

Bauhaus-Universität Weimar  
Fakultät Medien  
Studiengang Medieninformatik

# Korpuskonstruktion und Entwicklung einer Pipeline für Clickbait-Spoiling

## Bachelorarbeit

Bagrat Ter-Akopyan  
geb. am: 10.02.1992 in Jerewan, Armenien

Matrikelnummer 110151

1. Gutachter: Junior-Prof. Dr. Matthias Hagen
2. Gutachter: Prof. Dr. Volker Rodehorst

Datum der Abgabe: 23. Oktober 2017

# **Erklärung**

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

Weimar, 23. Oktober 2017

.....  
Bagrat Ter-Akopyan

## Zusammenfassung

Diese Arbeit widmet sich einem im Internet weitverbreiteten Phänomen – dem *Clickbaiting* (deutsch etwa „Klickköder“). Ziel der Arbeit ist die Entwicklung eines prototypischen, semi-automatisierten Verfahrens zur Auflösung (wird in dieser Arbeit als Synonym für Spoiling verwendet) von *Clickbait*-Nachrichten. Die Arbeit stellt den ersten *Clickbait-Spoiler*-Korpus vor, der aus 7229 *Clickbait-Spoiler*-Pärchen besteht. Die *Clickbait*-Schlagzeilen wurden auf der Basis linguistischer Merkmale manuell analysiert und in unterschiedliche Klassen eingeteilt. In Anlehnung an *Question Answering* wird in dieser Arbeit eine mögliche *Pipeline* zum *Spoiling* von *Clickbaits*, mit dem Fokus auf die Extraktion von Spoilerkandidaten aus dem verlinkten Dokument sowie die Entwicklung und Evaluation von Verfahren zur Gewichtung von Kandidaten für faktenbasierte *Clickbaits*, im Speziellen solche, die einen Eigennamen als Spoiler erwarten, vorgestellt. Ein statistisches Ranking-Verfahren von extrahierten Kandidaten, das nur auf der Häufigkeit des Kandidaten im verlinkten Dokument basiert, erreichte dabei eine Genauigkeit von 68,93%.

# Inhaltsverzeichnis

<b>1 Einleitung</b>	<b>1</b>
<b>2 Verwandte Arbeiten</b>	<b>5</b>
<b>3 Korpuskonstruktion und Analyse</b>	<b>8</b>
3.1 Rohdatenerhebung und Vorverarbeitung . . . . .	8
3.2 Korpusanalyse . . . . .	10
<b>4 Faktenbasiertes Clickbait-Spoiling</b>	<b>14</b>
4.1 Inhalt- und Metadaten-Extraktion des verlinkten Dokuments . .	16
4.2 Clickbait-Verarbeitung . . . . .	16
4.3 Spoiler-Verarbeitung . . . . .	24
<b>5 Experimente</b>	<b>32</b>
5.1 Versuchsaufbau . . . . .	32
5.2 Ergebnisse . . . . .	33
<b>6 Zusammenfassung und Ausblick</b>	<b>36</b>
<b>Literaturverzeichnis</b>	<b>39</b>

# Kapitel 1

## Einleitung

Spätestens seit der globalen Verbreitung des Internets und des *World Wide Web* ist der Prozess der *Digitalisierung* nicht mehr aus unserem Alltag wegzudenken. Mittlerweile gibt es kaum noch einen Bereich, der nicht von diesem Wandel betroffen ist – Politik, Bildung, Kunst und Kultur. Auch der Journalismus ist stark von der *Digitalen Revolution* betroffen. Immer mehr Zeitungen und Magazine stellen ihren Lesern eine Online-Ausgabe zur Verfügung. Öffentlich-rechtliche und private Sender ergänzen ihr Angebot durch die Online-Berichterstattung und es entstehen neue Medienunternehmen, wie beispielsweise *BuzzFeed*, die ihre Inhalte ausschließlich online anbieten.

Zusätzlich besitzen die Verleger auch mehrere Auftritte in den sozialen Medien, wie zum Beispiel auf Facebook und Twitter, um eine höhere Reichweite und somit mehr Leser mit ihren Inhalten zu erreichen. Die Verbreitung der Inhalte erfolgt unter anderem in Form von Schlagzeilen, die in den sozialen Medien platziert werden, um die Aufmerksamkeit der Leser zu erreichen. Fühlt sich der Nutzer durch die formulierte Schlagzeile angesprochen und wenn er mehr über das in der Schlagzeile angedeutete Thema erfahren möchte, gelangt er durch das Anklicken der zugehörigen URL auf die Internetpräsenz des jeweiligen Anbieters.

Mit der Entwicklung neuer Methoden zur Analyse des Nutzerverhaltens bezüglich des Erfolgs einer Schlagzeile, der sich in der Anzahl der Klicks wider spiegelt, den Fortschritten im Bereich des maschinellen Lernens, die sich bereits auf viele Bereiche unseres Alltags auswirken und den bereits vorhandenen Kenntnissen auf dem Gebiet der menschlichen Psychologie, kam es zu einem Phänomen, das in den Medien und unter Forschern als *Clickbaiting* bezeichnet wird, das schließlich durch die hohe Verbreitung im Internet eine mediale Relevanz bekam und so das Interesse der Forscher aus unterschiedlichen Bereichen weckte.

Als *Clickbaits* werden Schlagzeilen bezeichnet, die in erster Linie dem Leser

**SELECTED FOR YOU**

Sponsored Links by Taboola ▶



This Edible One-Use Razor Is Sweeping the Shave Market in Singapore  
RazurMatch



Forget the iPhone 7, This Is the iPhun And It's Only \$19.999.  
Up and Aiwa



Home Buyer 101: Top Seven Smells That Sell  
Realtorque



Who Is This? I'm In Love With Her And Need To  
LuvLorne



Why This Egg-Shaped Camper is Changing The Way We Think About Breast Cancer  
PopCamp



These Twins Invented The Internet 150 Years Ago and No One Knew  
Twin News



11 Things You Didn't Know About The Cambodian Genocide  
Genosi.de



See the Online Furniture Store That Has Microsoft Worried  
MicrosoftFurniture.biz

**Abbildung 1.1:** Clickbait-Beispiele.<sup>1</sup>

die relevanten Informationen vorenthalten, die zum Verständnis des Kontexts beitragen würden, und solche, die zu Übertreibungen neigen, um irreführende Erwartungen zu provozieren (Peysakhovich and Hendrix, 2016). Im Zusammenhang mit *Clickbaits* wird häufig von einer Neugierlücke (engl. *curiosity gap* (Loewenstein, 1994)) gesprochen, die den Leser dazu verleiten soll, auf den mitgelieferten Link zu klicken (Wikipedia, 2016). Ein weiteres Merkmal von *Clickbaits* zeichnet sich dadurch aus, dass es eine starke Diskrepanz zwischen dem, was die Schlagzeile verspricht und dem tatsächlichen Inhalt, der in dem verlinkten Artikel angeboten wird, besteht (Bilton, 2014). Der Informationsgehalt des vermittelten Artikels ist meist von eher niedriger Qualität und wirkt auf den Leser eher enttäuschend, da der Inhalt des Artikels bei Weitem nicht seinen Erwartungen entspricht. Des Öfteren werden solche *Clickbaits* in den sozialen Medien durch grafische Elemente, wie z.B. einem Bild, zusätzlich ergänzt. Abbildung 1.1 zeigt einige Beispielhafte *Clickbait*-Schlagzeilen.

Der, dank der Digitalisierung, uneingeschränkte Zugang zum Wissen erleichtert unser alltägliches Leben in sämtlichen Bereichen. Bei einer sorgfältigen

<sup>1</sup> <https://matthewcmariner.com/2016/05/11/selected-for-you-1/>. Zuletzt besucht: 23. Oktober 2017

und kritischen Quellenauswahl gibt es kaum eine Frage, auf die wir nicht eine qualitativ hochwertige Antwort finden könnten. Doch die Erfahrungen aus der Vergangenheit zeigen, dass die technologischen Neuentwicklungen nicht immer ausschließlich im Interesse der Bildung und der Aufklärung und zum Nutzen der Menschheit eingesetzt werden. Ein sehr bekanntes und weit verbreitetes Beispiel ist der E-Mail-Spam, durch den ungefragt werbender Inhalt in den Umlauf gebracht werden kann. Solche Nachrichten werden als lästig eingestuft und es wurden Spam-Filter entwickelt, um der Verbreitung solcher E-Mails vorzubeugen.

Eine ähnliche Entwicklung lässt sich auch im Fall von *Clickbaits* beobachten. Sie sind auf eine rasche Verbreitung in den sozialen Medien ausgelegt und so formuliert, dass sie möglichst viele Klicks erzielen. Durch die Schaltung von Online-Werbung lässt sich jeder Besuch der verlinkten Seite direkt monetarisieren. Daraus resultierend könnte die These aufgestellt werden, dass es bei der Verbreitung solcher Schlagzeilen nicht primär darum geht, Inhalte zu verbreiten, sondern um aus überspitzten Schlagzeilen auf Kosten des Nutzers via Werbung Profit zu schlagen.

Die Wissenschaft steht in der Pflicht, eine gesellschaftliche Verantwortung zu übernehmen, um diese ökonomischen Prinzipien der Nutzbarmachung und Steuerung von individuellen Interessen offenzulegen und damit negativen Entwicklungen entgegen zu wirken. In einem Interview mit Boris Rosenkranz für das Online-Magazin *Übermedien* betonte Martin Potthast die Relevanz dieser Verantwortung mit der Aussage, dass „[...] Technologien entwickelt werden müssen, die uns helfen, nicht ausgebeutet zu werden, zum Beispiel auf geistiger, mentaler Ebene“ (Rosenkranz, 2016). Daraus resultiert die Motivation dieser Arbeit, *Clickbaits* nicht nur als Techniken zu analysieren, die eine semantische Redundanz erzeugen (nämlich als betontes Zeichen, dessen schlichte Antwort ein Klick ist und mehr nicht), sondern vor allem als Techniken anzuerkennen, die an unserer Aufmerksamkeitsökonomie arbeiten und so auch an unserem Mißtrauen gegenüber digitalen Inhalten.

Komplementär zu den im Kapitel 2 aufgeführten Arbeiten, die sich einerseits mit der linguistischen Analyse der *Clickbaits* und andererseits mit der automatisierten Erkennung von *Clickbaits* beschäftigen, gibt es eine Reihe von engagierten Nutzern (Beckman, 2014; Mizrahi, 2012; Stempeck, 2013), die auf die in den sozialen Medien verbreiteten *Clickbait*-Schlagzeilen mit einem Spoiler reagieren. Der Spoiler ist in diesem Zusammenhang genau der Teil der Information, den eine *Clickbait*-Schlagzeile nicht erzählt und auf die sie im verlinkten Dokument verweist. Abbildung 1.2 zeigt ein Beispiel eines Spoilers als *Retweet* auf ein *Clickbait*.



Abbildung 1.2: Ein Spoilerbeispiel als Antwort auf einen Tweet von *Mashable*.<sup>2</sup>

Diese Arbeit greift die Idee der *Clickbait*-Auflösung auf und entwickelt in Anlehnung an die Verfahren aus dem Bereich *Question Answering* eine *Pipeline*, um einen Spoiler für einen *Clickbait* maschinell aus dem zugehörigen, verlinkten Dokument zu extrahieren.

Im Kapitel 3 dieser Arbeit wird die Konstruktion eines Datenkorpus sowie die linguistische Analyse der *Clickbait*-Schlagzeilen hinsichtlich ihres vorwärtsweisenden Charakters näher beschrieben. Das Augenmerk der linguistischen Auseinandersetzung liegt dabei auf Manifestationen einer Katapher als eine direkte Referenz auf eine konkrete Entität innerhalb des verlinkten Dokuments. Im Kapitel 4 wird die entwickelte *Pipeline* detailliert erläutert. In diesem Zusammenhang bildet die Extraktion der Inhalte der jeweils verlinkten Webseiten den ersten Schritt, der die Suche nach einem Spoiler erst ermöglicht. Des Weiteren unternimmt die Arbeit den Schritt der *Clickbait*-Verarbeitung, bei dem es um die Extraktion zusätzlicher Informationen aus der Schlagzeile geht, um die Menge der möglichen Spoilerkandidaten im weiteren Verlauf zu minimieren. Der letzte Schritt der *Pipeline* widmet sich der eigentlichen Spoilerverarbeitung, bei der es einerseits um die Extraktion der Spoilerkandidaten aus dem verlinkten Dokument und andererseits um die Entwicklung unterschiedlicher Ansätze für die Kandidatengewichtung geht. Im Kapitel 5 werden die entwickelten Ansätze anschließend anhand von Experimenten auf Testdaten evaluiert.

---

<sup>2</sup> <https://twitter.com/SavedYouAClick/status/883513293217951744>. Zuletzt besucht: 23. Oktober 2017

# Kapitel 2

## Verwandte Arbeiten

Aufgrund der weiten Verbreitung der Clickbait-Schlagzeilen und der daraus entstandenen medialen Aufmerksamkeit ist das Phänomen in den Blickwinkel sowohl kommerzieller *Social-Media*-Plattformen als auch der wissenschaftlichen Forschung gerückt.

Facebook erklärt in einer Meldung, dass es aktiv gegen solche Schlagzeilen vorgehen wird (El-Arini and Tang, 2014). In der Mitteilung werden zwei Aspekte erwähnt, die zur Clickbait-Erkennung in Betracht gezogen werden: die Zeit, die der Nutzer auf der verlinkten Seite verbringt und das Verhältnis zwischen der Anzahl der Klicks und der *Likes* der Schlagzeile.

In der Arbeit „*Clickbait Detection*“ (Potthast et al., 2016) werden Tweets manuell entweder als Clickbait oder nicht als solcher eingestuft. Basierend darauf wurde ein maschinelles Lernsystem entwickelt, das unter Berücksichtigung von 215 Merkmalen Clickbaits erkennt. Das entwickelte System betrachtet dabei nicht nur die Eigenschaften der Schlagzeile, sondern zieht ebenfalls die Eigenschaften des Inhalts im verlinkten Dokument sowie Metadaten über den Tweet in die Analyse mit ein. Das Verfahren erreicht mit dem *Random Forest Classifier* eine Performanz von 0.79 *ROC-AUC* zu 0.76 Präzision und 0.76 Trefferquote und stellt damit eine starke *Baseline* im Bereich der Clickbait-Erkennung auf.

Eine weitere Arbeit „„8 Amazing Secrets for Getting More Clicks‘: Detecting Clickbaits in News Streams Using Article Informality“ (Biyani et al., 2016) konstruiert ebenfalls einen Datenkorpus und erstellt eine eigene Kategorisierung von Clickbaits. Tabelle 2.1 schildert die jeweiligen Kategorien mit Definitionen und Beispielen. Auch diese Arbeit entwickelt ein maschinelles Lernverfahren zur Clickbait-Erkennung mit Hilfe von 7677 Merkmalen. Einen besonderen Wert legt die Arbeit auf die Auswahl von Merkmalen, wie Informalität und Lesbarkeit, die sich auf den Inhalt des verlinkten Dokuments beziehen. Die Arbeit zeigt, dass Informalität ein starker Indikator dafür ist, ob eine Schlag-

zeile als Clickbait eingestuft wird. Das in dieser Arbeit entwickelte Modell erreicht eine Performanz von 74.9 F-Maß. Das Bedürfnis nach Neuentwicklungen im Bereich der automatisierten Clickbait-Erkennung führte dazu, dass vom Lehrstuhl für Webtechnologien und Informationssysteme an der Bauhaus-Universität Weimar ein *Shared Task* in diesem Bereich ausgeschrieben wurde (Gollub et al., 2017).

Auch im Bereich der Psychologie und Linguistik wurde das Clickbait-Phänomen bereits mehrfach behandelt. Im Mittelpunkt der Arbeit von Vijgen (Bram, 2014) stehen Schlagzeilen, die auf eine Liste einer bestimmten Entität verweisen, auch *listicles* genannt. Ein wesentliches Merkmal solcher Schlagzeilen besteht darin, dass sie Kardinalzahlen enthalten und 85% solcher Schlagzeilen beginnen sogar mit solch einer Zahl. Die Sensationsgier und der Einfluss solcher lockenden Nachrichten wird durch den Gebrauch starker Nomen und Adjektive verstärkt. Blom und Hansen (Blom and Hansen, 2015) analysieren 2000 zufällig ausgewählte Schlagzeilen hinsichtlich der Verwendung von Vorwärtsreferenzen als Mittel, um die Neugierde zu wecken. Sie identifizieren zwei Arten von Vorwärtsreferenzen: Diskursdeixis (1) und Katapher (2). Während eine Diskursdeixis auf eine diskursartige Auseinandersetzung verweist, bezieht sich eine Katapher auf eine konkrete Entität im verlinkten Text. Die Erkenntnisse der Arbeit von Blom und Hansen werden im Abschnitt 3.2 näher erläutert und fließen in die Analyse dieser Arbeit ein.

Typ	Definition	Beispielsatz
Exaggeration	Title exaggerating the content on the landing page.	Cringeworthy tattoos that will destroy your faith in humanity.
Teasing	Omission of details from title to build suspense: teasing.	New twist in Panthers star's trial could end his season.
Inflammatory	Either phrasing or use of inappropriate/vulgar words.	Putin Punched at G20 Summit.
Formatting	Overuse of capitalization/punctuation, particularly ALL CAPS or exclamation points.	EXCLUSIVE: Top-Secret Method allowed a mother to break the world record: 12kg in 4 weeks!
Graphic	Subject matter that is salacious or disturbing or unbelievable.	Donatella Versace plastic surgery overload: Waxy face resembles melting candle.
Bait-and-switch	The thing promised/implied from the title is not on the landing page: it requires additional clicks or just missing.	Beers Americans No Longer Drink.
Ambiguous	Title unclear or confusing to spur curiosity.	Hands on: Samsung's iPhone 5 is absolutely beautiful.
Wrong	Just plain incorrect article: factually wrong.	Scientist Confesses: „Global Warming a \$22 Billion Scam“.

**Tabelle 2.1:** Typen von Clickbaits in Anlehnung an Biyani (Biyani et al., 2016).

# Kapitel 3

## Korpuskonstruktion und Analyse

In diesem Kapitel werden die Konstruktion des Datenkorpus und die linguistische Analyse der Clickbaits vorgestellt. Im ersten Abschnitt des Kapitels wird der Prozess der Rohdatenerhebung erläutert. Der zweite Abschnitt des Kapitels fokussiert sich auf die Analyse der Schlagzeilen aus der linguistischen Sicht sowie der zugehörigen Spoiler. In diesem Zusammenhang werden der Begriff der Vorwärtsreferenz und die sprachlichen Einheiten, Diskursdeixis und Kataphер, die zur Kohäsion eines Textes beitragen, präzisiert und mit Beispielen aus dem Korpus veranschaulicht.

### 3.1 Rohdatenerhebung und Vorverarbeitung

Der erste Schritt dieser Arbeit widmet sich der Konstruktion eines Datenkorpus mit Clickbaits und den dazugehörigen Spoilern. Wie in der Zielsetzung im Kapitel 1 bereits aufgeführt, gibt es in den sozialen Medien Twitter- und Facebook-Nutzer (Beckman, 2014; Mizrahi, 2012; rlshashSYAC, 2016; Stempeck, 2013; StopClickbaitOfficial, 2016), die aktiv gegen Clickbaits vorgehen, indem sie die in der Schlagzeile verschwiegene Information aus dem verlinkten Text manuell extrahieren und in einem neuen Post den Clickbait auflösen. Twitter und Facebook stellen für ihre Plattform jeweils eine Schnittstelle zur Anwendungsprogrammierung bereit und ermöglichen dadurch den Zugriff auf die jeweiligen Seiten und das Abfragen von Tweets und Facebook-Posts in Form einer HTTP-GET-Anfrage. Eine Übersicht zu den Seiten, die solche Spoiler veröffentlichen, und der Anzahl der gesammelten Posts wird in Tabelle 3.1 angegeben.

Die Tweets bzw. Facebook-Posts werden im *line-delimited-json*-Format für die

Onlineplattform	Benutzer	Anzahl Clickbaits
Twitter	rslashSYAC <sup>3</sup>	3024
	HuffPoSpoilers <sup>4</sup>	1886
	SavedYouAClick <sup>5</sup>	582
	UpworthySpoiler <sup>6</sup>	466
Facebook <sup>7</sup>	Stop Clickbait – Lifestyle	227
	Stop Clickbait – Science	197
	Stop Clickbait – Entertainment	160
	Stop Clickbait – WTF	132
	Stop Clickbait – Gaming	106
	Stop Clickbait – World News	104
	Stop Clickbait – Aww	93
	Stop Clickbait – India	63
	Stop Clickbait – Technology	62
	Stop Clickbait – Funny	46
Stop Clickbait – Sports		41
Stop Clickbait – Pilipinas		40
		Insgesamt: 7229

**Tabelle 3.1:** Anzahl der gesammelten Clickbait-Spoiler Pärchen pro Quelle.

weitere Nutzung abgespeichert. Dabei repräsentiert jede Zeile einen Tweet bzw. einen Facebook-Post mit unterschiedlichen Attributen, die dabei mitgeliefert wurden. So hat zum Beispiel jeder Facebook-Post ein *Key-Attribut* „*likes*“, das als Wert eine Liste mit Nutzernamen und *IDs* von Nutzern enthält, die den Beitrag mit „Gefällt mir“ markiert haben. Jeder Tweet hingegen hat beispielsweise ein *Key-Attribut* „*retweeted*“, das einen Booleschen Wert enthält, ob ein Tweet ein Retweet ist, oder nicht. In der weiteren Untersuchung werden aus der umfangreichen Menge an Informationen die notwendigen Attribute extrahiert. In Abbildung 3.1 sind diese repräsentativ aufgeschlüsselt. Jeder Tweet bzw. Facebook-Post hat eine eindeutige *ID*, unter der man ihn erneut im originalen Rohkorpus finden kann und die unter dem Attribut „*post\_id\_str*“ gespeichert wird. Die „*cb\_headline*“ ist die original Clickbait-Schlagzeile und der „*cb\_spoiler*“ ist der vom Menschen formulierte Spoiler. Unter „*target\_url*“ ist der Link abgespeichert, der auf das vom Clickbait verlinkte Dokument

<sup>3</sup> <https://twitter.com/rslashSYAC>. Zuletzt besucht: 23. Oktober 2017

<sup>4</sup> <https://twitter.com/HuffPoSpoilers>. Zuletzt besucht: 23. Oktober 2017

<sup>5</sup> <https://twitter.com/savedyouaclick>. Zuletzt besucht: 23. Oktober 2017

<sup>6</sup> <https://twitter.com/UpworthySpoiler>. Zuletzt besucht: 23. Oktober 2017

<sup>7</sup> <https://www.facebook.com/StopClickBaitOfficial>. Zuletzt besucht: 23. Oktober 2017

```
{
    "post_id_str" : "567763744483467265",
    "cb_headline" : "First California Republican wades
                    into 2016 Senate race",
    "cb_spoiler" : "CA Assemblyman Rocky Chávez",
    "spoiler_publisher" : "HuffPoSpoilers",
    "social_media_platform" : "Twitter",
    "target_url" : "http://huff.to/1yS0rZo"
}
```

**Abbildung 3.1:** Korpus-Auszug: ein Tweet mit den notwendigsten Attributen.

zeigt. Die Attribute „*spoiler\_publisher*“ und „*social\_media*“ werden der Vollständigkeit halber extrahiert und übernommen, um jeden Post eindeutig einer Plattform und einem Nutzerkonto zuordnen zu können.

Die oben genannten Attribute werden für jeden Tweet bzw. Facebook-Post aus den Rohdaten extrahiert und im selben Format *line-delimited-json* wie die Rohdaten organisiert. Der resultierende Korpus ist in einer Datei mit 7229 Zeilen gespeichert, wobei in jeder Zeile ein json-Objekt mit den bereits erwähnten Attributen einen Tweet bzw. einen Facebook-Post repräsentiert.

## 3.2 Korpusanalyse

Die folgende Analyse des aggregierten Korpus stellt eine wichtige Grundlage für diese Arbeit dar und wird aus der linguistischen Sicht, aufbauend auf den Erkenntnissen aus der Arbeit „Click bait: Forward-reference as lure in online news headlines.“ (Blom and Hansen, 2015), durchgeführt. Um die Erkenntnisse dieser Arbeit auf die linguistische Analyse des konstruierten Korpus anzuwenden, ist es notwendig, einige Begriffe aus dem Bereich der Linguistik und Pragmatik im Vorfeld zu definieren.

Wie bereits im Kapitel 1 beschrieben, wird die relevante Information in den Clickbait-Schlagzeilen oft verschwiegen, bzw. mit Hilfe von anderen sprachlichen Mitteln im verlinkten Dokument referenziert. Blom und Hansen stellen fest, dass Clickbait-Schlagzeilen zwei Arten von Vorwärtsreferenzen verwenden: *Diskursdeixis* und *Katapher*. In der *Pragmatik* und *Semantik* wird die *Diskursdeixis* als eine Referenz auf den jeweils nachfolgenden Diskurs definiert (Youwen, 2011). Im Fall von Clickbaits verweist die Referenz auf die thematische Auseinandersetzung im verlinkten Dokument. *Katapher* verweist ebenfalls auf eine bestimmten Stelle im nachfolgenden Text, jedoch nicht auf einen Dis-

(1) „*Here's why summer in New York City smells so awful*“

(2) „*She planted these tea bags in her garden*“

**Abbildung 3.2:** Verwendung einer Diskursdeixis (1) und einer Katapher (2) in Clickbaits.

Mittel	Beispiel
Demonstrativpronomen	<i>This state could be next to legalize pot...</i>
Personalpronomen	<i>She planted these tea bags in her garden</i>
Adverbien des Ortes	<i>Here Is One Thing You Can Do That...</i>
Bestimmter Artikel	<i>The world's busiest airport</i>
Imperativ	<i>See How Much Money Floyd Mayweather...</i>
Interrogativpronomen	<i>What Trump is Hiding About Canada...</i>
Allgemeine Nomen	<i>Girls star opens up about her eating disorder</i>

**Tabelle 3.2:** Sprachliche Mittel der Vorwärtsreferenz (Blom and Hansen, 2015).

kurs, sondern auf ein ganz konkretes Wort bzw. eine Phrase (Halliday and Hasan, 1976).

Die folgenden Clickbait-Beispiele aus dem Korpus verdeutlichen den Unterschied zwischen den zwei genannten Arten der Vorwärtsreferenz: Das Ortsadverb *Here* im Beispiel (1) aus Abbildung 3.2 verweist deiktisch auf den Inhalt im verlinkten Dokument. Der Leser ist gezwungen, sich mit dem verlinkten Dokument auseinanderzusetzen, um zu verstehen, worauf das Adverb referenziert. Es handelt sich hierbei um einen Textausschnitt, in dem mehrere Gründe für die Aussage genannt werden. Das Personalpronomen *She* im Beispiel (2) hingegen, fungiert als Katapher und referenziert einen Namen im verlinkten Dokument. Ein wesentlicher Unterschied zwischen einer Diskursdeixis und einer Katapher besteht darin, dass eine Katapher und der referenzierte Teil im Dokument auf ein und dieselbe Entität verweisen, zum Beispiel eine Person oder einen Ort, während eine Diskursdeixis einen Textabschnitt referenziert, der wiederum neue Zusammenhänge beinhaltet. Für die Konstruktion einer Diskursdeixis oder einer Katapher können unterschiedliche Sprachmittel benutzt werden. Diese werden in Tabelle 3.2 geschildert. Die Clickbait-Beispiele stammen aus dem konstruierten Korpus.

Eine andere Art von Clickbaits, die im Verlauf der Korpusanalyse identifiziert wurden, sind Sätze, die zwar in sich abgeschlossen sind und keine Vorwärtsreferenzen beinhalten, bei denen allerdings durch das absichtliche Weglassen relevanter Informationen eine irreführende Sensationsmeldung entsteht - nur mehr eine weitere Leerstelle (*curiosity gap*), die die Sehnsucht nach vollständi-

*Clickbait*: „Netflix will pay you to watch Netflix all day“

*Spoiler*: „If you live in the UK or Ireland“

**Abbildung 3.3:** Clickbait als eine irreführende Sensationsmeldung.

*Clickbait*: „If you buy a motorcycle, you will die“

*Spoiler*: „Even if you don’t“

**Abbildung 3.4:** Beispiel eines Clickbaits, der eine unbegründete Behauptung aufstellt.

ger Information beständig steigert. Die Clickbait-Schlagzeile in Abbildung 3.3 verdeutlicht dieses Phänomen. Im gezeigten Beispiel fehlt eindeutig eine für das Verständnis sehr relevante Information. Ohne diese Information, die der Leser erst durch den Spoiler erfährt, ist die Schlagzeile irritierend und führt zu überdimensionierten Erwartungen im Bezug auf das verlinkte Dokument, die wiederum in einem Klick resultieren.

Bei der Analyse des Korpus wurden zusätzlich Clickbaits identifiziert, die aus der Sicht dieser Arbeit nicht als Clickbaits zu betrachten sind. Die Clickbait-Schlagzeile in der Abbildung 3.4 kann als ein Witz interpretiert werden. Aus der Sicht dieser Arbeit entsteht keine Erwartung, dass im verlinkten Dokument eine signifikante Abhängigkeit zwischen dem Kauf eines Motorrads und dem Eintreten des Todes offenbart wird. Der Spoiler, der in diesem Fall keine spoilende Funktion erfüllt, da er nichts verrät, ist somit eine Tatsache, welche die Absurdität des formulierten Clickbaits unterstreicht.

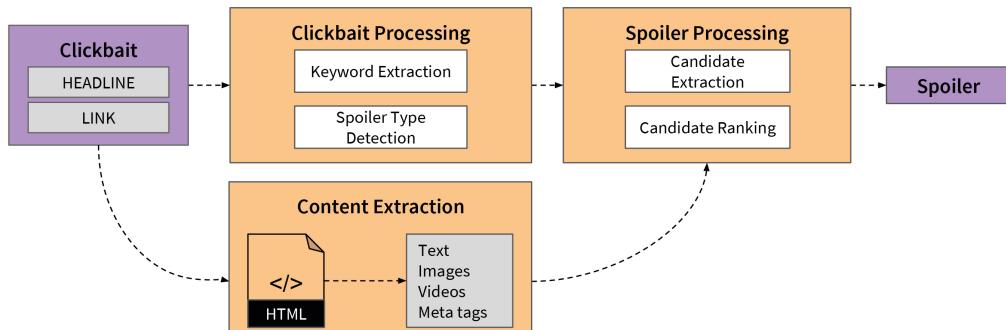
Da die Spoiler für diese Arbeit ebenfalls eine wichtige Rolle spielen, wird bei der Korpusanalyse ebenfalls darauf geachtet, dass der Spoiler tatsächlich eine spoilende Funktion erfüllt. In diesem Zusammenhang treten Antworten auf, die nicht als Spoiler verwertet werden können. Dazu gehören: nicht spoilende Repliken, wie beispielsweise Antworten, die nur aus Emoticons bestehen; Antworten, die nicht in englischer Sprache formuliert sind und Antworten, die nicht ausgeschrieben sind und als Teil der Vulgärsprache betrachtet werden. Die manuelle Annotation des Korpus erfolgt mit dem Korpus-Viewer (Komlossy, 2017). Tabelle 3.3 bildet die oben beschrieben Fälle der Clickbaits und Spoiler mit der jeweiligen Anzahl im Korpus ab. Die Clickbait-Spoiler-Pärchen, die unter die Kategorie „Sonstige“ fallen, werden im weiteren Verlauf dieser Arbeit nicht betrachtet. Somit verbleiben 5787 Clickbait-Spoiler-Pärchen mit den jeweiligen Metainformationen.

Clickbait-Typ	Anzahl
Vorwärtsverweis als Aussagesatz	4190
Vorwärtsverweis als Fragesatz	934
Abgeschlossener Satz	663
Sonstige	1442
$\sum$	7229

**Tabelle 3.3:** Annotierter Korpus. Zur Kategorie „Sonstige“ zählen Fälle mit nicht spoilenden Repliken, Spoiler in nicht englischer Sprache, Spoiler mit nicht ausgeschriebenen Wörtern sowie Schlagzeilen, die nach Auffassung dieser Arbeit keine Clickbaits darstellen.

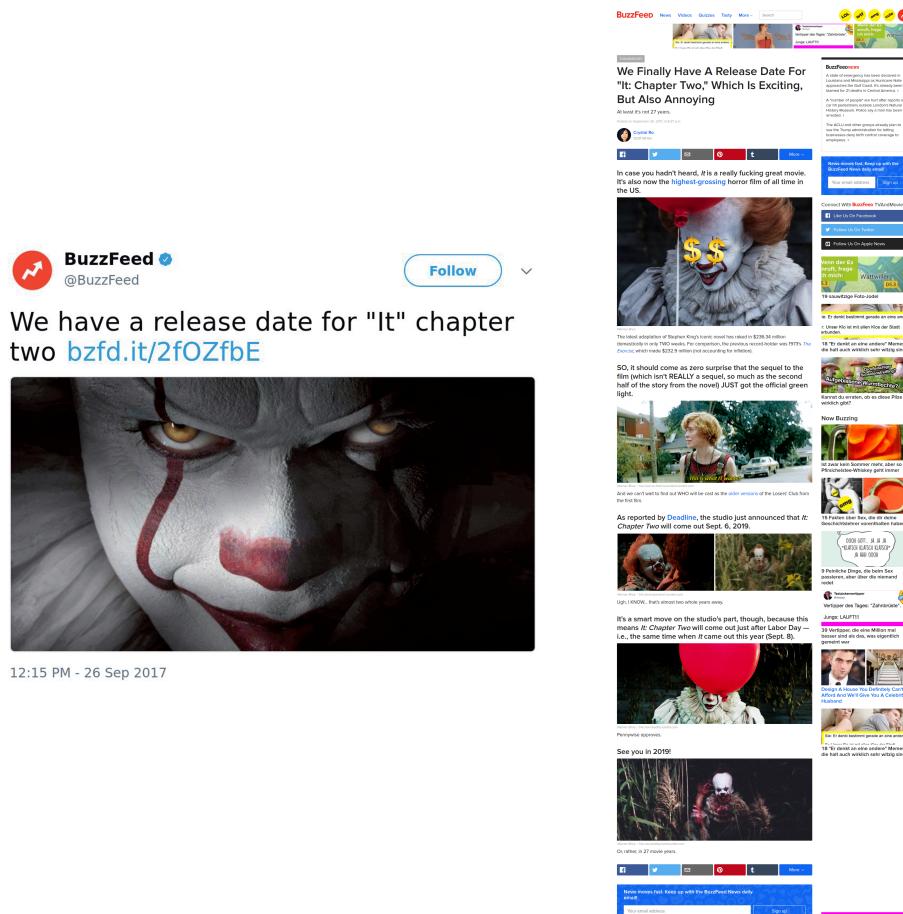
# Kapitel 4

## Faktenbasiertes Clickbait-Spoiling



**Abbildung 4.1:** Clickbait-Spoiling-Pipeline aufgeteilt in drei Bereiche: *content extraction*, *clickbait processing* und *spoiler processing*.

In diesem Kapitel wird, resultierend aus der für die Arbeit motivierten Forschungsfrage nach der inhaltlichen Verbesserung von Informationen und aufbauend auf der Analyse des Korpus im Abschnitt 3.2, das in dieser Arbeit entwickelte Verfahren zum automatisierten Auflösen von Clickbaits vorgestellt. Abbildung 4.1 zeigt die *Pipeline* des entwickelten Verfahrens. Im ersten Abschnitt dieses Kapitels wird der Prozess der Inhaltsextraktion (*Main Content Extraction*) beschrieben. Die Extraktion des Textes aus dem verlinkten Dokument bildet die Grundlage, um Spoilerkandidaten extrahieren zu können. Ein weiterer wichtiger Bestandteil der *Pipeline*, der im zweiten Abschnitt behandelt wird, ist die Clickbait-Verarbeitung (*Clickbait Processing*). Bei diesem Schritt geht es um Extraktion relevanter Informationen aus der Clickbait-Schlagzeile, um die Anzahl der möglichen Spoilerkandidaten einzuschränken. Im



**Abbildung 4.2:** Ein Tweet<sup>1</sup>(links) und das dazugehörige, verlinkte Dokument<sup>2</sup>(rechts).

dritten Abschnitt wird anschließend der Schritt der Spoilerverarbeitung (*Spoiler Processing*) beschrieben, bei dem es einerseits um die Extraktion der Spoilerkandidaten aus dem verlinkten Dokument und andererseits um das Ranking der Kandidaten geht. In diesem Zusammenhang werden vier Ranking-Ansätze vorgestellt, die im Rahmen dieser Arbeit entstanden sind.

<sup>1</sup> <https://twitter.com/BuzzFeed/status/912757496019914754>. Zuletzt besucht: 23. Oktober 2017

<sup>2</sup> [https://www.buzzfeed.com/crystalro/we-finally-have-a-release-date-for-it-chapter-two-which-is?bftw&utm\\_term=.ha1PP9K40#.bfP33Jlr6](https://www.buzzfeed.com/crystalro/we-finally-have-a-release-date-for-it-chapter-two-which-is?bftw&utm_term=.ha1PP9K40#.bfP33Jlr6). Zuletzt besucht: 23. Oktober 2017

## 4.1 Inhalt- und Metadaten-Extraktion des verlinkten Dokuments

Wie bereits erwähnt, ist jedem Clickbait eine URL beigefügt, die den Nutzer auf die zum Clickbait verlinkte Webseite weiterleitet, wie in Abbildung 4.2 veranschaulicht. Um eine Liste mit möglichen Spoilerkandidaten aus dem verlinkten Dokument gewinnen zu können, muss zuerst der Inhalt des verlinkten HTML-Dokuments extrahiert werden. Dieser Prozess der Informationsgewinnung wird als *Information Extraction* bezeichnet und wird sowohl in wissenschaftlichen Arbeiten behandelt, als auch in einer Vielzahl von Anwendungen eingesetzt. Diese Arbeit verwendet für diese Aufgabe die Python-Bibliothek *newspaper* (Ou-Yang, 2016). Ein wesentlicher Vorteil dieser Bibliothek im Vergleich zu Alternativen, wie beispielsweise *Boilerpipe* (Kohlschütter et al., 2010), ist, dass neben dem relevanten Text auch URLs zu Multimedia-Dateien, wie zu Bildern oder Videos und zusätzliche Metainformationen, wie der Autor des Artikels oder die Artikel-Beschreibung, aus dem HTML-Dokument extrahiert werden. Auf diese Weise wird der konstruierte Korpus von 5787 Clickbaits um den zugehörigen Artikeltitel und Artikeltext erweitert und um weitere Informationen ergänzt. In Abbildung 4.3 ist das Beispiel 3.1 um die ihm zugehörigen Informationen ergänzt. Da sich diese Arbeit auf die Spoiler-Extraktion aus dem Text begrenzt, wurden die URLs zu den Bildern aus dem verlinkten Dokument der Vollständigkeit halber extrahiert und können in zukünftigen Arbeiten verwendet werden.

## 4.2 Clickbait-Verarbeitung

Das Ziel der Clickbait-Verarbeitung besteht in der Extraktion relevanter Information aus der Clickbait-Schlagzeile, um die Suche nach einem passenden Spoiler in relevanten Dokumenten zu vereinfachen. Hierzu werden folgende Informationen aus der Schlagzeile bzw. dem verlinkten Dokument extrahiert:

- Spoilertyp
- Schlüsselwörter
- Stimmung
- Koreferenzen im verlinkten Dokument.

Diese Informationen sind einerseits notwendig, um eine Liste mit passenden Kandidaten zu erstellen und andererseits, um die Kandidaten im Anschluss zu gewichten.

```
{
    "post_id_str" : "567763744483467265",
    "cb_headline" : "First California Republican wades
                    into 2016 Senate race",
    "cb_spoiler" : "CA Assemblyman Rocky Chávez",
    "spoiler_publisher" : "HuffPoSpoilers",
    "social_media_platform" : "Twitter",
    "target_url" : "http://huff.to/1yS0rZo",
    "article_title" : "First California Republican wades
                      into 2016 Senate race",
    "article_text" : "California Assemblyman Rocky
                     Chávez announced Tuesday...",
    "article_imgs" : ["http://i.huffpost.com/...-facebook.jpg"],
    "article_description" : "California Assemblyman Rocky Chávez"
}
```

**Abbildung 4.3:** Ein Auszug aus dem Korpus: ein Tweet mit den jeweiligen Attributen und den Informationen aus dem verlinkten Dokument, die mit *newspaper* extrahiert wurden: *article\_title*, *article\_text*, *article\_imgs* und *article\_description*.

- (1) Faktenbasierte Clickbaits (Katapher als Vorwärtsreferenz)
- (2) Komplexe, narrative Clickbaits (Diskursdeixis als Vorwärtsreferenz).

**Abbildung 4.4:** Clickbait-Arten

#### 4.2.1 Erkennung des Spoilertyps

Das Ziel der Spoilertyp-Erkennung (*spoiler type detection*) besteht darin, für jeden Clickbait den Typen des erwarteten Spoilers zu definieren. Analog zum Bereich *Question Answering* (Jurafsky and Martin, 2017) und wie bereits im Abschnitt 3.2 gezeigt, können Clickbaits in zwei Arten unterteilt werden: Faktenbasierte Clickbaits wie zum Beispiel „*A cup of coffee will cost you 8\$ in this city*“ verlangen nach einem Ort als Spoiler, während Clickbaits wie „*How many millions did The Interview make online this weekend*“ nach einer Zahl bzw. einem Preis verlangen. Komplexe, narrative Clickbaits wie „*Here's why summer in New York City smells so awful*“ fordern hingegen keine konkrete Entität sondern eine Textpassage bzw. eine Argumentation. Ist der erwartete Spoilertyp bekannt, so beschränkt sich die Suche nach dem Spoiler auf die Sätze im verlinkten Dokument, die diesen Typen auch enthalten, während

Große Klasse	Feinere Unterteilung
LOCATION	country, city, state
HUMAN	individual, group, title
ENTITY	currency, food, animal
ABBREVIATION	abbreviation, expression
DESCRIPTION	definition, reason
NUMERIC	date, distance, size, money, percent

**Tabelle 4.1:** Untermenge der Taxonomie von Li und Roth (Li and Roth, 2006).

Spoilertyp	Anzahl
PERSON	429
LOCATION	219
ORGANIZATION	50
DATE	47
Sonstige	5042
$\Sigma$	5787

**Tabelle 4.2:** Manuelle Annotation des Korpus nach Spoilertyp.

die anderen Sätze ignoriert werden können. Die Klassifizierung der Clickbaits nach dem erwarteten Spoilertyp bedarf einer detaillierten Ausarbeitung einer Taxonomie, so wie man sie aus dem *Question-Answering*-Bereich kennt. Eine der bekanntesten Taxonomien in diesem Feld ist von Li und Roth (Li and Roth, 2002). Die Taxonomie von Li und Roth ist in zwei Schichten eingeteilt. Die erste Schicht umfasst die sechs allgemeinen Klassen der *Antworttypen*. Jede der Klassen wird in weitere, feiner definierte Unterklassen aufgeteilt. Eine Untermenge der Taxonomie ist beispielhaft in Tabelle 4.1 dargestellt. Zur Erkennung des Antworttyps werden mit Methoden des Überwachten Lernens *Classifier* entwickelt, die auf der Basis einer großen Menge an Fragen, die manuell mit dem entsprechenden *Antworttyp* markiert sind, trainiert werden. Die selben *Classifier* werden im späteren Verlauf für die Erkennung des gesuchten *Antworttyps* im Dokument, in dem sich die Antwort befinden soll, verwendet. Die Entwicklung solch einer Taxonomie sowie der benötigten *Classifier* würde den Rahmen dieser Arbeit übersteigen. Aus diesem Grund beschränkt sich diese Arbeit auf die Clickbaits, die Eigennamen als Spoiler erwarten, da diese Sequenzen im verlinkten Dokument mit Hilfe der Eigennamen-Erkennung (*Named Entity Recognition*) erkannt werden können. Hierzu verwendet diese Arbeit im weiteren Verlauf das *StanfordCoreNLP-Framework* (Manning et al., 2014), das im Abschnitt 4.3.2 näher erläutert wird.

In diesem Zusammenhang wurde der Datensatz erneut hinsichtlich folgen-

der Entitäten manuell annotiert: Person, Location, Organization, Date. Die Anzahl der Clickbaits, die den jeweiligen Typ als Spoiler erfordern, ist in Tabelle 4.2 dargestellt. Diese Arbeit behandelt im Folgenden die Clickbaits, die entweder nach einer *Person* oder einem *Ort* als Spoiler verlangen. Von den 648 Clickbaits werden für den weiteren Verlauf ebenfalls jene ausgelassen, die nach mehr als einem Spoiler verlangen (89 Fälle). Es verbleiben somit 559 Clickbaits.

#### 4.2.2 Extraktion der Schlüsselwörter

Bei der Extraktion von Schlüsselwörtern aus der Clickbait-Schlagzeile besteht die Aufgabe in der Identifikation von Begriffen, die die Schlagzeile am prägnantesten beschreiben. Die Schlüsselwörter können ein wichtiger Indikator dafür sein, ob es sich bei einem Spoilerkandidaten um einen richtigen oder falschen Kandidaten handelt. In der Schlagzeile „*First California Republican wades into 2016 Senate race*“ wird die tatsächliche Person durch den Ausdruck *First California Republican* ersetzt. Wird der Ausdruck durch den tatsächlichen Namen der umschriebenen Person ersetzt, entsteht ebenfalls eine valide Satzkonstruktion: „*Rocky Chávez wades into 2016 Senate race*“. Der korrekte Spoiler könnte also im verlinkten Dokument in einem Satz auftauchen, in dem ebenfalls die Wörter „*wade, 2016, senate, race*“ benutzt werden.

In dieser Arbeit wird der *keyword selection*-Algorithmus nach „*Lasso: A Tool for Surfing the Answer Net*“ (Moldovan et al., 1999) implementiert. Die Extraktion der Schlüsselwörter basiert auf einer Abfolge von Heuristiken, die jeweils eine Liste mit Schlüsselwörtern zurückliefert. Folgende Heuristiken werden implementiert:

- Heuristik 1: Existiert ein Ausdruck in Anführungszeichen, so werden alle Wörter, Stoppwörter ausgenommen, in die Liste der Schlüsselwörter eingefügt
- Heuristik 2: Alle erkannten Eigennamen aus der Schlagzeile
- Heuristik 3: Alle zusammengesetzte Substantive mit den Adjektiven, die unmittelbar davor stehen
- Heuristik 4: Alle restlichen zusammengesetzte Substantive
- Heuristik 5: Alle Nomen mit den Adjektiven, die unmittelbar davor stehen
- Heuristik 6: Alle restlichen Nomen
- Heuristik 7: Alle Verben.

- (1) „@Huffingtonpost: First California Republican wades into 2016 Senate race“
- (2) „First California Republican wades into 2016 Senate race“
- (3) [’NNP’, ’NNP’, ’NNP’, ’VBZ’, ’IN’, ’CD’, ’NNP’, ’NN’]

**Abbildung 4.5:** Clickbait-Beispiel aus den Rohdaten (1), bereinigter Clickbait (2) und die entsprechenden Wortarten (3).

Vor der Extraktion der Schlüsselwörter muss jede Schlagzeile bereinigt werden, da sie meist plattformspezifische Elemente enthält, die in diesem Zusammenhang irrelevant sind. Dazu werden alle Wörter entfernt, die entweder mit einer Raute (#) oder einem At-Zeichen (@) anfangen. Da die Facebook-Schlagzeilen zumeist eine URL enthalten, muss auch diese entfernt werden. Für die Identifikation von Stopwörtern verwendet diese Arbeit das *stopword*-Modul der Python-Bibliothek *nltk*. Als Stopwörter werden in der Informationsgewinnung jene Wörter bezeichnet, die bei der Analyse nicht betrachtet werden, da diese aufgrund des häufigen Auftretens keine inhaltliche Relevanz für das Dokument beitragen. Einige Beispiele solcher Wörter sind unter anderem Artikel (*a, the*), Adverbien (*here, there*) oder auch Personalpronomen (*I, you*). Aus der ursprünglichen Schlagzeile (1) wird für das weitere Vorgehen das „@HuffingtonPost.“ Element entfernt, da es lediglich ein twitterspezifisches Element ist und nicht zum Inhalt der Schlagzeile beiträgt.

Da es sich bei den Schlüsselwörtern um die Extraktion bestimmter Wortarten handelt, muss jedes Wort in der Schlagzeile mit der zugehörigen Wortart (*part of speech*) annotiert werden. Dazu verwendet diese Arbeit den *pos-annotator* aus dem *StanfordCoreNLP-Framework* (Manning et al., 2014). Auf diese Weise entsteht aus der bereinigten Schlagzeile (2) eine Liste mit entsprechenden Wortarten (3). Die Abkürzungen in (3) stehen für bestimmte Wortarten. Die Wortarten und ihre jeweiligen Abkürzungen, die bei der Extraktion der Schlüsselwörter für die Heuristiken 2-7 von Bedeutung sind, werden in Tabelle 4.3 dargestellt. Da bei der Heuristik 1 alle Wörter, die nicht in der Liste der Stopwörter auftauchen, in die Liste der Schlüsselwörter aufgenommen werden, kann die Wortart in diesem Zusammenhang ignoriert werden. Tabelle 4.4 verdeutlicht die Extraktion der Schlüsselwörter am Beispiel einer konkreten Schlagzeile (2).

<sup>3</sup> <https://cs.nyu.edu/grishman/jet/guide/PennPOS.html>. Zuletzt besucht: 23. Oktober 2017

Abkürzung	Wortart
JJ	Adjektiv, Normalform
JJR	Adjektiv, Komparativ
JJS	Adjektiv, Superlativ
NN	Nomen, Singular
NNS	Nomen, Plural
NNP	Eigenname, Singular
NNPS	Eigenname, Plural
RB	Adverb
RBR	Adverb, Komparativ
RBS	Adverb, Superlativ
VB	Verb, Infinitiv
VBD	Verb, Präteritum
VBG	Verb, Gerundium oder Partizip Präsens
VBN	Verb, Partizip Präteritum
VBP	Verb, Nicht dritte Person Singular, Präsens
VBZ	Verb, Dritte Person Singular, Präsens

**Tabelle 4.3:** Abkürzungen für Wortarten.<sup>3</sup>

Schlagzeile	<i>First California Republican wades into 2016 Senate race</i>
Schlüsselwörter	First California Republican
	First California Republican Senate
	First California Republican Senate race
	First California Republican Senate race wades

**Tabelle 4.4:** Extraktion der Schlüsselwörter am Beispiel eines Clickbaits.

### 4.2.3 Stimmungsanalyse

Bei der Annotation des Korpus wird festgestellt, dass es oft zur Verwendung von polarisierenden Adjektiven in den Clickbait-Schlagzeilen kommt. Die Frage, die sich daraus ergibt, ist, ob der Spoiler ebenfalls in einem polarisierenden Satz vorkommt. Die erkannte Eigenschaft der Schlagzeilen wird mit Hilfe der Stimmungserkennung messbar erfasst. Dazu verwendet diese Arbeit den *sentiment-annotator* des *StanfordCoreNLP-Frameworks* (Manning et al., 2014). Das verwendete *Framework* klassifiziert die Sätze hinsichtlich der Stimmung auf einer 4-stufigen Skala: von sehr negativ bis sehr positiv. Tabelle 4.5 zeigt die Ergebnisse der Stimmungsanalyse. Die Ergebnisse untermauern die Annahme, dass die meisten (348 von 559) Clickbaits tatsächlich stimmungsweisend sind. So scheint es der Fall zu sein, dass negative Clickbaits öfter Verwendung finden als die positiven.

Stimmung	Sehr negativ	Negativ	Neutral	Positiv	Sehr positiv
Stimmungswert	0	1	2	3	4
Anzahl	2	211	248	97	1

**Tabelle 4.5:** Clickbait-Annotation hinsichtlich der Stimmung.

- (1) „Er ist ein begabter amerikanischer Schauspieler. Er hat unzählige Preise und Auszeichnungen bekommen. Die Rede ist von *Johnny Depp*.“<sup>4</sup>

**Abbildung 4.6:** Beispielsatz für den Gebrauch einer Katapher.

Die Erkenntnisse werden beim Ranking der Spoilerkandidaten verwendet.

#### 4.2.4 Auflösung der Koreferenzen

Der Gebrauch von Anaphern und Kataphern trägt im Allgemeinen zur Kohäsion eines Textes bei. Das Verknüpfen solcher Mittel mit der gemeinten Entität innerhalb eines geschriebenen oder gesprochenen Textes stellt für den Menschen keine Schwierigkeiten dar, spielt jedoch eine bedeutende Rolle für das Verständnis von Texten.

Im Beispiel (1) ist eindeutig erkennbar, dass das Pronomen „er“ in beiden Sätzen auf „*Johnny Depp*“ verweist.

Im Bereich der natürlichen Sprachverarbeitung (*Natural Language Processing*) besteht die Aufgabe der Koreferenz-Auflösung (*coreference resolution*) in der Identifikation aller Ausdrücke innerhalb eines Textes, die auf die selbe Entität verweisen (Heeyoung et al., 2013).

Wie bereits im Abschnitt 3.2 beschrieben, werden in Clickbaits des Öfteren solche sprachlichen Mittel verwendet, die auf eine Entität im verlinkten Dokument verweisen. Diese Arbeit verwendet den *dcoref-annotator* des *StanfordCoreNLP-Frameworks* (Manning et al., 2014) für die Auflösung des Referenten im verlinkten Dokument. Hierfür muss vorher die Clickbait-Schlagzeile, wie bereits im Abschnitt 4.2.2, von irrelevanten Elementen, wie Rauten, At-Zeichen oder URLs, bereinigt werden. Im zweiten Schritt wird die bereinigte Schlagzeile an den Anfang des verlinkten Dokuments gestellt, um eine inhaltliche Einheit zu bilden. Abbildung 4.7 verdeutlicht diesen Schritt. Der *dcoref-annotator* liefert als Ergebnis einen Graphen mit Koreferenzen innerhalb des gesamten Textes mit dem *headword* als Knoten der jeweiligen Entität. Im Beispiel 4.7 ist das *headword* ‘*Republican*’, da es die erste Erwähnung der referenzierten Entität ist. Aus dem Graphen lassen sich die anderen Referenten

*Clickbait* „First California Republican wades into 2016 Senate race.“

*Text* „California Assemblyman Rocky Chávez announced Tuesday that he is exploring a bid for the Golden State’s open U.S. Senate seat in 2016. He is becoming the first sitting Republican...“

*Clickbait+Text* „First California Republican wades into 2016 Senate race. California Assemblyman Rocky Chávez announced Tuesday that he is exploring a bid for the Golden State’s open U.S. Senate seat in 2016. He is becoming the first sitting Republican...“

**Abbildung 4.7:** Zusammenführung der Schlagzeile mit dem Text des verlinkten Dokuments.

```
{
    "spoiler_type"      : "PERSON"
    "keywords"          : ["First California Republican",
                          "Senate", "race", "wades"]
    "sentiment"         : "Neutral",
    "sentimentValue"    : 2,
    "corefs"            : ["Rocky Chávez", "Chávez"]
}
```

**Abbildung 4.8:** Alle Ergebnisse aus der Clickbait-Verarbeitung werden separat gespeichert.

derselben Entität im Text ablesen. Für das Beispiel 4.7 existieren mehrere Erwähnungen im Text, die als Referenten derselben Entität erkannt werden: *Rocky Chávez* und *Chávez*. Für jede Schlagzeile wird eine Liste mit Referenten extrahiert, die auf die selbe Entität verweisen, wie die Erwähnungen in der Clickbait-Schlagzeile. In 293 von 559 Fällen liefert das Framework eine Liste mit Referenten, in den anderen 266 Fällen kommt eine leere Liste zurück.

Die Ergebnisse aus der Clickbait-Verarbeitung, Zuweisung des Spoilertyps, Extraktion der Schlüsselwörter, Stimmungsanalyse sowie die Auflösung der Koreferenzen werden separat als zusätzliche Information zum Korpus gespeichert. Abbildung 4.8 zeigt die Organisation der gesammelten Zusatzinformationen pro Clickbait-Fall. Diese extrahierten Informationen über die Clickbait-Schlagzeilen spielen eine wesentliche Rolle einerseits bei der Extraktion der möglichen Kandidaten und andererseits bei der Gewichtung der Kandidaten.

**Clickbait** „This was the unhealthiest U.S. President“

**Spoiler** William Henry Harrison. Dude died a month into office.

**Modifizierte Spoiler** [‘William Henry Harrison’, ‘William Henry’, ‘Henry Harrison’, ‘William Harrison’, ‘William’, ‘Henry’ ‘Harrison’]

**Abbildung 4.9:** Modifikation des originalen Spoiler: manuelle Extraktion des erwarteten Eigennamen. Teilmengen des Eigennamen werden ebenfalls als korrekte Spoiler akzeptiert.

## 4.3 Spoiler-Verarbeitung

Der letzte Schritt in der entwickelten *Pipeline* ist die Spoiler-Verarbeitung. Im ersten Abschnitt werden die originalen Spoiler modifiziert und durch mehrere Spoiler-Varianten ergänzt. Im zweiten Abschnitt dieses Kapitels wird gezeigt, wie die Kandidaten aus dem verlinkten Dokument extrahiert werden. Anschließend werden die Ansätze zum Ranking der extrahierten Kandidaten erläutert, um dem Nutzer einen Spoiler für einen Clickbait zurückzuliefern.

### 4.3.1 Modifikation der Originalspoiler

Da es sich bei den originalen Spoilern um vom Menschen geschriebene Repliken handelt, müssen diese in erster Linie auf ihre Richtigkeit überprüft und angepasst werden, da sie die Grundlage für die Evaluierung der extrahierten Kandidaten bilden.

Bei der Überprüfung auf das Vorkommen des originalen Spoilers im verlinkten Dokument, stellt sich heraus, dass in nur 129 von 512 Fällen der vom Menschen geschriebene Spoiler in exakt dieser Form im Dokument vorkommt. Da es sich bei den 512 Fällen um Clickbaits handelt, die nach einem Eigennamen verlangen, werden die originalen Spoiler erneut manuell dahingehend modifiziert, dass sie nur den relevanten Eigennamen beinhalten. In Abbildung 4.9 erkennt man, dass der ursprüngliche Spoiler auf den eigentlichen Eigennamen gekürzt wird. Zusätzlich wird auch jede Teilmenge des Eigennamens als ein korrekter Spoiler gewertet.

*Clickbait* „First California [Republican](#) wades into 2016 Senate race.“

*Textausschnitt* „California Assemblyman [Rocky Chávez](#) announced Tuesday that he is exploring a bid for the Golden State’s open U.S. Senate seat in 2016.“ He is becoming the first sitting Republican lawmaker to take a formal step toward running for retiring Democratic Sen. Barbara Boxer’s seat.

[Chávez](#), who represents part of San Diego County, has opened an exploratory committee, allowing him to fundraise for the statewide race.

**Abbildung 4.10:** Ein Clickbait und ein Textausschnitt aus dem verlinkten Dokument. Das Wort *Republican* ist eine Katapher auf den Eigennamen *Rocky Chávez* und *Chávez* im verlinkten Dokument.

### 4.3.2 Extraktion der Spoilerkandidaten

Um die Spoilerkandidaten aus dem verlinkten Dokument zu extrahieren, muss im Vorfeld definiert werden, welche sprachliche Einheit in diesem Zusammenhang als Spoiler bezeichnet wird. Der Spoiler ist als ein Textabschnitt aus dem verlinkten Dokument zu verstehen, in dem genau der Teil der Information enthalten ist, der den fehlenden Teil in der Clickbait-Schlagzeile auflöst. Wie bereits erwähnt, beschränkt sich diese Arbeit auf die Clickbait-Schlagzeilen, die einen Eigennamen, *Person* oder *Ort*, im verlinkten Dokument referenzieren.

In Abbildung 4.10 sieht man ein Clickbait-Beispiel mit einem Textausschnitt aus dem verlinkten Dokument. In der Clickbait-Schlagzeile fungiert das Wort *Republican* als eine Katapher und verweist auf den Referenten *Rocky Chávez*. Somit ist der Eigename *Rocky Chávez* ein möglicher Spoiler für diesen Clickbait und stimmt mit einem korrekten Spoiler aus der Liste der modifizierten Spoiler überein.

Eine andere Möglichkeit zum Auflösen eines Clickbaits besteht in der Extraktion eines semantisch passenden Satzes oder eines Absatzes, im Allgemeinen eines Textabschnitts, in dem der Eigennname enthalten ist. Somit wäre der erste Satz oder auch der ganze erste Absatz des Textabschnitts ein valider Spoiler auf diesen Clickbait. Hingegen ist der erste Satz des darauf folgenden Absatzes „[Chávez](#), who represents...“ kein inhaltlich passender Satz zu der Schlagzeile, auch wenn der korrekte Eigenname in dem Satz enthalten ist.

Diese Arbeit widmet sich dem ersten Ansatz – der Extraktion des konkreten Eigennamens, der in der Clickbait-Schlagzeile referenziert wird. Ein Spoilerkandidat ist dementsprechend jede Entität, die dem gesuchten *Spoilertypen*

**Satz** „California<sub>Location</sub> Assemblyman<sub>MISC</sub> Rocky<sub>Person</sub> Chávez<sub>Person</sub> announced Tuesday<sub>Date</sub> that he is exploring a bid for the Golden State’s open U.S.Senate<sub>Location</sub> seat in 2016<sub>Date</sub>.“

**Abbildung 4.11:** Ein Beispielsatz aus dem verlinkten Dokument und die zugewiesenen *Named Entity Tags*. MISC entspricht einer Kategorie, die zwar als *Named Entity* erkannt, aber nicht in die Kategorie der *Person*, *Location* oder *Organization* klassifiziert wird. Jedes Wort, das keiner Kategorie zugewiesen werden kann, wird mit einem ’O’-Tag markiert.

entspricht, der im Abschnitt 4.2.1 zugewiesen wurde. Um für jeden Clickbait die jeweiligen Kandidaten aus dem Text extrahieren zu können, muss vorher die Eigennamen-Erkennung (*Named Entity Recognition*) auf dem jeweiligen Dokument durchgeführt werden. *Named Entity Recognition*, *NER* ist eine Unteraufgabe aus dem Bereich der Informationsgewinnung (*Information Extraction*). Das Ziel von *NER*-Systemen besteht in der Identifikation und Klassifizierung von Entitäten innerhalb eines Textes in vordefinierte Klassen. Dafür wird in dieser Arbeit der *ner-annotator* des *StanfordCoreNLP-Frameworks* (Manning et al., 2014) verwendet. Das Framework implementiert unterschiedliche Ansätze für die Erkennung dieser Klassen. Für die Erkennung von *Named Entities* (Person, Location, Organization) setzt das Framework eine Kombination aus unterschiedlichen Modellen basierend auf *Conditional Random Field* (*CRF*) zur Segmentierung von Eingabesequenzen ein, die auf unterschiedliche Korpora trainiert sind (Finkel et al., 2005). Für die Erkennung von numerischen Entitäten (Money, Number) verwendet das Framework zwei regelbasierte Ansätze: ein System für die Erkennung von Geldangaben und eins für Zahlen. Für die Erkennung von Zeitangaben (Date, Time, Duration, Set) verwendet das Framework ein separates *state-of-art* System (Chang and Manning, 2012). In dem Beispiel in Abbildung 4.11 wird ersichtlich, dass jedes Wort im verlinkten Textdokument mit einem entsprechenden *Named Entity Tag* markiert wird. Entspricht der *Tag* der erkannten und markierten Sequenz im Text dem Spoilertypen, der dem Clickbait zugewiesen wird, so wird diese Sequenz als Spoilerkandidat abgespeichert.

In Abbildung 4.12 sieht man die Sequenzen, die als Kandidaten abgespeichert werden. Für jedes Dokument aus dem Korpus werden die Kandidaten extrahiert und in einem Wörterbuch als *Keys* abgespeichert. Der Wert des jeweiligen Kandidaten ist eine Liste mit Tupel. Die erste Stelle im Tupel ist der Index des Satzes im Dokument, in dem der Kandidat vorkommt, während die zweite Stelle den Startindex der Position des Kandidaten innerhalb des Satzes abbildet. Die Länge der Liste entspricht dabei der Häufigkeit des

*Clickbait* „First California Republican wades into 2016 Senate race.“

*Spoilertyp* Person

*Satz* „California<sub>Location</sub> Assemblyman<sub>MISC</sub> [Rocky<sub>Person</sub> Chávez<sub>Person</sub>] announced Tuesday<sub>Date</sub> that he is exploring a bid for the Golden State’s open U.S.Senate<sub>Location</sub> seat in 2016<sub>Date</sub>. He is becoming the first<sub>Ordinal</sub> sitting Republican<sub>MISC</sub> lawmaker to take a formal step toward running for retiring Democratic<sub>MISC</sub> Sen. [Barbara<sub>Person</sub> Boxer<sub>Person</sub>]’s seat.

[Chávez<sub>Person</sub>], who represents part of San<sub>Location</sub> Diego<sub>Location</sub> County<sub>Location</sub>, has opened an exploratory committee, allowing him to fundraise for the statewide race.“

*Kandidaten* {’Barbara Boxer’:[(1,19)], ’Chávez’:[(2, 0)], ’Rocky Chávez’:[(0, 2)]}

**Abbildung 4.12:** Extrahierte Spoilerkandidaten: Kandidat als *key*, mit einer Liste von Tupel als *value*. Die erste Stelle im Tupel repräsentiert den Satzindex im Dokument, während die zweite Stelle den Wortindex innerhalb des Satzes abbildet. Die Länge der Liste bildet die Häufigkeit des Kandidaten im Dokument ab.

Kandidaten im gesamten Dokument. Aus der Liste mit den Tupel ist ebenfalls die Reihenfolge ablesbar, in der die Kandidaten im Dokument vorkommen. All diese Informationen werden für das Ranking der Kandidaten im nächsten Kapitel verwendet. Abbildung 4.13 zeigt die Kandidatenverteilung über alle Dokumente. In der Grafik wird ersichtlich, dass aus zehn Dokumenten genau ein Kandidat extrahiert wurde. In all diesen Fällen handelt es sich um die korrekten Kandidaten für den jeweiligen Clickbait. Aus diesem Grund werden diese Fälle bei der Evaluation der Ranking-Ansätze nicht weiter betrachtet. Die weitere Betrachtung ergibt, dass unter den extrahierten Spoilerkandidaten von 37 Dokumenten kein einziger korrekter Kandidat, entsprechend des im jeweiligen Dokument aufgelisteten Goldstandards, vorhanden ist. Auch diese Fälle werden ausgelassen, da sie ansonsten die Ergebnisse der Evaluation der Ranking-Ansätze verfälschen und somit die Genauigkeit des Gesamtergebnisses herabmindern würden. Durch diese Reduktion verbleiben noch 512 Clickbaits. Darunter werden aus 58 Dokumenten mehr richtige als falsche Kandidaten, aus 57 Dokumenten genau so viele falsche wie richtige und aus 397 Dokumenten mehr falsche als richtige Kandidaten extrahiert.

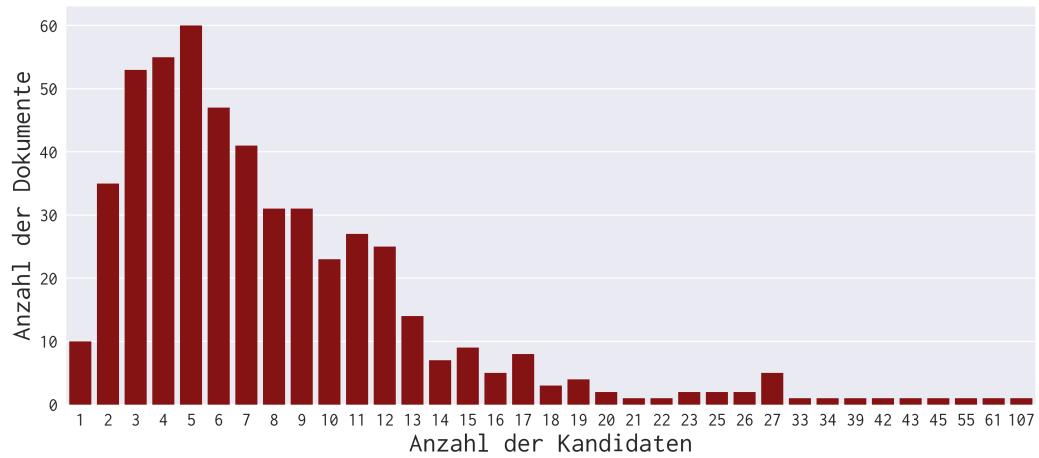


Abbildung 4.13: Verteilung der Kandidaten über 512 Dokumente.

### 4.3.3 Ranking der Spoiler-Kandidaten

Für das Ranking der Kandidaten werden in dieser Arbeit vier Ansätze implementiert und anschließend evaluiert. Im Folgenden werden die Ansätze genauer erläutert.

**Position des Kandidaten im Dokument** Bei diesem naiven Ansatz werden die Kandidaten in Abhängigkeit von ihrer Position im Dokument sortiert. Der erstgenannte Kandidat im Dokument wird somit als erster Spoiler vom System vorgeschlagen.

**Häufigkeit des Kandidaten im Dokument** Bei diesem Ansatz werden die Kandidaten in Abhängigkeit von ihrer Häufigkeit im Dokument sortiert. Der erste Kandidat, der vom System vorgeschlagen wird, ist somit der häufigste im gesamten Dokument. Kommen zwei oder mehrere Kandidaten gleich häufig im Dokument vor, dann werden diese wiederum, analog zum ersten Ansatz, in Abhängigkeit von ihrer Position im Dokument sortiert.

**Regelbasierter Ansatz** Der regelbasierte Ansatz selektiert einen bestimmten Kandidaten in Abhängigkeit von mehreren Regeln, die nacheinander angewendet werden und schlägt ihn als Spoiler vor. Da in diesem Ansatz als erster Schritt die Kosinus-Ähnlichkeit verwendet wird, wird diese im Vorfeld näher erläutert.

Die Kosinus-Ähnlichkeit ist ein Maß, das in *Information Retrieval* zum Ver-

gleich zweier Dokumente  $d_1, d_2$ , die im Vektorraum abgebildet sind, eingesetzt wird.

$$sim(d_1, d_2) := \frac{\langle v(d_1), v(d_2) \rangle}{|v(d_1)| \cdot |v(d_2)|} = 1 - \cos \phi$$

Dabei steht  $\phi$  für den Winkel zwischen den zwei Vektoren. Ist  $sim(d_1, d_2) = 1$ , so bedeutet dies, dass die zwei Dokumente eine identische Wortverteilung haben. Ist  $sim(d_1, d_2) = 0$ , haben die zwei Dokumente kein einziges Wort gemeinsam.

Zur Überführung des Dokuments in eine Vektor-Darstellung wird das Tf-idf-Maß (*Termfrequenz inverse Dokumentfrequenz*) eingesetzt, das in *Information Retrieval* zur Einschätzung der Termrelevanz in Dokumenten einer Dokumentenkollektion verwendet wird. In dieser Arbeit werden die einzelnen Sätze  $s_i$  des verlinkten Textes und die Clickbait-Schlagzeile  $cb$  als eine Dokumentenkollektion

$$C = \{cb, s_1 \dots s_l\}$$

betrachtet, die wiederum aus einzelnen Termen

$$s_1 = \{t_1 \dots t_n\}$$

bestehen. Darauf basierend lässt sich von jedem Term  $t$  seine Häufigkeit im Dokument, die *Termfrequenz*  $tf_s(t)$ , in diesem Fall im Satz bzw. in der Clickbait-Schlagzeile, ermitteln. Dieser Wert gibt an, wie relevant ein Term für das Dokument ist. Um eine Aussage darüber zu treffen, wie relevant ein Term unter allen Sätzen ist, muss die *Dokumentfrequenz* berechnet werden. In diesem Fall beschreibt die *Dokumentfrequenz*  $df_t$  die Anzahl der Sätze, in denen ein Term vorkommt. Daraus lässt sich die *inverse Dokumentfrequenz*

$$idf_t = \log \frac{N}{df_t}$$

berechnen. Kommt ein Term  $t$  in der Gesamtkollektion aller Sätze relativ selten vor, so ist dieser Wert hoch, kommt der Term  $t$  hingegen häufiger vor, so ist der Wert niedrig. Aus den beiden Werten lässt sich nun der *tf-idf*-Wert

$$tf\text{-}idf_t = tf_s(t) \cdot idf_t$$

berechnen. Der *tf-idf*-Wert eines Terms  $t$  ist hoch, wenn dieser in einem Dokument häufig und gleichzeitig in anderen Dokumenten kaum vorkommt. Der Wert ist niedrig, wenn der Term  $t$  entweder nicht im Dokument vorkommt oder nicht zur Unterscheidung zweier Dokumente beiträgt. Auf der Grundlage der *tf-idf*-Vektoren wird, wie oben beschrieben, die Kosinus-Ähnlichkeit berechnet. Für die Überführung des Textes aus der verlinkten Webseite in die

*tf-idf*-Darstellung und die Berechnung der Kosinus-Ähnlichkeit zwischen den Vektoren wird in dieser Arbeit die *Machine-Learning*-Bibliothek *scikit-learn* verwendet (Pedregosa et al., 2011).

Im ersten Schritt wird zwischen allen Sätzen, in denen mindestens ein Kandidat vorkommt und der Clickbait-Schlagzeile die Kosinus-Ähnlichkeit berechnet. Aus allen Sätzen des verlinkten Textes werden  $N$  Sätze mit der höchsten Kosinus-Ähnlichkeit zu der Clickbait-Schlagzeile ausgewählt. Die Anzahl  $N$  der extrahierten Sätze wird im Kapitel 5 ermittelt und erläutert. Aus den ähnlichen Sätzen, wird genau der Kandidat ausgewählt, der in diesen Sätzen am häufigsten vorkommt (lokale Häufigkeit). Sollten mehrere Kandidaten gleich oft vorkommen, wird unter diesen Kandidaten derjenige ausgewählt, der im gesamten Dokument am häufigsten vorkommt (globale Häufigkeit). Kommen durch diese Selektion weiterhin mehrere Kandidaten gleich häufig vor, so präferiert das System denjenigen Kandidaten, der unter den ausgewählten als erstgenannter im Dokument vorkommt.

**Machine-Learning-Ansatz** In jedem der 512 Dokumente existieren mehrere Kandidaten, die sich in zwei Kategorien einteilen lassen: korrekter Kandidat, falscher Kandidat. Man kann in diesem Zusammenhang auch von einer binären Klassifikation sprechen. Um mittels maschinellem Lernen einen Kandidaten in eine der Kategorien zu klassifizieren, muss ein Algorithmus auf Grundlage von Beispielen (vgl. Kapitel 5) trainiert werden. Ein Beispiel ist eine Kollektion von Merkmalen (engl. *Features*), die einen Kandidaten vom Anderen unterscheidet. Überlicherweise wird ein Beispiel in Form eines Vektors  $\mathbf{x} \in \mathbb{R}^n$  repräsentiert, wobei jeder Eintrag  $x_i$  des Vektors ein anderes *Feature* abbildet. Folgende *Features* werden zur Unterscheidung von Kandidaten implementiert:

- *frequency*: Häufigkeit des Kandidaten im Text
- *max-cos*: Maximale Kosinusähnlichkeit zwischen Kandidatensätzen und der Clickbait-Schlagzeile: zwischen allen Sätzen, in denen der jeweilige Kandidat vorkommt, wird die Kosinus-Ähnlichkeit zu der Clickbait-Schlagzeile berechnet. Unter diesem Merkmal wird der maximale Wert eingetragen
- *num-named-entities*: Anzahl der erkannten Eigennamen in allen Kandidatensätzen des jeweiligen Kandidaten
- *num-keywords*: Anzahl der Schlüsselwörter in der Clickbait-Schlagzeile aus allen Kandidatensätzen des jeweiligen Kandidaten
- *first-sent-idx*: Index des ersten Satzes, in dem der Kandidat vorkommt

*Kandidaten* {’Barbara Boxer’:[(1,19)], ’Chávez’:[(2, 0)], ’Rocky Chávez’:[(0, 2)]}

*Kandidat:Satzindex* {’Barbara Boxer’: 1, ’Chávez’ : 2, ’Rocky Chávez’: 0}

**Abbildung 4.14:** Sortierte Kandidaten nach Auftreten im Text unter der Betrachtung der ersten Erwähnung.

- *in-corefs*: Kommt der Kandidat in der Liste mit Koreferenzen für das jeweilige Dokument vor, so ist dieser Wert auf 1 gesetzt, ansonsten auf 0
- *sentiment*: Durchschnitt der Stimmungswerte aller Kandidatensätze des jeweiligen Kandidaten,

$$sentiment = \frac{\sum_{i=1}^N s_i}{N}$$

wobei  $N$  die Anzahl der Kandidatensätze abbildet und  $s_i$  den Stimmungswert des jeweiligen Kandidatensatzes.

Nach der Berechnung der *Features* entsteht eine Matrix, in der jede Zeile einen Kandidaten repräsentiert und jede Spalte ein *Feature*. Jedem *Feature*-Vektor ist ein Label zugeordnet. Repräsentiert der *Feature*-Vektor einen Kandidaten, der richtig ist, so hat der Vektor das Label *True*, anderenfalls *False*.

Anstatt im Folgenden die Kandidaten mit dem trainierten Classifier zu klassifizieren, werden in dieser Arbeit für jeden Kandidaten die *soft labels* mit der *softmax*-Funktion berechnet. Hierzu verwendet diese Arbeit die *predict\_proba*-Funktion der *Machine-Learning*-Bibliothek *sklearn* (Pedregosa et al., 2011). Die Ausgabe des Classifiers ist als eine Wahrscheinlichkeitsverteilung der jeweiligen Klassen zu interpretieren. Somit besteht die Ausgabe für einen Feature-Vektor beispielsweise aus zwei Wahrscheinlichkeitswerten: einer für die Klasse *True* und einer für die Klasse *False*. Die Kandidaten eines Dokuments werden daraufhin auf der Grundlage ihrer Wahrscheinlichkeitswerte, mit denen sie als *True* eingestuft werden, absteigend sortiert.

# Kapitel 5

## Experimente

In diesem Teil der Arbeit werden die im vorherigen Kapitel beschriebenen Ranking-Ansätze evaluiert und miteinander verglichen. Zuvor wird erklärt, wie der Datensatz in Training- und Testdaten aufgeteilt wird und welche Metriken zur Evaluierung verwendet werden.

### 5.1 Versuchsaufbau

Der Teil des Korpus, der in dieser Arbeit verwendet wird, besteht aus 512 Dokumenten mit insgesamt 888 richtigen und 3380 falschen Kandidaten. Für jedes der 512 Dokumente gibt es mindestens einen korrekten Kandidaten, wie bereits im Abschnitt 4.3.2 gezeigt. Für das Maschinelle Lernen wird der Datensatz vorher in einem Verhältnis von 80% Trainings- und 20% Testdaten aufgeteilt. Tabelle 5.1 schildert die genaue Verteilung der Dokumente und der Kandidaten. In dieser Arbeit werden im *Machine-Learning*-Ansatz vier Algorithmen verwendet: Logistic Regression, Decision Tree, Naive Bayes und Random Forest. Um die Ergebnisse der beschriebenen Ansätze untereinander vergleichen zu können, werden die Experimente ausschließlich auf den Testdaten durchgeführt.

Für die Evaluierung der Ergebnisse werden zwei Metriken verwendet. Alle Ansätze werden hinsichtlich ihrer *Accuracy* (Genauigkeit) untereinander verglichen.

$$\text{Accuracy} = \frac{\text{Anzahl korrekter Spoiler}}{\text{Anzahl der Dokumente}}$$

Die Accuracy gibt an, in wie vielen Fällen der höchstgerankte Kandidat bzw. der einzige vorgeschlagene Kandidat des regelbasierten Ansatzes einem tatsächlichen Spoiler des jeweiligen Dokuments entspricht. Die Ansätze, die die extrahierten Kandidaten ranken (alle außer dem regelbasierten Ansatz) werden zusätzlich mit dem *Mean Reciprocal Rank (MRR)* evaluiert. MRR ist eine

	#D	#KK	#FK
Gesamter Korpus	512	888	3380
Trainingsdaten	409	706	2634
Testdaten	103	182	746

**Tabelle 5.1:** Aufteilung des Korpus in Trainings- und Testdaten: Anzahl der Dokumente (#D), Anzahl korrekter Kandidaten (#KK) und Anzahl falscher Kandidaten (#FK).

gängige Metrik aus dem Bereich des *factoid question answering* (Jurafsky and Martin, 2017). Der Wert für jeden Clickbait ist der reziproke Rang, also die Position innerhalb der Liste von Spoilerkandidaten, des ersten korrekten Kandidaten. Befindet sich beispielsweise der erste korrekte Spoiler innerhalb der Liste mit Kandidaten auf dem zweiten Rang, so ist der reziproke Wert des Ranges  $\frac{1}{2}$ . Sollte innerhalb der ersten drei Kandidaten kein Spoiler korrekt sein, wird dem jeweiligen Clickbait der Wert 0 zugewiesen. Aus den Werten der einzelnen Clickbaits lässt sich der *MRR*

$$MRR = \frac{\sum_{i=1}^N \frac{1}{rank_i}}{N}$$

berechnen. Für die Berechnung des *MRR* wird von jedem Ansatz, bis auf den regelbasierten Ansatz, eine Liste mit den drei höchstgerankten Kandidaten zurückgeliefert. Liegt der MRR bei 0,5, so bedeutet dies, dass für jeden Clickbait der erste korrekte Kandidat innerhalb der zurückgelieferten Liste immer auf dem zweiten Rang ist. Liegt der Wert über 0,5, so bedeutet dies, dass mehr Kandidaten auf dem ersten Rang platziert sind, als auf dem dritten und jeweils umgekehrt.

## 5.2 Ergebnisse

Der erste naive Ansatz, die Kandidaten ausschließlich auf der Grundlage ihrer Reihenfolge im Dokument zu ranken, erweist sich als nicht besonders aussagekräftig. In einem Viertel der Fälle entspricht der höchstgerankte Kandidat einem der modifizierten Spoiler aus dem Goldstandard für das jeweilige Dokument. Die Berechnung des Mean Reciprocal Rank liefert einen Wert von 0,45. In 31 Fällen ist unter den ersten drei Kandidaten kein korrekter Kandidat vorhanden. In 26 Fällen ist der erste korrekte Kandidat auf Rang 1, in 31 Fällen auf Rang 2 und in 16 Fällen auf Rang 3.

Deutlich besser fallen die Ergebnisse mit dem Ranking nach der Häufigkeit aus. In 68,93% der Fälle entspricht der höchstgerankte Kandidat einem der

	$N = 1$	$N = 2$	$N = \{3, 4\}$	$N = 5$	$N = \{6, 7, 8\}$	$N > 8$
<b>Accuracy</b>	0,6019	0,6504	0,6893	0,7087	0,6990	0,6893

**Tabelle 5.2:** Ergebnisse des regelbasierten Ansatzes in Abhängigkeit von der Anzahl  $N$  der selektierten Sätze.

möglichen Spoiler für das jeweilige Dokument. Der MRR der ersten drei Vorschläge liegt bei 0,80. In 7 Fällen ist kein richtiger Kandidat unter den ersten drei Vorschlägen. In 71 Fällen liegt der erste richtige Kandidat auf dem Rang 1, in 19 Fällen auf dem Rang 2 und in sechs Fällen auf dem Rang 3.

Die Ergebnisse des regelbasierten Ansatzes variieren in Abhängigkeit von der Anzahl  $N$  der Sätze, die nach der Kosinus-Ähnlichkeit selektiert werden. In Tabelle 5.2 sind die Ergebnisse in Abhängigkeit von  $N$  dargestellt. Durch die Auswahl des ähnlichsten Satzes erreicht der Ansatz eine Genauigkeit von 60,19%. Bei  $N = \{3, 4\}$  liefert der Ansatz die selbe Accuracy von 68,93% wie das häufigkeitsbasierte Ranking. Dieses Ergebnis lässt sich darauf zurückführen, dass unter den selektierten Sätzen kein Kandidat mit einer lokalen Häufigkeit vorkommt und deshalb der Kandidat mit der höchsten globalen Häufigkeit vom Verfahren ausgewählt wird. Auch bei  $N > 8$  hat die Accuracy den selben Wert, was darauf zurückzuführen ist, dass in diesem Fall die lokale Häufigkeit der globalen Häufigkeit entspricht, da in dieser Selektion bereits alle Sätze enthalten sind, die mindestens einen Kandidaten haben. Die höchste Accuracy erreicht das Verfahren bei  $N = 5$ . Dieses Ergebnis zeigt, dass die Selektion der ähnlichsten Sätze und die darauf basierende Auswahl des Kandidaten die Performanz im Vergleich zum naiven häufigkeitsbasierten Verfahren verbessern kann.

Für den Ansatz des Maschinellen Lernens werden alle Feature-Kombinationen evaluiert. An den Ergebnissen der einzelnen Features wird ersichtlich, dass das Ranking der *soft labels* auf Grundlage der Häufigkeit der Kandidaten im Text am aussagekräftigsten ist und eine Accuracy von 69,90% erreicht. Die Evaluierung der Feature-Kombinationen ergibt, dass das beste Ergebnis der *Decision Tree*-Classifier, auf Basis der Kombination aus der Häufigkeit des jeweiligen Kandidaten im Dokument, seiner ersten Position im Dokument (Satzindex) und des Vorkommens des Kandidaten in der Liste mit den Koreferenzen, aufweist. Dabei liegt die Accuracy bei 73,78% und der *MRR* bei 0,8171. Tabelle 5.3 gibt einen Überblick über die Ergebnisse der Experimente auf Basis des Rankings der *soft labels*. Die Ergebnisse lassen sich jedoch darauf zurückführen, dass der Datensatz relativ klein ist, und die Klassen in einem starken Ungleichgewicht sind, da es deutlich mehr falsche als richtige Kandidaten gibt.

Die Ergebnisse aller Ansätze, dargestellt in Abbildung 5.4, zeigen, dass die Häufigkeit des Kandidaten im verlinkten Dokument bereits ein gutes Merkmal dafür darstellt, ob ein Kandidat korrekt ist oder nicht. Im regelbasierten An-

Feature	Accuracy				MRR@3			
	DT	RF	NB	LR	DT	RF	NB	LR
Alle	0,4660	0,6019	0,6310	0,6699	0,6051	0,7297	0,7459	0,7637
1,5,6	0,7378	0,7184	0,6699	0,6893	0,8171	0,8090	0,7637	0,7766
1,4,6	0,7184	0,7184	0,6504	0,6699	0,7912	0,7961	0,7540	0,7605
1,4,5,6	0,7087	0,7087	0,6310	0,6893	0,7702	0,7928	0,7524	0,7766
1 frequency	0,6990	0,6990	0,6990	0,6990	0,7912	0,7912	0,7912	0,7912
2 max-cos	0,4271	0,4757	0,5825	0,5825	0,5695	0,6165	0,7152	0,7152
3 num-named-entities	0,5436	0,5533	0,5728	0,5728	0,6618	0,6763	0,7071	0,7071
4 num-keywords	0,5922	0,5922	0,5825	0,5825	0,6990	0,6990	0,6990	0,6990
5 first-sent-idx	0,5533	0,5533	0,4757	0,5533	0,6990	0,6990	0,6148	0,6957
6 in-corefs	0,4271	0,4271	0,4271	0,4271	0,5663	0,5663	0,5663	0,5663
7 sentiment	0,6310	0,6407	0,5145	0,6990	0,7508	0,7589	0,6553	0,7912

**Tabelle 5.3:** Ergebnisse des ML-Ansatzes: Ranking auf der Grundlage der *soft labels* für die Klasse *True*. Vier *Classifier* im Vergleich: Decision Tree (DT), Random Forest (RF), naive Bayes (NB) und Logistic Regression (LR).

Ansatz	Accuracy	MRR@3
Ranking nach der Position	0,2524	0,4498
Ranking nach der Häufigkeit	0,6893	0,8009
Regelbasierter Ansatz ( $N = 5$ )	0,7087	-
ML-Ansatz ( <i>Decision Tree</i> )	0,7378	0,8171

**Tabelle 5.4:** Ergebnisse der Evaluierung aller Ansätze im Vergleich.

satz wird eine minimale Steigerung der Performanz durch die Minimierung der Kandidaten auf der Basis der Kosinus-Ähnlichkeit erreicht.

# Kapitel 6

## Zusammenfassung und Ausblick

Die vorliegende Arbeit befasst sich mit dem Entwurf einer prototypischen, semi-automatisierten *Pipeline* zum Auflösen von Clickbait-Schlagzeilen. Im Rahmen dieser Arbeit wird ein Korpus konstruiert, der aus 7229 Clickbait-Spoiler-Pärchen besteht. Die Daten werden von bestimmten Nutzerkonten (siehe Kapitel 3.1) der *Social Media* Plattformen Twitter und Facebook extrahiert. Die hinter den Kulissen agierenden Redakteure verfolgen das Ziel, durch aktives Auflösen der Clickbait-Schlagzeilen vorrangig die Banalität und die Oberflächlichkeit der verbreiteten Meldungen aufzuzeigen, sowie die tatsächliche Anzahl der Klicks, um die damit verbundenen Werbeeinnahmen der jeweiligen Webseiten zu minimieren.

Zusätzlich zum Clickbait-Spoiler-Korpus wird der Inhalt der jeweiligen Zielseiten extrahiert und gespeichert. Neben dem Titel und dem Text des verlinkten Dokuments werden unter anderem die URLs zu Bild- und Videodateien sowie Metainformationen, wie Artikelbeschreibungen und Artikel-*Keywords*, abgespeichert. Somit stellt die Korpuskonstruktion einen wichtigen Beitrag für diese und zukünftige Arbeiten im Bereich des Clickbait-Spoilings dar.

Die linguistische Analyse der Clickbait-Schlagzeilen hinsichtlich ihres vorwärts-verweisenden Charakters zeigt, dass kataphorische Sprachmittel eine breite Verwendung in solchen Schlagzeilen finden: Mehr als zwei Drittel der Clickbait-Schlagzeilen (5124 Schlagzeilen) aus dem Korpus werden in die Kategorie der Vorwärtsreferenzen eingeordnet. Diese Auswertung bestärkt erneut die These, dass die meisten Clickbait-Schlagzeilen durch fehlende Informationen auf die Neugier der Menschen und weniger auf die Qualität der Artikel setzen, um auf diese Art und Weise die Nutzer auf die eigene Webseite zu locken.

Das in dieser Arbeit vorgestellte Verfahren lehnt sich stark an die Methodik aus dem Forschungsfeld des *Question Answering* an und konzentriert sich vor allem auf die Kandidatenextraktion aus dem verlinkten Dokument sowie das Ranking von Kandidaten. Aus den Ergebnissen der Evaluation ist eindeutig ablesbar,

dass bereits das häufigkeitsbasierte Ranking der Kandidaten eine Accuracy von 68,93% erzielt. Somit ist die Häufigkeit des Kandidaten im Dokument ein aussagekräftiges Merkmal, um den Kandidaten als korrekt einzustufen. Sowohl die Schritte der Selektion im regelbasierten Verfahren als auch die zusätzlichen *Features* im *Machine-Learning*-Ansatz tragen zur Verbesserung der Accuracy bei, weisen jedoch eine geringe Signifikanz hinsichtlich der Performanz auf. Besonders beim *Machine-Learning*-Ansatz sind die Ergebnisse auf die Größe und das unausgeglichene Verhältnis der falschen und richtigen Kandidaten in den Trainingsdaten zurückzuführen.

Diese Erkenntnisse motivieren, weitere Untersuchungen an einem größeren und ausbalancierteren Datensatz hinsichtlich der Kandidatenverteilung durchzuführen, um die tatsächliche Signifikanz der einzelnen *Features* zu ermitteln.

Die aus dem Question-Answering-Bereich angepasste Pipeline bietet aufgrund ihrer Modularität Anknüpfungspunkte für zukünftige Arbeiten. Der Schritt der Clickbait-Verarbeitung könnte durch die Ausarbeitung einer aussagekräftigeren Taxonomie der Clickbaits sowie die Entwicklung von Classifiern zur Klassifizierung der Clickbaits automatisiert werden. Die Erweiterung der Taxonomie sowie die automatisierte Clickbait-Verarbeitung würde zudem die Anwendung der Pipeline auf andere Klassen von Clickbaits ermöglichen, und zwar solche, die kein *Named Entity* als Spoiler verlangen.

In zukünftigen Projekten könnten andere Ansätze für die Extraktion von Spoilern im Vergleich getestet werden. Die Erkenntnis über den vorwärtsweisenen Charakter der Clickbait-Schlagzeilen motiviert, das Problem des Spoilings mit der Technik der Koreferenz-Auflösung anzugehen, wobei es ebenfalls nur auf die Klasse der faktenbasierten Clickbaits angewendet werden kann. Die automatisierte Textzusammenfassung (*automatic summarization*) stellt ebenfalls einen validen Ansatz zur Lösung des Spoilings-Problems dar, auch im Hinblick auf die narrativen Clickbaits.

# Danksagung

Ein herzlicher Dank geht an dieser Stelle an Junior-Prof. Dr. Matthias Hagen, Dr. Martin Potthast, sowie Tim Gollub für Ihre intensive Betreuung und die konstruktive Kritik, die sie mir von Woche zu Woche mit auf den Weg gegeben haben. Außerdem bedanke ich mich bei Vinc, Manu, Henning, Isabelle, Jonas und Giuli dafür, dass sie mir während der gesamten Zeit mit Rat und Tat zur Seite standen.

# Literaturverzeichnis

- Beckman, J. (2014). Saved You A Click - Don't click on that. <https://twitter.com/savedyouaclick>. [Online; letzter Zugriff: 23. Oktober 2017]. (siehe S. 3, 8)
- Bilton, R. (2014). The many different ways publishers define 'clickbait'. <https://digiday.com/media/many-many-ways-publishers-define-clickbait/>. [Online; letzter Zugriff: 23. Oktober 2017]. (siehe S. 2)
- Biyani, P., Tsoutsouliklis, K., and Blackmer, J. (2016). "8 Amazing Secrets for Getting More Clicks": Detecting Clickbaits in News Streams Using Article Informality. In *AAAI*, pages 94–100. (siehe S. 5, 7)
- Blom, J. N. and Hansen, K. R. (2015). Click bait: Forward-reference as lure in online news headlines. *Journal of Pragmatics*, 76:87–100. (siehe S. 6, 10, 11)
- Bram, V. (2014). The listicle: An exploring research on an interesting shareable new media phenomenon. *Studia Universitatis Babes-Bolyai, Ephemerides*, 59(1). (siehe S. 6)
- Chang, A. X. and Manning, C. D. (2012). Sutime: A library for recognizing and normalizing time expressions. In *LREC*, volume 2012, pages 3735–3740. (siehe S. 26)
- El-Arini, K. and Tang, J. (2014). News Feed FYI: Click-baiting. <https://newsroom.fb.com/news/2014/08/news-feed-fyi-click-baiting/>. [Online; letzter Zugriff: 23. Oktober 2017]. (siehe S. 5)
- Finkel, J. R., Grenager, T., and Manning, C. (2005). Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd annual meeting on association for computational linguistics*, pages 363–370. Association for Computational Linguistics. (siehe S. 26)

- Gollub, T., Potthast, M., Hagen, M., and Stein, B. (2017). Clickbait Challenge 2017. <http://www.clickbait-challenge.org/>. [Online; letzter Zugriff: 23. Oktober 2017]. (siehe S. 6)
- Halliday, M. A. and Hasan, R. (1976). Cohesion in. *English, Longman, London.* (siehe S. 11)
- Heeyoung, L., Angel, C., Yves, P., Nathanael, C., Mihai, S., and Dan, J. (2013). Deterministic coreference resolution based on entity-centric, precision-ranked rules. *Computational Linguistics*, 39(4):885–916. (siehe S. 22)
- Jurafsky, D. and Martin, J. H. (2017). Speech and Language Processing (3rd ed. draft). <http://web.stanford.edu/~jurafsky/slp3/28.pdf>. [Online; letzter Zugriff: 23. Oktober 2017]. (siehe S. 17, 33)
- Kohlschütter, C., Fankhauser, P., and Nejdl, W. (2010). Boilerplate detection using shallow text features. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 441–450. ACM. (siehe S. 16)
- Komlossy, K. (2017). Corpus-Viewer. <https://git.webis.de/webisstud/wstud-viewer-framework-django>. [Online; letzter Zugriff: 23. Oktober 2017]. (siehe S. 12)
- Li, X. and Roth, D. (2002). Learning question classifiers. In *Proceedings of the 19th international conference on Computational linguistics- Volume 1*, pages 1–7. Association for Computational Linguistics. (siehe S. 18)
- Li, X. and Roth, D. (2006). Learning question classifiers: the role of semantic information. *Natural Language Engineering*, 12(3):229–249. (siehe S. 18)
- Loewenstein, G. (1994). The psychology of curiosity: A review and reinterpretation. *Psychological bulletin*, 116(1):75. (siehe S. 2)
- Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S. J., and McClosky, D. (2014). The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60. (siehe S. 18, 20, 21, 22, 26)
- Mizrahi, A. (2012). HuffpoSpoilers - I give in to clickbait so you don't have to. <https://twitter.com/huffpospoilers>. [Online; letzter Zugriff: 23. Oktober 2017]. (siehe S. 3, 8)

- Moldovan, D. I., Harabagiu, S. M., Pasca, M., Mihalcea, R., Goodrum, R., Girju, R., and Rus, V. (1999). LASSO: A Tool for Surfing the Answer Net. In *TREC*, volume 8, pages 65–73. (siehe S. 19)
- Ou-Yang, L. (2016). Newspaper3k: Article scraping & curation. <https://github.com/codelucas/newspaper>. [Online; letzter Zugriff: 23. Oktober 2017]. (siehe S. 16)
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830. (siehe S. 30, 31)
- Peysakhovich, A. and Hendrix, K. (2016). News Feed FYI: Further Reducing Clickbait in Feed. <https://newsroom.fb.com/news/2016/08/news-feed-fyi-further-reducing-clickbait-in-feed/>. [Online; letzter Zugriff: 23. Oktober 2017]. (siehe S. 2)
- Potthast, M., Köpsel, S., Stein, B., and Hagen, M. (2016). Clickbait Detection. In Ferro, N., Crestani, F., Moens, M.-F., Mothe, J., Silvestri, F., Di Nunzio, G., Hauff, C., and Silvello, G., editors, *Advances in Information Retrieval. 38th European Conference on IR Research (ECIR 16)*, volume 9626 of *Lecture Notes in Computer Science*, pages 810–817, Berlin Heidelberg New York. Springer. (siehe S. 5)
- rslashSYAC (2016). The official twitter feed of the subreddit /r/savedyouaclick. We click on stuff so you don't have to. <https://twitter.com/rslashSYAC>. [Online; letzter Zugriff: 23. Oktober 2017]. (siehe S. 8)
- Rosenkranz, B. (2016). Es muss Technologien geben, die uns helfen, nicht geistig ausgebeutet zu werden. <http://uebermedien.de/3827/es-muss-technologien-geben-die-uns-helfen-nicht-geistig-ausgebeutet-zu-werden/>. [Online; letzter Zugriff: 23. Oktober 2017]. (siehe S. 3)
- Stempeck, M. (2013). Upworthy Spoiler - Words that describe the links that follow. <https://twitter.com/UpworthySpoiler>. [Online; letzter Zugriff: 23. Oktober 2017]. (siehe S. 3, 8)
- StopClickbaitOfficial (2016). Mission: To rid the Internet of clickbait. We click so you don't have to. [https://www.facebook.com/pg/StopClickBaitOfficial/about/?ref=page\\_internal](https://www.facebook.com/pg/StopClickBaitOfficial/about/?ref=page_internal). [Online; letzter Zugriff: 23. Oktober 2017]. (siehe S. 8)

Wikipedia (2016). Clickbaiting — Wikipedia, The Free Encyclopedia.  
<https://de.wikipedia.org/w/index.php?title=Clickbaiting&oldid=167604893>. [Online; letzter Zugriff: 23. Oktober 2017]. (siehe S. 2)

Youwen, Y. (2011). A cognitive interpretation of discourse deixis. *Theory and Practice in Language Studies*, 1(2):128–135. (siehe S. 10)