Projekt Big Data

Wintersemester 2017/18

Projektbericht –
Analyse von News-Artikeln – Sentimentanalyse mit Word2vec

Raffael Diestel, Maike Schubert
{5diestel, 5mschube}@informatik.uni-hamburg.de

Abstrakt. Im Laufe des Projekts wurde von uns untersucht, wie es möglich ist eine Sentimentanalyse mit Hilfe von Word2vec Modellen durchzuführen und inwiefern sich diese auf Artikel verschiedener Nachrichtenportale anwenden lässt. In diesem Projektbericht werden in der Einleitung zunächst die Aufgabenstellung und die Ziele dargestellt, bevor dann das Vorgehen und die Realisierung beschrieben werden. Es werden zwei verschiedene Ansätze vorgestellt und deren Ergebnisse analysiert. Zum Schluss werden die wichtigsten Aspekte noch einmal zusammengefasst und ein Fazit gezogen.

1 Einleitung und Aufgabenstellung

1.1 Aufgabenstellung

Die Aufgabe war es, eine Sentimentanalyse von News-Artikeln unter Zuhilfenahme von Word2vec, einem Modell zur Vektordarstellung von Wörtern, durchzuführen.
Verwendet haben wir dazu Artikel verschiedener Nachrichtenportale, die über einen zur Verfügung gestellten Crawler bezogen wurden.
Diese Artikel waren die Grundlage für die Sentimentanalyse.

Zunächst haben wir uns genauer darüber informiert, was eine Sentimentanalyse ist und worum es sich bei Word2vec handelt, um diese beiden Dinge dann miteinander kombinieren zu können.

1.2 Was ist eine Sentimentanalyse?

Eine Sentimentanalyse ist eine automatische Analyse eines Textes, bei der es darum geht dessen Stimmung zu erkennen. Dabei geht es um die Analyse der Polarität. Ziel ist es also, eine in einem Text ausgedrückte Haltung zu erkennen und diese als positiv, negativ oder auch neutral einzuordnen.
Das genaue Vorgehen für eine Sentimentanalyse in Verbindung mit Word2vec wird später noch erläutert.
Generell kann bei einer Sentimentanalyse verschiedene Methoden kombinieren. Zunächst wird eine Grundmenge von Begriffen ausgewertet, deren Polarität man bestimmen muss (positiv oder negativ). Diese kann man dann heranziehen, um Methoden des Machine Learnings darauf anzuwenden, sodass die genutzten Algorithmen für weitere Texte lernen können.
Sentimentanalyse ist dem Text Mining zuzuordnen. Es geht also um das Analysieren von Daten in natürlichsprachlicher Form.

1.3 Was ist Word2vec?

Für die Sentimentanalyse soll Word2vec genutzt werden. Dabei handelt es sich um ein Modell zur Vektordarstellung von Wörtern. Diese Vektordarstellungen werden als „word embeddings“, also Worteinbettungen bezeichnet.
Dadurch können verschiedene Ähnlichkeiten und semantische und syntaktische Beziehungen festgestellt und dargestellt werden.
Als Input des Modells dient ein großer Textkorpus, in unserem Fall waren es die gecrawlten News-Artikel, aus denen vorher noch der Reintext extrahiert wurde.

Aus diesem wird dann ein Vektorraum konstruiert. Die einzelnen Wörter werden auf Vektoren reeller Zahlen abgebildet, wobei die Wortvektoren so positioniert werden, dass Wörter mit einem gemeinsamen Kontext im Textkorpus auch im konstruierten Vektorraum nahe beieinander liegen, da davon ausgegangen wird, dass diese auch eine gemeinsame semantische Bedeutung haben.

1.4 Ziele

Für die Arbeit mit den gecrawlten Artikeln haben wir uns mehrere Ziele überlegt.
Zum einen sollte herausgearbeitet werden ob und wie eine Sentimentanalyse mit Word2vec funktioniert und konkret ob wie wie eine Sentimentanalyse von News-Artikel möglich ist und welche Rolle Word2vec dabei spielt.
Ein weiteres Ziel war es, die Frage zu beantworten was genau sich erkennen lässt, also spezielle Sentiments.
Zuletzt wollten wir herausfinden, ob sich mit Hilfe der Sentimentanalyse Unterschiede bei bestimmten Themen oder den verschiedenen Nachrichtenportalen erkennen lassen.

2 Vorgehen und Realisierung

2.1 Starten des Crawlers

Zur Beschaffung der News-Artikel wurde der von Max Lübbering zur Verfügung gestellte Crawler[1] verwendet. Dieser greift in Intervallen auf die übergebenen RSS-Feeds zu, prüft, ob neue Artikel vorhanden sind, und lädt diese dann anschließend herunter und speichert sie in einer CSV-Datei.

Es wurden die RSS-Feeds von neun verschiedenen Nachrichtenportalen verwendet:

-FOCUS (www.focus.de) -http://rss.focus.de/fol/XML/rss_folnews.xml
-T-Online (http://www.t-online.de) -http://rss1.t-online.de/c/11/52/20/74/11522074.xml
-Tagesschau (www.tagesschau.de) -http://www.tagesschau.de/xml/atom/

[1] *https://github.com/le1nux/crawly*

-WELT (www.welt.de) -https://www.welt.de/feeds/latest.rss
-Spiegel Online (www.spiegel.de) -http://www.spiegel.de/schlagzeilen/index.rss
-ZEIT online (http://newsfeed.zeit.de) -http://newsfeed.zeit.de/index
-ZDF (www.zdf.de) -https://www.zdf.de/rss/zdf/nachrichten
-FAZ (www.faz.net) -http://www.faz.net/rss/aktuell/
-Sueddeutsche (www.sueddeutsche.de) -http://rss.sueddeutsche.de/app/service/rss/alles/index.rss?output=rss

Die einzelnen Feeds sind nicht weiter nach Themen oder Kategorien gefiltert, da die Übereinstimmungen bei den verschiedenen Nachrichtenportalen zu gering waren. Die Kategorien zu denen die einzelnen Artikel gehören, waren nicht immer deutlich genug abgrenzbar, da die Themen auf den verschiedenen Seiten unterschiedlich zugeordnet werden. Auch die Anzahl der Kategorien stimmt nicht überein. Daher haben wir also im Vorhinein keine weitere Einteilung vorgenommen.
Durch die Website-Downloads von Artikeln wurden durch den Crawler insgesamt 108 GB an Daten gesammelt.

2.2 Extraktion des Textes

Die Extraktion des Textes aus den Website-Downloads wurde mit Hilfe von Beautiful Soup 4[2] durchgeführt. Dabei handelt es sich um eine Python-Library zum Extrahieren von Informationen aus HTML- und XML-Dateien. Sie unterstützt den HTML-Parser aus der Python Standard-Library und ermöglicht den Zugriff auf die einzelnen HTML-Tags, sodass die Paragraphen der Artikel direkt ausgelesen werden konnten. Ziel war es, somit die unwichtigen Informationen zu entfernen und den daraus resultierenden Reintext erneut abzuspeichern.

[2] *https://www.crummy.com/software/BeautifulSoup/bs4/doc/*

Aus den insgesamt 108 GB Website-Downloads entstand so schließlich ein Reintext mit einer Größe von 158 MB:

- Welt 29.3 MB
- Focus 29.1 MB
- Süddeutsche 28.4 MB
- Spiegel 23.6 MB
- FAZ 16.4 MB
- Zeit 15.3 MB
- T-Online 10.3 MB
- Tagesschau 5.6 MB
- ZDF 4.3 MB

```
def focus(soup):
    article = soup.find(id='article')
    if article is None:
        return None
    else:
        paragraphs = article.find_all('p')
        content_list = []
        for paragraph in paragraphs:
            content_list.append(paragraph.get_text().strip())
        content = ' '.join(content_list)
        return content
```

Beispiel für die Extraktion des Textes eines Focus-Artikels aus dem HTML-Download

2.3 Word2vec-Modelle

Mit den Artikeln, dem extrahierten Text, wurden dann Word2vec-Modelle erstellt. Jeweils ein Modell für jedes Nachrichtenportal und anschließend noch ein Modell mit dem Text von allen Nachrichtenportalen kombiniert. So kann man dann diese für die Sentimentanalyse nutzen und die einzelnen Portale analysieren und miteinander vergleichen.
Hierzu wird folgendermaßen vorgegangen:

1. Laden der Texte aller Artikel einer Nachrichtenseite aus Datei

```
with open('{}'.format(self.file_path), 'r', encoding='utf-8') as text_file:
    self.text += text_file.read()
```

2. Aufsplitten der Sätze (Tokenizing) mit Hilfe von NLTK

```
tokenizer = nltk.data.load('file://' + nltk_path)
raw_sentences = tokenizer.tokenize(self.text)
```

3. Entfernen von Stoppwörtern und unerwünschten Zeichen und anschließendes Speichern in einer Wortliste (pro Satz)

```
stop_words = stopwords.words('german')
for raw_sentence in raw_sentences:
    clean_sentence = re.sub('[^a-zA-ZäöüÄÖÜß]', ' ', raw_sentence)
    word_list = [w for w in clean_sentence.split() if not w in stop_words]
    self.sentences.append(word_list)
```

4. Erstellen und Speichern des Word2vec-Modells

```
self.model = gensim.models.Word2Vec(sentences=self.sentences, min_count=25, size=500,
                                    workers=multiprocessing.cpu_count(), sample=1e-3)
self.model.save('trained/{}_model.w2v'.format(self.site))
```

5. Zur späteren Wiederverwendung kann das Modell geladen werden

```
gensim.models.Word2Vec.load('trained/{}_model.w2v'.format(self.site))
```

Somit erhält man Wortvektoren und trainiert die Modelle. Word2vec-Modelle kann man mit Skip-Gram oder Continous Bag of Words (CBOW) erstellen. In unserem Fall wird Skip-Gram (default) genutzt, da bei der Verwendung von CBOW keine deutlichen Verbesserungen erkennbar waren.

Word2vec Parameter

sentences

➢ Liste aus Wortlisten

min_count

➢ Anzahl wie oft ein Wort vorkommen muss, um in das Vokabular aufgenommen zu werden

size

➢ Dimensionalität der Wortvektoren

workers

➢ Anzahl der verwendeten Threads zum Trainieren des Modells

sample

➢ Schwellwert für das zufällige Downsampling von häufig vorkommenden Wörtern

Eine Liste aller möglichen Parameter und deren Erklärung sind auf der offiziellen Seite von gensim zu finden.

2.4 Sentimentanalyse - erster Ansatz

Word2vec bietet verschiedene Funktionen.
Die Funktion von Word2vec, die als Ansatz für die Sentimentanalyse verwendet wird, ist die Ähnlichkeitsfunktion. Mit dieser kann ermittelt werden, wie nahe die entsprechenden Wörter miteinander in Verbindung stehen. Je näher der Wert an 1 liegt, desto ähnlicher sind sich die beiden Wörter.

Beispiele für diese Funktion:

```
>>> w2v.model.wv.similarity('Kanzlerin', 'Merkel')
0.9176394683663345
```
→ Ähnlichkeit der Wörter „Kanzlerin" und „Merkel"

```
>>> w2v.model.wv.most_similar('GroKo')
('Koalition', 0.9391289949417114)
```
→ Wort, welches die größte Ähnlichkeit zu „GroKo" hat. → „Koalition"

```
>>> w2v.model.wv.most_similar_cosmul(['Kanzlerin',
Präsident'], ['Putin'])
('Merkel', 0.9637267589569092)
```
→ Das Wort, dass am ehesten im gleichen Zusammenhang zu „Kanzlerin" steht in dem „Putin" zu „Präsident" steht, ist „Merkel".

Für die Sentimentanalyse wird ein Sentiment-Score für jedes Wort eines Artikels berechnet. Dieser Wert basiert auf der Ähnlichkeit zu den Wörtern und deren Polarität aus dem Sentiment-Wortschatz den wir nutzen. Man benötigt also ein gelabeltes deutsches Dataset. Wir haben uns nach längerer Suche für SentiWS[3] entschieden.
Dieser wird von der Universität Leipzig zur Verfügung gestellt. Er enthält positiv und negativ gelabelte Wörter. Die Werte mit denen die Wörter gelabelt sind, befinden sich im Intervall [-1;1], wobei -1 negativ und 1 am positivsten ist.
Der SentiWS enthält 1818 negative und 1650 positive Grundformen. Daneben noch weitere andere Formen der Wörter.
Insgesamt bietet der von uns verwendete Wortschatz so 15.632 negative und 15.649 positive Wörter.

[3] *http://wortschatz.uni-leipzig.de/de/download*

Beispiele für Wörter mit positiver Polarität im SentiWS:

```
motivieren|VVINF        0.2248
motiviert|ADJX  0.3541  motiviertest,motivierteres,motiviertestes,motivierterem
mustergültig|ADJX       0.0040  mustergültigem,mustergültigstem,mustergültigsten
mutig|ADJX      0.3435  mutige,mutiges,mutigstem,mutigsten,mutigere
mächtig|ADJX    0.2277  mächtigste,mächtigstem,mächtigerem,mächtiger
mögen|VVINF     0.345
```

Beispiele für negativ gelabelte Wörter im SentiWS:

```
Frechheit|NN    -0.4665 Frechheiten
Frust|NN        -0.469  Frustes,Frusts
Frustration|NN  -0.3354 Frustrationen
Furcht|NN       -0.5012
Fälschung|NN    -0.3377 Fälschungen
```

Die Beispiele machen deutlich wie die die unterschiedlichen Wörter gelabelt sind. ‘Furcht’ ist ein Wort mit einer meist negativen Bedeutung und hat hier im SentiWS einen Wert von -0.5012. Ein positiver Begriff wie ‘liebevoll’ dagegen bekommt einen Wert von 0.4657.

Erhält man bei der Analyse der Artikel der verschiedenen Nachrichtenportale als Output für einen bestimmten Artikel einen hohen Sentiment Score im positiven Bereich, handelt es sich um einen eher positiven Artikel, befindet sich der Wert im negativen Bereich, ist die Stimmung des Artikels eher negativ.

Beispiel:

```
Analysis of sueddeutsche:
Average sentiment score for sueddeutsche article #1 (230 words): -1888.54% Min: -4696.07 (versagt), Max: 293.75 (Alten)
Average sentiment score for sueddeutsche article #2 (88 words): -1809.36% Min: -4464.78 (Rauch), Max: 228.33 (Jahrgang)
Average sentiment score for sueddeutsche article #3 (189 words): -1890.17% Min: -4696.39 (Raser), Max: 593.96 (Felix)
Average sentiment score for sueddeutsche article #4 (143 words): -2423.06% Min: -4759.71 (rassistischen), Max: 317.95 (arbeitet)
Average sentiment score for sueddeutsche article #5 (101 words): -2366.36% Min: -4743.48 (Vertraege), Max: -64.03 (Platz)
```

Berechnung des durchschnittlichen Sentiment Scores

2.5 Sentimentanalyse: zweiter Ansatz

Da die erste Lösung noch nicht ganz zufriedenstellend war, haben wir mit einem weiteren Ansatz versucht, noch bessere Ergebnisse zu erreichen und den Sentiment Score zu verbessern.

Bis jetzt war es schwierig, ein neutrales Ergebnis zu erhalten. Alles was nicht sehr positiv ist, wie zum Beispiel eine Werbeanzeige, wird als negativ gewertet.

Eine Überlegung war für einen neuen Ansatz zunächst wie bei der alten Berechnung erst einmal zu prüfen, ob ein Wort eher positiv oder negativ ist und zusätzlich in einem nächsten Schritt zu versuchen einen guten Schwellenwert zu finden, um wirklich nur stark positive oder stark negative Wörter mit in die Berechnung einzubeziehen mit denen es dann möglich ist, zu überprüfen wie positiv oder negativ ein Text ist. Ein Wert, der einen Artikel als gut klassifiziert würde so also nur mit den positiven Wörtern von SentiWS berechnet werden.

Für die eher neutralen Wörter, die aber trotzdem einen Wert haben der leicht negativ ist, muss auch eine Lösung gefunden werden, da solch ein Wort durch die Berechnungen bei der Analyse dann schließlich auch negativ gewertet wird. Also auch hier wäre die Überlegung, einen Grenzwert zu finden, ab welchem ein negativ gelabeltes Wort erst als negativ gewertet wird und ein neutrales Wort nicht automatisch als negativ gilt. Also war eine Überlegung den neutralen Wörtern vielleicht den Wert 0 zu geben und aus der Bewertung rauszulassen.

Schließlich sieht der zweite Ansatz aber folgendermaßen aus:

Es gibt jetzt einzelne Wörter die als neutral gewertet werden. Artikel werden jedoch immer noch entweder als negativ oder positiv klassifiziert. Einen Grenzwert festzulegen ist inhaltlich sehr schwierig.

Es wird bei diesem Ansatz nun überprüft, ob ein Wort prozentual eine größere Ähnlichkeit zu positiven oder negativen Wörtern aus dem SentiWS besitzt. Sind beide Ähnlichkeiten sehr niedrig und eher unbestimmt, so wird das Wort als neutral eingestuft.

Am Ende wird aus der Differenz der Werte die Gewichtung bestimmt.

```
SentScore für Tod: -1 (negativ) // 65.77% positiv | 90.5% negativ

SentScore für Haus: 0 (neutral) // 74.47% positiv | 86.43% negativ
```

Abbildung: Beispiel für ein negatives und neutrales Wort

Nicht miteinbezogen wird bei diesem Ansatz allerdings die Polarität des Wortes bei SentiWS selbst.

Trotzdem ist jetzt nicht mehr automatisch alles außer Werbung, also alles außer Artikeln mit extrem positiven Wörtern, negativ.

2.6 Realisierung des zweiten Ansatzes

Zuerst einmal mussten die beiden Listen mit den positiven bzw. negativen Wörtern aus SentiWS initialisiert werden. Hierzu wurden die dementsprechenden SentiWS-Dateien zeilenweise gelesen und weiterverarbeitet. Jede Form eines Wortes wird zusammen mit der gegebenen Gewichtung als Liste in der dazugehörigen Liste (positiv/negativ) gespeichert.

```
def init_sent(sentiment, word_list):
    with open('{}{}.txt'.format(sent_path, sentiment), encoding='utf-8') as file:
        sent_words_all = file.readlines()
    for line in sent_words_all:
        word_info = line.split()
        base_word = word_info[0].split('|')[0]  # splits first word into 'word','|','XX' and then accesses word
        base_word = base_word.replace('ß', 'ss').replace('ä', 'ae').replace('ö', 'oe').replace('ü', 'ue')
        sent_value = float(word_info[1])
        word_list.append([base_word, sent_value])
        if len(word_info) > 2: # check if there are word forms other than the base form
            for word in word_info[2].split(','):
                word = word.replace('ß', 'ss').replace('ä', 'ae').replace('ö', 'oe').replace('ü', 'ue')
                word_list.append([word, sent_value])
```

Im Anschluss wird für jedes Nachrichtenportal ein Word2vec-Objekt erstellt und das Modell trainiert, falls dies noch nicht getan wurde.

```
for site in ['faz', 'focus', 'spiegel', 'sueddeutsche', 'tagesschau', 'tonline', 'welt', 'zdf', 'zeit']:
    w2v = word2vec(site)
    if w2v.model is None:
        w2v.get_text_from_files()
        w2v.create_sentence_list()
        w2v.create_model()
```

Nun wurden die Test-Artikel der jeweiligen Seite aus einer Datei gelesen und in einer Liste gespeichert. Für jedes Wort in jedem Artikel wurde nun überprüft, ob es im Vokabular des dazugehörigen Word2vec-Modells enthalten ist und falls dies zutraf, wurde es in einer neuen Liste gespeichert. Diese Liste wurde anschließend für die Analyse verwendet. Bei der Analyse wurde nun für jedes Wort ein Sentiment Score mit Hilfe von SentiWS berechnet. Es wurde also einmal für jedes Wort in der Liste positiver SentiWS-Wörter und der Liste negativer SentiWS-Wörter die Ähnlichkeit zu dem gegebenen Artikelwort mit Hilfe von Word2vecs Ähnlichkeitsfunktion geprüft und je nachdem, ob die Ähnlichkeit positiv oder negativ war, der dazugehörige Zähler inkrementiert.

```
for pos_word in sent_pos:
    similarity = w2v.model.wv.similarity(word, pos_word[0]) * 100
    if similarity > 0:
        pos_pos += 1 # positive similarity to a positive word
    else:
        pos_neg += 1 # negative similarity to a positive word
```

(selbe Schleife für negative Wörter aus SentiWS)

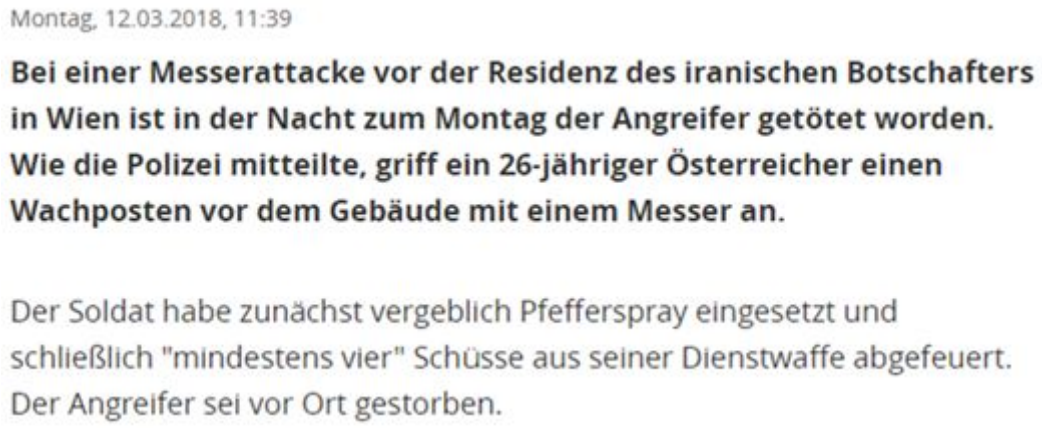
Somit hat beispielsweise das Wort „Tod” eine positive Ähnlichkeit zu 90.5% der negativen SentiWS-Wörter und eine positive Ähnlichkeit zu nur 65.77% der positiven SentiWS-Wörter. Die Differenz aus diesen beiden Ähnlichkeitswerten wird anschließend zur Bestimmung des Sentiment Scores verwendet.
Es ist aufgefallen, dass Artikelwörter, welche sowohl zu den positiven, als auch den negativen Wörtern aus SentiWS im Durchschnitt eine Ähnlichkeit von unter 90% hatten, eher neutrale Wörter waren. Allerdings gab es hier auch Ausnahmen, bei denen die Ähnlichkeit zwar unter 90% lagen, aber die Differenz zwischen positiv und negativer Ähnlichkeit so hoch war, dass sie somit teilweise zu identifizieren waren. Die Schwierigkeiten lagen hier darin, gute Schwellwerte zu finden.
Auf diese Weise wurde somit für jedes Wort in einem Artikel ein Sentiment Score gebildet, welche letztendlich aufsummiert wurden und den Artikel Score darstellten.

3 Auswertung

3.1 Ergebnisse des ersten Ansatzes

Erste Ergebnisse bei der Analyse haben gezeigt, dass die meisten Artikel als eher negativ eingeordnet werden, also einen durchschnittlichen Sentiment Score im negativen Bereich haben. Das ist auch ein sinnvolles Ergebnis, da Nachrichten allgemein eher über negative als über positive Vorkommnisse berichten.

Ein Beispiel für einen Artikel mit einem negativen durchschnittlichen Sentiment Score ist folgender Artikel über einen Messerangriff:



Montag, 12.03.2018, 11:39

Bei einer Messerattacke vor der Residenz des iranischen Botschafters in Wien ist in der Nacht zum Montag der Angreifer getötet worden. Wie die Polizei mitteilte, griff ein 26-jähriger Österreicher einen Wachposten vor dem Gebäude mit einem Messer an.

Der Soldat habe zunächst vergeblich Pfefferspray eingesetzt und schließlich "mindestens vier" Schüsse aus seiner Dienstwaffe abgefeuert. Der Angreifer sei vor Ort gestorben.

Quelle: focus.de

Der durchschnittliche Sentiment Score dieses Artikels liegt bei -1911,82. Ausschlaggebende Wörter die diesen beeinflussen sind hier beispielsweise „Angreifer“, „getötet“ oder „Schüsse“.

Hier hat die Einordnung also funktioniert.
Der Artikel wird zurecht als negativ gewertet.

Eindeutig positiv waren von Anfang an Werbeanzeigen, die auffällig viele sehr positive Wörter verwenden.
Ein Beispiel dafür ist dieser Artikel, eine Werbung von Media Markt:

Donnerstag, 15.03.2018, 13:32

Hochwertige PCs und Tablets, interessante Spiele oder neue Kaffeevollautomaten - Das Angebot von Media Markt ist vielfältig. Entdecken Sie im aktuellen Media Markt-Katalog die Top-Deals der Woche.

Im aktuellen Media Markt-Prospekt finden Sie die besten Angebote der Woche. Media Markt bietet ein interessantes und vielfältiges Sortiment an innovativen Produkten, bei dem sowohl Anfänger als auch Technikspezialisten fündig werden. Nicht umsonst gehört Media Markt zu den erfolgreichsten Elektrofachmärkten in ganz Europa.

Quelle: focus.de

Bei diesem Werbeartikel ist der durchschnittliche Sentiment 999,12. Beeinflusst wird diese Wertung durch Wörter mit starker positiver Polarität wie „hochwertige“, „interessante“ oder „bester“.

Auch in diesem Fall war das Ergebnis der Analyse zufriedenstellend.
Jedoch ist eine Einstufung eines Artikels als neutral noch nicht gut möglich.

Aufgefallen ist des Weiteren, dass der durchschnittliche Sentiment Score zwischen den einzelnen Nachrichtenportalen variiert.
Als Beispiel ein Vergleich zwischen FAZ und Focus:

```
Analysis of faz:
Average sentiment score for faz article #1 (119 words): -2413.21%
Average sentiment score for faz article #2 (53 words): -2240.44%
Average sentiment score for faz article #3 (106 words): -2454.93%
Average sentiment score for faz article #4 (95 words): -2486.43%
```

Abbildung: Auswertung von FAZ-Artikeln

```
Analysis of focus:
Average sentiment score for focus article #1 (201 words): -972.95%
Average sentiment score for focus article #2 (34 words): -336.93%
Average sentiment score for focus article #3 (29 words): -291.58%
Average sentiment score for focus article #4 (60 words): -1166.26%
```

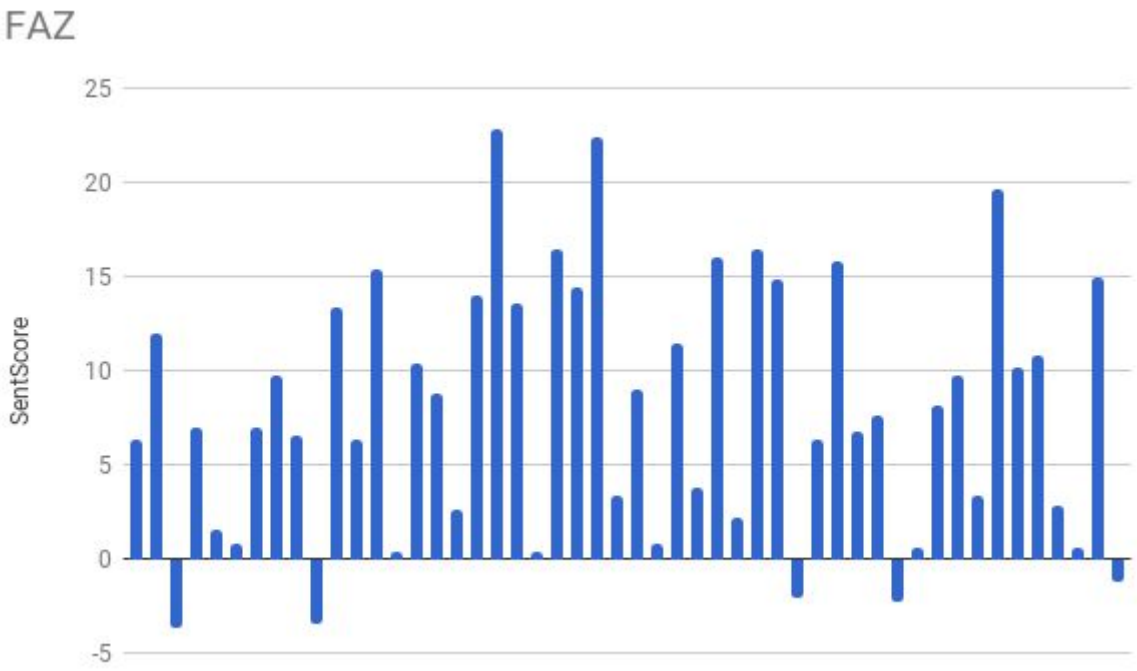
Abbildung: Auswertung von Focus-Artikeln

Man kann hier erkennen, dass der Score bei den Artikeln der FAZ negativere Werte annimmt als bei den Artikeln von focus.de.
Dies kann ein erster Hinweis darauf sein, dass die Artikel in der FAZ allgemein negativer formuliert sind als Focus-Artikel oder zumindest Wörter verwendet werden, die einen negativeren Sentimentwert besitzen.

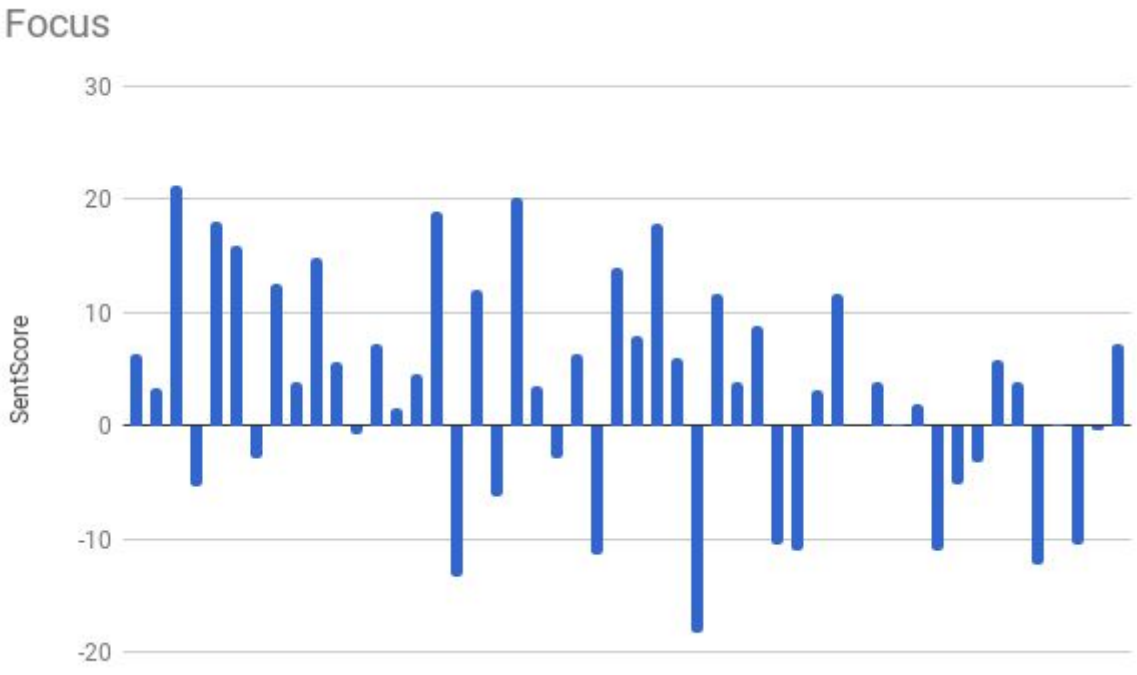
3.2 Ergebnisse des zweiten Ansatzes

Mit der neuen Methode wurde nun ebenfalls die Sentimentanalyse durchgeführt und die Artikel analysiert.
Zum einen wurden die Artikel eines Nachrichtenportale in das entsprechende Word2vec-Modell gegeben.
Ein weiteres Modell (Gesamtmodell) wurde mit Artikeln aller Nachrichtenseiten trainiert, um einen Vergleich zu haben.

Die folgenden Diagramme zeigen die Sentiment Scores von 50 Artikeln der jeweiligen Seiten und ihren Word2vec-Modellen, wobei das Diagramm Focus (Gesamtmodell) die Sentiment Scores von 50 Focus-Artikeln basierend auf dem oben genannten Word2vec Gesamtmodell zeigt.

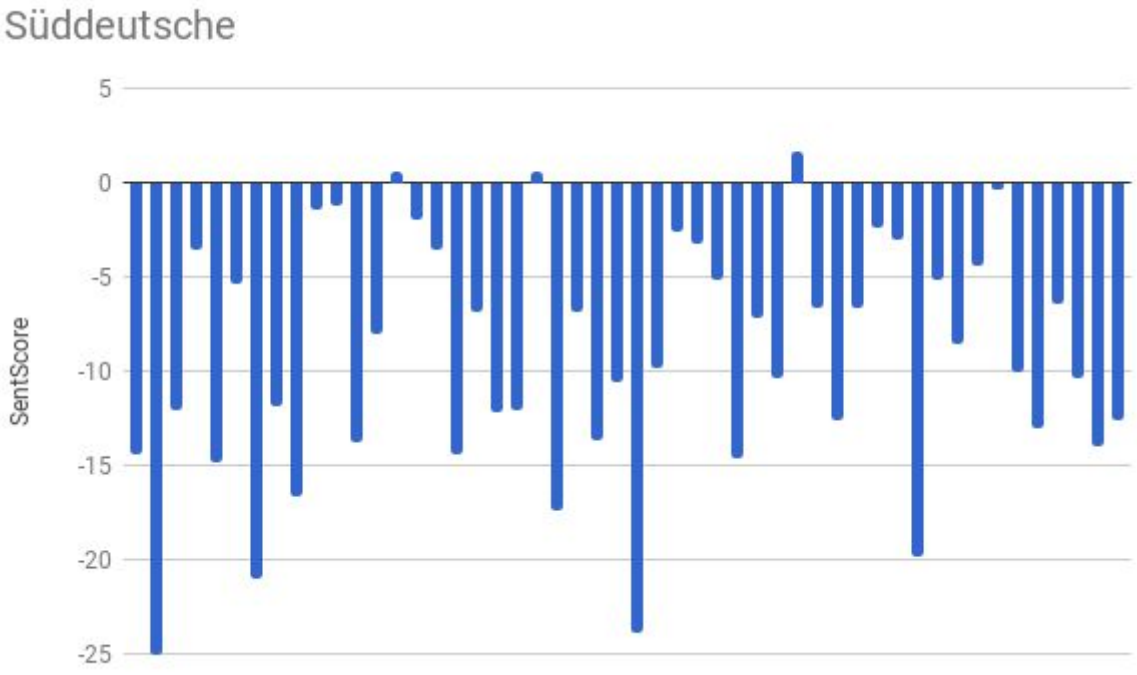


Minimaler Sentiment Score: -3,6, Maximaler Sentiment Score: 22,8
positive Artikel: 45, negative Artikel: 5, neutrale Artikel: 0

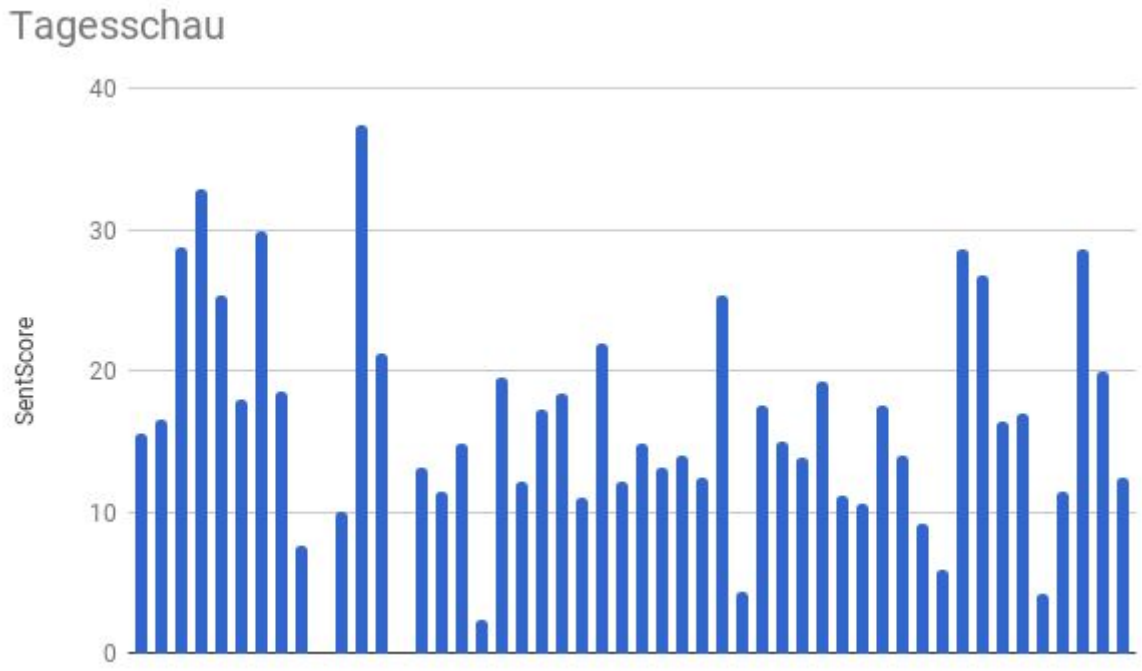


Minimaler Sentiment Score: -18,2, Maximaler Sentiment Score: 21,2
positive Artikel: 33, negative Artikel: 16, neutrale Artikel: 1

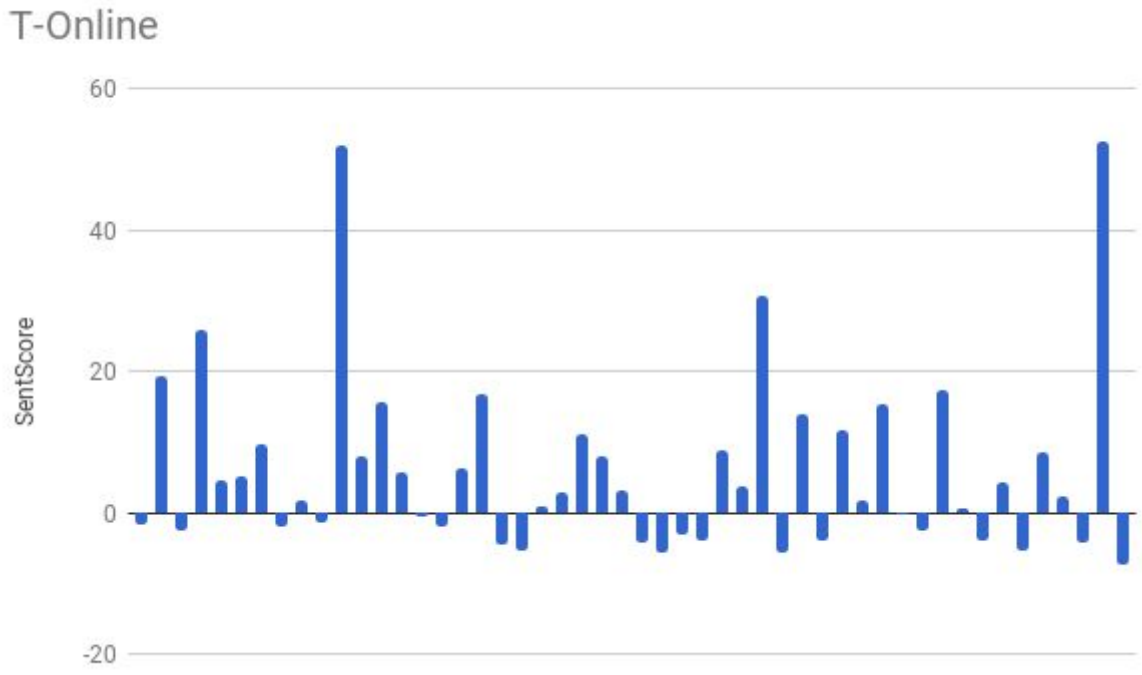
Minimaler Sentiment Score: -80,8, Maximaler Sentiment Score: 14
positive Artikel: 18, negative Artikel: 32, neutrale Artikel: 0



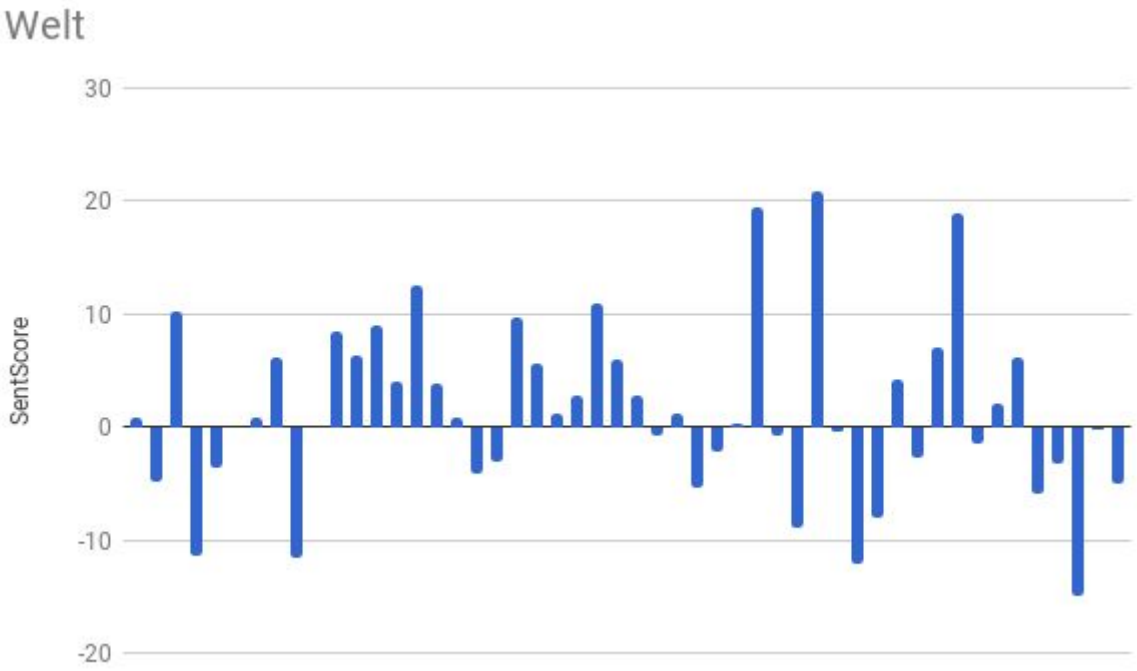
Minimaler Sentiment Score: -25, Maximaler Sentiment Score: 1,6
positive Artikel: 3, negative Artikel: 47



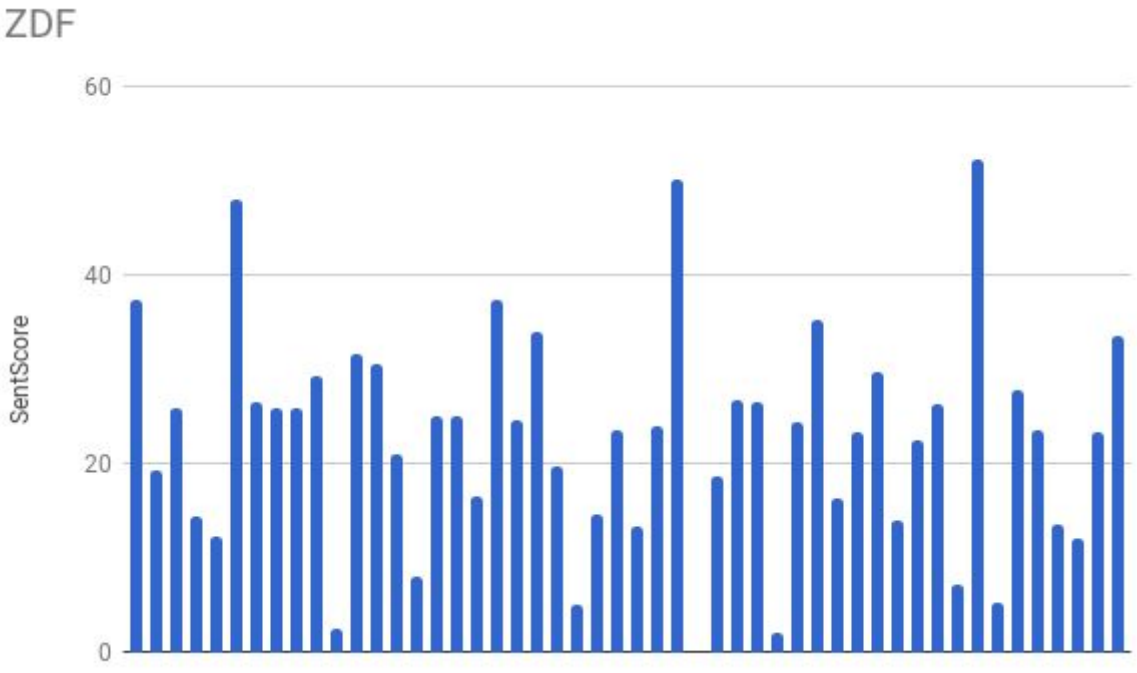
Minimaler Sentiment Score: 0, Maximaler Sentiment Score: 37,4
positive Artikel: 48, negative Artikel: 0, neutrale Artikel: 2



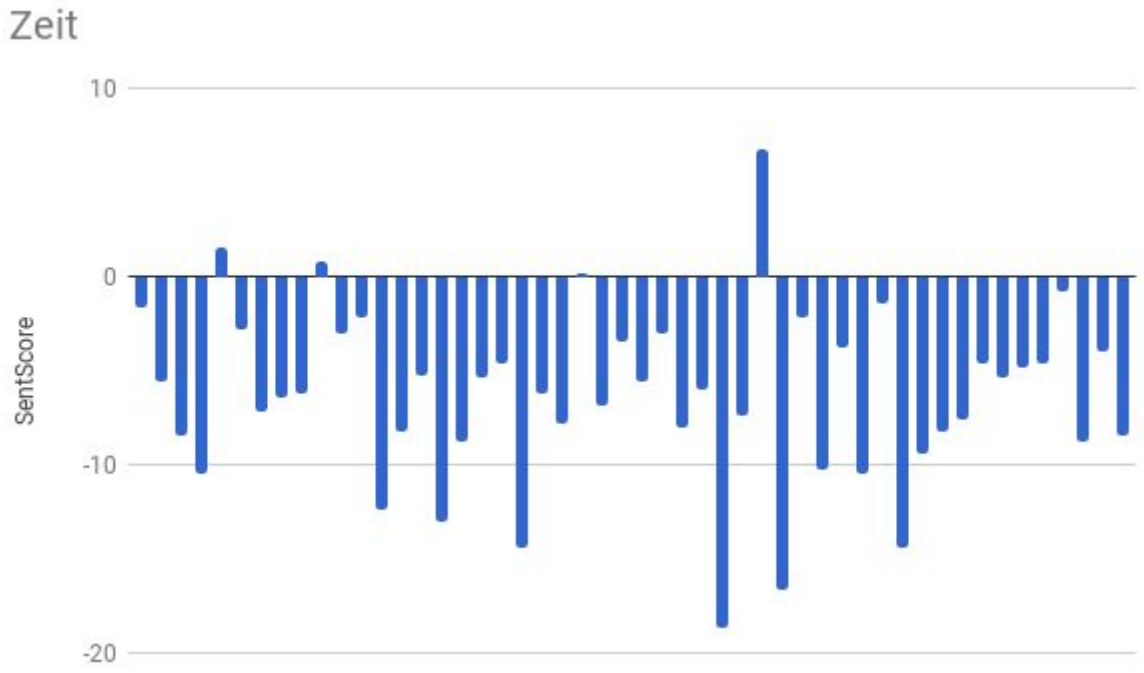
Minimaler Sentiment Score: -7,4, Maximaler Sentiment Score: 52,4
positive Artikel: 30, negative Artikel: 20, neutrale Artikel: 0



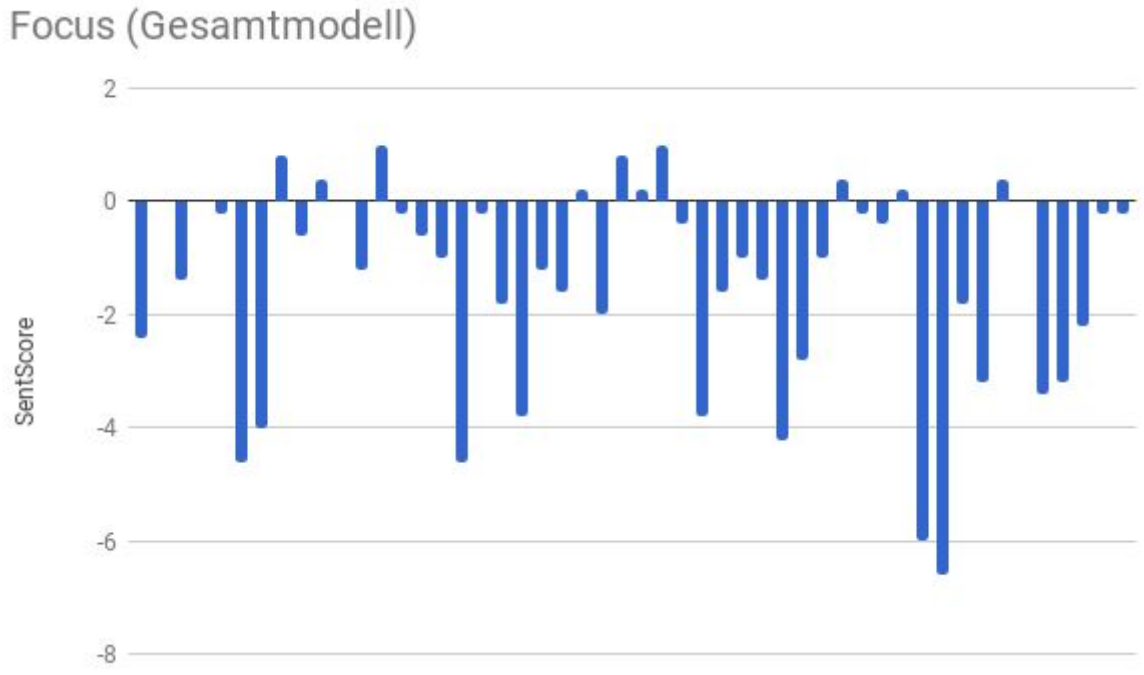
Minimaler Sentiment Score: -14,8, Maximaler Sentiment Score: 20,8
positive Artikel: 27, negative Artikel: 21, neutrale Artikel: 2



Minimaler Sentiment Score: 0, Maximaler Sentiment Score: 52,2
positive Artikel: 49, negative Artikel: 0, neutrale Artikel: 1



Minimaler Sentiment Score: -18,6, Maximaler Sentiment Score: 6,8
positive Artikel: 4, negative Artikel: 46, neutrale Artikel: 0



Minimaler Sentiment Score: -6,6, Maximaler Sentiment Score: 1
positive Artikel: 10, negative Artikel: 36, neutrale Artikel: 4

Minimaler Sentiment Score: -8,1, Maximaler Sentiment Score: 18,5
positiv: FAZ, Focus, Tagesschau, T-Online, Welt, ZDF
negativ: Spiegel, Süddeutsche, Zeit, Focus (Gesamt)

Die Analyse zeigt, dass die Ergebnisse sehr unterschiedlich ausfallen.
Für die meisten Nachrichtenportale wird die Mehrzahl der Artikel als positiv bewertet. Dies ist der Fall bei der FAZ, Focus, Tagesschau, T-Online, Welt und ZDF.
Mehrheitlich negative Artikel wurden bei Spiegel, Süddeutsche, Zeit und Focus (Gesamt) festgestellt.

Auch bei den Werten des Sentiment Scores der verschiedenen Nachrichtenseiten sind Unterschiede zu erkennen. Sehr extreme Werte sind beispielsweise der besonders hohe maximale Score bei der Analyse von T-Online oder der mit Abstand niedrigste Score von -80,8 beim Spiegel.

Besonders auffällig bei diesen Ergebnissen ist, dass bei ZDF und Tagesschau keiner der analysierten Artikel als negativ eingeordnet wurde.
Gleichzeitig sind dies die beiden Nachrichtenseiten von denen im Vergleich zu den anderen Seiten mit Abstand die wenigsten Daten vorhanden waren. Nach der Extraktion lagen für die Tagesschau-Artikel 5.6 MB und für ZDF 4,3 MB vor. Eventuell besteht darin ein Zusammenhang.

Hier einige Beispielartikel:

Ihr steht auf Hip Hop oder Rap und wollt die Bühne mal ganz für euch haben? Dann habt ihr im April die Chance, das Publikum von euch zu überzeugen! Bevor es im Sommer beim **Juicy Beats Festival** so richtig heiß hergeht, gibt es am Donnerstag, 19. April eine eigene Bühne für Nachwuchs-Acts im FZW: "Bring Your Own Beats" heißt die Initiative des Dortmunder Jugendamts und UPop e.V..

Juicy Beats-Tickets zu gewinnen

Zuschauen lohnt sich: Mit am Start sind Rapper **Sylabil Spill, P. Hightower** und **Lyrico** als Live-Acts! Gastgeber und **DJ Max Gyver** sorgt zwischen den Gigs für die richtige Stimmung. Zu gewinnen gibt's auch noch was: Unter allen Zuschauern werden zwei Tickets für das Juicy Beats Festival im Sommer verlost!

Sentiment Score: 14
Vorhersage: positiv
tatsächliches Sentiment: positiv

Zwei Unbekannte haben am Samstagabend einen 33-Jährigen in Garbsen mit einem Messer angegriffen und lebensgefährlich verletzt. Der Mann musste noch am Abend notoperiert werden. Sein Zustand ist derzeit stabil, er schwebt jedoch noch immer in Lebensgefahr.

Nach Informationen der "Hannoverschen Allgemeinen" hatte sich der 33-Jährige gegen 18:45 Uhr an der Haltestelle Planetenring der Stadtbahn 4 aufgehalten, als er von den beiden südosteuropäisch aussehenden Tätern unvermittelt angegriffen wurde. Die Männer schlugen und stachen auf ihr Opfer ein, verletzten es im Gesicht und am Oberkörper. Dann flüchteten sie zu Fuß über den Planetenring in Richtung Siriushof.

Passanten eilten dem Verletzten zur Hilfe und alarmierten umgehend **Polizei** und Rettungsdienst. Der Mann wurde schließlich zur Not-OP ins Krankenhaus gebracht. Eine erste Fahndung nach den Messerstechern verlief erfolglos. Die Polizei bittet daher die Bevölkerung um Mithilfe.

Sentiment Score: -18,2
Vorhersage: negativ
tatsächliches Sentiment: negativ

Schon seit langem ist vom horrenden Lehrermangel im Land die Rede, dabei ist auch klar: Es mangelt nicht nur an Lehrkräften, sondern auch an Erziehern.

Und es ist damit zu rechnen, dass sich das Problem noch verschärft, auch in Halle. Denn die Stadt will in den kommenden Jahren mehr Kita-Plätze schaffen - ob sie dafür aber auch genügend Personal findet, ist unklar.

Insgesamt fehlen der Stadt in diesem Jahr 1.059 Betreuungsplätze. Das geht aus der Bevölkerungsprognose 2018 hervor. Demnach werden in diesem Jahr 7.762 Kinder von drei Jahren bis zum Schuleintritt einen Kita-Platz benötigen - geplant hatte die Stadt aber mit nur 7.374 Kindern. Geht man davon aus, dass 95 Prozent dieser Kinder in die Kita gehen, fehlen in diesem Altersbereich 323 Plätze. Analog zu dieser Berechnung fehlen 466 Krippen- und 270 Hortplätze.

Sentiment Score: 12
Vorhersage: positiv
tatsächliches Sentiment: negativ

Die Beispiele zeigen drei Artikel, von denen bei der Sentimentanalyse einer als negativ und zwei als positiv eingeordnet wurden. Die Auswertung der Ergebnisse hat gezeigt, dass deutlich positive und negative Artikel erkannt werden. Jedoch hängt die richtige Kategorisierung von den verwendeten Wörtern und deren Häufigkeit ab.
Der eigentliche Inhalt wird dabei nicht mit einbezogen und nicht als Kontext selber als negativ oder positiv klassifiziert. Ein Artikel der viele als negativ gelabelte Wörter enthält, wie beispielsweise Artikel über Terrorattacken oder Messerangriffe, erhält einen negativen Score. Der Beispielartikel über den Messerangriff wurde mit einem Score von -18,2 daher korrekterweise als negativ klassifiziert. Der erste Artikel, der über ein kommendes Festival berichtet, erhält einen Sentiment Score von 14. Auch hier hat die Einordnung korrekt funktioniert. Werden dagegen eher neutrale Wörter verwendet, um über etwas zu berichten, ist der Score trotz des nicht positiven Inhalts im unteren positiven Bereich. Wie in dem dritten Artikel. In diesem geht es um Lehrer- und Erziehermangel. Das Thema und die Stimmung des Artikels sind negativ. Trotzdem liegt der Sentiment Score bei 12. Es werden wahrscheinlich zu wenige eindeutig als negativ klassifizierte Wörter verwendet. Einen Grund sehen wir darin, dass keine gelabelten deutsch Artikel genutzt werden konnten, die einen gesamten Kontext als negativ oder positiv klassifizieren, sondern wir schließlich den SentiWS verwendet haben, der Polaritäten für einzelne Wörter festlegt.

3.3 Probleme

Für die Sentimentanalyse benötigt man gelabelte Daten mit denen das Word2vec Modell trainieren kann. Allerdings müssen diese in Deutsch sein, da wir uns für deutsche Nachrichtenportale und Artikel entschieden haben.
Hier war es schwierig etwas Passendes in deutscher Sprache zu finden. Am besten wäre es gewesen, ein Dataset mit gelabelten Nachrichtenartikeln zu finden, um dann einen genau passenden Vergleich zu haben.
Wir haben uns schließlich für SentiWS entschieden.
Dieser Sentiment Wortschatz ist eventuell aber nicht optimal, da es sich um einzelne Wörter und eben nicht um zusammenhängende Texte wie bei den gecrawlten Artikeln handelt.

Der Kontext des Wortes muss mit in die Bewertung einbezogen werden. Aktuell werden noch die Wörter einzeln bewertet.

Bei den Word2vec-Modellen gibt es verschiedene Parameter, die verändert werden können. So kann man versuchen ein besseres Ergebnis zu erhalten. Die optimalen Parameter zu finden ist schwierig.
Ein ähnliches Problem bestand bei der Berechnung des Sentiment Scores, da es sehr schwer ist einen möglichst optimalen Schwellwert für die Einordnung manuell zu finden.

Auch bei dem zweiten, verbesserten Ansatz ist es jedoch immer noch so, dass es sich eher um eine Bewertung der einzelnen Wörter handelt. Der Artikel wird nicht als Ganzes bewertet, sondern die Klassifizierung ergibt sich aus der Häufigkeit der einzelnen Wörter.

Nicht als positiv erkannt werden würde aber beispielsweise eine Formulierung wie „nicht schlecht“, da diese in der Sentimentanalyse durch das Wort „schlecht“ als negativ erkannt wird, obwohl der Kontext positiv ist.
In diesem Fall würde dann aber vermutlich anhand der Anzahl der anderen negativen oder positiven Wörter in einem Artikel eine korrekte Kategorisierung vorgenommen.

4 Zusammenfassung und Fazit

Die Aufgabe war es, eine Sentimentanalyse von Nachrichtenartikeln mit Hilfe von Word2vec durchzuführen.

Dabei war es das Ziel, folgendes herauszufinden:

- Wie ist es möglich, mit Word2vec eine Sentimentanalyse durchzuführen?
- Inwiefern ist eine Sentimentanalyse mit Word2vec bei News-Artikeln möglich?
- Was lässt sich erkennen?
- Sind Unterschiede bei bestimmten Themen oder den unterschiedlichen News Portalen erkennbar?

Dafür haben wir zunächst mit dem zur Verfügung gestellten Crawler Daten gesammelt. Dies ergab ein Dataset mit einer Größe von 108 GB. Nach der Extraktion erhielten wir einen Reintext mit einer Größe von 158 MB.
Anschließend wurden Word2vec-Modelle trainiert und zwei verschiedene Ansätze der Sentimentanalyse weiter verfolgt und die Ergebnisse analysiert.

Der erste Ansatz lieferte schon erste gute Ergebnisse.
Besonders positiv geschriebene Artikel wie Werbeartikel wurden zuverlässig als positiv identifiziert.
Alles anderen Artikeln waren durchgängig negativ. Allerdings mit verschieden hohen Werten des Sentiment Scores.
Neutrale Artikel zu identifizieren ist schwierig.

Da die Ergebnisse dieser ersten Lösung noch verbessert werden sollten, haben wir mit einem weiteren Ansatz versucht, dies zu erreichen und den Sentiment Score zu verbessern.
Es wird bei diesem Ansatz nun überprüft, ob ein Wort prozentual eine größere Ähnlichkeit zu positiven oder negativen Wörtern aus dem SentiWS besitzt. Sind beide Ähnlichkeiten sehr niedrig und eher unbestimmt, so wird das Wort als neutral eingestuft. Am Ende wird aus der Differenz der Werte die Gewichtung bestimmt.

Die Ergebnisse dieses neuen Ansatzes liefern korrekte Werte für Artikel, welche positive oder negative Schlagwörter in einer ausreichend hohen Anzahl beinhalten.
Eine immer korrekte Einbeziehung des Kontextes wurde auch hier noch nicht erreicht und ist schwierig. Vermutlich da wir keine gelabelten Nachrichtenartikel in deutscher Sprache zur Verfügung hatten und deshalb auf den SentiWS zurückgegriffen haben.

Generell ist eine Sentimentanalyse mit Word2vec also möglich. Die hier vorgestellten Ansätze liefern viele korrekte Ergebnisse, gerade bei der Identifizierung eindeutig positiver Artikel.
Auch, wenn Nachrichtenartikel häufig eher neutral gehalten sind und die Meinung der Autoren selten mit einbezogen wird, haben die Ergebnisse gezeigt, dass auch diese generell für eine Sentimentanalyse geeignet sind.
Für eine vollständige Einbeziehung des Kontextes wäre es aber im Nachhinein betrachtet, vorteilhafter und einfacher gewesen, wenn englische Nachrichtenseiten verwendet worden wären.
Aufgrund der deutschen Texte, wäre auch ein Set mit gelabelten deutschsprachigen Artikeln nötig gewesen, um ein optimales Ergebnis zu erzielen.
Aber auch mit der Verwendung eines deutschen Wortschatzes (SentiWS), konnten richtige Analysen durchgeführt werden.
Die Berechnung des Sentiment Scores stellt dafür eine gute Möglichkeit dar, die eine gute Grundlage für weitere Verbesserungen liefert und mit anderen Artikeln, vorzugsweise in englischer Sprache oder mit einem entsprechenden deutschen gelabelten Dataset in Zukunft die Sentimentanalyse noch deutlich vereinfachen kann und bessere Ergebnisse möglich macht.

5 Quellen

Word2vec:

- https://www.tensorflow.org/tutorials/word2vec
- https://radimrehurek.com/gensim/models/word2vec.html
- https://en.wikipedia.org/wiki/Word2vec
- https://en.wikipedia.org/wiki/Word_embedding

Sentimentanalyse:

- https://de.wikipedia.org/wiki/Sentiment_Detection
- http://www.enzyklopaedie-der-wirtschaftsinformatik.de/lexikon/daten-wissen/Business-Intelligence/Analytische-Informationssysteme--Methoden-der-/sentimentanalyse/sentimentanalyse

Beautiful Soup 4:

- https://www.crummy.com/software/BeautifulSoup/bs4/doc/

SentiWS:

- wortschatz.informatik.uni-leipzig.de/de/download

Bildquellen:

- https://www.focus.de/panorama/welt/wien-wachsoldat-erschiesst-messerangreifer-vor-irans-botschafterresidenz_id_8596180.html
- https://www.focus.de/shopping/prospekte/der-neue-media-markt-prospekt-die-top-angebote-der-kw-11-2018_id_7424451.html
- https://www.focus.de/regional/dortmund/dortmund-dieser-contest-macht-bock-aufs-juicy-beats_id_8593059.html

- https://www.focus.de/regional/niedersachsen/garbsen-mann-33-bei-ueberfall-an-haltestelle-lebensgefaehrlich-verletzt-polizei-jagt-messerstecher-duo_id_8593012.html

- https://www.focus.de/regional/sachsen-anhalt/halle-saale-neue-kitas-fuer-halle-aber-wo-sollen-mehr-erzieher-herkommen_id_8592858.html

6 Anhang

verwendete Software:

- Website-Crawler (Crawly) von Max Lübbering, https://github.com/le1nux/crawly
- PyCharm Community Edition 2017.2, Jetbrains, https://www.jetbrains.com/pycharm/

verwendete (nicht-standard) Python-Libraries:

- nltk, NLTK, https://www.nltk.org/
- gensim.word2vec, Radim Rehurek, https://radimrehurek.com/gensim/models/word2vec.html
- Beautiful Soup 4, Leonard Richardson, https://www.crummy.com/software/BeautifulSoup/

verwendete Hardware:

- Cluster des DKRZ Hamburg