Topics springt der Graph stark. Auch der beste Wert von rund -150 liegt absolut außerhalb des akzeptablen Bereichs.[1]

Modeloptimierung B

Die Methode `FindTopicsNumber()` hingegen zielt auf eine Modeloptimierung mittels mehrerer Parameter. Dazu testet sie deren Kombination und führt eine Kreuzvalidierung durch.

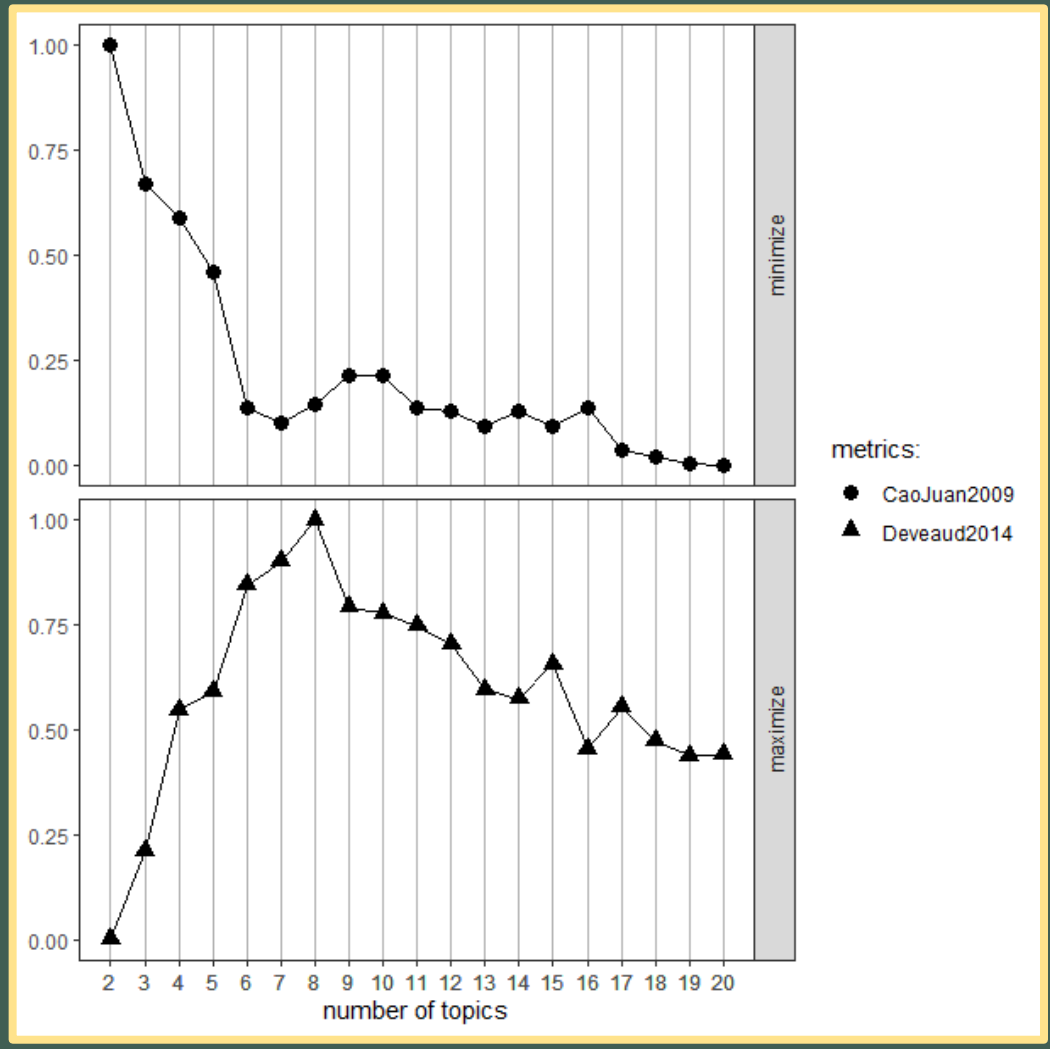
Es stehen dafür 4 Ansätze zur Verfügung, wobei für dieses Projekt 2 gewählt wurden:

- CaoJuan2009 misst die Kohärenz eines Themas über die Wahrscheinlichkeiten der Wörter im Thema[4]
- Deveaud2014 zieht zudem externe Daten hinzu, um die Semantik von Wörtern mit in die Berechnung miteinfließen zu lassen[5]

Modeloptimierung B

Je niedriger der score bei CaoJuan2009 und je höher bei Deveaud2014, desto besser ist die gewählte Zahl an Topics.

→ die beste Zahl Topics liegt bei 8



4 Visualisierung

Nach Bedarf lassen sich zudem die Verteilung der Topics über die Dokumente, sowie

- θ (Theta): Verteilung der Themen in einem bestimmten Dokument
- ϕ (Phi): Verteilung der Wörter innerhalb eines Themas

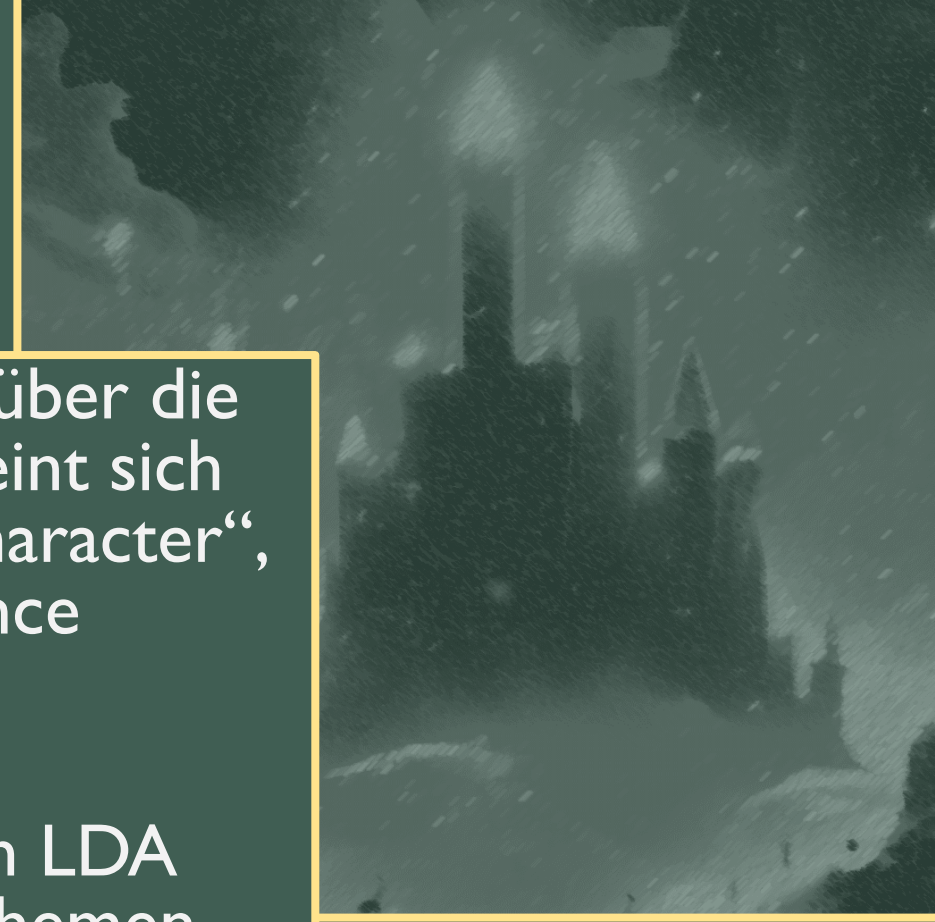
ausgeben.

Die 8 Topics mit ihren 10 wahrscheinlichsten Wörtern:

	Topic 1	Topic 2	Topic 3	Topic 4
[1,]	"character"	"performance"	"game"	"game"
[2,]	"main"	"issues"	"games"	"play"
[3,]	"quests"	"fps"	"dont"	"time"
[4,]	"spells"	"drops"	"buy"	"hours"
[5,]	"quest"	"settings"	"people"	"fix"
[6,]	"feel"	"review"	"single"	"playing"
[7,]	"dont"	"run"	"player"	"bugs"
[8,]	"spell"	"runs"	"makes"	"hope"
[9,]	"feels"	"patch"	"life"	"wait"
[10,]	"characters"	"stuttering"	"shit"	"enjoying"

	Topic 5	Topic 6	Topic 7	Topic 8
[1,]	"game"	"game"	"harry"	"world"
[2,]	"fun"	"amazing"	"potter"	"story"
[3,]	"awesome"	"recommend"	"love"	"hogwarts"
[4,]	"pretty"	"played"	"wizard"	"combat"
[5,]	"nice"	"ive"	"fan"	"beautiful"
[6,]	"magic"	"worth"	"avada"	"experience"
[7,]	"lots"	"absolutely"	"youre"	"gameplay"
[8,]	"revelio"	"graphics"	"kedavra"	"lot"
[9,]	"pet"	"fantastic"	"fans"	"explore"
[10,]	"bit"	"games"	"dream"	"immersive"

Fazit



Das LDA-Model liefert einen ersten Überblick über die in den Rezensionen genannten Aspekte. So scheint sich Topic 1 um den Inhalt des Spiels zu drehen („character“, „quest“, „spell“) während Topic 2 auf Performance Probleme hindeutet („performance“, „issues“, „stuttering“).

Somit finden sich durch Topic Modeling mit dem LDA logische Zusammenhänge in den errechneten Themen. Diese lassen sich in einer weiterführenden Beschäftigung mit anderen Spielen vergleichen oder zu weiteren Informationen der Ursprungsdaten, wie Spielzeit oder Bewertung, ins Verhältnis setzen.

Fazit

Es gibt aber auch Optimierungsbedarf, der sich im Laufe der Arbeit ergeben hat:

- Die Liste an Stoppwörtern sollten mit spezifischen für diese Fragestellung erweitert werden, etwa „game“ und „games“
- Einige der Rezensionen sind umgangssprachlich und zum Teil absichtlich falsch formuliert (vgl. erstes Dokument: „greattt Game“). Für genauere Ergebnisse sind diese in ihre ursprüngliche Form zu bringen.
- Für noch bessere Ergebnisse bei der Optimierung des Models sollte – bei ausreichend Rechenleistung – alle 4 Varianten der `FindTopicsNumber()` –Methode verwendet werden.

Datensatz

Verwendet wurde folgender Datensatz von kaggle.de an 46405 Bewertungen zum Spiel ‚Hogwarts Legacy' auf Steam:

https://www.kaggle.com/datasets/georgescutelnicu/hogwarts-legacy-reviews

Quellen

Die verwendeten Graphen und Code-Beispiele stammen aus dem beigefügten R-Skript.

Der Aufbau der Arbeit orientiert sich an folgenden Tutorials:

[1]Farren tang (2019). Beginner‘s Guide to LDA Topic Modeling. Towardsdatascience.com. URL: https://towardsdatascience.com/beginners-guide-to-lda-topic-modelling-with-r-e57a5a8e7a25

[2]Martin Schweinberger (2023). Topic Modeling with R. lada.edu.au. URL: https://ladal.edu.au/topicmodels.html

Sofern nicht anders angegeben wurden die Bilder generiert von Dall-E:

[3]https://labs.openai.com/

Quellen

Weitere Quellen:

[4]Juan Cao et al. (2007). A density-based method for adaptive LDA model selection. URL: https://doi.org/10.1016/j.neucom.2008.06.011

[5]Romain Deveaud et al. (2014). Accurate and effective latent concept modeling for ad hoc information retrieval. In Document numérique. Revue des sciences et technologies de l'information. Série Document numérique, 2014, pp.61-84. URL: https://dn.revuesonline.com/article.jsp?articleId=19419

[6]Github-Issue Thread (2017). Error: "Each row of the input matrix needs to contain at least one non-zero entry". URL: https://github.com/nikita-moor/ldatuning/issues/6

[7]Julia Silge und David Robinson (2022). Text Mining with R: A Tidy Approach. tidytextmining.com. URL: https://www.tidytextmining.com/topicmodeling.html

[8] Peter Steinlechner (2023). 12 Millionen Muggel kaufen Hogwarts Legacy. Golem.de. URL: https://www.golem.de/news/harry-potter-12-millionen-muggel-kaufen-hogwarts-legacy-2302-172150.html

[9]Anku Tomar (2018). Topic modeling using Latent Dirichlet Allocation (LDA) and Gibbs Sampling explained!. Medium.com. URL: https://medium.com/analytics-vidhya/topic-modeling-using-lda-and-gibbs-sampling-explained-49d49b3d1045