

# Sprawozdanie projektu: Analiza tweetów dotyczących filmu „Avatar”

Anna Ledworowska

nr indeksu: 266376

# 1. Wstęp

Celem projektu była analiza i porównanie tweetów dotyczących pierwszej i drugiej części filmu „Avatar” reżyserii Jamesa Camerona.

## 1.1. Twitter

Twitter jest to serwis społecznościowy założony w 2006 roku, który daje użytkownikom możliwość publikowania krótkich wpisów nazywanych tweetami. Twitterzy dzielą się tam swoimi przemyśleniami na dowolne tematy, również na te związane z rozrywką.

## 1.2. „Avatar”

Pierwszy film o tytule „Avatar” miał premierę w 2009, a na kolejny fani musieli czekać, aż 13 lat. Obie części zostały stworzone przy użyciu zaawansowanych technologii, przez co wzbudzają duże zainteresowanie widzów. Aktywni użytkownicy Twittera komentowali swoje spostrzeżenia na jego temat.

## 1.3. Użyte narzędzia oraz założenia projektu

Tweety zostały pozyskane za pomocą snsrape. To wpisy w języku polskim oraz angielskim umieszczone miesiąc przed i miesiąc po premierze filmu zarówno dotyczące pierwszej części filmu oraz drugiej. Porównywane zostaną statystyczne własności tweetów. Za pomocą vaderSentiment oraz textblob sprawdzone zostanie jak zmieniał się sentyment wpisów przed i po premierze. Cały projekt został napisany w języku Python w Jupyter Notebook.

### 1.3.1. Potrzebne paczki

Do stworzenia projektu potrzebne było zaimportowanie poniższych paczek.

```
from snsrape.modules.twitter import TwitterSearchScraper
import pandas as pd
from datetime import timedelta, datetime
import os
from os.path import exists
import numpy as np
from textblob import TextBlob
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
import pandas as pd
from googletrans import Translator
import matplotlib.pyplot as plt
from wordcloud import WordCloud, ImageColorGenerator
from stop_words import get_stop_words
from PIL import Image
import numpy as np
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.stem import WordNetLemmatizer
from nltk.probability import FreqDist
from nltk.corpus import wordnet as wn
```

### 1.3.2. Pozyskiwanie tweetów

By pozyskać tweety potrzebne do analizy została stworzona funkcja *twitterScrapper*, w której za pomocą *TwitterSearchScrapper* wyszukiwane są tweety:

- związane z „avatar”,
- napisane w przedziale danych dat,
- wybranym języku.

Ilość pobranych tweetów ograniczona jest przez podanie maksymalnej ilości wpisów, którą funkcja ma zbierać. Funkcja upewnia się czy w tweetcie jest wzmianka o avatarze.

Zbierane informacje z każdego tweeta to:

- czas jego udostępnienia,
- treść,
- użyte hasztagi,
- nazwa użytkownika,
- lokalizacja osoby piszącej,
- język w którym został napisany.

Następnie zebrane dane są zapisywane w pliku csv. Funkcja zwraca ilość zebranych tweetów.

```
def twitterScrapper(jezyk,od,do,maxTweets,tytul_csv,sumaTweetow):
    tweets_list = []

    for i,tweet in enumerate(TwitterSearchScrapper('Avatar OR avatar since:'+od+' until:'+do+' lang:'+jezyk+'').get_items()):
        if i>maxTweets:
            break
        if "avatar" in tweet.rawContent or "Avatar" in tweet.rawContent:
            sumaTweetow=sumaTweetow+1
            tweets_list.append([tweet.date, tweet.rawContent, tweet.hashtags, tweet.user.username, tweet.user.location, tweet.lang])

    tweets_df = pd.DataFrame(tweets_list, columns=['Datetime', 'Text', 'Hashtags', 'Username', 'Location', 'Language'])

    if(exists(tytul_csv)==0):
        tweets_df.to_csv(tytul_csv, mode='a',sep=',', index=False)
    else:
        tweets_df.to_csv(tytul_csv, mode='a',sep=',', index=False, header=False)

    return sumaTweetow
```

Kolejna funkcja *twitterScrapperIterator* została stworzona by wywoływać *twitterScrapper* dla kolejnych dni, tak by dane były zbierane z każdego dnia w zadanym przedziale dat.

```
def twitterScrapperIterator(odData,doData,jezyk,maxTweets,tytul_csv):
    czas_start = datetime.now()
    sumaTweetow = 0

    while odData != doData:
        sumaTweetow = twitterScrapper(jezyk,str(odData),str(odData+timedelta(days=1)),maxTweets,tytul_csv,sumaTweetow)
        odData += timedelta(days=1)

    czas_stop = datetime.now()
    print('Czas potrzebny na zebranie tweetow: {}'.format(czas_stop - czas_start))
    print("Liczba zebranych tweetow: "+str(sumaTweetow))
```

Poniżej zaprezentowane jest wywołanie funkcji tak by uzyskać tweety:

- o pierwszej części w języku polskim, miesiąc przed i po premierze ( premiera: 25.12.2009 )
- o pierwszej części w języku angielskim, miesiąc przed i po premierze ( premiera: 10.12.2009 )
- o drugiej części w języku polskim, miesiąc przed i po premierze ( premiera: 16.12.2022 )
- o drugiej części w języku angielskim, miesiąc przed i po premierze ( premiera: 14.12.2022 ).

Nałożono ograniczenie 250 tweetów na dzień.

```
print("Czesc pierwsza:")
print("\ntweety w jezyku polskim:")
twitterScraperIterator(datetime(2009,11,25,0,0,0),datetime(2010,1,25,0,0,0),"pl",250,"pierwsza_czesc_pl.csv")
print("\ntweety w jezyku angielskim:")
twitterScraperIterator(datetime(2009,11,10,0,0,0),datetime(2010,1,10,0,0,0),"en",250,"pierwsza_czesc_en.csv")
print("\n\nCzesc druga:")
print("\ntweety w jezyku polskim:")
twitterScraperIterator(datetime(2022,11,16,0,0,0),datetime(2023,1,16,0,0,0),"pl",250,"druga_czesc_pl.csv")
print("\ntweety w jezyku angielskim:")
twitterScraperIterator(datetime(2022,11,14,0,0,0),datetime(2023,1,14,0,0,0),"en",250,"druga_czesc_en.csv")
```

Następnie ukazany jest wydruk z konsoli, gdzie zostało zaprezentowane ile tweetów udało się zebrać i ile czasu to zajęło.

Czesc pierwsza:

tweety w jezyku polskim:  
Czas potrzebny na zebranie tweetow: 0:01:46.752078  
Liczba zebranych tweetow: 1007

tweety w jezyku angielskim:  
Czas potrzebny na zebranie tweetow: 0:09:42.886665  
Liczba zebranych tweetow: 14731

Czesc druga:

tweety w jezyku polskim:  
Czas potrzebny na zebranie tweetow: 0:03:35.684186  
Liczba zebranych tweetow: 3227

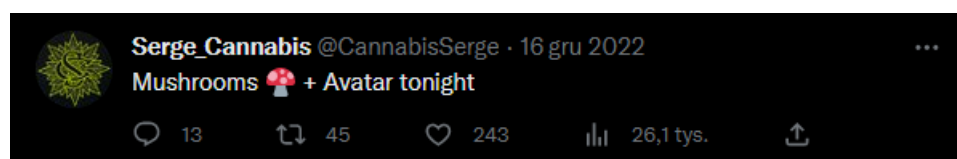
tweety w jezyku angielskim:  
Czas potrzebny na zebranie tweetow: 0:11:25.664401  
Liczba zebranych tweetow: 13782

Po mimo możliwości zebrania dziennie 250 tweetów nie każdego dnia udało się tyle zebrać. Widocznie nie zostało umieszczonych tyle wpisów na badany temat. Nie jest zaskakujące fakt, że tweetów w języku angielskim jest zdecydowanie więcej niż w polskim. Chociaż widać wzrost liczby wpisów w języku polskim, co może sugerować większą popularność Twittera w 2022 wśród polskich użytkowników niż to było w 2009.

Poniżej kilka przykładowych ze zebranych tweetów oraz jeden z nich wyszukany na Twitterze.

```
df_avatar_2en[8000:8005]
```

	Datetime	Text	Hashtags	Username	Location	Language
8000	2022-12-16 23:53:16+00:00	I finally watching avatar	NaN	xobriiii1	NaN	en
8001	2022-12-16 23:53:16+00:00	@AAD101192 @awk_ashleigh Slap "in your opinion...	NaN	lowkee120	The In-Between	en
8002	2022-12-16 23:53:16+00:00	Mushrooms 🍄 + Avatar tonight	NaN	CannabisSerge	Los Angeles, CA	en
8003	2022-12-16 23:53:15+00:00	avatar 2 masterclass	NaN	ello_gl	Grenoble, France	en
8004	2022-12-16 23:53:15+00:00	It's actually so depressing, no one I know is ...	NaN	huntzworld	NaN	en



## 2. Część praktyczna

### 2.1. WordNet

Przed analizą tweetów sprawdzono czy słowo „avatar” istnieje w WordNecie – leksykalnej bazie danych języka angielskiego.

```
avatar = wn.synsets('avatar', pos=wn.NOUN)
print(avatar[0])

print(avatar[0].name())
print(avatar[0].definition())

print(avatar[1].name())
print(avatar[1].definition())
```

```
Synset('embodiment.n.01')
embodiment.n.01
a new personification of a familiar idea
avatar.n.02
the manifestation of a Hindu deity (especially Vishnu) in human or superhuman or animal form
```

„Avatar” znajduje się w WordNecie, lecz pod znaczeniami: wcielenie i hinduskie bóstwo. Są one różne od znaczenia słowa „avatar”, który jest tematem projektu.

## 2.2. Analiza długości tweetów

Funkcja `srednia_dlugosc_tweetow` sprawdza ile średnio znaków miały tweety związane z „Avatarem”.

```
def srednia_dlugosc_tweetow(df_avatar):  
    srednia = 0  
    for index, row in df_avatar.iterrows():  
        srednia = srednia+len(row['Text'])  
  
    srednia = srednia/len(df_avatar)  
    return srednia
```

Poniżej zaprezentowany jest wynik wywołania funkcji dla zebranych tweetów.

Średnia długość tweetów (ilość znaków):

po polsku o I części:       76.69215491559086  
po angielsku o I części:   89.6967619306225

po polsku o II części:       108.06321660985435  
po angielsku o II części:   129.10375852561313

Tweety o drugiej części są zdecydowanie dłuższe, zarówno w języku polskim jak i angielskim. Może to świadczyć o tym, że kiedyś użytkownicy Twittera pisali krótsze tweety niż teraz lub rzeczywiście druga część wzbudziła większe emocje i sprawiła, że użytkownicy mieli więcej do napisania na jej temat.

## 2.3. Występowanie wzmianki o technologii 3D

Filmy 3D w 2009 roku były czymś nowym i niezwykłym dla wielu widzów. Wydaje się, że w dzisiejszych czasach ludzie przywykli do tej technologii i nie robi to już takiego dużego wrażenia na odbiorcach. Dlatego sprawdzono czy w tweetach jest różnica w ilości wzmianek o 3D.

Funkcja `szukanie_3D` wyszukuje w każdym tweetcie frazy „3D” i zlicza ilość jej występowania.

```
def szukanie_3D(df_avatar):  
    licznik=0  
    for index, row in df_avatar.iterrows():  
        licznik+=row['Text'].lower().count('3d')  
    return([licznik,round(licznik/df_avatar.shape[0]*100,2),df_avatar.shape[0]])
```

Poniżej wynik wywołania funkcji dla zebranych tweetów.

Ilość wzmianek o technologii 3D w tweetach:

po polsku w I części:	115	jest to 11.42% wszystkich tweetow (1008)
po angielsku w I części:	1325	jest to 8.99% wszystkich tweetow (14731)

po polsku w II części:	117	jest to 3.63% wszystkich tweetow (3227)
po angielsku w II części:	396	jest to 2.87% wszystkich tweetow (13782)

W drugiej części filmu ilość wzmianek o technologii 3D jest mniejsza. Różnicę dobrze widać w tweetach w języku angielskim. W pierwszym i drugim filmie zebrano zbliżoną liczbę tweetów w tym języku, a ilość wzmianek w części z 2022 roku jest o około 1 tysiąc mniejsza. W polskich tweetach również widać różnicę, gdyż wzmianek o 3D jest ok 115 i w jednej i drugiej części, jednak na temat młodszego filmu było około 3 razy więcej tweetów.

## 2.4. Tłumaczenie

By móc przeanalizować sentyment tweetów w języku polskim zostały one przetłumaczone na język angielski przy użyciu googletrans.

Funkcja *tlumaczenie* tłumaczy treść tweetów i zapisuje do pliku csv.

```
translator = Translator()

def tlumaczenie(df_avatar,tytul_csv):
    czas_start = datetime.now()

    for index, row in df_avatar.iterrows():

        tlumaczenie = translator.translate(row['Text'],dest="en")
        df_avatar.at[index,'Text']=tlumaczenie.text

    df_avatar.to_csv(tytul_csv, mode='w',sep=',', index=False)

    czas_stop = datetime.now()
    print('czas potrzebny na tłumaczenie tweetów: {}'.format(czas_stop - czas_start))
```

Wykorzystywana funkcja sprawdzała ile czasu zajęło przetłumaczenie zebranych tweetów w języku polskim.

Część pierwsza:  
czas potrzebny na tłumaczenie tweetów: 0:03:17.812292

Część druga:  
czas potrzebny na tłumaczenie tweetów: 0:10:37.938260

## 2.5. Chmury tagów

Chmura tagów to graficzne zobrazowanie zawartości tekstu, jest to ciekawe ukazanie najczęściej występujących wyrazów.

W projekcie za tworzenie chmur tagów odpowiedzialna jest funkcja *tworzenie\_wordClouds*. By stworzyć taki rezultat dokonuje się tokenizacji, czyli rozbicia tekstu na pojedyncze wyrazy. Następnie w funkcji usuwane są słowa ze stop listy. Są to popularne słowa które mają małe znaczenie dla tekstu. Potem robiona jest lematyzacja, czyli sprowadzenie słowa do jego formy podstawowej. Informacyjnie wyświetlana jest lista 10 najczęściej występujących słów w danych tweetach. Kolejnym krokiem jest stworzenie i zapisanie chmury tagów.

```
def tworzenie_wordClouds(df_avatar,tytul):
    text = ""
    for index, row in df_avatar.iterrows():
        text = text + " "+row['Text'].lower()

    word_tokens = word_tokenize(text)
    print("Ilość słów bez stop words: "+str(len(word_tokens)))

    stop_words = get_stop_words('en')
    for s in my_stop_words:
        stop_words.append(s)

    text_bez_stop_words = [w for w in word_tokens if not w in stop_words]
    print("Ilość słów po eliminacji stop words: "+str(len(text_bez_stop_words)))

    lemmatizer = WordNetLemmatizer()

    lemmatize_sentence = []

    for w in text_bez_stop_words:
        lemmatize_sentence.append(lemmatizer.lemmatize(w))

    fdist = FreqDist(lemmatize_sentence)

    fdist_10_most_common = fdist.most_common(10)

    df = pd.DataFrame(fdist_10_most_common)
    df.columns = ['word', 'amount']
    print("\nLista 10 najczęściej występujących słów:")
    print(df)

    coloring = np.array(Image.open("avatar_warstwa.png"))
    wc = WordCloud(background_color="white", max_words=500, mask=coloring)
    wc.generate(' '.join(text_bez_stop_words))
    image_colors = ImageColorGenerator(coloring)
    wc.recolor(color_func=image_colors)
    wc.to_file(tytul)
```



By uatrakcyjnić wygląd chmury tagów zostaje jej nadany kształt jak na poniższej grafice (główni bohaterzy filmu).



Wywołanie funkcji skutkuje poniższym wydrukiem.

#### CZĘŚĆ PIERWSZA

Tweety po polsku:  
Ilość słów bez stop words: 16034  
Ilość słów po eliminacji stop words: 7834

Lista 10 najczęściej występujących słów:

	word	amount
0	avatar	1016
1	movie	128
2	3d	108
3	film	98
4	2009	64
5	cinema	62
6	will	53
7	cameron	48
8	d	47
9	new	42

Tweety po angielsku:  
Ilość słów bez stop words: 73738  
Ilość słów po eliminacji stop words: 35400

Lista 10 najczęściej występujących słów:

	word	amount
0	avatar	3346
1	2	504
2	water	399
3	movie	383
4	will	354
5	new	321
6	cinema	313
7	like	242
8	one	239
9	time	191

#### CZĘŚĆ DRUGA

Tweety po polsku:  
Ilość słów bez stop words: 284511  
Ilość słów po eliminacji stop words: 133047

Lista 10 najczęściej występujących słów:

	word	amount
0	avatar	14858
1	now	3347
2	support	3029
3	add	2993
4	see	1713
5	movie	1286
6	3d	1244
7	new	1140
8	just	951
9	like	816

Tweety po angielsku:  
Ilość słów bez stop words: 362496  
Ilość słów po eliminacji stop words: 181049

Lista 10 najczęściej występujących słów:

	word	amount
0	avatar	14023
1	movie	1776
2	2	1705
3	like	1514
4	way	1276
5	just	1259
6	see	1177
7	water	1080
8	one	1028
9	new	941

Poniżej znajdują się stworzone chmury tagów.

Dla pierwszej i drugiej części w języku polskim (tweety zostały przetłumaczone wcześniej na język angielski)



Dla pierwszej i drugiej części w języku angielskim.



## 2.6. Analiza sentymentu

### 2.6.1. Funkcja służąca do analizy sentymentu

Najciekasz statystyką jest analiza sentymentu, np. czy zmieniało się nastawienie użytkowników do filmu w tweetach przed i po premierze filmu. W projekcie tą analizą zajmuje się funkcja *analizaSentymentu*. Sprawdzany jest sentyment dla kolejnych dni i z wyników tworzony jest średni wynik dla każdego dnia. Sentyment zostaje badany za pomocą vaderSentiment oraz textblob, by można było porównać dane narzędzia. Zliczana jest liczba pozytywnych i negatywnych tweetów na temat filmu.

```
analyzer = SentimentIntensityAnalyzer()

def analizaSentymentu(data, tweet_df, tytul):
    data2 = data
    data2 += timedelta(days=1)
    licznik = 1
    licznik_neg, licznik_poz = 0, 0
    sentiment1, sentiment2 = 0, 0
    pozytywne, negatywne = "", ""

    tweets_list = []
    for i in range(0, tweet_df.shape[0]):

        if(tweet_df.iloc[i]['Datetime'][0:10] == str(data)[0:10]):
            licznik += 1

            blob = TextBlob(tweet_df.iloc[i]['Text'])
            vs = analyzer.polarity_scores(tweet_df.iloc[i]['Text'])

            sentiment1 += blob.sentiment.polarity
            sentiment2 += vs["compound"]

            if vs["compound"] <= 0:
                negatywne = negatywne + " " + tweet_df.iloc[i]['Text'].lower()
                licznik_neg += 1
            elif vs["compound"] >= 0:
                pozytywne = pozytywne + " " + tweet_df.iloc[i]['Text'].lower()
                licznik_poz += 1
            if i+1 < tweet_df.shape[0]:
                if(tweet_df.iloc[i+1]['Datetime'][0:10] == str(data2)[0:10]):

                    tweets_list.append([tweet_df.iloc[i]['Datetime'][0:10], licznik, sentiment1/licznik, sentiment2/licznik])

                    data += timedelta(days=1)
                    data2 += timedelta(days=1)
                    licznik = 1
                    sentiment1 = 0
                    sentiment2 = 0
                elif(tweet_df.iloc[i+1]['Datetime'][0:10] != str(data)[0:10]):
                    while(tweet_df.iloc[i+1]['Datetime'][0:10] != str(data)[0:10]):
                        data += timedelta(days=1)
                        data2 += timedelta(days=1)

    tweets = pd.DataFrame(tweets_list, columns=['Datetime', 'Counter', 'Sentiment1', 'Sentiment2'])

    ax = tweets.plot.bar(x='Datetime', y='Sentiment1', figsize=(15, 7))
    ax = tweets.plot.bar(x='Datetime', y='Sentiment2', figsize=(15, 7))
    plt.tight_layout()
    plt.savefig(tytul)

    return([licznik_poz, round(licznik_poz/tweet_df.shape[0]*100, 2), licznik_neg, round(licznik_neg/tweet_df.shape[0]*100, 2), tweet
    print("Pozytywnych tweetów było: " + str(licznik_poz) + "to ")
    print("Negatywnych tweetów było: " + str(licznik_neg))
```

Wydruk po wywołaniu funkcji dla zebranych tweetów o „Avatarze”

```
Ilość tweetów:

po polsku w I części - pozytywnych :      485      jest to 48.12% wszystkich tweetow (1008)
po polsku w I części - negatywnych :      523      jest to 51.88% wszystkich tweetow (1008)

po polsku w II części - pozytywnych :     1550      jest to 48.03% wszystkich tweetow (3227)
po polsku w II części - negatywnych :     1677      jest to 51.97% wszystkich tweetow (3227)

po angielsku w I części - pozytywnych :    9232      jest to 62.67% wszystkich tweetow (14731)
po angielsku w I części - negatywnych :    5499      jest to 37.33% wszystkich tweetow (14731)

po angielsku w II części - pozytywnych :   6409      jest to 46.5% wszystkich tweetow (13782)
po angielsku w II części - negatywnych :   7373      jest to 53.5% wszystkich tweetow (13782)
```

## 2.6.2. Porównanie działania vaderSentiment oraz textblob dla jednego tweeta

Treść badanego tweeta: „Avatar - The creature of water Wonderful special effects create a great spectacle worth the money spent. It's good to watch from start to finish, but the content of the film is too simple and that's why I left the cinema with a slight unsatisfaction. I recommend. This is a must watch movie.”

Treść tweeta jest raczej pozytywna, autor zachwala film lecz ma pewne zastrzeżenia co do filmu.

Poniżej widzimy jak z jego oceną poradziły sobie badane narzędzia.

```
text2 = "Avatar - The creature of water Wonderful special effects create a great spectacle worth the money spent. It's good to
blob = TextBlob(text2)
print("Ocena textblob      : "+str(blob.sentiment.polarity))

analyzer = SentimentIntensityAnalyzer()
vs = analyzer.polarity_scores(text2)
print("Ocena vaderSentiment : "+str(vs["compound"]))
```

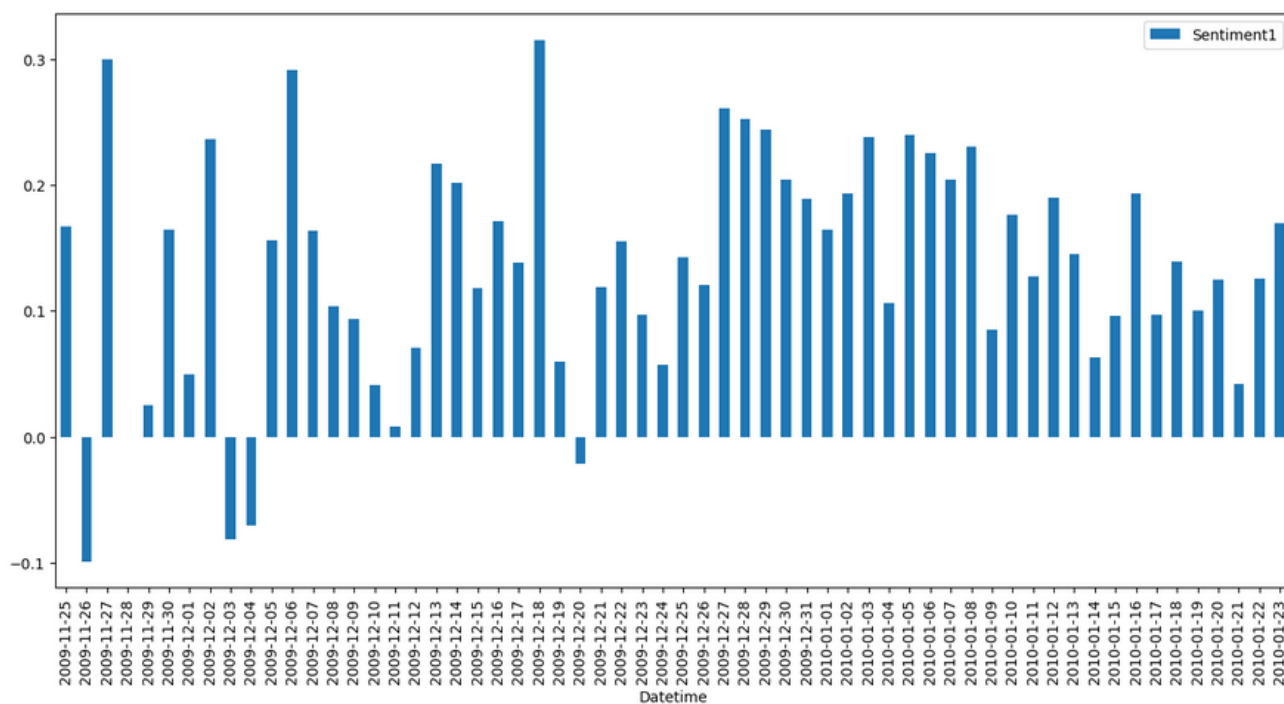
Ocena textblob : 0.32116402116402115  
Ocena vaderSentiment : 0.8899

Wydaje się, że lepszej oceny dokonało narzędzie vaderSentiment.

## 2.6.3. Analiza sentymentu tweetów na przestrzeni miesiąca przed i po premierze

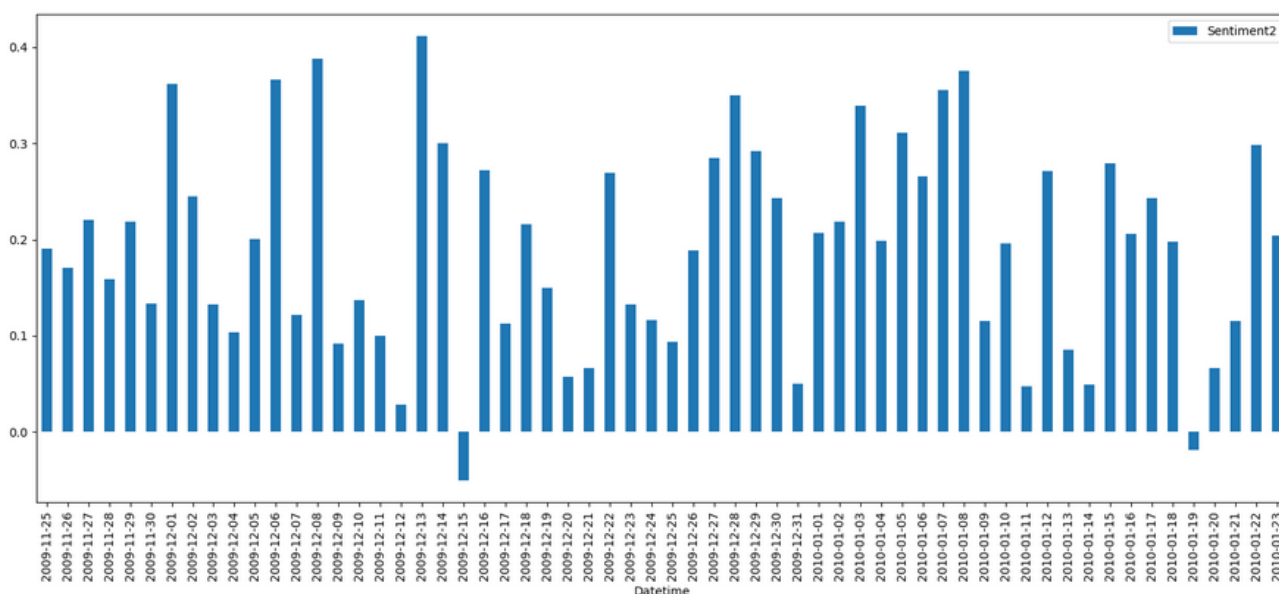
### 2.6.3.1 Analiza sentymentu tweetów w języku polskim dotyczących pierwszej części filmu

Poniżej znajduje się wykres obrazujący analizę sentymentu stworzoną przy pomocy narzędzia textblob.



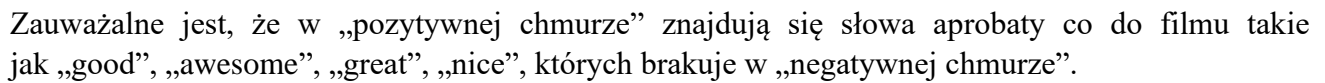
Analizując wykres widoczne jest, że przez większość dni średnia sentymentu tweetów przyjmuje pozytywne wyniki. Zauważalne jest również, że po premierze filmu (25.12.2009) średnia sentymentu przyjmuje bardzo zbliżone wyniki między dniami, gdzie przed premierą wachania wyników były bardzo duże.

Poniżej znajduje się wykres obrazujący analizę sentymentu stworzoną przy pomocy narzędzia vaderSentiment



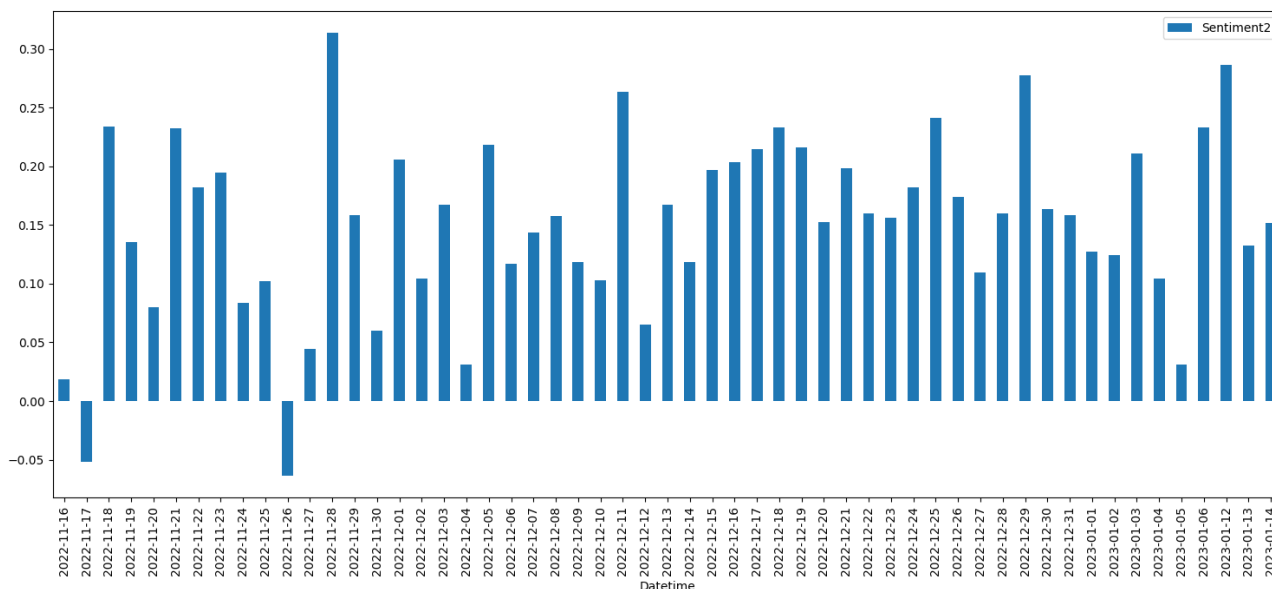


Poniżej chmura tagów dla tweetów ocenionych jako pozytywne i negatywne.



### 2.6.3.2 Analiza sentymentu tweetów w języku polskim dotyczących drugiej część filmu

Dla kolejnych przykładów zostanie omówiona tylko ocena syntementu stworzona dzięki vaderSentiment, gdyż narzędzie wydaje bardziej wiarygodne zwracając uwagę na wcześniejszą ocenę pojedynczego tweetu.



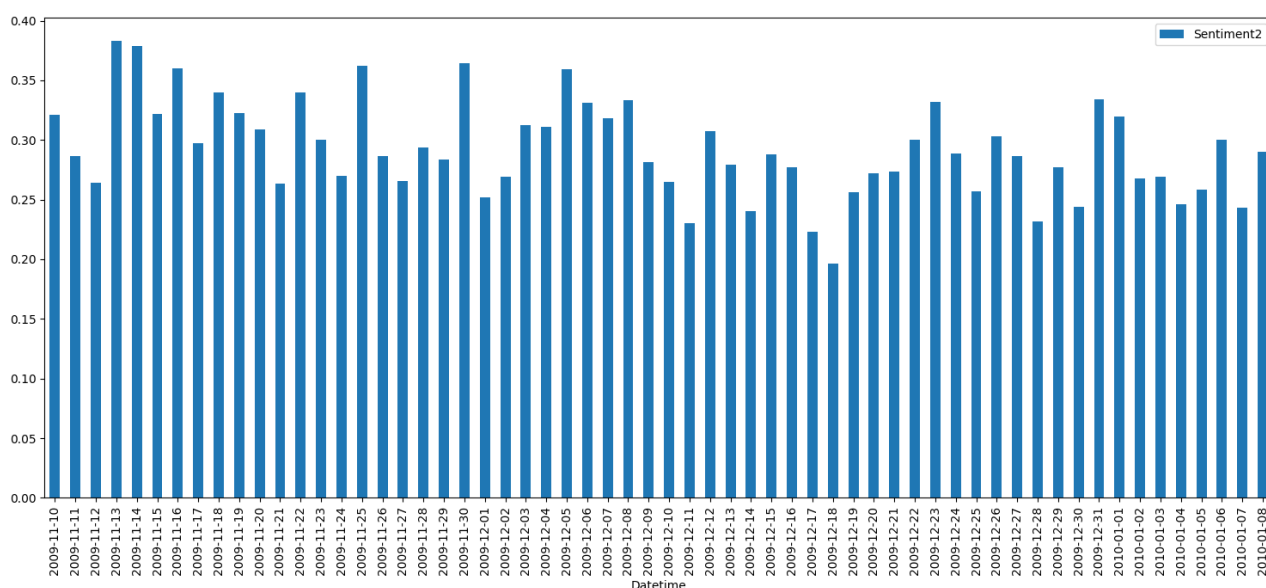
Ponownie przez większość dni średnia sentymentu tweetów przyjmuje pozytywne wyniki. Po premierze filmu (16.12.2022) widzimy, że wyniki sentymentu są na ogół wyższymi wartościami niż te przed premierą oraz wachania wyników między dniami są mniejsze.

Poniżej chmura tagów dla tweetów ocenionych jako pozytywne i negatywne.



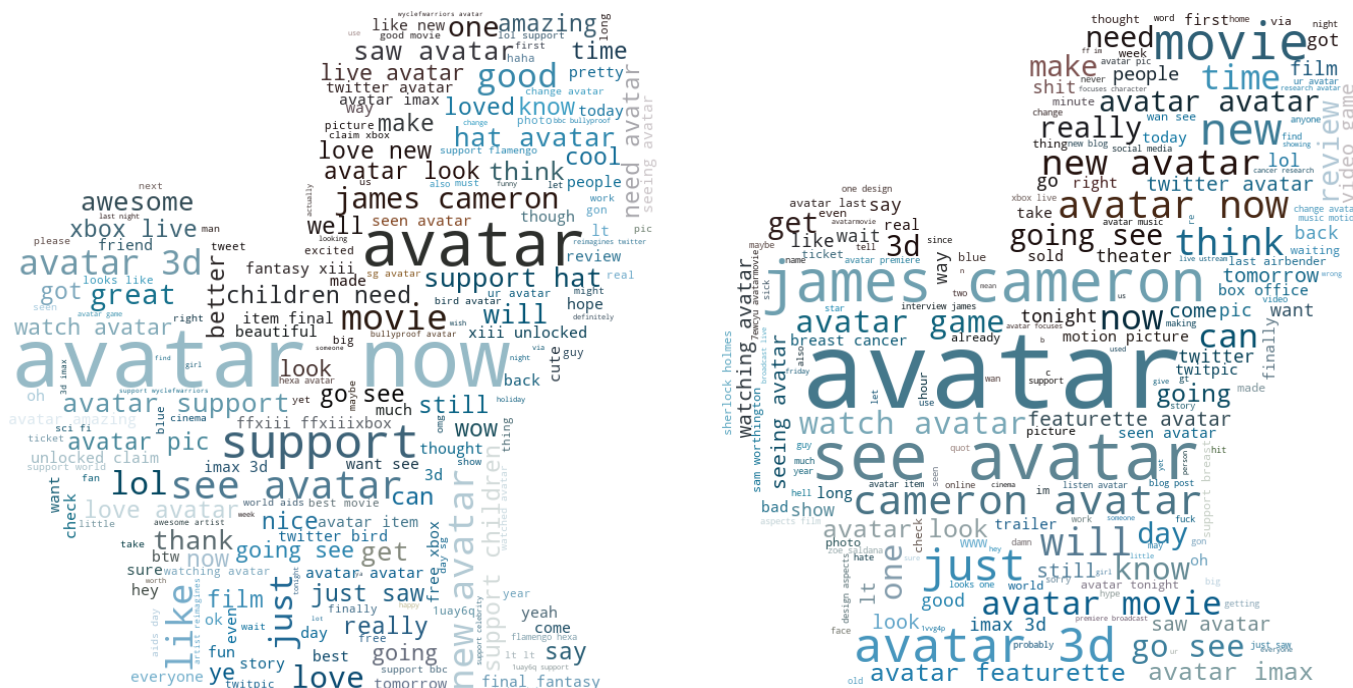
W chmurze stworzonej z tweetów ocenionych jako pozytywne rzucają się w oczy słowa takie jak „beautiful”, oraz „like” czy „love”. W drugiej chmurze znajduje się słowo „long”, co może świadczyć, że użytkownicy narzekali na długość trwania filmu, który trwa ponad 3h.

### 2.6.3.3 Analiza sentymentu tweetów w języku angielskim dotyczących pierwszej część filmu



Analizując wykres widać, że większość tweetów była pozytywna, gdyż żadnego dnia wynik nie jest poniżej 0.2. Co prawda w dzień premiery (10.12.2009) i przez kilka następnych dni, wynik przyjmuje najniższe wartości dla badanego przedziału czasowego.

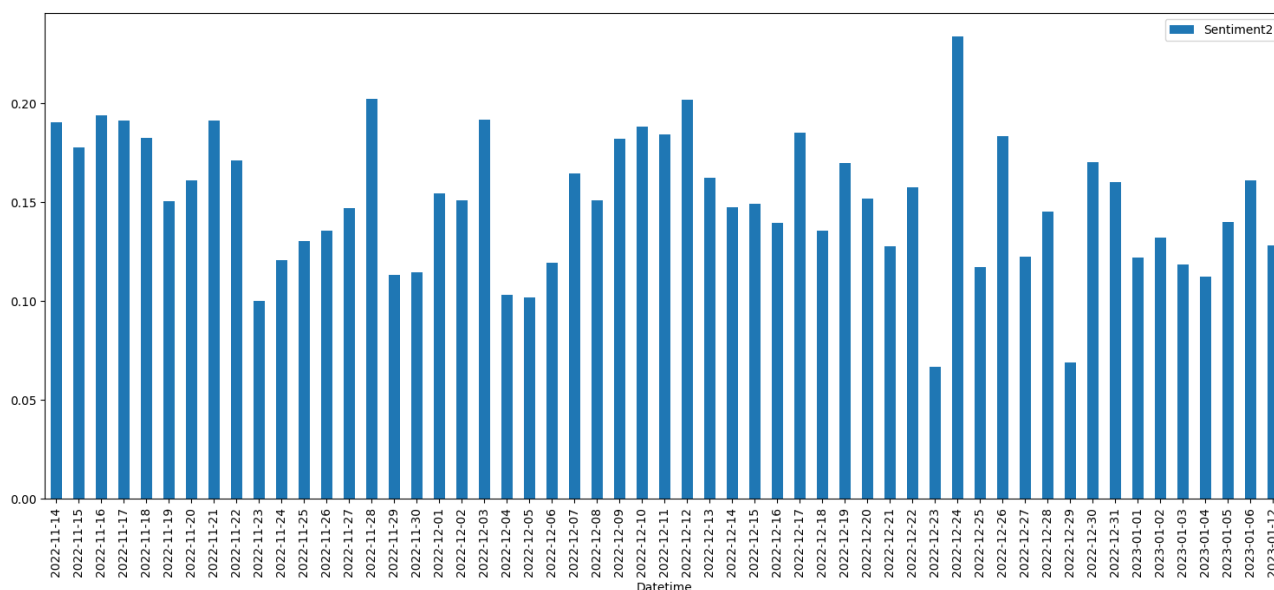
Poniżej chmura tagów dla tweetów ocenionych jako pozytywne i negatywne.



Chmura z pozytywnymi tweetami wypełniona jest dużą ilością pozytywnych wyrazów np. „beautiful”, „great”, „amazing”, chmura z negatywnymi tweetami nie zawiera negatywnych słów lub nie są widoczne na pierwszy rzut oka.



#### 2.6.3.4 Analiza sentymentu tweetów w języku angielskim dotyczących drugiej część filmu



Oceniając po wykresie znowu większość tweetów była pozytywna. Nie ma zauważalnych zmian dotyczących premiery filmu (14.12.2022).

Poniżej chmura tagów dla tweetów ocenionych jako pozytywne i negatywne.



W chmurze stworzonej z tweetów ocenionych jako pozytywne znajdują się słowa takie jak „like”, „love”, „great”, których nie ma w chmurze stworzonej z tweetów negatywnych, lecz np. znajduje się tam słowo „long”, co sugeruje, że użytkownicy narzekali na długość trwania filmu.

### 3. Podsumowanie

Film „Avatar” cieszył się większym zainteresowaniem wśród anglojęzycznych użytkowników Twittera, niż wśród polskojęzycznych. Mniejsza ilość tweetów w języku polskim utrudniała ich analizę. Tweety pisane na temat pierwszej części filmu są zauważalnie krótsze. W drugiej części jest zdecydowanie mniej razy poruszana kwestia technologii 3D. Różnica w ilości tweetów ocenionych jako negatywne i jako pozytywne jest zauważalna tylko wpisach na temat pierwszej części w języku angielskim, gdzie około 60% to tweety pozytywne, a około 40% negatywne. Dla reszty pozytywnych i negatywnych wpisów jest prawie tyle samo. Analiza sentymentu rozbita na dni daje obraz na to jak film był postrzegany przed i po premierze, lecz niestety dla żadnej z badanych opcji nie widać dużej różnicy spowodowanej wypuszczeniem filmu do kin.

### 4. Bibliografia

- <https://betterprogramming.pub/how-to-scrape-tweets-with-snsrape-90124ed006af>
- <https://py-googletrans.readthedocs.io/en/latest/>