Politechnika Warszawska

WYDZIAŁ ELEKTRYCZNY

Instytut Elektrotechniki Teoretycznej i Systemów Informacyjno-Pomiarowych
Zakład Elektrotechniki Teoretycznej i Informatyki Stosowanej

Praca dyplomowa magisterska

na kierunku Informatyka
w specjalności Inżynieria oprogramowania

Analiza wiadomości z serwisów społecznościowych na użytek rozpoznawania nastrojów.

Jakub Rzepliński
Numer albumu: 233608

promotor
dr inż. Marcin Kołodziej

Warszawa, 2018

Streszczenie

Abstract

Oświadczenie autora (autorów) pracy

Świadom odpowiedzialności prawnej oświadczam, że przedstawiona praca dyplomowa:

– została napisana przeze mnie samodzielnie i nie zawiera treści uzyskanych w sposób niezgodny z obowiązującymi przepisami,

– nie była wcześniej przedmiotem procedur związanych z uzyskaniem tytułu zawodowego lub stopnia naukowego w wyższej uczelni

Oświadczam ponadto, że niniejsza wersja pracy jest identyczna z załączoną wersją elektroniczną.

....
data podpis autora (autorów) pracy

Oświadczenie

Wyrażam zgodę / nie wyrażam zgody*1 na udostępnianie osobom zainteresowanym mojej pracy dyplomowej. Praca może być udostępniana w pomieszczeniach biblioteki wydziałowej. Zgoda na udostępnienie pracy dyplomowej nie oznacza wyrażenia zgody na jej kopiowanie w całości lub w części.
Brak zgody nie oznacza ograniczenia dostępu do pracy dyplomowej osób:

– reprezentujących władze Politechniki Warszawskiej,

– członków Komisji Akredytacyjnych,

– funkcjonariuszy służb państwowych i innych osób uprawnionych, na mocy odpowiednich przepisów prawnych obowiązujących na terenie Rzeczypospolitej Polskiej,

do swobodnego dostępu do materiałów chronionych międzynarodowymi przepisami o prawach autorskich. Brak zgody nie wyklucza także kontroli tekstu pracy dyplomowej w systemie antyplagiatowym.

....
data podpis autora (autorów) pracy

*1 - niepotrzebne skreślić

Spis treści

Rozdział 1

Wstęp

Czynnikiem powodującym rozwój świata są informacje i umiejętne ich wykorzystanie. Od jakiegoś czasu źródłem informacji stał się internet, a szczególnie portale społecznościowe, gdzie ludzie wymieniają się informacjami na różne tematy. To na serwisach tego typu często posiadają konta ważne globalne instytucje i stało się normalne, że za ich pomocą wydają oficjalne komunikaty. Ilość informacji generowanych przez użytkowników takich serwisów jest tak duża, że wymaga użycia specjalnych narzędzi *Big Data*. Jednak decydującym czynnikiem jest umiejętne wykorzystanie wiedzy zawartej w zgromadzonych informacjach, do czego potrzebny jest czynnik ludzki.
Jednym z ciekawych sposobów wykorzystania informacji z serwisów takich jak np. *Twitter* jest badanie wydźwięku wpisów zamieszczanych przez użytkowników zwanego sentymentem od angielskiego sformułowania *sentiment analysis*. Aby było to możliwe potrzebne jest przetworzenie języka, którym posługują się ludzie czyli *języka naturalnego* na postać, którą mogą posługiwać się systemy NLP (ang. *Natural Language Processing*), a następnie określenie sentymentu za pomocą podejścia słownikowego lub technik maszynowego uczenia się (ang. *machine learning*). Wykorzystanie takich informacji pozwala zbadać opinie ludzi na praktycznie dowolny temat.
Przedstawiona praca dyplomowa opisuje także aplikację internetową stworzoną na potrzeby niniejszej pracy, która spełnia wymagania systemu czasu rzeczywistego - przetwarza i analizuje wiadomości przychodzące w czasie rzeczywistym z serwisu Twitter zawierające podane słowo kluczowe oraz bada wydźwięk opinii użytkowników tego serwisu.

Rozdział 2

Serwis społecznościowy Twitter

Serwis społecznościowy Twitter jest globalnym serwisem internetowym służącym głównie do zamieszczania wiadomości tzw. *tweet*, które użytkownicy tego serwisu mogą także czytać, komentować lub przekazywać dalej. Od kilku lat Twitter jest serwisem gdzie dochodzi do wymiany zdań na różny temat, dotyczących np. polityki, sportu, produktów, wydarzeń społecznych, a profile posiada wiele osób znanych publicznie oraz instytucji.

2.1 Historia

Serwis ten, nazywany SMS internetu, został założony w 2006 r. w Stanach Zjednoczonych przez Jacka Dorseya, Ev Williamsa, Noah Glassa oraz Biza Stone'a i od początku powstania sukcesywnie zwiększał swoją popularność poprzez wzrost liczby użytkowników odwiedzających jego witrynę oraz wysyłających wiadomości. W 2012 r. osiągnął ponad 100 milionów użytkowników, którzy zamieszczali łącznie ponad 340 milionów wiadomości dziennie oraz obsługiwał średnio około 1.6 miliarda wyszukujących zapytań dziennie. W 2013 r. Twitter stał się jedną z najczęściej odwiedzanych stron w całym internecie. W tym samym roku inżynierowie Twittera podali informację, że serwis ten obsługuje ok. 143 tys. wiadomości na sekundę. Na początku 2016 r. serwis ten posiadał ponad 319 milionów użytkowników aktywnych podczas każdego miesiąca. Od listopada 2013 r. akcje Twittera są obecne na nowojorskiej giełdzie.

2.2 Tweet

Tweet, czyli krótka wiadomość tekstowa, była początkowo ograniczona do 140 znaków, ale limit ten został podwojony w 2017 r. dla wszystkich języków oprócz chińskiego, japońskiego i koreańskiego. Użytkownicy mają możliwość wyróżniania wybranych przez siebie tematów przez dodanie do nich znaku '#', co czyni takie wyrażenie tagiem. Inną możliwością oferowaną przez Twittera jest odpowiadanie innym użytkownikom lub zamieszczenie referencji do nich przez dodanie znaku '@' poprzedzającego nazwę profilu innej osoby.

2.3 Architektura

Serwis społecznościowy Twitter opierał się początkowo o typową architekturę trójwarstwową składająca się z warstwy prezentacji, logiki biznesowej oraz warstwy danych. Do napisania tej aplikacji został użyty framework Ruby on Rails wykorzystujący język Ruby, a warstwa bazodanowa opierała się o technologię MySQL. Jednak wraz ze wzrostem ilości przetwarzanych danych inżynierowie Twittera podjęli decyzję w 2011 r. o zmianie technologii na język Scala, który działa na maszynie wirtualnej Javy oraz zrezygnowano z dotychczasowej architektury na

rzecz budowy rozproszonych serwisów komunikujących się między sobą. Wraz z przeprowadzonymi zmianami zanotowano ponad 10-krotne polepszenie obsługi tweetów.

2.4 Interfejs programistyczny aplikacji Twitter

Twitter jest platformą otwartą i udostępnia programowalny interfejs (ang. API - *Application Programming Interface*) w dwóch postaciach: Search API oraz Streaming API.

2.4.1 Zakres działania

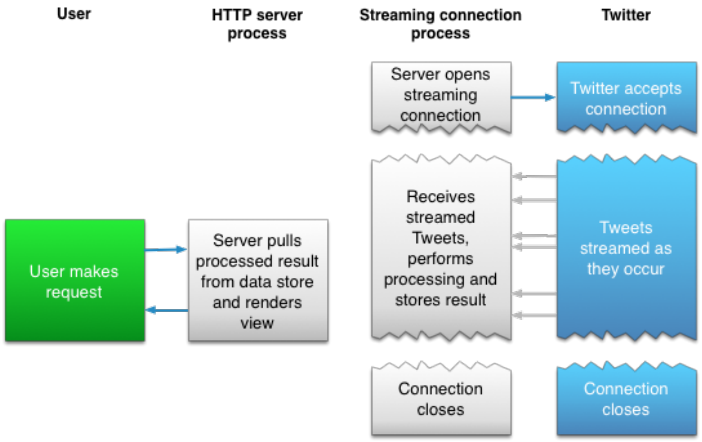
Programiści korzystający z Search API są w stanie uzyskać dostęp tylko do danych historycznych, które zostały już wcześniej zamieszczone na łamach serwisu Twitter. Natomiast w przypadku Streaming API dostajemy możliwość śledzenia strumienia danych, które są do naszego dostępu nawet już kilka sekund po zamieszczeniu w serwisie Twitter. Po podłączeniu do takiego strumienia możemy cały czas obserwować nowe wiadomości. Przyjęło się stosować nazywnictwo, że analiza Search API to analiza *back in time*, a Streaming API to śledzenie *real time*.

2.4.2 Rejestracja

Obie formy API wymagają wcześniejszej rejestracji na stronie *https://developer.twitter.com/en/apply-for-access* przeznaczonej dla deweloperów zainteresowanych wykorzystywaniem Twitter API. Po przejściu pomyślnej rejestracji dostajemy dane, które po nawiązaniu połączenia z serwisem Twitter umożliwiają mu jednoznacznie określić, że możemy uzyskać dostęp do API.

2.4.3 Sposób działania

Search API powstało z wykorzystaniem standardu REST - *Representational State Transfer*. Oba rodzaje API wykorzystują protokół HTTP: do poprawnego działania Streaming API potrzebne jest ciągłe połączenie HTTP, a w przypadku drugiego z nich każda operacja jest wykonywana przy nawiązaniu oddzielnego połaczenia.



Rysunek 2.1: Schemat działania dwóch rodzajów programistycznego interfejsu API udostępnianego przez serwis społecznościowy Twitter: Search API i Streaming API.

Search API posiada ściśle określone paramtery, które mogą być przesłane w żądaniu. W tabeli 3.1 zaprezentowano ich wykaz.

Tablica 2.1: Parametry żądania Twitter Search API

Parametr	Wymagany/Opcjonalny	Opis	Przykład
q	wymagany	zapytanie wyszukujące o maksymalnej długości 500 znaków	nasa
geocode	opcjonalny	zwraca wiadomości użytkowników oddalonych o podany promień od podanej szerokości i długości geograficznej, promień może być podany w milach lub kilometrach	37.781157 -122.398720 1mi
lang	opcjonalny	ogranicza wiadomości do wybranego języka spośród dostępnych kodów ISO 639-1	pl
locale	opcjonalny	specyfikuje język wysyłanego zapytania, obecnie tylko *j*a jest skuteczny	ja
result_type	opcjonalny	określa typ zwracanych wiadomości, obecnie dostępne są trzy wartości tego parametru: *recent* (zwracane są najnowsze wiadomości), *popular* (zwracane są najbardziej popularne wiadomości) *mixed* (wartość domyślna, zwracane wyniki obejmują najnowsze i najbardziej popularne wiadomości)	mixed
count	opcjonalny	specyfikuje ilość zwracanych wiadomości; maksymalna wartość to 100, a domyślna to 15	100

until	opcjonalny	ten parametr odpowiada za zwracanie wiadomości, których data utworzenia jest starsza o maksymalnie tydzień niż podana; obowiązuje format YYYY-MM-DD	2015-07-19
since_id	opcjonalny	dzięki temu parametrowi zwracane są wiadomości o ID większym niż podane czyli nowsze wiadomości niż określono	12345
max_id	opcjonalny	dzięki temu parametrowi zwracane są wiadomości o ID mniejszym lub równym niż podane czyli starsze lub takie same wiadomości niż określono	54321

Streaming API nie posiada takich ograniczeń. W języku programowania Java dostępny jest pakiet *twitter4j* zawierający interfejsy *User* oraz *Status*, na które mapowane są przychodzące ze strumienia informacje. W tabeli 3.2 zamieszczono ich dokumentację.

2.4.4 Ograniczenia

Korzystając z Search API mamy możliwość wysłania 720 zapytań na godzinę, a maksymalna ilość wiadomości jaka może być zwrócona na jedno zapytanie to 100. Jeśli wykorzystalibyśmy ten limit w maksymalny sposób to daje nam to 72 000 wiadomości na godzinę. W przypadku Streaming API głównym ograniczeniem jest dostęp do ok. 1 % danych ze strumienia, a maksymalna ilość wiadomości w czasie jednej minuty to 3 000. W przypadku tego API w ciągu godziny możemy uzyskać 180 000 wiadomości na godzinę. Są to ograniczenia, które obowiązują dla rozwiązań typu *open-source*.

2.5 Dostępne narzędzia analityczne

Udostępnienie API przez serwis społecznościowy Twitter oraz rosnące znaczenie danych generowanych przez użytkowników tego serwisu spowodowało, że wiele firm oraz instytucji zaczęło przywiązywać dużą wagę do analizy opinii wyrażanych na swój temat lub na tematy pokrewne, zainteresowania pewnymi tematami oraz kształtujących się trendów. Dlatego powstały aplikacje internetowe służące do wyświetlania takich informacji i przeprowadzające wstępną analizę zebranych danych. W dalszej części pracy znajduje się omówienie najważniejszych z nich.

2.5.1 Twitter Analytics

Pierwszym z narzędzi, którym warto poświęcić uwagę jest *Twitter Analytics*. Aplikacja ta posiada trzy zakładki. Na pierwszej z nich wyświetla statystyki dotyczące profilu z serwisu Twitter, którym logujemy się do niej: naszą najbardziej popularną wiadomość, najpopularniejszą wzmiankę o naszym profilu oraz wykresy trendów m. in. liczby osób śledzących nasz profil i odwiedzin. Na kolejnej zakładce mamy możliwość tworzenia wiadomości, które zostaną wygenerowane na naszym profilu oraz dodania do nich plików graficznych, materiałów audio lub wideo. Na ostatniej stronie użytkownik ma szansę poznania informacji takich jak np. lokalizacja geograficzna i wiek osób śledzących nasz profil czyli tzw. *followers*.

2.5.2 Hootsuite

Kolejną aplikacją jest *Hootsuite*. Jest to narzędzie, które umożliwia prowadzić kampanie w kilku mediach społecznościowych na raz np. *Facebook*, *Instagram*, *Linkedin*. Posiada wiele rozbudowanych funkcji. Co ciekawe Hootsuite pozwala na analizę nastrojów społecznych dla wybranych

Tablica 2.2: Dokumentacja interfejsu Status

Typ zwracany	Nazwa metody	Opis
long[]	getContributors()	
java.util.Date	getCreatedAt()	zwraca datę utworzenia wiadomości
long	getCurrentUserRetweetId()	zwraca id użytkownika, którego wiadomość została podana dalej
int	getDisplayTextRangeEnd()	
int	getDisplayTextRangeStart()	
int	getFavoriteCount()	zwraca informację ile razy została polubiona wiadomość
GeoLocation	getGeoLocation()	zwraca lokalizację użytkownika zamieszczającego tą wiadomość
long	getId()	zwraca id wiadomości
java.lang.String	getInReplyToScreenName()	zwraca nazwę użytkownika, do którego kierowana jest odpowiedź
long	getInReplyToStatusId()	zwraca id wiadomości, do którego kierowana jest odpowiedź
java.lang.String	getLang()	zwraca język zamieszczonej wiadomości
Place	getPlace()	zwraca obiekt Place przypisany do tej wiadomości
Status	getQuotedStatus()	zwraca obiekt Status cytowanej wiadomości
long	getQuotedStatusId()	zwraca id cytowanej wiadomości
URLEntity	getQuotedStatusPermalink()	zwraca obiekt URLEntity reprezentujący bezpośredni odnośnik do cytowanej wiadomości
int	getRetweetCount()	zwraca ile razy wiadomość została podana dalej
Status	getRetweetedStatus()	zwraca oryginalny status, który jest podany dalej w tej wiadomości
Scopes	getScopes()	zwraca obiekt typu Scopes posiadający informację o id miejsc, do których odnosi się ta wiadomość
java.lang.String	getSource()	zwraca źródło wiadomości
java.lang.String	getText()	zwraca tekst wiadomości
User	getUser()	zwraca obiekt typu User powiązany z tą wiadomością
java.lang.String[]	getWithheldInCountries()	zwraca tablicę nazw krajów, w których wiadomość została wstrzymana

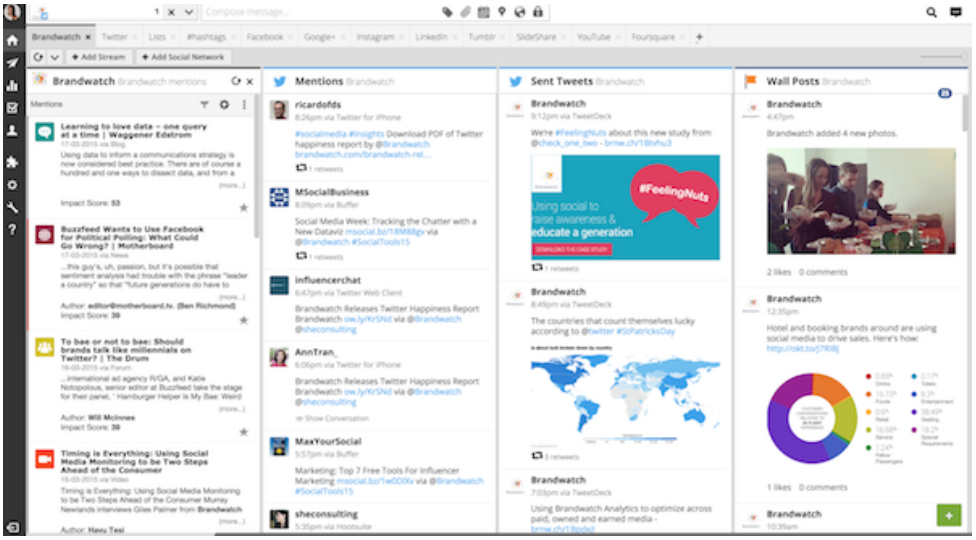
boolean	isFavorited()	zwraca informację czy wiadomość została polubiona
boolean	isPossiblySensitive()	zwraca informację czy wiadomość zawiera link do chronionych informacji
boolean	isRetweet()	zwraca informację czy tweet jest podaną dalej wiadomością
boolean	isRetweeted()	informuje czy wiadomość jest podana dalej
boolean	isRetweetedByMe()	informuje czy wiadomość jest podana dalej przez tego użytkownika
booleand	isTruncated()	informuje czy wiadomość jest skrócona (zakończona znakiem "...")

Tablica 2.3: Dokumentacja interfejsu User z pakietu twitter4j. Zamieszczone zostały tylko najważniejsze z metod.

Typ zwracany	Nazwa metody	Opis
java.util.Date	getCreatedAt()	zwraca datę utworzenia profilu użytkownika
java.lang.String	getDescription()	zwraca opis konta użytkownika
java.lang.String	getEmail()	zwraca adres e-mail powiązany z tym kontem
int	getFavouritesCount()	zwraca liczbę wiadomości, którą polubił ten użytkownik
int	getFollowersCount()	podaje ilość użytkowników śledzących profil
int	getFriendsCount()	podaje ilość śledzonych profili
long	getId()	zwraca id użytkownika
java.lang.String	getLang()	zwraca język preferowany przez użytkownika
java.lang.String	getLocation()	zwraca lokalizację użytkownika
java.lang.String	getName()	podaje nazwę użytkownika
java.lang.String	getScreenName()	zwraca nazwę konta
Status	getStatus()	zwraca obiekt typu Status reprezentujący wiadomość wysłaną przez użytkownika
int	getStatusesCount()	podaje ilość wiadomości wysłanych przez użytkownika
java.lang.String	getTimeZone()	podaje strefę czasową użytkownika
java.lang.String	getURL()	zwraca URL do profilu
boolean	isVerified()	podaje informację czy profil jest zweryfikowany

Rysunek 2.2: Narzędzie Twitter Analytics.

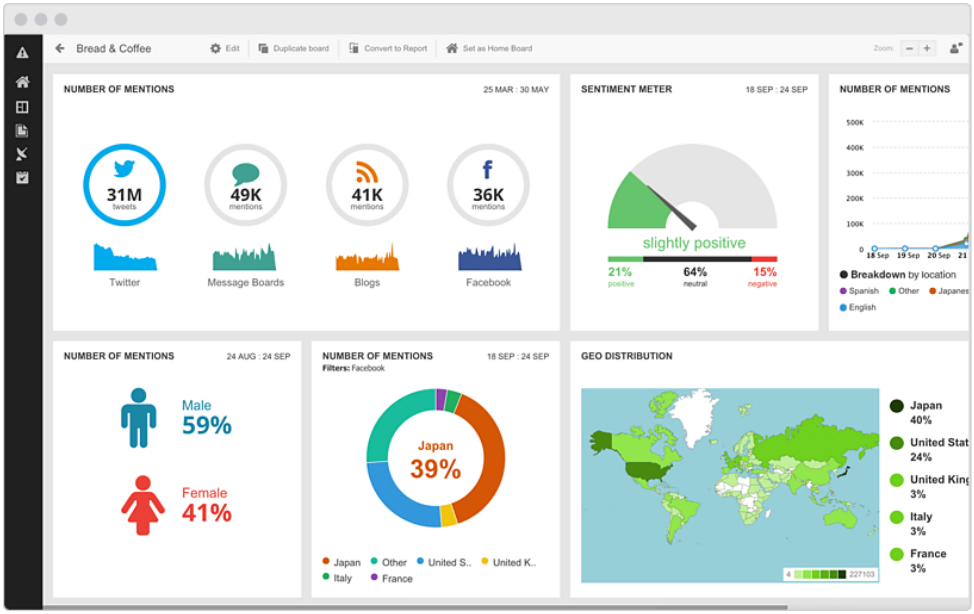
słów kluczowych. Wyświetlane informacje zawierają ogólny zarys użytkowników zamieszczających wiadomości na dany temat np. lokalizację geograficzną, podział ze względu na płeć i język oraz wykres trendu. Aplikacja ta wyświetla wszystkie wiadomości, w których użytkownicy używają wybranego słowa kluczowego. Jak podają twórcy tego narzędzia ma to główne zastosowanie jako pomoc w kampaniach marketingowych w dotarciu do użytkowników krytykujących produkt i posiadających największą ilością osób śledzących.



Rysunek 2.3: Narzędzie Hootsuite - główny pulpit.

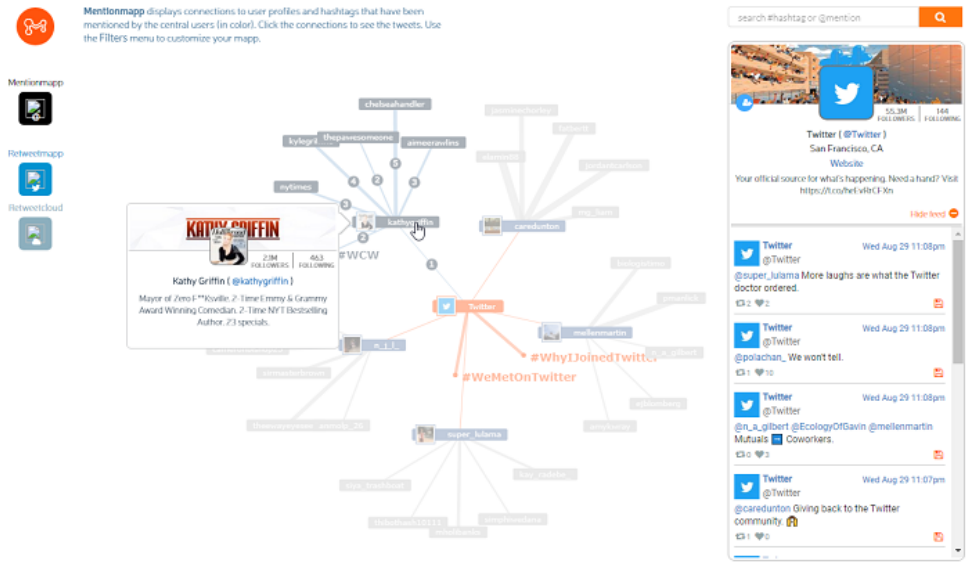
2.5.3 Mentionmap

Trzecim narzędziem wartym omówienia jest *Mentionmap*. Jest to aplikacja, która wyróżnia się spośród innych tym, że rysuje wykres powiązań pomiędzy użytkownikami wchodzącymi w interakcje z danym profilem, a także z ich profilami. Posiada także podstronę umożliwiającą badanie nastrojów społecznych osób zamieszczających wiadomości z konkretnym słowem klu-



Rysunek 2.4: Narzędzie Hootsuite - badanie nastrojów społecznych.

czowym. Nastroje te są przedstawione w postaci chmury nazw kont użytkowników, gdzie kolor nazwy zależy od nastoroju prezentowanego przez użytkownika pod kątem słowa kluczowego.



Rysunek 2.5: Narzędzie Mentionmap - wykres powiązań.

2.5.4 Podsumowanie

Podsumowując warto zauważyć, że nie ma obecnie na rynku aplikacji, która umożliwiałaby śledzenie występowania dowolnego słowa w wiadomościach zamieszczanych w czasie rzeczywistym w serwisie Twitter, rysowałaby wykres zależności pomiędzy użytkownikami tego serwisu, analizowałaby nastroje społeczne użytkowników z wyświetleniem informacji o nastroju wyrażanym w poszczególnych wiadomościach oraz pozwalałaby na analizowanie danych historycznych i zamieszczanych w czasie rzeczywistym. Taka sytuacja pozwala na utworzenie nowej aplikacji, która udostępniałaby te funkcjonalności.

Rysunek 2.6: Narzędzie Mentionmap - badanie nastrojów społecznych.

Rozdział 3

Analiza danych Big Data

Termin *Big Data* odnosi się do dużych, zmiennych i różnorodnych zbiorów danych, których przetwarzanie i analiza jest pracochłonne, ale może prowadzić do ciekawych wniosków oraz pozyskania nowej wiedzy. Zbieranie oraz przechowywanie dużej ilości danych do analizy było praktykowane od bardzo dawna, jednak dokładniejsza koncepcja Big Data została poznana w 2001 roku kiedy to analityk Doug Laney zaprezezentował znaną dzisiaj definicję *3V*: *volume*, *velocity*, *variety* czyli: ilość, szybkość, złożoność, a później dodano jeszcze czwarty atrybut *veracity* czyli: wiarygodność.

3.1 Pojęcie Big Data

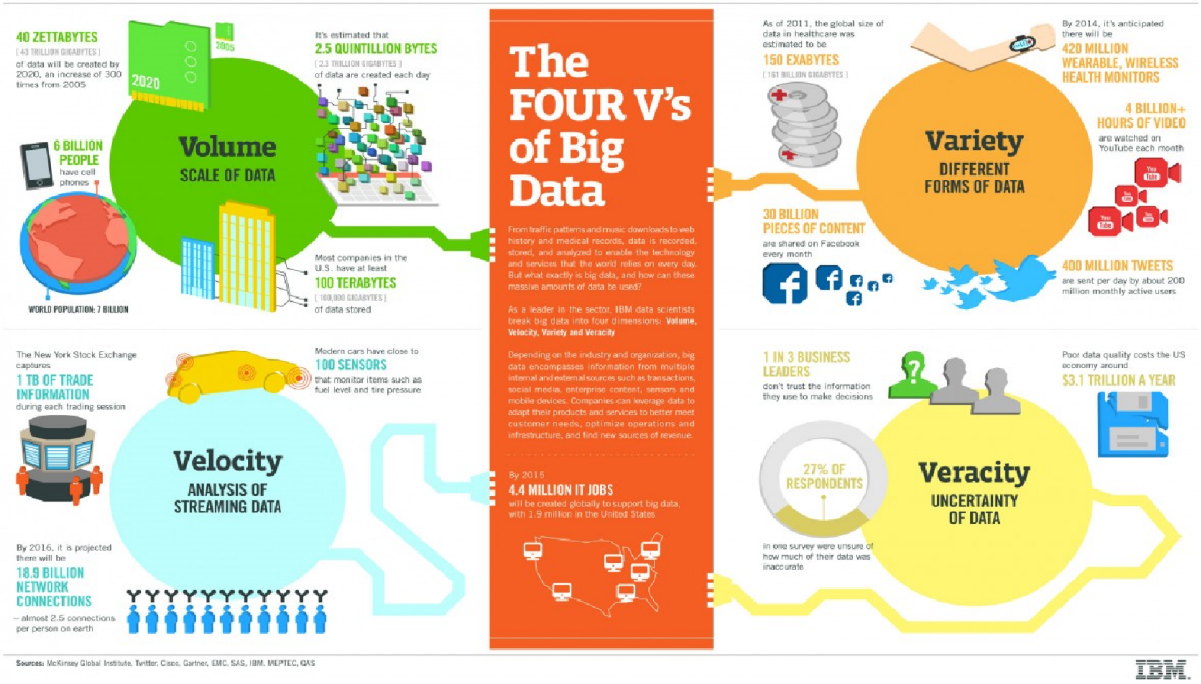
Wraz ze wzrostem zainteresowania Big Data podjęto próby dokładniejszego opisania tego terminu. Obecnie definiując to pojęcie trzeba odnieść się do nowych rozwiązań technologicznych dotyczących wielkich wolumenów danych o innym charakterze ilościowym oraz jakościowym niż dotychczas.
Jedna z pierwszych definicji Big Data została zaprezentowana przez M. Cox i D. Ellsworth w 1997 r. jako duża ilość danych, którą należy zwiększać, aby wydobyć wartości informacyjne. Inna i najbardziej popularna, została przestawiona w 2001 r. przez pracującego dla firmy analityczno-doradczej wspomnianego analityka D. Laney, opiera się o trzy atrybuty: ilość, szybkość i złożoność. W 2012 r. ta sama firma dodała do swojej definicji kolejne dwa atrybuty: zmienność i złożoność. Autorzy innej publikacji *Big Data: Issues, Challenges, Tools and Good Practices* z 2013 r. definiują pojęcie Big Data jako wymagające stosowania nowych technologii i architektur z powodu potrzeby ekstrakcji wartości płynącej z tych danych.
Podsumowując, określenie Big Data to pojęcie odnoszące się do zbiorów danych, które jednocześnie charakteryzują się dużą objętością, różnorodnością, strumieniowym napływem w czasie rzeczywistym, zmiennością, złożonością oraz wymagają stosowania innowacyjnych technologii i narzędzi, aby możliwe było wydobycie z nich wartościowych informacji.

3.2 Charakterystyka

Termin Big Data charakteryzują atrybuty: objętość, szybkość, różnorodność, zmienność, złożoność i wartość. W dalszej cześci tej pracy dyplomowej przedstawiono omówienie każdego z tych atrybutów.

3.2.1 Objętość

Atrybut ten odnosi się do dużej ilości danych, które wymagają nowych technologii. Rozmiar danych zależy od dziedziny i może wynosić od terabajtów lub petabajtów w zagadnieniach ta-



Rysunek 3.1: Definicja *Big Data* w ujęciu 4V.

kich jak np. analiza zderzeń cząstek elementarnych w fizyce do megabajtów lub gigabajtów np. w telekomunikacji przy analizie połączeń wykonywanych przez abonentów. Najnowsze badania prognozują, że ilość danych wzrośnie do 2020 r. o 40% zeta bajtów, co będzie skutkować 50-krotnym wzrostem od początku 2010 r..

3.2.2 Szybkość

Duża szybkość napływających danych charakteryzuje się strumieniowym napływem wymagającym analizy w czasie rzeczywistym. Z powodu ograniczeń przepustowości sieci dane takie należy pobierać porcjami i filtrować pod kątem przydatności oraz wartości informacji jaką ze sobą niosą.

3.2.3 Różnorodność i złożoność

Kolejne atrybuty czyli różnorodność i złożoność są ze sobą powiązane i odnoszą się do dużego zróżnicowania źródeł pochodzenia danych i ich formatu. Informacje mogą przychodzić w postaci ustrukturyzowanej jak i m.in. niestrukturalnych dokumentów tekstowych, wiadomości e-mail, materiałów audio i video, transakcji finansowych. Natomiast źródłami ich pochodzenia mogą być wszystkie systemy, które generują i gromadzą dane czyli np.:

- wewnętrzne systemy organizacji takie jak systemy księgowe, kadrowe i transakcyjne,
- źródła zewnętrzne - dane ze stron internetowych (także te nieindeksowane przez wyszukiwarki - pochodzące z głębokiego internetu), blogów, wiadomości tweet i forów internetowych,
- sklepy, usługodawcy i instytucje finansowe,
- instytucje zdrowia.

3.2.4 Zmienność

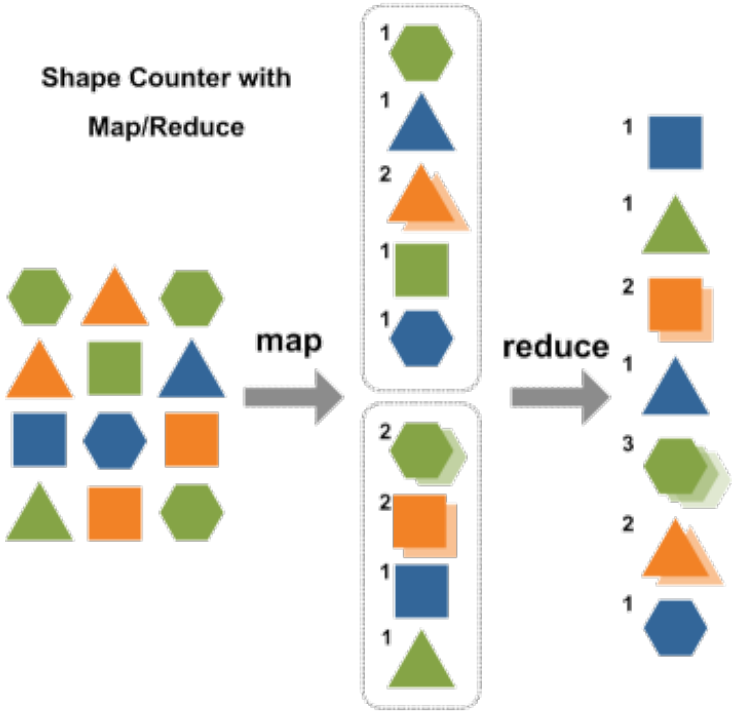
Dane typu Big Data podlegają okresowym trendom i wahaniom. Napływają ze zmiennym w czasie natężeniem. Mają na to wpływ wydarzenia z wielu dziedzin naszego życia - od pór roku do wydarzeń z dziedziny polityki czy sportu. Przykładem jest np. zainteresowanie inną tematyką w czasie Bożego Narodzenia niż w czasie lata lub bardziej dynamiczne zmiany na rynkach finansowych.

3.2.5 Wartość

Wartość danych napływających z systemów jest trudna do analizy, ponieważ struktura tych danych charakteryzuje się dużą złożonością, a korzyść płynąca z analizy jest ukryta. Obecnie istnieją już systemy, które potrafią radzić sobie z dużym wolumenem informacji, ale to jak te informacje zostaną wykorzystane zależy już od człowieka.

3.3 Paradygmat MapReduce

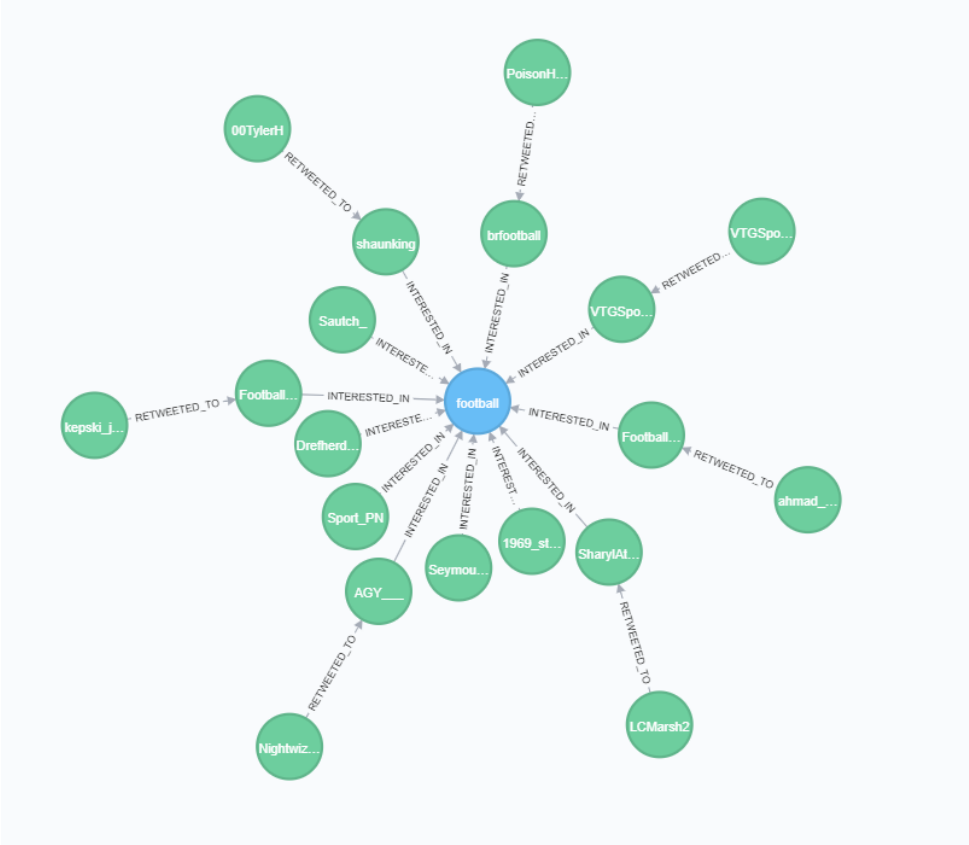
Większość systemów przetwarzających duże ilości danych opiera swoje działanie o paradygmat *MapReduce*. MapReduce polega na dzieleniu zbioru danych wejściowych na mniejsze niezależne od siebie podzbiory, które są następnie przetwarzane przez równoległe zadania typu *map*. Wynik wyjścia z zadań typu map jest sortowany, a następnie podawany na wejście zadań typu *reduce*. Zastosowanie tego paradygmatu skutkuje zwiększeniem wydajności dzięki przetwarzaniu równoległemu niezależnych bloków danych co może zostać wykorzystane przez zastosowanie wielu procesorów.
MapReduce przetwarza wyłącznie pary *<klucz, wartość>*. Dlatego większość operacji polega na znalezieniu elementów spełniających pewne kryteria, umieszczenie ich w zbiorze oraz zwiększenie licznika sygnalizującego częstotliwość występowania elementu tak jak pokazano na ilustracji 4.2..



Rysunek 3.2: Przykład zastosowania paradygmatu *MapReduce*.

3.4 Wymagania stawiane przed systemami Big Data

Przetwarzanie danych napływających w sposób strumieniowy wiąże się z dużym wyzwaniem związanym z koniecznością niezwłocznej obsługi danych pojawiających się w dużej ilości i z dużą prędkością. Są to wymagania stawiane przede wszystkim przed platformami zarządzania danymi - ich przetwarzaniem oraz przechowywaniem. Jednymi z najbardziej znanych obecnie rozwiązań tego typu są *Apache Hadoop* oraz *Apache Spark*, które zostaną omówione w dalszej części pracy przy okazji sformułowania wymagań stawianych przed aplikacją zbudowaną na potrzeby tej pracy dyplomowej i dostępnych rozwiązań.
Innym napotkanym problemem jest duża złożoność danych, szczególnie danych niestrukturalnych, która stawia wyzwania związane z efektywnym przechowywaniem informacji oraz przeszukiwaniem bazy danych. Sztywno zdefiniowane relacyne bazy danych nie odpowiadają wymaganiom stawianym przed systemami Big Data. Bazami, które znalazły swoje zastosowanie w takich systemach są bazy *NoSQL*. Dzięki ich zastosowaniu można uzyskać korzyści związane z poprawą zrozumienia danych, możliwość gromadzenia danych niestrukturalnych, skalowalność oraz elastyczność bazy danych. Przykładem baz danych typu NoSQL są *Cassandra* oraz *Neo4j*, które tak jak i Apache Hadoop oraz Apache Spark zostaną omówione w dalszej części pracy.



Rysunek 3.3: Przykład zastosowania grafowej bazy danych Neo4j.

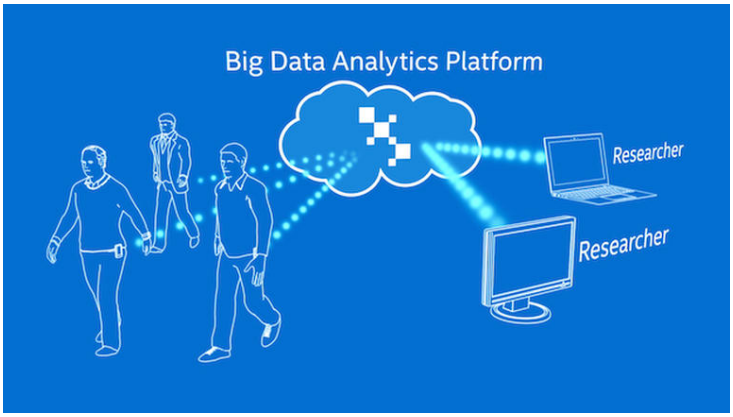
3.5 Przykłady zastosowania systemów Big Data

Systemy Big Data mogą mieć bardzo szerokie zastosowanie. Poniżej podano listę istniejących już zastosowań, są to np.:

– wykorzystanie danych pogodowych generowanych przez Interdyscyplinarne Centrum Mo-

delowania Matematycznego i Komputerowego ICM Uniwersytetu Warszawskiego do m.in. prognozowania ilości energii produkowanej przez elektrownie wiatrowe i słoneczne, prognozowania zagęszczenia ruchu, pomoc sieciom handlowym w prognozowaniu jakie produkty będą się lepiej sprzedawały i powinny być w promocyjnej cenie,

– wykorzystanie zegarków, należących do układów typu wearables, wraz z smartfonami oraz systemów Big Data do analizy zdrowia pacjenta np. przy leczeniu choroby Parkinsona, ale też w innych chorobach gdzie ważne jest codzienne monitorowanie,



Rysunek 3.4: Przykład zastosowania układów wearables, telefonów komórkowych i systemu Big Data do analizy stanu zdrowia pacjentów.

– dzięki zbieraniu i analizie danych takich jak np. temperatura, czas otwarcia drzwi, ilość kursów z wind oraz specyfikacji sprzętu przewidywany jest czas kiedy należy wykonać naprawę, a służby serwisowe wysyłane są tylko wtedy i tylko tam gdzie są rzeczywiście potrzebne,

– gromadzenie danych ze stron internetowych wraz z wyrażanymi przez internautów opiniami i analiza ich za pomocą systemów wykorzystujących sztuczną inteligencję służy firmom, ale także partiom politycznym, do monitorowania opinii na temat produktów oraz nastrojów społecznych.

Rozdział 4

Analiza sentymentu wypowiedzi

Zagadnieniem dobrze ilustrującym postęp technologiczny jest prędkość rozprzestrzeniania się informacji. W starożytności informacja była przekazywana zazwyczaj przez posłańców, od których losów zależało czy i kiedy informacja zostanie dostarczona do nadawcy. Tak było w przypadku posłańca Filippidesa, który według legendy podążał po wygranej bitwie pod Maratonem do Aten, aby uprzedzić Greków przed zbliżającym się atakiem Persów. W dzisiejszych czasach coraz częściej informacja jest dostępna na ekranach urządzeń mobilnych takich jak telefon komórkowy w czasie rzeczywistym zaraz po wystąpieniu jakiegoś zdarzenia. Dlatego bardzo ważne stało się monitorowanie opinii i nastojów społecznych. Dostępne obecnie narzędzia pozwalają klasyfikować wydźwięk tekstu jako pozytywny, negatywny albo neutralny.

4.1 Przetwarzanie języka naturalnego

4.1.1 Ustalanie znaczenia

Podstawową operacją języka naturalnego jest ustalenia znaczenia zdania zwane także semantyką. Jego zrozumienie wymaga w systemach informatycznych określenia znaczenia zdania oraz opracowania metody jego zapisu do czego potrzebne są opis faktów danych w zdaniu oraz wyciąganie wniosków z posiadanych już informacji.
Ustalenie znaczenia wyrażenia zdania w języku naturalnym polega na wyznaczeniu występujących w nim obiektów oraz zachodzących między nimi relacji. Do analizy pełnego znaczenia wymagana jest szeroka wiedza o świecie odnosząca się do konretnego kontekstu. Na tym etapie przetwarzania tekstu pomocne są informacje o powiązaniach między słowami. W zależności od zastosowania przydatna może być na przykład wiedza o tym, że kot jest ssakiem, a rafineria rodzajem przedsiębiorstwa.
Do reprezentacji semantyki języka naturalnego można wykorzystać mechanizmy formalne, które odpowiadają praktycznym potrzebom dokonania interpretacji. Jendym z przykładów takich mechanizmów jest rachunek predykatów I rzędu, który pozwala zapisać czy fakt jest prawdziwy lub fałszywy, umożliwia zapisywanie pytań za pomocą użycia zmiennych, a także posiada opracowane metody wnioskowania. Inną metodą reprezentowania znaczenia zdania jest *Teoria Reprezentacji Dyskursu - DRT* (ang. *Discourse Representation Theory*), która polega na przekształcaniu drzewa rozbioru znaczenia zdania na strukturę zwaną *DRS* (ang. *Discourse Representation Structure*). Niestety żaden ze znanych mechanizmów nie jest w stanie odzwierciedlić całej złożoności procesów powiązanych z rozumieniem języka naturalnego. Przyjmując jedną z istniejących metod lub tworząc nową musimy zmierzyć się z wybraniem mechanizmu, który wobec postawionych wymagań najlepiej poradzi sobie z rozległością skali znaczeń jakie chcemy reprezentować, stopniem skomplikowania semantyki oraz kosztem jej uzyskania. Dlatego większość systemów informatycznych nie korzysta z wyrafinowanych reprezentacji znaczenia, ale ogranicza ją do najprostszych scenariuszy.

4.1.2 Powiązania międzyzdaniowe

Analiza tekstu jest złożonym zadaniem. Wymaga zrozumienia zależności pomiędzy występującymi faktami oraz do jego pełnego zrozumienia potrzebna jest wiedza, którą dysponuje człowiek. Dlatego analiza tekstu jest uznawana za problem *AI-zupełny* (ang. *AI - Artifical Intelligence* lub po polsku *SI - Sztuczna Inteligencja*). Zrozumienie kolejnej części tekstu wymaga umiejętnego wyciągania wniosków z poprzedniej części tekstu i powiązania ich z wiedzą nabytą z innych źródeł.
Opis dłuższego tekstu wymaga odtworzenia powiązań pomiędzy kolejnymi zdaniami, ale z powodu ilości możliwych kombinacji sekwencji zdań można opisać tylko wybrane zjawiska oraz wykluczyć niektóre typy powiązań. Kolejną trudnością jest rozstrzyganie do jakich obiektów odnoszą się wyrażenia wskazujące, ponieważ w tekstach stosowane są sposoby opisu tego samego obiektu w różny sposób np. wszystkie frazy odnoszące się do tej samej osoby - *Janek, kolega Maćka, mały chłopiec, ten z prawej, najmłodszy w rodzinie*. Inną ważną kwestią są części wpływające na ciągłość tekstu (nawiązanie do porzedniego tekstu lub wypowiedzi, ciągłość opisów).

4.1.3 Reprezentacja języka

Analiza nastrojów społecznych wypowiedzi jest możliwa po poddaniu tekstu zamianie na reprezentację, którą mogą posługiwać się systemy informatyczne. Poniżej przedstawiono opis kilku takich reprezentacji:

- bag of words - najpopularniejszy sposób reprezentacji tekstu w postaci zestawu wyrazów z przyporządkowanymi im liczbami wystąpień w tekście np. zdanie *Jan lubi oglądać filmy, a Maria także lubi je oglądać* można zapisać w notacji *JSON* (ang. *JSON - JavaScript Object Notation*) jako *{"Jan": 1, "lubi": 2, "oglądać": 2, "filmy": 1, "a": 1, "Maria": 1, "także": 1, "je": 1}*; Innym przykładem zastosowania takiej reprezentacji jest wykorzystanie jej w mechanizmach odpowiadających za filtrowanie wiadomości e-mail.; Wadą takiego podejścia jest utrata informacji o kolejności wyrazów w zdaniu i powiązaniach między nimi.

- reprezentacja wektorowa - sposób reprezentacji tekstu w postaci wektorów słów występujących w dokumencie tekstowym w przestrzeni N-wymiarowej; Wadą takiego rozwiązania jest w niektórych przypadkach ilość występujących słów.

- reprezentacja grafowa - podejście rozszerzające reprezentację wektorową, w której słowa są węzłami połączonymi ze sobą krawędziami jeśli występują razem w tekście; Istnieje możliwość zaobrazowania kolejności występowania słów z wykorzystaniem krawędzi skierowanych.

4.2 Sentyment wypowiedzi

Jako istoty ludzkie posiadamy zdolność do rozpoznawania cudzych emocji po treści wypowiedzi i towarzyszących jej czynników pozawerbalnych, której wydźwięk określa się jako stosunek lub postawa pozostająca w korelacji do pewnego zdarzenia lub sytuacji. Treści publikowane w internecie w formie tekstu niosą ze sobą tylko reprezentację tekstową, dlatego odczytanie nastroju wyrażanego w taki sposób jest trudnym zadaniem. Tym bardziej złożonym procesem wymagającym dokładnego przeanalizowania każdego słowa jest zadanie odczytania wydźwięku wypowiedzi przez napisany system.
Dziedziną zajmującą się analizą wypowiedzi jest przetwarzanie języka naturalnego (ang. *NLP - natural language processing*), gdzie językiem naturalnym określa się język stosowany przez ludzi do komunikacji interpersonalnej. Zadaniem systemów rozumiejących język naturalnych jest

przekształcenie go na formę bardziej przyjazną dla komputerów. Natomiast podzbiór tych systemów służący do określania sentymentu zwraca ogólną informację o wydźwięku w ustalonej skali. Przykładowo analiza sentymentu w uważanej za punkt odniesienia dla takich algorytmów bibliotece CoreNLP, napisanej przez pracowników i studentów amerykańskiego Uniwersytetu Stanforda, zwraca informację o sentymencie w pięciostopniowej skali: bardzo negatywny, negatywny, neutralny, pozytywny, bardzo pozytywny. Jest to ogólna informacja, ale nawet taka w postaci statystyk jest w stanie posłużyć jako cenne źródło informacji.

Rysunek 4.1: Tweet firmy *Google* zamieszczony z okazji Dnia Dziękczynienia jako przykład tekstu o pozytywnym wydźwięku.

4.2.1 Wymagania związane z oceną sentymentu

Algorytmy analizujące sentyment muszą poradzić sobie z następującymi wymaganiami:

- złożoność języka naturalnego,
- niejednoznaczność wypowiedzi np. wieloznaczność słowa zamek: *akcja Zemsty Fredry dzieje się na zamku w Odrzykoniu* lub *zamek w moich drzwiach nie chce się otworzyć* lub syntaktyczna niejednoznaczność gdy w jednej części zdania znajduje się pozytywny wydźwięk, a w kolejnej negatywny: *pogoda była okropna, ale obiad bardzo nam smakował*,
- specyfika stosowanego języka np. w internecie z powodu niejednokrotnie nie zachowanych zasad gramatycznych lub wiadomości nie niosących ze sobą żadnej treści,
- neologizmy np. retweet,
- idiomy np. "urwanie głowy"
- problemy z rozpoznawaniem nazw np. *byliśmy wczoraj na K2* - czy chodzi o film czy o szczyt górski,
- problem wyrwania wypowiedzi z kontekstu,
- tekst o charakterze sarkastycznym.

4.2.2 Techniki badania sentymentu

Techniki badania sentymentu wypowiedzi dzielimy na dwa rodzaje: korzystające ze słowników, które korzystają ze zbudowanych wcześniej list słów kluczowych z określonym sentymentem oraz oparte na klasyfikatorach, które wykorzystują techniki maszynowego uczenia się (ang. *machine learning*). Podstawowe techniki badania sentymentu to:

- słownik (ang. *lexicon based approach*) - klasyfikator ten bada tekst na podstawie wystąpień słów przy wykorzystaniu słownika składającego się ze słów z przypisanym pozytywnym lub negatywnym wydźwiękiem; sentyment pojedynczego słowa określa się jako iloraz prawdopodobieństwa wystąpienia z pozytywnym sentymentem do prawdopodobieństwa wystąpienia z negatywnym; Wadą takiego podejścia jest brak analizy powiązania między słowami.

- naiwny klasyfikator Bayesa (ang. *naive Bayes classifier*) - probabilistyczny klasyfikator wykorzystujący techniki maszynowego uczenia się, opierający się o założenie o niezależności słów oraz wykorzystujący założenie, że teksty o charakterze pozytywnym charakteryzują się określonym słownictwem, a te o charakterze negatywnym charakteryzują się innym; Podejście to zakłada także, że tekst, w którym występuje więcej słów o charakterze z kategorii pozytywnej lub negatywnej powinien zostać zaklasyfikowany do tej kategorii;

- technika maksymalnej entropii (ang. *maximum entropy technique*) - podejście szacujące rozkład prawdopodobieństwa opierające się na założeniu, że rozkład ma maksymalną entropię, jeśli dane nie są dobrze znane; Entropia zwana także miarą niepewności rozkładu jest kolejną metodą wykorzystującą techniki maszynowego uczenia się.

- metoda wektorów nośnych (ang. *support vector machines*) - technika *machine learning* opierająca się o ideę hiperpłaszczyzny dzielącej teksty na pozytywne oraz negatywne z jak najmniejszym marginesem; Celem jest znalezienie funkcji, dla której błąd sklasyfikowania tekstu będzie najmniejszy; Programy wykorzystujące tą metodę dobrze radzą sobie z dużą ilością słów, ale oznaczenie części z nich jako nieistotne może powodować utratę cześci informacji.

4.2.3 Zastosowanie badania wydźwięku wypowiedzi z internetu

Jak wynika z badań przeprowadzonych na grupie ponad 2000 dorosłych Amerykanów 81% internautów przynajmniej raz szukało opinii o produkcie w internecie, a 20% Amerykanów robi to na co dzień. Około 80% użytkowników internetu spośród wspomnianej grupy badawczej przyznało, że opinie przeczytane w internecie miały znaczny wpływ na dokonane przez nich zakupy. Wyniki tych badań pokazują jak bardzo wydźwięk informacji, opinii, recenzji oraz poglądów w internecie ma wpływ na zachowania ludzi. Najczęściej spotykane zastosowania analizy sentymentu wypowiedzi zamieszczanych w internecie to:

- analizowanie opinii wyrażanych na temat produktów np. krótko po premierze nowego produktu firmy chcą wiedzień jak są one odbierane tak aby móc w porę zaplanować wprowadzenie poprawek lub aby prognozować wyniki sprzedaży oraz cenę akcji, firmy próbują także docierać do osób krytykujących aby przekonać ich do zmiany zdania lub żeby krytyków konkurencji zachęcić do zakupu swoich produktów,

- badanie opinii na temat firm np. jako pracodawców lub ich odbiór na rynku,

- analiza opinii na temat ugrupowań politycznych i polityków np. dotycząca obecnej sytuacji lub przyszłych decyzji oraz nastawienia badanych grup społecznych,

– badanie nastrojów społecznych podczas wydarzeń sportowych może posłużyć np. do uzyskania wiedzy jak odbierane są decyzje właścicieli klubów piłkarskich.

Rozdział 5

Wymagania funkcjonalne i niefunkcjonalne

Jak już zostało wspomniane w rozdziale dotyczącym serwisu Twitter, przy okazji omówienia dostępnych narzędzi, nie ma obecnie na rynku aplikacji, która umożliwiałaby śledzenie występowania dowolnego słowa w wiadomościach zamieszczanych w czasie rzeczywistym w tym serwisie, rysowałaby wykres zależności pomiędzy użytkownikami, analizowałaby nastroje społeczne użytkowników z wyświetleniem informacji o nastroju wyrażanym w poszczególnych wiadomościach oraz pozwalałaby na analizowanie danych historycznych i zamieszczanych w czasie rzeczywistym. Głównym celem tej pracy dyplomowej jest stworzenie aplikacji, która posiadałaby wspomniane funkcjonalności.

5.1 Wymagania funkcjonalne

Główną funkcjonalnością, którą powinna zapewnić budowana aplikacja jest dostęp do usystematyzowanych danych pochodzących z sewisu Twitter, które będą nieść ze sobą informację o sentymencie. Dane te powinny być także przechowywane w taki sposób, aby móc zapewnić do nich dostęp w dowolnym momencie oraz spoza zaimplementowanego narzędzia.
Wymagania funkcjonalne, które dotyczą budowanego rozwiązania to:

- **przetwarzanie danych z serwisu Twitter** - aplikacja powinna przetwarzać tweety użytkowników serwisu Twitter, do których dostęp można uzyskać przez Streaming API tego serwisu, które zostało szczegółowo omówione w rozdziale 2.;
- **filtrowanie napływających danych po słowie kluczowym** - dane napływające w czasie rzeczywistym powinny być filtrowane pod względem zawartości w treści wiadomości słowa kluczowego, które zostało określone przez użytkownika za pomocą graficznego interfejsu aplikacji;
- **zapis napływających danych** - budowane narzędzie powinno zapisywać przetworzone, uporządkowane i dotyczące wybranego słowa kluczowego dane w lokalnej bazie danych, która umożliwiałaby dostęp do nich przez swój wbudowany pulpit w dowolnym momencie oraz spoza zaimplementowanego narzędzia;
- **duża częstotliwość pobierania danych** - aplikacja powinna pobierać informacje w krótkich odstępach czasu, ponieważ serwis Twitter charakteryzuje duża ilość informacji przesyłanych w każdej sekundzie;
- **prezentowanie danych historycznych** - narzędzie powinno prezentować dane historyczne zgromadzone podczas przetwarzania danych napływających wówczas w czasie rzeczywistym;

- **prezentowanie podstawowych danych napływających w czasie rzeczywistym** - aplikacja powinna umożliwiać analizowanie podstawowych informacji o wiadomościach i użytkownikach, które będą napływać w czasie rzeczywistym;
- **dostęp do szczegółowej informacji o użytkowniku** - implementowane narzędzie powinno umożliwiać dostęp do informacji o każdym użytkowniku zainteresowanym wybranym słowem kluczowym, którego wiadomość udało się zarejestrować podczas gromadzenia danych z wykorzystaniem Streaming API serwisu Twitter;
- **dostęp do szczegółowej informacji o wiadomości** - aplikacja powinna wyświetlać szczegółową informację, o każdej wiadomości zawierającej wybrane słowo kluczowe i zapisanej podczas gromadzenia danych z wykorzystaniem Streaming API;
- **prezentowanie informacji statystycznej o sentymencie na dany temat** - narzędzie powinno prezentować statystyki sentymentu użytkowników serwisu Twitter, którzy w swoich wiadomościach zawarli wybrane słowo kluczowe i których wiadomości udało się zapisać podczas analizy danych napływających w czasie rzeczywistym;
- **możliwość jednoczesnej analizy danych historycznych i napływających w czasie rzeczywistym** - aplikacja powinna umożliwiać jednoczesną analizę danych historycznych i napływających w czasie rzeczywistym bez konieczności ponownego uruchamiania aplikacji lub zatrzymywania jednej z analiz;
- **przyjazny interfejs graficzny** - narzędzie powinno posiadać wygodny, łatwy do nauczenia oraz prosty interfejs graficzny.

5.2 Wymagania niefunkcjonalne

Tworzone rozwiązanie będzie także analizowało wydźwięk wypowiedzi użytkowników. Wymagania niefunkcjonalne, które powinna spełniać przygotowywana aplikacja to:

Spełnienie głównych założeń systemu czasu rzeczywistego

Głównym przypadkiem biznesowym, dla którego tworzona jest wspomniana aplikacja jest sytuacja dużego i globalnego zainteresowania pewnym tematem, które objawia się odnoszeniem się do niego w wiadomościach zamieszczanych przez użytkowników serwisu Twitter. Można stwierdzić, że przygotowywane narzędzie będzie przykładem systemem czasu rzeczywistego jeśli system ten będzie "*urządzeniem techniczne, którego wynik i efekt działania będzie zależny od chwili wypracowania tego wyniku*". Wspólną cechą definicji takiego systemu jest "*zwrócenie uwagi na równoległość w czasie zmian w środowisku oraz obliczeń realizowanych na podstawie stanu środowiska*".

Płynna obsługa danych

Budowany system powinien przetwarzać dane w czasie nie większym niż tempo napływania nowych informacji. Koniecznością jest zatem skorzystanie z narzędzi umożliwiających sprawne przetwarzanie danych, ale także ich zapis do bazy w czasie nie większym niż czas trwania określonego okna czasowego.

Możliwość działania aplikacji i analizowania danych historycznych w trybie offline

Dane zapisane podczas sesji korzystania ze strumienia danych serwisu Twitter będą zapisywane w lokalnej bazie danych, dlatego stworzone narzędzie powinno umożliwiać analizowanie

danych historycznych bez połączenia z internetem.

Jednorazowe przetwarzanie informacji ze strumienia danych

Aplikacja powinna tylko jeden raz przetwarzać i zapisywać do bazy danych informacje pozyskane ze strumienia. Jeśli w bazie istnieje już informacja o użytkowniku to nowe wiadomości powinny być z nim powiązane.

Niezawodność
System powinien charakteryzować się niezawodnością podczas pracy z dużą ilością danych napływających w krótkich odstępach czasu oraz jak najbliższym prawdy określaniem sentymentu wypowiedzi zawartej w wielu wiadomościach.

5.3 Podsumowanie

System napisany na potrzeby tej pracy dyplomowej powinien spełniać wszystkie z wymienionych wymagań niefunkcjonalnych, ponieważ nie spełnienie nawet jednej z nich może spowodować, że aplikacja będzie nieużyteczna. Postawione wymagania funkcjonalne i niefunkcjonalne, łącząc się z opisem serwisu Twitter, narzędzi Big Data oraz przetwarzania języka naturalnego, definiują potrzeby jakie powinny umożliwiać narzędzia wybrane do jej budowy oraz samo narzędzie. Zostanie to omówione w następnych rozdziałach.

Rozdział 6

Wybór narzędzi

Rozdział 7

Aplikacja Twitter Analyser

Rozdział 8

Badania i wnioski

Rozdział 9

Podsumowanie

Bibliografia