Automatisierte Auswahl von positiven Nachrichten Anhand der Schlagzeilen

01.02.2021

Klemens Gerber
klemens.gerber@siemens.com

Florian Gemmer
florian.gemmer@siemens.com

Modul: Neue Konzepte II: Natural Language Processing
Dozent: Prof. Dr. Tobias Günther
Duale Hochschule Baden-Württemberg - Mannheim

1 Inhaltsverzeichnis

2 Einleitung

Das vorgestellte Projekt befasst sich mit der automatisierten Auswahl von positiven Nachrichten mithilfe von Natural Language Processing (NLP). Dabei soll sich nur auf die Schlagzeile der Artikel bezogen werden, nicht aber auf den Inhalt selbst. Hierfür soll der Naive Bayes Algorithmus genutzt werden.

Weshalb sollte ein solcher Algorithmus verwendet werden? - Die Berichterstattung wird dominiert von negativen Meldungen. Ein Blick in die Nachrichten reicht meistens schon aus, um diese These zu bestätigen. Aber auch Studien belegen dies: Hermanni und Schiller haben über den Zeitraum von einer Woche vier große deutsche Nachrichtensendungen analysiert. Das Ergebnis waren 146 negativ geprägte, 123 neutrale und 9 positive Beiträge.[1] Warum das so ist lässt sich mit dem sogenannten “Negativity Bias” erklären. Menschen tendieren zu einer stärkeren emotionalen Reaktion auf negative, als auf positive Nachrichten. Deshalb verkaufen sich Zeitungen besser, erhalten Online Artikel mehr Klicks und bekommen Nachrichtensendungen bessere Einschaltquoten, wenn sie sich auf negative Nachrichten fokussieren. Dies ist kein deutsches oder “westliches” Phänomen. Eine groß angelegte Studie von Soroka et. al. über 17 Länder auf sechs Kontinenten konnte den Negativity Bias länderübergreifend feststellen.[2]

Es ist aber auch ein gegenläufiger Trend zu beobachten: Nachrichtenportale wie die deutsche Webseite “nur-positive-nachrichten.de” oder die in London ansässige Medienorganisation Positive News sind darauf spezialisiert ausschließlich positive Nachrichten zu verbreiten. Dass es bedarf gibt, zeigt die im Jahr 2015 von Positive News gestartete Crowdfunding Kampagne, an der über 1500 personen teilnahmen.

An diesem Punkt setzt die Arbeit an. Während die Nachrichten der genannten Anbieter kuratiert werden, würde ein Algorithmus die Arbeit vereinfachen. Durch einen NLP basierten Ansatz kann die Auswahl der Nachrichten auf einem Portal dieser Art nicht ersetzt werden.Es könnte jedoch zumindest eine Vorverarbeitung getroffen werden. Ein weiterer Ansatz wäre es, traditionelle Nachrichtenseiten für den Nutzer zu filtern und nur vom Algorithmus als positiv bewertete Nachrichten anzuzeigen.

---

[1] vgl. Hermanni, Schiller 2019

[2] vgl. Soroka et. al. 2019

3 Datenquelle

Als Datenquelle dient der “A Million News Headlines” Datensatz von Kaggle.com[3]. Der Datensatz beinhaltet eine Millionen Nachrichtenüberschriften des australischen Ablegers der ABC. Die Daten Bilden einen Zeitraum von 18 Jahren, von 2003 bis 2020, ab. Im Schnitt veröffentlicht ABC 200 Artikel pro Tag.

Die ABC Australia ist eine traditionelle Rundfunkgesellschaft mit einigem Fokus auf internationalen Nachrichten. Meldungen beziehen sich auf alle Lebensbereiche, von Wirtschaft über Politik bis hin zu Lifestyle und Sport. Damit ist ein guter Querschnitt der Nachrichtenwelt gegeben.

4 Theorie

In der Theorie sollten sich, nach der Auswertung eines ausreichend großen Datensatzes, Wörter abzeichnen, welche auf eine eher positive oder eher negative Konnotation einer Überschrift hinweisen. Diese Wörter sollten gewichtet werden. Es ist zu erwarten, dass eine gewisse, geringe, Anzahl an Begriffen einen sehr starken Indikator für die Neigung eines Textes darstellen. Der Großteil der verwendeten Begriffe sollte eine fast neutrale Gewichtung haben.

Die meisten der von uns verwendeten Wörter sind weder mit positiver noch mit negativer Bedeutungen assoziiert. Ebenso verhält es sich mit Überschriften. Beim händischen klassifizieren von über 2000 Überschriften konnte festgestellt werden, dass deren Großteil neutral gehalten ist (über 900 der gelabelten Überschriften wurden als neutral klassifiziert). Entsprechend werden bei diesem Ansatz viele Nachrichten aussortiert, die möglicherweise einen positiven beiklang haben, der sich jedoch nicht aus der Überschrift erschließen lässt.

Ebenfalls wurde beim händischen labeln der Daten erkannt, dass positiv und negativ selbstverständlich starker subjektivität unterworfen sind. Da ABC dem liberalen Spektrum zuzuordnen ist, kann erwartet werden, dass zum Beispiel Nachrichten zu konservativen Politikern oder Waffenrechten häufig einen negativen Anklang haben. Demgegenüber werden liberale Themen wie Umweltschutz positiv gesehen. Daraus lässt sich schließen, dass das hier erarbeitete Klassifikationsmodell einen besseren Nutzen für ein liberales Publikum darstellt. Um den gleichen Wert für ein konservatives Publikum zu erreichen, sollte eine andere Datengrundlage verwendet werden.

---

[3] vgl. Kulkarni, 2020

5 Naive Bayes'

"[The idea] of Bayesian inference has been known since the work of Bayes (1763), and was first applied to text classification by Mosteller and Wallace (1964)."[4]

In der Literatur finden sich verschiedene Arten von Bayes' Classifiern. Grundsätzlich lässt sich aber zwischen drei Varianten unterscheiden: Binomial, Multinomial und Gaussian. Da es sich bei dem hier vorgestellten Klassifizierungsproblem nicht um ein kontinuierliches Problem handelt, scheidet die Gaussian Variante aus. In einem Binomial Bayes Classifier wird nur in Betracht gezogen, ob ein Wort in einem Text vorkommt, nicht aber wie oft es vorkommt. Daher wurde sich in dieser Arbeit für die Multinomiale Variante entschieden.

Konkret handelt es sich hier um einen multinomialen naive Bayes classifier. Der Klassifikator wird aus zwei Gründen naiv genannt:

1. Der Klassifikator zieht die Reihenfolge der Wörter in den untersuchten Dokumenten nicht in Betracht. Diese Annahme wird auch "bag of words" assumption genannt.
2. Es wird davon ausgegangen, dass die betrachteten Wahrscheinlichkeiten für die Features unabhängig sind. Diese Annahme wird auch "Naive Bayes Assumption" genannt.

Multinomial bedeutet, dass der Classifier nicht nur darauf fokussiert ist, ob ein bestimmtes Wort in einem Text vorkommt, sondern auch wie häufig es vorkommt. Die Standardgleichung zur Klassen Vorhersage im Multinomial Naive Bayes lautet

$$c_{NB} = argmax[log(P(c)) + log(\prod_{i \in positions} P(w_i|c))]$$

Gleichung 4.1

Dabei steht c_{NB} für die von Bayes vorhergesagte Klasse. $P(c)$ist die Prior Probability einer bestimmten Klasse. Die Prior Probability gibt an, wie oft eine Klasse im Trainingsdatensatz, im Verhältnis zur Gesamtheit aller Klassen, vorkommt. Diese Prior Probability wird multipliziert mit den Wort Wahrscheinlichkeiten für jedes im untersuchten Dokument vorkommende Wort unter der Voraussetzung, dass die Klasse des Dokuments c ist. Die Berechnung findet im logarithmischen Raum statt, um Underflow (die zur Berechnung verwendeten Zahlen sind so klein, dass ein Rechner nicht mit ihnen umgehen kann) zu

[4] Jurafsky, Martin, 2020

vermeiden und die Rechengeschwindigkeit zu erhöhen. Der Unterschied zu einem Binomial Classifier ist dabei, dass ein Binomial Classifier nur untersucht ob ein Wort in einem Dokument vorkommt, nicht aber wie oft.

Nachdem Gleichung 4.1 aufgestellt ist und erklärt wurde, wie die Prior Probability berechnet wird, gilt es die Wahrscheinlichkeit für das Vorkommen eines Wortes für jede Klasse zu berechnen. Dies geschieht im Training des Naive Bayes' wie folgt:

$$P(w_i|c) = \frac{count(w_i, c)}{\sum_{w \in V} count(w, c)}$$

Gleichung 4.2

Für $count(w_i, c)$ werden alle Dokumente der Klasse c konkateniert und anschließend gezählt, wie häufig w in c vorkommt. Das Ergebnis wird geteilt durch die Anzahl des Wortes w über alle Klassen (V ist eine Liste der Anzahl der Vorkommnisse aller Worte über alle Klassen).

Folgende Frage bleibt offen: Was passiert, wenn das Wort w_i nicht in Klasse c vorkommt?

Gleichung 4.2 würde null ergeben, entsprechend eingesetzt würde auch Gleichung 4.1 null ergeben. Damit wäre, sobald ein Wort w aus einem Testdokument d nicht im Trainingsdatensatz für Klasse c vorkam, die Wahrscheinlichkeit dass d in c liegt automatisch Null. Um das zu verhindern gibt es zwei Ansätze:

1. Smoothing

   Häufig wird Laplace Smoothing eingesetzt um oben beschrieben Fall zu verhindern. Dabei wird Gleichung 4.2 wie folgt angepasst:

$$P(w_i|c) = \frac{count(w_i, c) + 1}{\sum_{w \in V} count(w, c) + |V|}$$

Gleichung 4.3

2. Eine Alternative vorgehensweise ist es, unbekannte Worte schlicht zu ignorieren. Kommt ein Wort aus dem Testdatensatz nicht im Trainingsdatensatz vor, ihm kann also kein Gewicht zugewiesen werden, wird es aus dem Testdatensatz ausgeschnitten. Damit wird es in der Klassifizierung nicht weiter berücksichtigt.

6 Datentransformation

Zu Beginn sollte die Klassifikation auf Basis von ca. 2000 händisch mit drei Klassen annotierten Headlines aufgesetzt werden. Dies erwies sich jedoch nach einer ersten Implementierung als nicht möglich, da der Algorithmus eine signifikant größere Datenbasis benötigt. Um trotzdem die angegebene Datenbasis nutzen zu können, wurden die Daten automatisiert mit dem Sentiment Modell NLTK Vader annotiert. Dies soll einen proof of Concept ermöglichen, der eventuell mit einer nachträglich von Hand annotierten, größeren Datenbasis umgesetzt werden kann.

Vader steht für Valence Aware Dictionary for sEntiment Reasoning. Vader verwendet ein Sentiment Lexikon in Verbindung mit grammatikalischen und syntaktischen Konventionen um den Sentiment Wert eines Textes zu ermitteln. Das von Vader verwendete Sentiment Lexikon beinhaltet 7.500 Wörter, denen ein Wert zwischen -4 (sehr negativ) und +4 (sehr positiv) zugewiesen ist. Vader scannt den übergebenen Text nach diesen Wörtern und modifiziert den Sentiment Wert für einzelne Wörter mithilfe seiner grammatikalischen und syntaktischen Regeln. Die sich ergebenden Werte werden aufsummiert und anschließend nach folgender Gleichung normalisiert:

$$CompScore = \frac{x}{\sqrt{x^2 + \alpha}}$$

Gleichung 5.1

x steht hier für den errechneten sentiment Wert für das Dokument. α wird von Vader mit Wert 15 verarbeitet. Dies ist der maximale Wert den x Erwartungsgemäß annimmt. Das Ergebnis ist der Compound Score, der zwischen -1 und +1 liegt.[5]

200.000 Headlines aus dem “A Million News Headlines ” wurden so annotiert. Headlines mit einem Compound Score zwischen -0,3 und +0,3 wurden als neutral klassifiziert, alles darüber positiv, darunter negativ.

[5] vgl. Hutto, Gilbert 2014

Die von Vader Annotierten Daten wurden wie folgt vorverarbeitet:

1. Entfernen von Null Values aus dem Data Frame
    a. Leere Einträge können das Ergebnis verfälschen und werden deshalb entfernt.
2. Alle Buchstaben im Dataframe werden in lowercase umgewandelt
    a. Um bei späteren Verarbeitungsschritten nicht auf Groß- und Kleinschreibung achten zu müssen, wird sie von Anfang an nicht beachtet.
3. Zahlen werden aus den Text Strings entfernt
    a. Zahlen werden aus dem Text entfernt, da sie nicht im Kontext gesehen werden können. Deshalb würden sie das Ergebnis verfälschen.
4. Satzzeichen werden aus den Text strings entfernt
    a. Mit hilfe von Regular Expressions werden Satzzeichen aus dem Text entfernt. Im Falle von Methoden, die grammatische Strukturen schätzen sollte dies unterlassen werden, aber da ein Bag of Words Approach genutzt wird, würden Satzzeichen das Ergebnis verfälschen.
5. Tokenization
    a. Die Tokenization ist der wichtigste Schritt der Vorverarbeitung. Die Headline Strings werden an Leerzeichen aufgeteilt, was die Sätze in ihre Einzelteile zerlegt, die Tokens genannt werden und nun einzeln betrachtet werden können.
6. Tokens mit einer Länge < 2 werden in der Classification aussortiert
7. Stopwords entfernen
    a. Stopwords sind Worte, die besonders oft vorkommen und gewöhnlich keine Relevanz für die Klassifikation haben. Sie werden mithilfe einer statischen Liste aussortiert.

Es wurde weder eine Lemmatization, noch Stemming verwendet, da keiner dieser Schritte bei den durchgeführten Tests eine signifikante Verbesserung der Ergebnisse brachte. Gleichzeitig haben diese Verarbeitungsschritte jedoch die Durchlaufzeit des Trainings und Klassifikationsprozesses stark erhöht. Deshalb wurden sie im finalen Prozess nicht eingesetzt.

7 Implementierungsdetails

Die Implementierung sieht zwei Klassen “positiv” und “negativ” zur Klassifizierung vor. Hierfür wird ein Train - Test Split von 60/40 auf dem vorher bereits genannten ABC News Datensatz verwandt. Hierbei werden 120.000 Headlines für das Training und 80.000 für das Testing verwendet. Es werden, abgesehen von NLTK Vader als Werkzeug zum Labelling, lediglich numpy und pandas als Dependencies benötigt.

Zuerst werden die ”rohen” Daten in einen Pandas Dataframe eingelesen. In diesem folgt die zentrale Vorverarbeitung aller Daten gemeinsam (Siehe Kapitel Datentransformation). Anschließend wird der ursprüngliche Dataframe in einen Training und einen Testing Dataframe aufgeteilt. Danach kann die Menge der Test und Trainingsdaten manuell festgelegt werden, um die Klassenverteilung der Trainingsdaten beispielsweise anzupassen, sollte der Usecase dies erfordern. Anhand speziell hierfür erstellter Dataframes wird anschließend die anzahl von positiv und negativ klassifizierten Trainingsdaten ermittelt, die später im Prediction Modell genutzt wird. Zudem wird die Anzahl aller Trainingsdaten berechnet. Diese Werte werden anschließend in der Trainingsfunktion genutzt. In dieser werden die ganzzahligen Worthäufigkeiten aller Tokens pro Klasse ermittelt. Hierfür wird über alle Tokens im Training Dataframe iteriert. Die Tokens werden, sollten sie noch nicht in diesem vorhanden sein, in einem Dictionary gespeichert. Dieses Dictionary enthält Informationen über die Häufigkeit des vorkommens jedes vorhandene Tokens in den jeweiligen Klassen “positiv” und “negativ. Ist ein Token bereits in diesem “Vocabulary” Dictionary vorhanden, wird bei einem erneuten vorkommen die betreffende Anzahl seines vorkommens in der jeweiligen Klasse erhöht. So wird ein Dictionary mit Worthäufigkeiten aller Tokens nach Klasse geschlüsselt aufgestellt Siehe Code 6.1). Ist dieses Dictionary vollständig, kann zur Prediction Phase übergegangen werden.

```
for headline, label in zip(dfTraining["headline"],dfTraining["Vader_Label"]):
    for word in headline:
        if word in Vocab:
            if label == "p":
                Vocab[word][1] +=  1
            elif label == "n":
                Vocab[word][0] += 1
        elif word not in Vocab:
            if label == "p":
                Vocab[word] = [0,1]
            if label == "n":
                Vocab[word] = [1,0]
```

Code 6.1

In der Prediction Phase wird über die Tokens jeder Headline im Testing Dataframe Iteriert. Dabei müssen die Tokens eine Länge > 2 haben, aus Buchstaben bestehen und nicht in der Liste der Stopwords vorkommen, um in der Prediction berücksichtigt zu werden. Wird eine dieser Bedingungen nicht erfüllt, so wird der betreffende Token übersprungen. Für Tokens, die diese Bedingungen erfüllen, wird der Class Posterior wie folgt berechnet:

$$\sum log \frac{|Token(class)|}{|Headlines(train)|}$$

Gleichung 6.2

Dabei steht $|Token(class)|$für die Häufigkeit von Token in Klasse class und $|Headlines(train)|$für die Anzahl aller Headlines im Training Dataframe. Anschließend werden die beiden so erhaltenen Werte mit dem logarithmus des Class Priors Addiert, der sich aus der Anzahl der Headlines der Jeweiligen Klasse geteilt durch die Gesamtanzahl der Headlines in der Trainingsbasis ergibt. (Siehe Code 7.2). Anhand der so errechneten Werte wird eine Voraussage getätigt, indem diejenige Klasse vorausgesagt wird, die durch die vorhergegangene Rechnung den größeren Wert ergibt.

```
for word in headlines:
if WordNotZero(word) == True and (len(word) > 2) and word.isalpha() and word not in StopWords:

PNegTokens=PNegTokens +np.log(Vocab[word][0]/AmountNegativeHeadlines)
PPosTokens=PPosTokens+np.log(Vocab[word][1]/AmountPositiveHeadlines)

PNegTokens=PNegTokens+np.log(AmountNegativeHeadlines/length)
PPosTokens=PPosTokens+np.log(AmountPositiveHeadlines/length)
```

Code 6.3

Im Prototyp für das Produktivsystem werden positiv und negativ klassifizierte Headlines in separate Dictionaries gespeichert. Über einen Index kann so eine Liste mit der klassifizierten Headline, sowie ein Link zu deren Quelle abgerufen und anschließend in HTML eingebunden oder anderweitig verwandt werden. Dies ermöglicht eine schnelle Nutzung und weiterverarbeitung der Ergebnisse.

8 Evaluierung der Umsetzung

Der trainierte Naive Bayes algorithmus performt bei einem Train Test Split von 60:40 mit einem Trainingsvolumen von ca. 57.400 Trainingsdaten mit einer Accuracy von 91,5%. Es konnte eine Precision von 92,5% (92,5% aller positiv klassifizierten Headlines sind tatsächlich positiv) und einem Recall von 95,2% (95,2% der gegebenen Headlines werden korrekt klassifiziert) erreicht werden. So lässt sich ein F1 Score von 93,8% errechnen.

	Predicted Positive	Predicted Negative
Actual Positive	22.331	1804
Actual Negative	1130	9439

7.1 Confusion Matrix

Die abgebildete Konfusionsmatrix zeigt, dass ca 5% aller positiv klassifizierten Headlines tatsächlich negativ sind. Diese könnten also fälschlicherweise als positiv angezeigt werden. Gleichzeitig sind im Testing 16% aller negativ klassifizierten Headlines tatsächlich positiv. Dieser Wert scheint zunächst sehr hoch, da für unseren Business Case jedoch ausschließlich positiv klassifizierte Headlines genutzt werden, ist dieser Parameter nicht ausschlaggebend für die Performance des Algorithmus.

Allgemein lässt sich sagen, dass negative Headlines verlässlich gefiltert werden. Trotzdem kann es in eher seltenen Fällen dazu kommen, dass eine eher negative Headline als positiv klassifiziert wird. Die Erfahrung zeigt jedoch, dass diese Headlines tatsächlich weniger stark negativ sind und meist auch einen schwach positiven Unterton haben. So können beispielsweise Headlines, die eine negative Thematik mit allgemein positiv konnotierten Wörtern beschreiben, weniger gut erkannt werden. Das liegt an dem Bag of Words Approach, der keine grammatikalischen Zusammenhänge erkennen kann.

Abschließend lässt sich sagen, dass der Kern des Systems angemessen genau performt und gut genug ist, um die Business Vision zu realisieren. Jedoch sind in Zukunft sicherlich noch bessere Ergebnisse mit einem komplexeren Modell und einer händisch annotierten Datenbasis möglich. Als Prototyp wird das Projekt von uns jedoch als erfolgreich betrachtet, der bei der Realisierung unserer Ziele eingesetzt werden kann.

9 Anhang

Das Projekt ist unter folgendem Link zu finden:
https://github.com/gerbklee/Headline_Classification_Florian_Klemens_WWI18DSA

Die Trainingsdaten sind unter folgendem Link zu finden:

https://drive.google.com/file/d/111kD-GVgK45NMthHxWgllxdroQCqaR9p/view?usp=sharing

10 Quellenangaben

Hermanni, Alfred-Joachim; Schiller, Tatjana (2019): Negative News dominieren die Nachrichtensendungen de Fernsehens, Masterarbeit SRH Fernhochschule

Hutto, C.J. ; Gilbert, Eric (2014): Vader: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text In: Proceedings of the Eighth International AAAI Conference on Weblogs and Social Media

Jurafsky, Daniel; Martin, James H. (2020/Kapitel 4 S.3): Naive Bayes and Sentiment Classification In: Speech and Language Processing

Kulkarni, Rohit (2020): A Million News Headlines - News headlines published over a period of 18 Years In: https://www.kaggle.com/therohk/million-headlines, 2020

Soroka, Stuart; Fournier, Patrick; Nir, Lilach (2019): Cross National Evidence of a negativity bias in psychophysiological reactions to news In: www.pnas.org/cgi/doi/10.1073/pnas.1908369116