Politechnika Wrocławska

Wydział Informatyki i Zarządzania
kierunek studiów: Inżynieria Systemów
specjalność: brak

Praca dyplomowa – magisterska

Metoda i narzędzie do wydobywania znaczenia z zadanego tekstu i wyszukiwania tekstów o zbliżonym znaczeniu

Piotr Grzesiak

słowa kluczowe:
Reprezentacja znaczenia tekstu
Podobieństwo tekstów
Rekomendacja piosenek katolickich
Przetwarzanie języka naturalnego

Celem pracy jest zbadanie oraz porównanie wybranych metod wydobywania znaczenia z zadanego tekstu i wyszukiwania tekstów o zbliżonym znaczeniu. Zadanym tekstem jest fragment biblii a tekstami, wśród których odbywa się wyszukiwanie są teksty piosenek o tematyce religijnej. Zwieńczeniem pracy jest implementacja najlepszej metody w postaci narzędzia rekomendacji piosenek katolickich - PiosenKatoR.

opiekun pracy dyplomowej	dr inż. Paweł Stelmach *Tytuł/stopień naukowy/imię i nazwisko*	..... *ocena*	..... *podpis*

Ostateczna ocena za pracę dyplomową

Przewodniczący Komisji egzaminu dyplomowego	..... *Tytuł/stopień naukowy/imię i nazwisko*	..... *ocena*	..... *podpis*

*Do celów archiwalnych pracę dyplomową zakwalifikowano do:**
a) kategorii A (akta wieczyste)
b) kategorii BE 50 (po 50 latach podlegające ekspertyzie)
** niepotrzebne skreślić*

pieczątka wydziałowa

Wrocław 2019

Streszczenie

Niniejsza praca zawiera opis przeprowadzonych badań nad różnymi metodami reprezentacji znaczenia z zadanego tekstu i wyszukiwania tekstów o zbliżonym znaczeniu. Zadanym tekstem jest fragment biblii odczytywany w poszczególne dni podczas celebracji mszy świętej w kościele katolickim a tekstami, wśród których odbywa się wyszukiwanie podobnego znaczenia są teksty piosenek o tematyce religijnej. Ideą pracy było wyznaczenie najlepszej metody rekomendującej piosenki do biblijnych czytań pod względem znaczenia.

W pracy opisany został przegląd istniejących metod wydobywania znaczenia oraz podobieństwa między tekstami wraz z krytyczną analizą źródeł i możliwym zastosowaniem w systemach rekomendacji. Opisano również proces pracy z tekstem w dziedzinie przetwarzania języka naturalnego oraz cały przebieg badań wraz z opisem implementacji.

Zwieńczeniem pracy jest implementacja najlepszego rozwiązania w postaci narzędzia do rekomendacji katolickich piosenek do czytań biblijnych - PiosenKatoR.

Abstract

Following thesis contains a description of performed experiments on different methods of meaning representations of a given text and searching texts with similar meaning. The given text is a text of the bible read on particular days during the holy mass celebration in catholic church, and the texts among which the searching is performed are religious songs. The main idea was to find the best method for recommending songs for bible readings in the matter of meaning.

The paper contains an overview of existing methods for meaning representation and similarity between texts with critical sources analysis and possible usage in recommendation systems. The workflow in natural language processing and the whole set of experiments with implementation included was also described.

The culmination of work is an implementation of the best solution as a tool for catholic songs recommendation to bible readings - PiosenKatoR.

Spis treści:

1. Wstęp

1.1. Wprowadzenie i pojęcia podstawowe

Analiza i przetwarzanie danych odgrywa współcześnie kluczową rolę w działaniu większości organizacji. Począwszy od małych prywatnych przedsiębiorstw, poprzez międzynarodowe korporacje a na instytucjach publicznych kończąc. Niezależnie od szczebla i zakresu działalności – dane, czyli zapisane i możliwe do przetworzenia informacje, stanowią fundament, z którego wynikają wszelkie aktywności i modele biznesowe [1, 2].

Jednym z przykładów danych są *dane tekstowe*. Są to różnego rodzaju artykuły, dokumenty czy wypowiedzi zapisane w formie tekstu – ciągu znaków zrozumiałych dla człowieka. O takich tekstach można powiedzieć, że sformułowane zostały w *języku naturalnym*, czyli języku tworzonym i rozwijanym przez człowieka podczas procesu komunikacji [2].

Jak podaje literatura, przetwarzanie i analiza takich danych z wykorzystaniem komputera nazywa się *przetwarzaniem języka naturalnego* (ang. natural language processing), *eksploracją tekstu* (ang. text mining) lub po prostu *analizą tekstu* (ang. text analytics) [2].

Dane tekstowe są trudne do przeanalizowania ze względu na swoją nieustrukturyzowaną postać. Większość znanych technik uczenia maszynowego i metod statystycznych związanych z przetwarzaniem danych dostosowana jest bowiem do pracy z danymi liczbowymi [2].

Obecnie jednak dzięki zastosowaniu różnych metod reprezentacji tekstu w formie liczbowej, efektywne przetwarzanie i analiza tekstu nie tylko staje się możliwa, lecz także niesie ze sobą dotychczas nieosiągalne korzyści. Wielkie firmy takie jak Google, Spotify czy Netflix wykorzystują rozmaite zadania przetwarzania języka naturalnego do udoskonalania procesu wyszukiwania informacji czy proponowania nowych produktów [3].

Szczególnym rodzajem zadań związanych z analizą tekstu jest badanie i porównywanie *znaczenia*. Najbardziej intuicyjne wydaje się rozumienie znaczenia tekstu jako sposobu na wyrażenie jego treści – pewnego ‘uchwycenia’ tego o czym dany tekst traktuje. Istnieją różne metody i algorytmy sprowadzające teksty do pewnej reprezentacji znaczenia – najczęściej w postaci zbioru terminów lub wektora liczb. Wektory takie można potem porównywać ze sobą stosując różne miary odległości i *podobieństwa* [4].

Mając zbiór przetworzonych tekstów wraz ze znajomością ich znaczenia możliwe jest by nowy, nieznany wcześniej dokument tekstowy ocenić czy jest on podobny pod względem znaczeniowym do któregoś z posiadanych już przetworzonych tekstów w *korpusie*, czyli zbiorze *dokumentów*.

Jednym z możliwych zastosowań takiego rozwiązania jest *rekomendacja* (inaczej propozycja) piosenek w oparciu o podobne znaczeniowo teksty [5].

1.2. Motywacja

Motywacją do zajęcia się tematem znaczenia i wyszukiwania tekstów podobnych jest brak narzędzia wspomagającego muzyków kościelnych, np. organistów przy wyborze repertuaru pasującego pod względem znaczeniowym do fragmentów biblii odczytywanych danego dnia podczas celebracji mszy świętej. Muszą oni samodzielnie oceniać podobieństwo biblijnych tekstów i piosenek ze śpiewnika. Istnienie takiego narzędzia nie tylko przyspieszyłoby proces doboru pieśni, lecz także poprawiło jego jakość.

1.3. Cel i zakres pracy

Celem pracy jest zbadanie skuteczności i porównanie wybranych metod wydobywania znaczenia z zadanego tekstu i wyszukiwania tekstów o podobnym znaczeniu. Zadanym tekstem jest fragment biblii (dalej zwany czytaniem) a tekstami, wśród których odbywa się wyszukiwanie są teksty piosenek o tematyce religijnej. Zwieńczeniem badań jest implementacja najlepszej metody w postaci narzędzia, którego wynikiem działania będzie zbiór zarekomendowanych tytułów piosenek najlepiej pasujących do danego czytania pod względem podobieństwa w znaczeniu.

Rysunek 1 - Idea kryjąca się pod tematem pracy

Zakres pracy:

- Przegląd literatury, w tym:
  - Zapoznanie z istniejącymi metodami przetwarzania języka naturalnego i analizy tekstu
  - Zapoznanie się z istniejącymi metodami reprezentacji znaczenia tekstu
  - Zapoznanie się z istniejącymi metodami badania podobieństwa między tekstami
  - Zapoznanie się z istniejącymi technikami rekomendacji piosenek
- Zdobycie danych
- Przetworzenie danych wraz z czyszczeniem
- Implementacja różnych metod reprezentacji znaczenia
- Implementacja obliczania podobieństwa między różnymi reprezentacjami znaczenia tekstów
- Porównanie i ewaluacja metod na podstawie zbioru rekomendacji ewaluacyjnych
- Implementacja najlepszej metody w postaci narzędzia

2. Przegląd literatury

Wydobywanie znaczenia z tekstu i wyszukiwanie tekstów podobnych było tematem badań wielu naukowców.

Najstarszym i najpowszechniejszym podejściem jest koncepcja opierająca się o model przestrzeni wektorowej (ang. vector space model, w skróce: VSM) [6]. Reprezentuje on zbiór tekstów jako wielowymiarową przestrzeń wektorową. Pojedynczy tekst jest charakteryzowany poprzez liczby wystąpień każdego z wyrazów wchodzących w skład całego zbioru dokumentów zapisane w postaci wektora. Porównywanie znaczenia tekstów polega na obliczeniu odległości ich wektorów za pomocą przyjętej miary. Najbardziej podstawowymi przedstawicielami modelu przestrzeni wektorowej jest Bag of words (BOW) i jego modyfikacja nadająca wagi TF-IDF.

Minusem takiego rozwiązania jest brak możliwości uwzględnienia zależności czy kolejności występowania poszczególnych wyrazów. Teksty są traktowane jako zbiory niezależnych słów co sprowadza ocenę podobieństwa do oceny podobieństwa w występowaniu tych samych terminów [7]. Rozwiązaniem takiego problemu może być użycie n-gramów, ale z kolei prowadzi to do zwiększenia wymiarów wektorów i zjawiska tzw. rzadkości danych [8].

Obecnie wielką popularnością w komercyjnych zastosowaniach przetwarzania języka naturalnego cieszą się metody tzw. osadzania słów (ang. word embbeddings). Opierają się one o założenia semantyki dystrybucyjnej, której podstawowa hipoteza głosi, iż słowa występujące w podobnych kontekstach w dużych zbiorach danych mają podobne znaczenie [9]. Każde słowo czy wyrażenie rzutowane jest na n-wymiarowy wektor liczb rzeczywistych [10]. W 2013 roku naukowcy z firmy Google pod przewodnictwem Tomasa Mikolova, w publikacjach *Efficient Estimation of Word Representations in Vector Space* [11] oraz *Distributed Representations of Words and Phrases and their Compositionality* [12] zaproponowali i udoskonalili model Word2vec. Wykorzystując architekturę płytkich sieci neuronowych z jedną warstwą ukrytą, pozwala on na reprezentację wyrazów w postaci wektorów oddających znaczenie i wzajemne podobieństwa syntaktyczne i semantyczne o wiele lepiej niż dotychczasowe znane metody. Co więcej, zaproponowane rozwiązanie pozwala na szybkie uczenie modeli na bardzo dużych zbiorach [9]. Inne powszechnie znane modele osadzania słów to GloVe rozwijany na Uniwersytecie Stanforda [13] oraz FastText opracowany przez firmę Facebook [14]. W Internecie dostępne są gotowe do użycia modele, w tym w języku polskim.

Niestety, osadzanie słów – jak wskazuje nazwa metody – jest metodą reprezentacji w postaci wektorów pojedynczych słów, a nie całych tekstów. W celu wykorzystania wyżej wspomnianych metod do reprezentacji dokumentów można zastąpić cały wektor danego słowa poprzez zsumowanie jego elementów lub obliczenie średniej. Jednakże, wciąż ignorowany będzie porządek słów czy ogólne znaczenie całych zdań [15].

Na bazie metod pozwalających na reprezentację słów w postaci wektorów, wspominany już Tomas Mikolov wraz ze współpracownikiem Quoc Le zaproponowali w 2014 roku inny sposób na wektorową reprezentację zdań, akapitów a nawet całych dokumentów [8]. Model zwany Doc2vec (lub Paragraph Vector) może być skutecznie wykorzystywany do mierzenia podobieństwa między tekstami, co potwierdziły badania prowadzone przez IBM [16].

Z drugiej strony, w przypadku trenowania modelu na zbyt małym zbiorze danych, jego skuteczność może nie być zadowalająca. Również słowa unikalne występujące w dokumencie znacznie zaburzają całą reprezentację.

Innowacyjne podejście do reprezentacji tekstów prezentują Jernej Flisar i Vili Podgorelec w artykule z 2018 roku pt. *Document Enrichment using DBPedia Ontology for Short Text Classification* [17]. W przypadku zadań klasyfikacji krótkich tekstów,

wymienione wyżej metody nie są aż tak skuteczne ze względu na mały zasób słów. Aby poradzić sobie z problemem badania podobieństwa między krótkimi tekstami naukowcy użyli narzędzia DBpedia Spotlight w celu wzbogacenia dokumenty o wiedzę reprezentowaną w ontologii DBpedia. Badania wykazały, że dodanie do reprezentacji Bag of words dodatkowych słów z bazy wiedzy zwiększa skuteczność klasyfikacji.

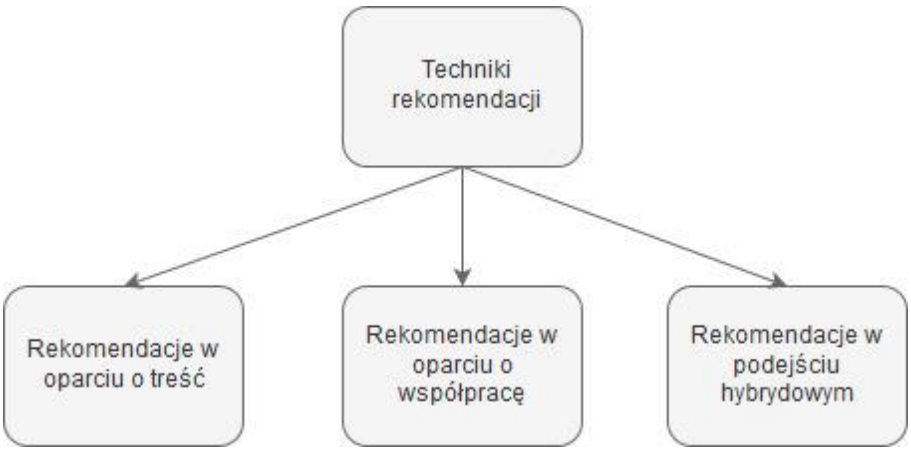
Całkiem inną koncepcję w badaniu znaczenia i podobieństwa tekstów prezentują Ziwon Hyuong i inni w artykule *Music Recommendation based on Text Mining* [18] i jego późniejszej wersji *Music recommendation using text analysis on song requests to radio stations* [5]. Opisują wykorzystanie analizy tekstu do budowy systemu rekomendacji piosenek. W celu zbadania podobieństwa tekstów wykorzystana została metoda modelowania tematycznego (ang. topic modeling) zwana Ukrytą Analizą Semantyczną (ang. Latent Semantic Analysis, w skrócie: LSA) [19] oraz jej rozszerzenie – Probabilistyczna Ukryta Analiza Semantyczna (ang. Probabilistic Latent Semantic Analysis, pLSA) [20]. Naukowcy zebrali życiowe historie słuchaczy koreańskiego radia wraz z przypisanymi do nich zamówieniami piosenek. Następnie przeprowadzili Ukrytą Analizę Semantyczną w celu identyfikacji znaczenia i znalezienia historii podobnych. Zakładając, że ludzie dzielący podobne życiowe historie słuchają podobnej muzyki, system proponował te same piosenki najbardziej podobnym tekstom.

Ewaluacja systemu z wykorzystaniem różnych miar odległości pomiędzy tekstami potwierdziła, iż istnieje pozytywna korelacja pomiędzy podobieństwem historii i piosenek, oraz że możliwe jest rekomendowanie muzyki bazując tylko i wyłącznie na analizie danych tekstowych [5].

Niestety w opisanym w artykule systemie rekomendacji piosenek nie wykorzystano żadnej z wcześniej wspomnianych metod reprezentacji znaczenia. Stanowi to solidne podstawy do zbadania ich skuteczności przy tego typu zastosowaniu, co dokładnie zawiera cel i zakres niniejszej pracy.

3. Typologia technik rekomendacji i ulokowanie celu pracy

Według literatury [21] wyróżnia się trzy techniki rekomendacji: rekomendacje w oparciu o treść (ang. content-based filtering), rekomendacje w oparciu o współpracę użytkowników (ang. colaborative filtering) oraz podejście hybrydowe łączące oba poprzednie. Każde z nich może zostać wykorzystane do rekomendacji piosenek [21].

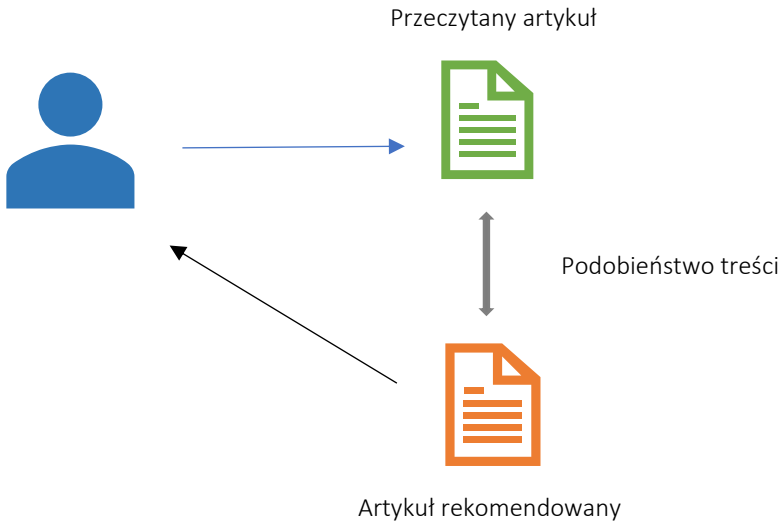


Rysunek 2 - Techniki rekomendacji

3.1. Rekomendacje w oparciu o treść

Technika rekomendacji bazująca na treści – jak wskazuje sama nazwa – opiera się o cechy i właściwości polecanego przedmiotu, obiektu. Rekomendacje tworzone są na podstawie profilów użytkownika tworzonych z cech ekstrahowanych z treści, które użytkownik ocenił. W przypadku rekomendacji utworów muzycznych treścią będzie tekst danej piosenki, ale także częstotliwość czy metadane takie jak gatunek i artysta [5].

Minusem takiego rozwiązania jest wysokie zapotrzebowanie na moc obliczeniową, co w przypadku dużych zbiorów danych może stanowić spory problem [5].



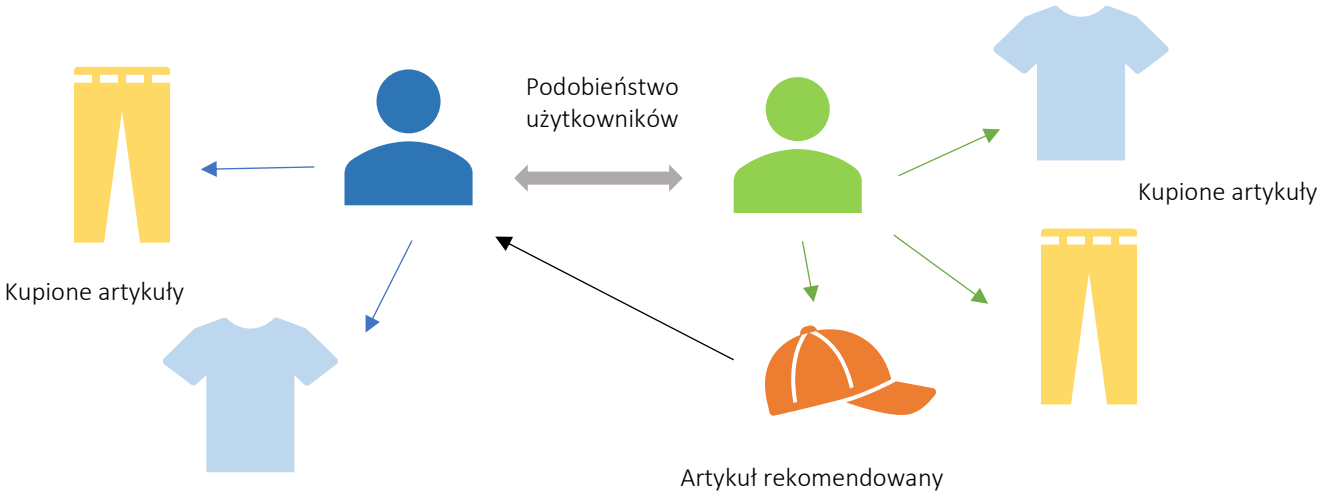
Rysunek 3 - Rekomendacja w oparciu o treść

3.2. Rekomendacje w oparciu o współpracę

Rekomendacje w oparciu o współpracę polegają na znalezieniu wspólnych cech użytkowników, a następnie proponowanie obiektów nowym użytkownikom na podstawie historycznych wyborów użytkowników podobnych do nich. W przypadku rekomendacji piosenek wspólnymi cechami będą najchętniej słuchane utwory, płeć czy wiek.

W takim podejściu niezwykle ważne jest posiadanie informacji dotyczących wyborów użytkowników w przeszłości. Brak takich danych określany jest w literaturze jako problem tzw. „zimnego startu" (ang. cold start) [5]. Według literatury [21], jest to jeden z głównych powodów, który redukuje skuteczność systemów rekomendacyjnych. Profil danego użytkownika, jego preferencje i wybory nie będą znane systemowi, dopóki faktycznie nie dokona on pewnego wyboru.

Innym problemem jest tzw. rzadkość danych (ang. data sparsity). Pojawia się wtedy, gdy system posiada już zapisane wybory kilku użytkowników, lecz jest ich niewiele. W takiej sytuacji możliwości systemu są ograniczone, ponieważ dokonuje rekomendacji tylko i wyłącznie takich obiektów, o których ma informacje.

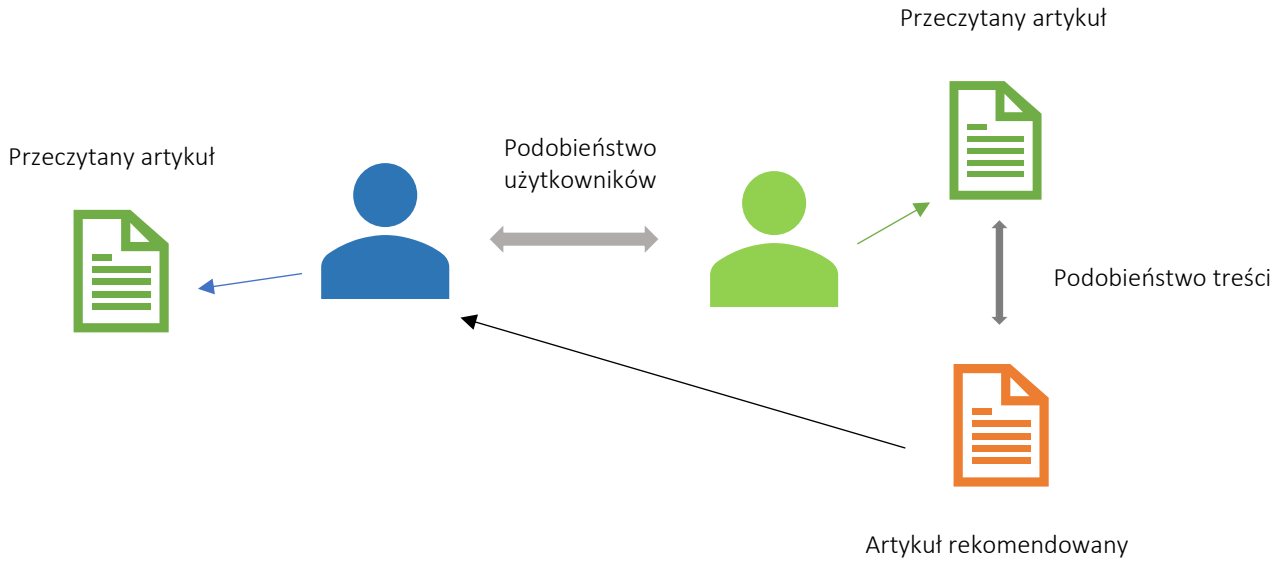


Rysunek 4 - Rekomendacja w oparciu o współpracę użytkowników

3.3. Rekomendacje w podejściu hybrydowym

Trzecia z technik rekomendacji polega na połączeniu techniki rekomendacji opierającej się o treść oraz techniki opierającej się o współpracę. Ma na celu uniknięcie problemów i ograniczeń wynikających ze stosowania jednej z dwóch powyższych technik samodzielnie [21]. Zakłada, iż połączenie różnych algorytmów zapewni dokładniejsze i bardziej efektywne rekomendacje oraz że minusy jednego algorytmu zostaną zniwelowane przez działanie drugiego.

Istnieje wiele podejść łączenia technik rekomendacji. Najpopularniejsze z nich to: osobna implementacja technik i łączenie wyniku, użycie techniki rekomendacji bazującej na treści w podejściu rekomendacji w oparciu o współpracę (i odwrotnie poprzez analogię) oraz utworzenie ujednoliconego podejścia wykorzystującego oba jednocześnie [21].



Rysunek 5 - Rekomendacja hybrydowa

3.4. Rekomendacje w oparciu o kontekst

Omawiane w rozdziale 2 artykuły *Music Recommendation based on Text Mining* [18] i *Music recommendation using text analysis on song requests to radio stations* [5] bazują na niespotykanej w klasycznej typologii systemów rekomendacyjnych technice – rekomendacji w oparciu o kontekst (ang. context-aware) [5].

Konieczność użycia w systemach rekomendujących piosenki informacji dotyczących kontekstu użytkowników wprowadził jako pierwszy Gordon Reynolds i inni w pracy pt. *Interacting with large music collections: towards the use of environmental metadata* [22]. Jako kontekst rozumiał on ogół sytuacji i środowisko, w którym znajduje się słuchający podczas słuchania muzyki. Skorzystał z takich cech jak lokalizacja, temperatura czy nastrój użytkownika.

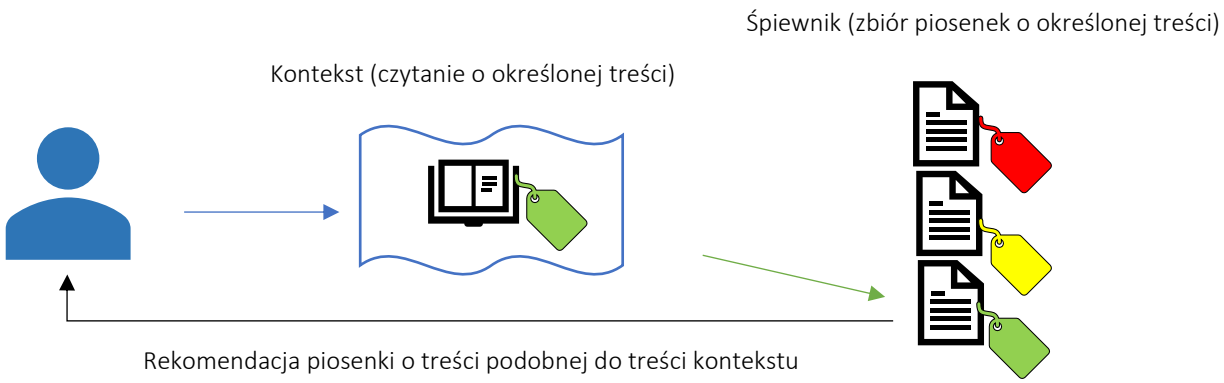
Owe dwa artykuły natomiast uwzględniają wykorzystanie do rekomendacji piosenek spisane historie użytkowników w postaci listów. Jak twierdzą autorzy, dokumenty zawierają tło dla zażyczonych przez słuchaczy piosenek i mogą być interpretowane jako kontekst [18]. Co więcej, dzięki wykorzystaniu w rekomendacjach danych tekstowych pozbyli się wspomnianych wcześniej problemów takich jak wspomniany już tzw. „zimny start" czy złożoność obliczeniowa.

3.5. Ulokowanie celu pracy w systemie rekomendacji

Jak wspomniano pod koniec rozdziału 2, koreańskie badania wykazały, iż podobieństwo znaczenia tekstów i piosenek można bezpośrednio wykorzystywać w budowaniu systemów rekomendacji utworów muzycznych [18].

Na tej podstawie zaproponowano hybrydowy system rekomendacji piosenek w oparciu o kontekst i treść, wykorzystujący wyniki badań niniejszej pracy. Jako kontekst rozumiane są bowiem fragmenty biblii (czytania) odczytywane w poszczególne dni w kościele katolickim, przy okazji których wykorzystywane są piosenki, a treścią - znaczenie obojga z nich.

Narzędzie nazwano *PiosenKatoR – Rekomendator Piosenek Katolickich.*



Rysunek 6 - PiosenKatoR - Rekomendator Piosenek Katolickich - koncepcja

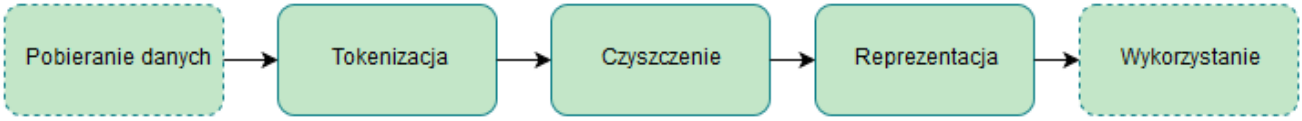
4. Przetwarzanie języka naturalnego

Badanie znaczenia i podobieństwa tekstu wchodzi w skład zadań z dziedziny przetwarzania języka naturalnego.

Niniejszy rozdział ma na celu przedstawienie ogólnego podejścia do pracy z analizą tekstu i przetwarzaniem języka naturalnego, a w szczególności do zadań związanych z badaniem podobieństwa znaczenia między tekstami.

4.1. Proces w przetwarzaniu języka naturalnego

Niezależnie od rodzaju problemów i zagadnień spotykanych w dziedzinie analizy tekstu, praca z danymi tekstowymi wymaga szeregu powtarzalnych czynności w celu doprowadzenia danych do postaci nadającej się do realnego wykorzystania. Typowy proces większości zadań z zakresu przetwarzania języka naturalnego można podzielić na następujące etapy: pobieranie danych, tokenizację (sprowadzenie do formy pojedynczych elementów, ang. token - znak), czyszczenie oraz reprezentację. Dopiero po uzyskaniu pożądanej reprezentacji (najczęściej w formie wektora) następuje wykorzystanie z użyciem modeli [10, 23].



Rysunek 7 - Proces w przetwarzaniu języka naturalnego

Dodatkowo na każdy z etapów składa się zestaw mniejszych kroków odpowiednio stosowanych w zależności od natury problemu.

4.1.1. Pobieranie danych

Pierwszy krok w procesie przetwarzania języka naturalnego to zdobycie odpowiednich danych. Muszą one być odpowiednio dobrane do rozwiązywanego zadania. O ile współcześnie wiele danych liczbowych jest udostępniana w wygodnych do pobrania i przetwarzania formatach, w przypadku danych tekstowych czasami wymagane jest zaprojektowanie całego mechanizmu wydobywania ich z różnych źródeł od zera [24]. W przypadku pozyskiwania tekstów z Internetu, często stosuje się metodę *scrapingu* (ang. scrape – zeskrobywać), polegającą na ściąganiu treści przedstawianych na stronach internetowych.

Ważnym aspektem przy pobieraniu danych jest sposób ich zapisu i przechowywania. Wpływa to nie tylko na szybkość przeprowadzania operacji, lecz także na bezpieczeństwo. Dane mogą być przechowywane na dysku jako pliki .txt, w relacyjnych bazach danych czy hurtowniach. Obecnie, ze względu na wzrost znaczenia danych tekstowych dużą popularnością cieszą się nierelacyjne bazy danych i tzw. jeziora danych (ang. data lake) umożliwiające przetrzymywanie wielkich wolumenów dokumentów w nieprzetworzonej formie [24].

Etap pobierania danych to pierwszy okazja weryfikacji ich jakości i poprawności. Jak podaje literatura, większość błędów napotykana podczas fazy gromadzenia danych jest łatwa do wychwycenia. Jednakże, będąc zbyt nieostrożnym można spędzić wiele godzin na

rozwiązywaniu problemów, których można było uniknąć już na etapie importu. Są to zazwyczaj kwestie dotyczące porównywania ściągniętych tekstów ze źródłami [24].

4.1.2. Tokenizacja

W odróżnieniu od standardowego procesu badania danych, w przypadku danych tekstowych po pobraniu nie następuje proces ich oczyszczania. Poprzedza je bowiem etap tokenizacji [24].

Tokenizacja (ang. token – znak) w przetwarzaniu języka naturalnego to metoda pozwalająca na przekształcenie ciągłego tekstu na słowa, wyrażenia lub inne znaczące elementy zwane tokenami. Według literatury nie istnieje żadne ogólne porozumienie ani ścisła definicja tego procesu [25, 26].

Głównym jego celem jest wyodrębnienie z danego tekstu poszczególnych słów - tokenów. Za token uznawane są także znaki interpunkcyjne i liczby. Powtórzone wyrazy nie są usuwane.

Przykład:

tekst: „Ala ma kota. Kot ma Alę. Czy na pewno?”
tokeny: [‘Ala’, ‘ma’, ‘kota’, ‘.’, ‘Kot’, ‘ma’, ‘Alę’, ‘.’, ‘Czy’, ‘na’, ‘pewno’, ‘?’]

4.1.3. Czyszczenie

Według autorów książki *Introducing Data Science* [24] podczas pracy z danymi należy spodziewać się spędzenia nawet do 80% czasu na ich poprawianiu i czyszczeniu. Dane uzyskane na etapie pobierania można utożsamić z „nieoszlifowanym diamentem”. Należy doprowadzić je do postaci nadającej się do raportowania czy modelowania. W dziedzinie przetwarzania języka naturalnego i analizy tekstów, etap ten jest niezwykle ważny.

Większość danych tekstowych zawiera sporo niepotrzebnych słów czy znaków, które utrudniają wydobycie znaczenia. Są to m.in. przyimki, spójniki, znaki interpunkcyjne czy przejęzyczenia [10]. Często również w tekstach ściągniętych ze stron internetowych pojawiają się znaczniki HTML czy emotikony. Zdarza się, że podczas przetwarzania dokumentów naukowych można napotkać równania matematyczne, chemiczne czy kody programów. Nie należą one do języka naturalnego i muszą zostać wyeliminowane [25].

Poniższe kroki pozwalają na przekształcenie tekstu do formy gotowej do zastosowania różnych metod reprezentacji umożliwiających dalsze przetwarzanie.

- Usuwanie szumu

Zazwyczaj dane tekstowe zawierają wiele niepotrzebnych z punktu widzenia analizy tekstu znaków takich jak znaki interpunkcyjne czy specjalne [10]. Są one traktowane jako zbędny szum i usuwane.

Przykład:

tokeny: [‘Ala’, ‘ma’, ‘kota’, ‘.’, ‘Kot’, ‘ma’, ‘Alę’, ‘.’, ‘Czy’, ‘na’, ‘pewno’, ‘?’]
tokeny po usunięciu szumu: [‘Ala’, ‘ma’, ‘kota’, ‘Kot’, ‘ma’, ‘Alę’, ‘Czy’, ‘na’, ‘pewno’]

- „Stop words”

Teksty i dokumenty zawierają wiele słów, który nie wnoszą niczego ważnego pod względem znaczenia i interpretacji ich treści. Są to m.in. przyimki, spójniki, przedrostki czy partykuły. Literatura przedmiotu określa je jako „stop words”. Najpowszechniejsza metoda pracy z tekstami zawierającymi takie tokeny to po prostu pozbycie się ich [10].

Przykład:

tokeny po usunięciu szumu: [‘Ala’, ‘ma’, ‘kota’, ‘Kot’, ‘ma’, ‘Alę’, ‘Czy’, ‘na’, ‘pewno’]
tokeny po usunięciu szumu bez „stop words”: [‘Ala’, ‘ma’, ‘kota’, ‘Kot’, ‘ma’, ‘Alę’, ‘pewno’]

- Ujednolicenie wielkości liter

Często w zdaniach w obrębie jednego lub kilku dokumentów znajdują się te same wyrazy (tokeny), różniące się jednak wielkością liter. Podczas pracy z dużymi zbiorami tekstu może to nastręczyć wiele problemów, gdyż słowa odnoszące się do tych samych obiektów mogą zostać uznane jako nietożsame. Dlatego też najczęściej stosuje się ujednolicenie wielkości każdej z liter do małej (ang. lower case).

Przykład:

tokeny po usunięciu szumu bez „stop words”: [‘Ala’, ‘ma’, ‘kota’, ‘Kot’, ‘ma’, ‘Alę’, ‘pewno’]
ujednolicone tokeny po usunięciu szumu bez „stop words”:
[‘ala’, ‘ma’, ‘kota’, ‘kot’, ‘ma’, ‘alę’, ‘pewno’]

Istnieje ryzyko, że podczas ujednolicenia wielkości liter utracone zostaną faktycznie istotne różnice.

Przykład:

Łódź (miasto) i łódź (mały statek)

W takich przypadkach stosowane są tzw. dezambiguatory, które rozróżniają kontekst występowania danego słowa.

- Slang i skróty

Użycie w tekście slangu lub skrótów jest kolejnym przykładem anomalii, które należy przepracować. Skrót jest to sprowadzenie wyrazu lub wyrażenia do ciągu najczęściej pierwszych jego wyrazów [10].

Przykład:

PWr – Politechnika Wrocławska

Slang natomiast, jest nieformalnym podzbiorem danego języka naturalnego używanym w mowie potocznej. Popularna metoda radzenia sobie z powyższymi przypadkami to konwersja do języka formalnego [10].

- Autokorekta

Błędy (ortograficzne lub w pisowni) występują w tekstach dość często – zwłaszcza, gdy dane pochodzą z mediów społecznościowych. W celu uniknięcia następstw z nich wynikających stosuje się wiele algorytmów autokorekcji, które przekształcają błędne wyrazy w poprawne [10].

- Stemming

W procesie przetwarzania języka naturalnego częstym przypadkiem jest występowanie tych samych wyrazów w różnych formach, mimo iż ich znaczenie semantyczne pozostaje to samo. Przykładem jest występowanie w języku polskim rzeczowników w liczbie pojedynczej i mnogiej [27].

Jedną z metod sprowadzających różne formy wyrazu do jednej jest stemming (ang. stem – rdzeń). Zazwyczaj polega ona na mechanicznym odcięciu końcówek wyrazów [27].

Przykład:

ujednolicone tokeny po usunięciu szumu bez „stop words”:
[‘ala’, ‘ma’, ‘kota’, ‘kot’, ‘ma’, ‘alę’, ‘pewno’]

po zastosowaniu stemmingu:
[‘al’, ‘ma’, ‘kot’, ‘kot’, ‘ma’, ‘al’, ‘pewn’]

- Lematyzacja

Lematyzacja jest procesem bardzo zbliżonym do stemmingu, lecz opiera się o sprowadzenie wyrazu do podstawowej słownikowej formy (lemmy), a nie samym odcięciu końcówek [28]. Jest dużo bardziej wymagająca obliczeniowo niż stemming. Podczas lematyzacji tekstów w języku polskim rzeczowniki zazwyczaj sprowadza się do mianownika, a czasowniki do bezokolicznika.

Przykład:

ujednolicone tokeny po usunięciu szumu bez „stop words”:
[‘ala’, ‘ma’, ‘kota’, ‘kot’, ‘ma’, ‘alę’, ‘pewno’]

po zastosowaniu lematyzacji:
[‘ala’, ‘mieć’, ‘kot’, ‘kot’, ‘mieć’, ‘ala’, ‘pewność’]

4.1.4. Reprezentacja

By móc efektywnie wykorzystać wcześniej przetworzony i wyczyszczony tekst, należy rozwiązać jeden z najbardziej fundamentalnych problemów w dziedzinie analizy i przetwarzania informacji – problem reprezentacji. Ma on na celu przekształcenie nieustrukturyzowanego dokumentu tekstowego do postaci numerycznej by umożliwić obliczenia matematyczne [23]. W nauce o danych takich zabieg nazywa się ogólnie ekstrakcją lub kodowaniem cech [29].

Dla danego zbioru dokumentów $D = \{d_i, i=1, 2, \dots, n\}$, gdzie każdy d_i stanowi dokument, problem reprezentacji polega na reprezentacji każdego d_i należącego do D jako s_i w przestrzeni S, gdzie dystans lub podobieństwo pomiędzy każdą parą punktów w przestrzeni S są ściśle określone [23].

Na przestrzeni lat zaproponowano wiele strategii reprezentacji tekstów [23]. Współcześnie najpowszechniej stosowane jest podejście reprezentacji tekstu jako model przestrzeni wektorowej – Vector Space Model, w skrócie VSM) [23].

Poniżej przedstawiono podstawowe modele reprezentacji tekstów.

- Bag of words (BOW)

Najstarszym sposobem reprezentacji tekstu w postaci wektorowej jest model Bag of words (ang. torba słów) lub BOW. Nazywany jest tak dlatego, ponieważ jakakolwiek informacja co do kolejności czy struktury wyrazów w dokumencie nie zostaje uwzględniona [29]. Model zlicza wszystkie słowa występujące w danym zbiorze dokumentów i traktuje każde z unikalnych słów jako indeks (cechę) tworząc wektor dokumentów. Następnie traktuje każdy dokument jako osobny wektor o długości najwyższego indeksu [23].

Dla zbioru dokumentów (poddanych procesowi tokenizacji i czyszczenia) $D = \{d_i, i=1, 2,\dots,n\}$ istnieje m unikalnych terminów indeksujących pojawiających się w tym zbiorze. Matematycznie dany korpus dokumentów może być przedstawiony w postaci macierzy S o wymiarach m na n, $S \in R^{m \times n}$. Każdy dokument tekstowy zapisywany jest w postaci wektora kolumny s_i, $i= 1, 2, \dots, n$ a każdy termin w postaci wektora wiersza. *J-ty* element wektora s_i oznaczany jest jako $s_{j,i}$, $j = 1, 2, \dots, m$ [23].

Przykład:

Dany jest trzyelementowy zbiór stokenizowanych i wyczyszczonych dokumentów $D = \{d_1, d_2, d_3\}$.

d_1: *['ala', 'mieć', 'kot']*
d2: ['kot', 'mieć', 'ala']
d3: ['pewność']

W zbiorze występują w sumie 4 unikalne terminy: 'ala', 'mieć', 'kot' i 'pewność'. Zbiór dokumentów może być zatem przedstawiony jako macierz S o wymiarach 4 na 3. Jeśli termin występuje w dokumencie, elementowi wektora reprezentującego dany termin w danym dokumencie przypisywana jest wartość 1, w przeciwnym wypadku - wartość 0. Transponowana macierz S wygląda następująco:

$$S^T = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Powyższy przykład przedstawia binarny model Bag of Words. Reprezentuje on każdy dokument jako wektor zer i jedynek. Gdy dane słowo występuje w tekście, jego pozycji odpowiadającej w wektorze dokumentu $s_{i,j}$ przypisywana jest liczba „1". W przeciwnym wypadku – liczba „0".

Inną odmianą modelu Bag of Words jest model podający zamiast zer i jedynek liczbę występujących słów.

Przykład:

Dany jest dwuelementowy zbiór stokenizowanych i wyczyszczonych dokumentów D = {d_1, d_2}.

d_1: ['ala', 'mieć', 'kot', 'kot', 'mieć', 'ala']
d2: ['pewność']

W zbiorze również występują w sumie 4 unikalne terminy: 'ala', 'mieć', 'kot' i 'pewność'. Zbiór może być przedstawiony jako macierz S o wymiarach 4 na 2. Tym razem jednak elementowi wektora reprezentującego dany termin w danym dokumencie przypisywana jest liczba wystąpień danego terminu w danym dokumencie. Transponowana macierz S wygląda następująco:

$$S^T = \begin{pmatrix} 2 & 2 & 2 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- TF-IDF

Model reprezentacji o nazwie Term Frequency – Inversed Document Term Frequency lub TF-IDF został zaproponowany jako rozszerzenie modelu BOW przez K. Sparck Jones'a [10]. Ma on na celu zmniejszenie ważności często występujących słów w całym korpusie dokumentów a podkreślenie ważności słów rzadkich. Jak sama nazwa wskazuje opiera się on o dwie miary: Term Frequency i Inversed Document Term Frequency.

TF (ang. Term Frequency – częstotliwość terminów) to miara występowania słowa w tekście. Bez pozbycia się tzw. „stop words" sama w sobie nie jest dobrym odzwierciedleniem ważności słowa w tekście, gdyż najczęstszymi słowami są zazwyczaj przyimki, spójniki i partykuły.

Zakładając, że dany jest zbiór dokumentów (poddanych procesowi tokenizacji i czyszczenia) $D = \{d_j, j=1, 2,..., N\}$ i termin t_i występuje w n_i spośród nich. Miara $TF_{i,j}$ występowania terminu t_i w dokumencie d_j definiowana jest w następujący sposób:

$$TF_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}}, \quad (1)$$

gdzie

$n_{i,j}$ – liczba wystąpień terminu t_i w dokumencie d_j
$\sum_k n_{k,j}$ – suma wszystkich wystąpień wszystkich terminów w dokumencie d_j

IDF (ang. Inversed Document Frequency) to miara odwrotności częstotliwości dokumentu. Pozwala określić wagę danego słowa w całym zestawie dokumentów (korpusie). Matematyczna reprezentacja wagi terminu t_i w dokumencie reprezentowana jest przez poniższe równanie:

$$IDF_i = log \frac{N}{n_i}, \tag{2}$$

gdzie

N – liczba wszystkich dokumentów
n_i – liczba dokumentów, w których występuje termin t_i

Na podstawie równań (1) oraz (2) , waga *TF-IDF* terminu t_i w dokumencie d_j definiowana jest jako:

$$w_{i,j} = TF_{i,j} \times IDF_i \tag{3}$$

Reprezentacja dokumentu tekstowego danego korpusu w modelu TF-IDF polega na przedstawieniu go jako wektora składającego się z odpowiednich wag wyliczonych wzorem (3). Wektor można wyrazić zapisem:

$$v(d_j) = (w_{1,j}, \ldots, w_{k,j}) \tag{4}$$

- N – gramy

Metodą reprezentacji pozwalającą zachować kolejność występujących po sobie słów jest model oparty o tzw. n-gramy – sekwencje n-słów. I tak 2-gram to sekwencja dwóch słów, 3-gram to sekwencja odpowiednio 3 słów [30].

Przykład:

Dany jest dwuelementowy zbiór stokenizowanych i wyczyszczonych dokumentów $D = \{d_1, d_2\}$.

d_1: *['ala', 'mieć', 'kot']*
d2: ['kot', 'mieć', 'ala']

Wszystkie możliwe bigramy (n=2) dla zbioru dokumentów D to:
['ala mieć', 'mieć kot', 'kot mieć', 'mieć ala'].

Wszystkie możliwe trigramy (n=3) dla dokumentu d_2 *to:*
['ala mieć kot', 'kot mieć ala'].

Model reprezentacji polega na budowie wektorów, których cechami (kolumnami) są ciągi n występujących obok siebie słów.

Reprezentacja w postaci wektora S^T dokumentów ze zbioru D z użyciem bigramów wygląda następująco:

$$S^T = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix}$$

Reprezentacja w postaci wektora S^T dokumentów ze zbioru D z użyciem trigramów wygląda następująco:

$$S^T = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

Bag of words jest przykładem reprezentacji z użyciem 1-gramów (unigramów). Najbardziej popularnymi w dziedzinie przetwarzania języka naturalnego są modele wykorzystujące 2-gramy (bigramy) i 3-gramy (tri-gramy) [23,26].

- Osadzanie słów / Wektory dystrybucyjne słów (ang. word embedding)

Osadzanie słów to technika wyuczania cech wywodząca się z dziedziny semantyki dystrybucyjnej, w której to każde słowo lub wyrażenie odzwierciedlane jest jako N-wymiarowy wektor liczb rzeczywistych. Tomas Mikolov i inni w pracach *Efficient estimation of word representations in vector space* [11] oraz *Distributed representations of words and phrases and their compositionality* [12] zaprezentowali, a następnie udoskonalili model "word to vector" skrótowo nazwany Word2vec.

W podejściu tym wektory powstają na podstawie dwóch architektur płytkich, dwuwarstwowych sieci neuronowych trenowanych do rozwiązywania zadania rekonstrukcji kontekstów wyrazów [31]. Przyjmują one jako wejście duży korpus słów i przekształcają go w wielowymiarową przestrzeń wektorową, w której to każde unikalne słowo w korpusie przypisane jest do wektora w tej przestrzeni. Wektory są tak skonstruowane, że słowa dzielące wspólne konteksty są zlokalizowane blisko siebie [31].

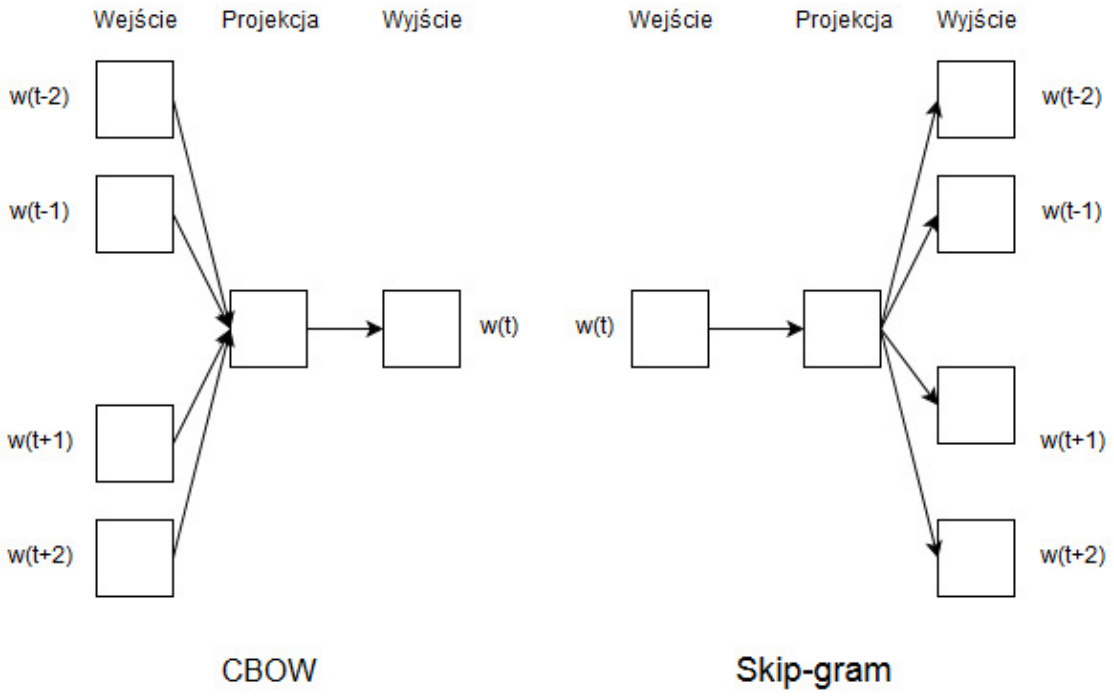
Pierwsza architektura nazywa CBOW (ang. continuous bag of words) a druga Skip-gram. Model CBOW próbuje znaleźć dane słowo *w(t)* bazując na słowach poprzedzających *w(t-1), w(t-2)* i następujących *w(t+1), w(t+2)*. Skip-gram odwrotnie - próbuje znaleźć sąsiedztwo danego słowa. Koncepcję przedstawia Rys.8.

Mimo, iż sieć uczona jest dla konkretnych zadań predykcji wyrazów, dla modelu Word2vec liczą się tylko wyuczone wagi warstwy ukrytej wyliczone podczas minimalizacji funkcji starty [31]. Wagi te stanowią docelową wektorową reprezentację słów [10, 11, 12].

$$h = v^T \tag{5}$$

gdzie

h – wyuczone wagi warstwy ukrytej sieci



Rysunek 8 - Architektura CBOW i Skip-gram [11]

Metoda Word2vec dostarcza bardzo skuteczne narzędzie służące do odkrywania zarówno wzajemnych związków w korpusie tekstów jak i podobieństwa pomiędzy słowami. Znając wektory poszczególnych słów, reprezentację całego dokumentu można przedstawić jako wektor składający się z ich średnich arytmetycznych. Stanowi on wtedy centroid wszystkich słów w przestrzeni [32, 33].

Najbardziej popularne metody tworzenia wektorów słów używane współcześnie w wielu rozwiązaniach komercyjnych to Word2Vec, GloVe i FastText [10]. W Internecie dostępne są do pobrania wytrenowane na wielkich zbiorach danych modele gotowe do wykorzystania bez konieczności przeprowadzania procesu uczenia.

- Osadzanie dokumentów / wektory dystrybucyjne dokumentów (ang. document embedding)

Rok po zaproponowaniu i rozwinięciu metody dystrybucyjnej reprezentacji słów, wspomniany już Tomas Mikolov i Quoc Le przedstawili jej rozszerzenie w postaci modelu do reprezentacji całych dokumentów jako wektory. Rozwiązanie zostało nazwane Paragraph Vectors a jego powszechniejsza nazwa to Doc2vec. Podobnie jak w przypadku pojedynczych słów, model również opiera się o dwie architektury: PV-DM (Paragraph Vector – Distributed Memory) oraz PV-DBOW (Paragraph Vector - Distributed Bag of Words) [33, 34] .

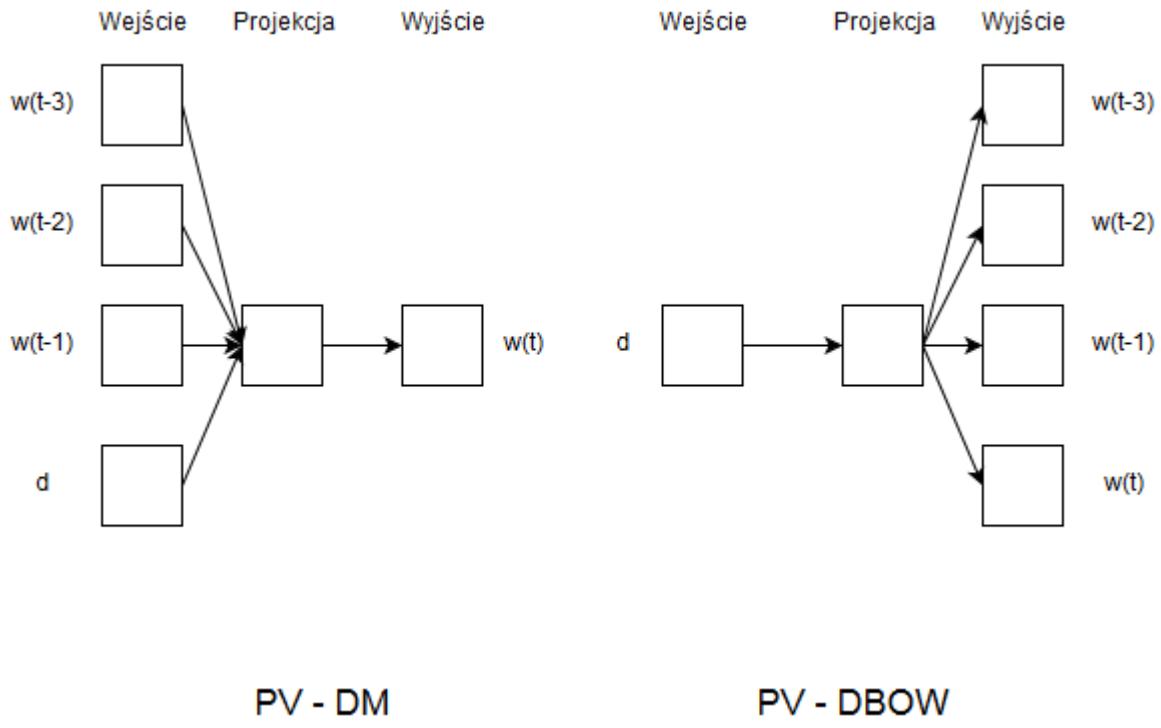
Idea stojąca za architekturą PV-DM inspirowana jest omawianym wyżej modelem CBOW, w którym to sieć uczy się przewidywać centralne słowo w kontekście. Podobnie tutaj, główna idea polega na wykorzystaniu występujących po sobie słów z danego dokumentu i przewidzeniu słowa centralnego. Tym razem jednak jako wejście oprócz kontekstu (otaczających słów *w*) wprowadza się również dany dokument (*d*) [33, 35].

Macierz *D* dokumentu *d* to macierz reprezentująca w każdej kolumnie wektor słów z danego dokumentu. Następnie poszczególne wektory są uśredniane lub sumowane tworząc wektor warstwy ukrytej wchodzący jako wejście do klasyfikatora, który przewiduje słowo

centralne *w(t)* [36]. Podczas gdy wektory słów reprezentują znaczenie słów, wektor dokumentu reprezentuje znaczenie całego dokumentu [35].

Druga z architektur, Paragraph Vector - Distributed Bag of Words (PV-DBOW) działa analogicznie do architektury Skip-gram w modelu Word2vec. Jak podaje literatura, „ignoruje ona kontekst słów na wejściu, ale zmusza model do predykcji losowych słów dokumentu na wyjściu".

Koncepcję działania dwóch architektur modelu Doc2vec przedstawia Rys.9.



Rysunek 9 - Architektura PV-DM i PV-DBOW [33]

4.1.5. Wykorzystanie

Po wstępnym przetwarzaniu i czyszczeniu, końcowa reprezentacja tekstu może zostać użyta do wielu zadań związanych z modelowaniem. Dzięki formie reprezentacji znaczenia pozwalającej na obliczenia matematyczne, teksty mogą zostać użyte do rozwiązywania takich zadań jak obliczanie wzajemnego podobieństwa.

5. Podobieństwo tekstów

Wybór miary podobieństwa tekstów jest jednym z ważniejszych problemów pojawiających się w zadaniach przetwarzania języka naturalnego i analizie tekstów [2, 37]. W literaturze znajdują się dziesiątki algorytmów pozwalających na porównanie ze sobą dwóch tekstów. Istnieje również wiele różnych podejść do mierzenia podobieństwa. Każdy z nich ma swoje wady i zalety i jest używany tylko do problemów z odpowiadającego zakresu.

W przypadku porównywania ze sobą dwóch tekstów pod względem znaczeniowym, najczęściej przytaczane są dwie miary: współczynnik podobieństwa Jaccarda oraz podobieństwo kosinusowe.

- Współczynnik podobieństwa Jaccarda

Współczynnik podobieństwa Jaccarda, inaczej zwany indeksem Jaccarda, mierzy podobieństwo między dwoma zbiorami i jest definiowany jako iloraz mocy części wspólnej zbiorów i mocy sumy tych zbiorów. Przyjmując, że że dany jest zbiór dokumentów (poddanych procesowi tokenizacji i czyszczenia) $D = \{d_j, j=1, 2,..., N\}$ a $T(d_j)$ oznacza zbiór wszystkich terminów występujących w dokumencie d_j indeks Jaccarda może być przedstawiony następująco:

$$P_{Jaccard}(T(d_j), T(d_{j+1})) = \frac{card(T(d_j) \cap T(d_{j+1}))}{card(T(d_j) \cup T(d_{j+1}))} \quad (6)$$

gdzie

$card(T(d_j))$ – liczność zbioru wszystkich terminów występujących w dokumencie d_j

Współczynnik podobieństwa Jaccarda używany jest podczas porównywania reprezentacji tekstów w postaci zbioru terminów. W przypadku reprezentacji tekstów jako wektorów liczbowych powszechniej stosowane jest podobieństwo kosinusowe [2, 38, 39].

- Podobieństwo kosinusowe

Podobieństwo kosinusowe oblicza kosinus kąta pomiędzy dwoma wektorami. Jest to miara kierunku, a nie wielkości. Dwa wektory o tym samym kierunku mają podobieństwo kosinusowe równe 1, z kolei wektory leżące wobec siebie prostopadle mają podobieństwo wynoszące 0 [37].

$$P_{Cosinus} = \frac{card(T(d_j) \cap T(d_{j+1}))}{\sqrt{card(T(d_j))} * \sqrt{card(T(d_{j+1}))}} \quad (7)$$

gdzie

$card(T(d_j))$ – liczność zbioru wszystkich terminów występujących w dokumencie d_j

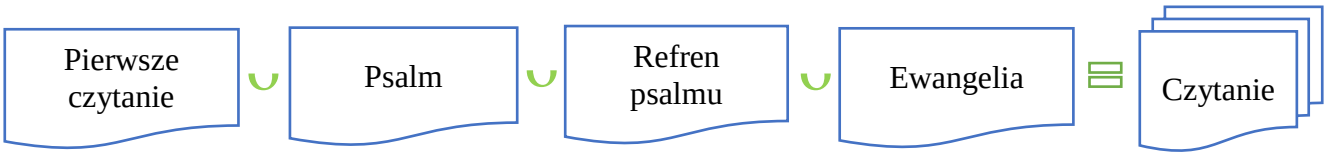
6. Wykorzystane dane

Niniejszy rozdział ma na celu przedstawienie danych użytych do badania poprzez opisanie podstawowych charakterystyk i statystyk.

6.1. Opis i charakterystyka

Dane tekstowe stanowiące bazę do przeprowadzonych badań zawierają teksty w języku polskim. Można podzielić je na trzy zestawy.

Pierwszy z nich stanowią fragmenty biblijne odczytywane w różnych dniach w kościele rzymskokatolickim podczas celebracji mszy świętej. Zazwyczaj składają się na nie: pierwsze czytanie, psalm, refren psalmu, cytat przed Ewangelią i Ewangelia. W przypadku świąt dochodzi jeszcze czytanie drugie. Fragmenty pobrano ze strony internetowej: www.mateusz.pl. W celu ujednolicenia wyników badań wykorzystano tylko pierwsze czytania, psalmy, refreny i Ewangelie. Służyły one jako wskazany w temacie pracy tekst zadany. Dane zostały potraktowane całościowo, tzn. wszystkie odczytywane teksty z danego dnia traktowano jako całość, tj. jakby stanowiły jeden nieprzerwany tekst. Jak wspomniano w rozdziale 1., takie połączone fragmenty biblii stanowiące zadany tekst z tematu pracy, nazwano ogólnie *Czytaniem*.

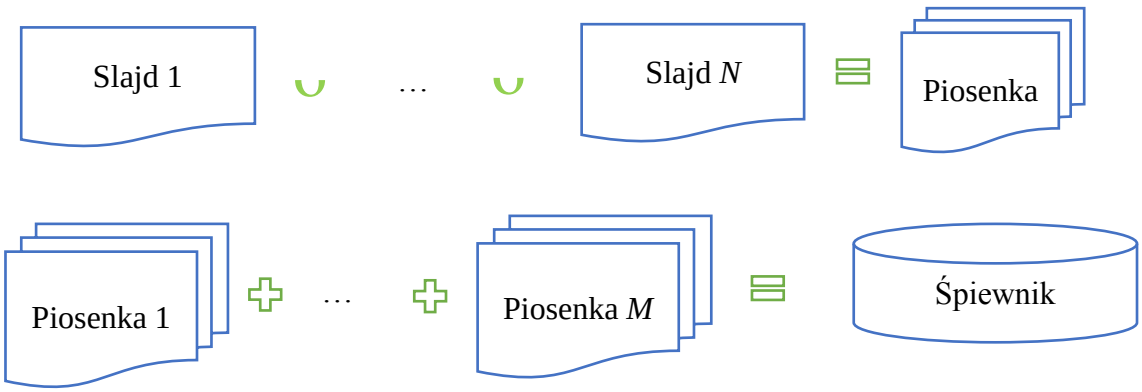


Rysunek 10 – Koncepcja zestawu pierwszego - Czytania

Przykład Czytania:

"Pan mówi: „Oto nadchodzą dni, kiedy wypełnię pomyślną zapowiedź, jaką obwieściłem domowi izraelskiemu i domowi judzkiemu. W owych dniach i w owym czasie wzbudzę Dawidowi potomka sprawiedliwego; będzie on wymierzał prawo i sprawiedliwość na ziemi. W owych dniach Juda dostąpi zbawienia, a Jerozolima będzie mieszkała bezpiecznie. To zaś jest imię, którym ją będą nazywać: ""Pan naszą sprawiedliwością"". Do Ciebie, Panie, wznoszę moją duszę Daj mi poznać, Twoje drogi, Panie, naucz mnie chodzić Twoimi ścieżkami. Prowadź mnie w prawdzie według Twych pouczeń, Boże i Zbawco, w Tobie mam nadzieję. Dobry jest Pan i prawy, dlatego wskazuje drogę grzesznikom. Pomaga pokornym czynić dobrze, pokornych uczy dróg swoich. Wszystkie ścieżki Pana są pewne i pełne łaski dla strzegących Jego praw i przymierza. Bóg powierza swe zamiary tym, którzy się Go boją, i objawia im swoje przymierze. Jezus powiedział do swoich uczniów: „Będą znaki na słońcu, księżycu i gwiazdach, a na ziemi trwoga narodów bezradnych wobec szumu morza i jego nawałnicy. Ludzie mdleć będą ze strachu, w oczekiwaniu wydarzeń zagrażających ziemi. Albowiem moce niebios zostaną wstrząśnięte. Wtedy ujrzą Syna Człowieczego, przychodzącego na obłoku z wielką mocą i chwałą. A gdy się to dziać zacznie, nabierzcie ducha i podnieście głowy, ponieważ zbliża się wasze odkupienie. Uważajcie na siebie, aby wasze serca nie były ociężałe wskutek obżarstwa, pijaństwa i trosk doczesnych, żeby ten dzień nie przypadł na was znienacka jak potrzask. Przyjdzie on bowiem na wszystkich, którzy mieszkają na całej ziemi. Czuwajcie więc i módlcie się w każdym czasie, abyście mogli uniknąć tego wszystkiego, co ma nastąpić, i stanąć przed Synem Człowieczym”."

Drugi zestaw to zbiór pieśni i piosenek chrześcijańskich (dalej zwany *śpiewnikiem*), do których porównywany był pod względem znaczeniowym zadany tekst i wśród których odbywało się wyszukiwanie. Pochodzą one ze zbioru prezentacji multimedialnych wykorzystywanych do wyświetlania tekstu podczas mszy i nabożeństw we wrocławskim duszpasterstwie akademickim Wawrzyny. Każda prezentacja składała się z kilku slajdów, na których umieszczono poszczególne zwrotki piosenek. Tekst ze wszystkich slajdów danej piosenki był ze sobą łączony w całość – podobnie jak w przypadku Czytań. Zbiór wszystkich tak skonstruowanych piosenek nazwano *Śpiewnikiem*.

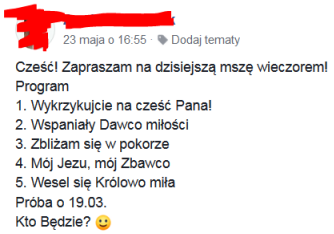


Rysunek 11 – Koncepcja zestawu drugiego - Śpiewnik

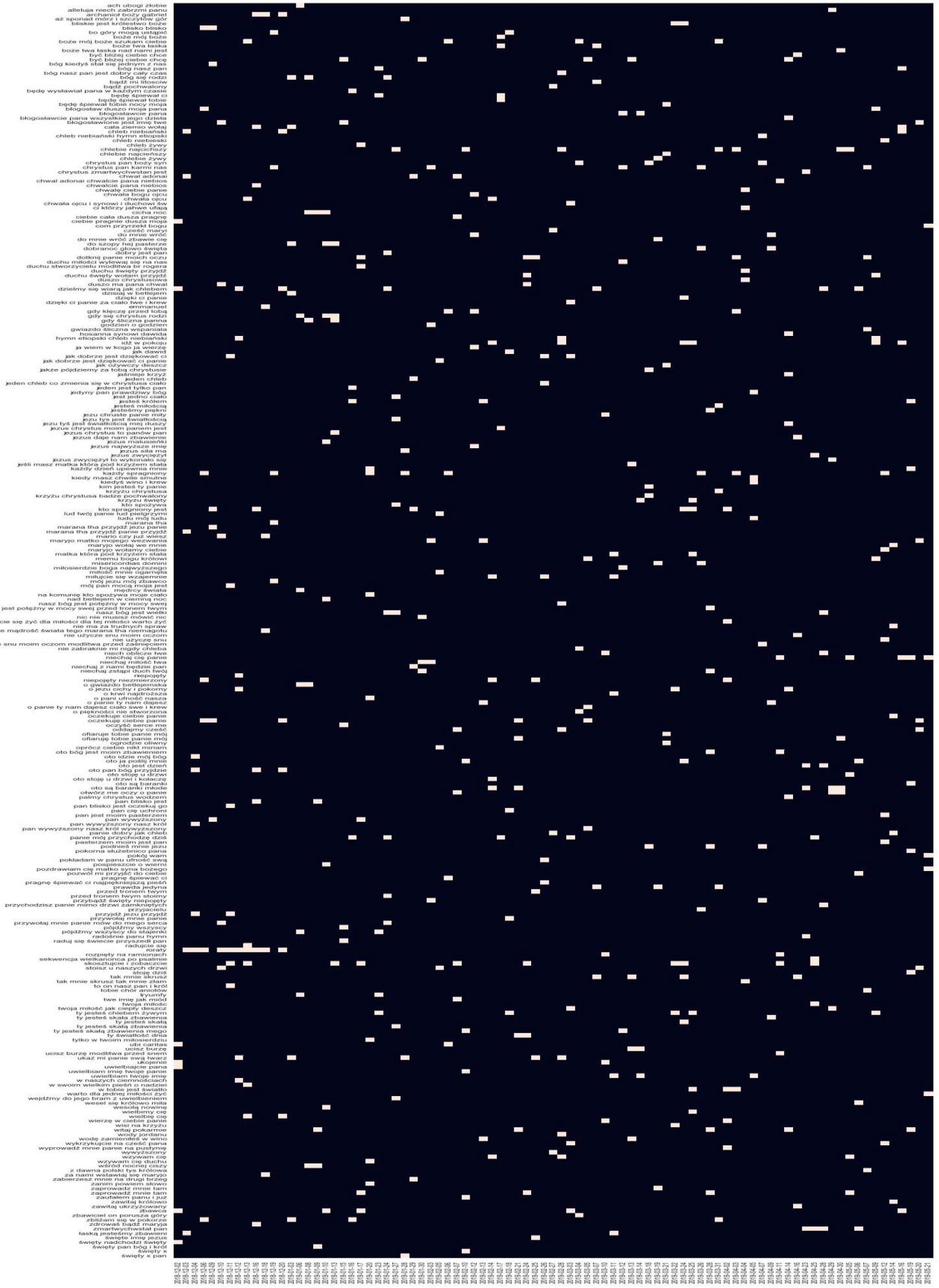
Przykład piosenki:

"Ty wyzwoliłeś nas Panie z kajdan i samych siebie, a Chrystus stając się bratem nauczył nas wołać do Ciebie: Abba Ojcze! Abba Ojcze! Abba Ojcze! Bo Kościół jak drzewo życia w wieczności zapuszcza korzenie, przenika naszą codzienność i pokazuje nam Ciebie. Abba Ojcze! Abba Ojcze! Abba Ojcze! Bóg hojnym Dawcą jest życia, on wyswobodził nas z śmierci i przygarniając do siebie uczynił swoimi dziećmi. Abba Ojcze! Abba Ojcze! Abba Ojcze! Wszyscy jesteśmy braćmi, jesteśmy jedną rodziną. Tej prawdy nic już nie zaćmi i teraz jest jej godzina. Abba Ojcze! Abba Ojcze! Abba Ojcze! "

Trzeci zestaw potraktowano jako zestaw ewaluacyjny. Zawierał tytuły wykorzystanych w poszczególne dni pieśni i piosenek wraz z datą. Dane pochodzą z grupy założonej na portalu społecznościowym Facebook. Skupia ona osoby odpowiedzialne za oprawę muzyczną i dobór repertuaru podczas mszy we wspomnianym wyżej duszpasterstwie. Można zatem traktować wybory tych osób co do piosenek używanych danego dnia jako specjalistyczną wiedzę ekspercką i przyjąć jako zawsze poprawne. Na potrzeby badań jeden zbiór proponowanych piosenek wybranych danego dnia został nazwany *rekomendacją ewaluacyjną*.



Rysunek 12 - Przykładowy post na portalu Facebook [39]



Rysunek 13 - Rozkład wyborów pieśni w poszczególne dni

6.2. Podstawowe statystyki

W Tabeli 1. zestawiono kilka podstawowych statystyk dotyczących wszystkich trzech zestawów.

Liczba dni	87
Liczba piosenek w śpiewniku	185
Liczba wszystkich piosenek w zbiorach rekomendacji ewaluacyjnych	494
Liczba unikalnych piosenek w zbiorach rekomendacji ewaluacyjnych	279
Liczba unikalnych piosenek w zbiorach rekomendacji ewaluacyjnych, które zawiera śpiewnik	87
Procentowy udział unikalnych piosenek ze zbiorów rekomendacji ewaluacyjnych, które zawiera śpiewnik w liczbie wszystkich piosenek w śpiewniku	47%
Suma wszystkich piosenek ze zbiorów rekomendacji ewaluacyjnych, które zawiera śpiewnik	215
Procentowy udział sumy wszystkich piosenek ze zbiorów rekomendacji ewaluacyjnych, które zawiera śpiewnik w liczbie wszystkich piosenek w zbiorach rekomendacji ewaluacyjnych	44%

Tabela 1 - Podstawowe statystyki danych badawczych

7. Metodyka badań

Niniejszy rozdział ma na celu przedstawienie celu, zakresu i planu przeprowadzonych badań.

7.1. Cel badań

Celem badań eksperymentalnych było zbadanie skuteczności i porównanie różnych metod reprezentacji znaczenia i podobieństwa tekstów przedstawionych w rozdziale 4. Jako miarę podobieństwa tekstów przyjęto podobieństwo kosinusowe. Do oceny skuteczności wykorzystano zbiór rekomendacji ewaluacyjnych opisany w rozdziale 6. przyjmując propozycje podane przez muzyków za poprawne.

7.2. Plan badań

Zaproponowano następujący plan badań:

- Implementacja pierwszych etapów procesu przetwarzania języka naturalnego, w tym:
  - pobranie danych
  - tokenizacja
  - czyszczenie danych
- Implementacja różnych metod reprezentacji piosenek
- Implementacja różnych metod reprezentacji czytań - wnioskowanie wektorów czytań na podstawie wyuczonych modeli reprezentacji piosenek

Dla każdej metody reprezentacji:

- Obliczenie odległości kosinusowej reprezentacji czytań od reprezentacji piosenek
- Zaproponowanie *k* najbliższych znaczeniowo piosenek do czytań dla *k* równego liczbie piosenek ze zbioru rekomendacji ewaluacyjnych dla czytania z danego dnia
- Ewaluacja wyliczonych rekomendacji na podstawie porównania z propozycją człowieka (obliczenie sumy tożsamych tytułów dla danego dnia piosenek proponowanych i pochodzących ze zbioru rekomendacji ewaluacyjnych).

7.3. Zakres badań

Przeanalizowano następujące metody reprezentacji znaczenia tekstów:

- Bag of words
  - wersja zliczająca sam fakt wystąpienia (binarna)
  - wersja zliczająca liczbę wystąpień (niebinarna)
- TF-IDF
  - wersja bez ingerencji w hiperparametry modelu
  - wersja z minimalnym współczynnikiem IDF = 0.05
- N-gramy
  - bigramy (n=2) w wersji binarnej
  - bigramy (n=2) w wersji zliczającej liczbę wystąpień (niebinarnej)

    - trigramy (n=3) w wersji binarnej
    - trigramy (n=3) w wersji zliczającej liczbę wystąpień (niebinarnej)

- Word2vec (z uwzględnieniem dwóch różnych algorytmów uczących: Negative Sampling i Hierarchical Softmax)
    - uśrednione wektory słów modelu przetrenowanego na Narodowym Korpusie Języka Polskiego, 100 cech, architektura CBOW
    - uśrednione wektory słów modelu przetrenowanego na Narodowym Korpusie Języka Polskiego, 100 cech, architektura Skip-gram
    - uśrednione wektory słów modelu przetrenowanego na Narodowym Korpusie Języka Polskiego, 300 cech, architektura CBOW
    - uśrednione wektory słów modelu przetrenowanego na Narodowym Korpusie Języka Polskiego, 300 cech, architektura Skip-gram

- Doc2vec
    - model przetrenowany na zbiorze piosenek, domyślne hiperparametry
    - model przetrenowany na zbiorze piosenek, architektura PV-DM

7.4. Metoda przeprowadzenia badań

Badania przeprowadzono z wykorzystaniem mocy obliczeniowej komputera w celu obliczenia podobieństwa między poszczególnymi reprezentacjami tekstów oraz porównania wyników z ewaluacyjnym zbiorem danych.

8. Przedstawienie platformy badawczej oraz narzędzi wraz z opisem implementacji.

Implementację, zgodnie z procesem przetwarzania języka naturalnego opisanym w podrozdziale 4.1 podzielono na kilka części: pobranie danych, czyszczenie danych, reprezentację danych oraz obliczanie podobieństwa. Końcową fazą badań była ewaluacja.

Rozwiązania zaimplementowano w języku *Python* w wersji 3.6. z wykorzystaniem następujących bibliotek do analizy tekstów: *nltk*, *gensim*. Do wizualizacji danych użyto bibliotek *matplotlib* i *seaborn* a reprezentacja była wspierana częściowo przez pakiet *scikit-learn* oraz *pandas*. Jako środowiska użyto interaktywnych notatników *Jupyter*.

8.1. Pobranie danych

Do pobierania fragmentów biblijnych napisano skrypt, który *scrapuje* odpowiednie czytania ze strony internetowej www.mateusz.pl/czytania. Użytkownik podaje liczbę a program zwraca wszystkie czytania od dnia teraźniejszego do podanej liczby dni wstecz. W przypadku święta uwzględniona jest zwiększona liczba czytań. Wykorzystane biblioteki to: *requests* oraz *beautiful soup 4*. Ściągnięte teksty zapisywane były w surowej formie do ramki danych. Zawierały one wiele znaczników html oraz zbędnych fragmentów. Stworzono więc kilka metod pomocniczych, które z wykorzystaniem wyrażeń regularnych ostatecznie sprowadzały teksty do postaci języka naturalnego.

	Dzień tygodnia	Pierwsze czytanie	Psalm_ref	Psalm	Drugie czytanie	Werset przed Ewangelią	Ewangelia
2019-05-22	Wednesday	W Antiochii niektórzy przybysze z Judei naucza...	Idźmy z radością na spotkanie Pana	Ucieszyłem się, gdy mi powiedziano: „Pójdziemy...	None	Trwajcie we Mnie, a Ja w was trwał będę. Kto...	Jezus powiedział do swoich uczniów: „Ja jestem...
2019-05-21	Tuesday	Do Listry nadeszli żydzi z Antiochii i z Ikoni...	Niech wierni Twoi głoszą Twe królestwo	Niech Cię wielbią, Panie, wszystkie Twoje dzie...	None	Trzeba, by wywyższono Syna Człowieczego, aby...	Jezus powiedział do swoich uczniów: „Pokój zos...

Rysunek 14 - Pobrane i sprowadzone do języka naturalnego fragmenty biblijnych czytań z danego dnia

Bazę piosenek w postaci śpiewnika zbudowano w oparciu o 185 prezentacji multimedialnych programu *Power Point*. Każdy slajd zawiera zwrotkę lub refren jednej piosenki. Przy pomocy biblioteki *python-pptx* napisano program, który zapisuje tytuły i treści całych piosenek w ramce danych.

	Tytuł	Tekst
0	Abba Ojcze	Ty wyzwoliłeś nas Panie z kajdan i samych sie...
1	Alleluja (Niech zabrzmi Panu)	Alleluja, Alleluja, Alleluja, Alleluja. Nie...

Rysunek 15 - Pobrane tytuły i tekst piosenek stanowiących śpiewnik

Jak wspomniano w rozdziale 4. zestaw rekomendacji ewaluacyjnych stworzony został na podstawie postów z grupy na portalu Facebook. Na początku dane zostały ręczne skopiowane do arkusza kalkulacyjnego programu *Microsoft Excel*.

Lp.	Data	Zestaw
1	2018-12-20	Roraty Archanioł Boży Gabriel Oto Pan Bóg przyjdzie Chleb niebiański Dzielmy się wiarą jak chlebem Wielbię Cię Oczekuję Ciebie Panie
2	2018-12-19	Marana tha Bo góry mogą ustąpić Chrystus Pan karmi nas Każdy spragniony Mój Jezu, mój Zbawco

Rysunek 16 - Początkowy zestaw rekomendacji ewaluacyjnych

Następnie doprowadzono arkusz do postaci gotowej do wczytania jako ramka danych.

	dayID	date	title
0	1	2018-12-20	Roraty
1	1	2018-12-20	Archanioł Boży Gabriel

Rysunek 17 - Rekomendacje ewaluacyjne jako ramka danych

8.2. Tokenizacja i czyszczenie danych

Dane zostały podanne procesowi tokenizacji a następnie czyszczeniu. Usunięte zostały znaki interpunkcyjne i litery. Pozbyto się także wyrazów wchodzących w skład tzw. „stop words". Do lematyzacji wykorzystano rozwiązanie LEM stworzone przez konsorcjum CLARIN-PL na Politechnice Wrocławskiej. Na potrzeby jego użycia napisano prosty skrypt przerabiający czytania i piosenki do plików *txt*.

	title	txt_pl_lem_tokenized
abba ojcze	Abba Ojcze	['wyzwolić', 'pan', 'kajdany', 'siebie', 'chry...
alleluja niech zabrzmi panu	Alleluja (Niech zabrzmi Panu)	['alleluja', 'alleluja', 'alleluja', 'alleluja...
alleluja alleluja amen amen alleluja	Alleluja, Alleluja, Amen Amen, Alleluja	['bóg', 'umiłować', 'świat', 'syn', 'swój', 'j...
blisko blisko blisko jesteś	Blisko, blisko, blisko jesteś	['piękność', 'niestworzony', 'raz', 'poznać', ...
bo góry mogą ustąpić	Bo góry mogą ustąpić	['stworzyciel', 'duch', 'przyjść', 'nawiedzić'...

Rysunek 18 - Przykładowe dane ze śpiewnika po tokenizacji i czyszczeniu

	txt_pl_lem_tokenized
2018-12-02	['pan', 'mówić', 'nadchodzić', 'dzień', 'wypeł...
2018-12-03	['ow', 'dzień', 'odrośl', 'pan', 'stanąć', 'oz...
2018-12-04	['wyrosnąć', 'różdżka', 'pień', 'jessego', 'wy...
2018-12-06	['dzień', 'śpiewać', 'pieśń', 'ziemia', 'judzk...
2018-12-09	['złoże', 'jeruzalem', 'szata', 'smutek', 'utr...

Rysunek 19 - Przykładowe dane ze zbioru czytań po tokenizacji i czyszczeniu

8.3. Reprezentacja danych

Kolejnym krokiem było stworzenie różnych reprezentacji danych według planu badań z rozdziału 5. Wykorzystano w tym celu biblioteki *nltk*, *gensim* i *sciki-learn*. Wszystkie cechy tworzone były na podstawie piosenek ze śpiewnika. Reprezentację czytań tworzono dokładnie na podstawie tych samych cech.

	abba	abraham	adonai	alleluja	amen	amor	anioł	archanioł	ave	ażebym	...	żeś	żeśmy	życie	życiodajny	żywioła	żywić
abba ojcze	12	0	0	0	0	0	0	0	0	0	...	0	0	2	0	0	0
alleluja niech zabrzmi panu	0	0	0	20	1	0	0	0	0	0	...	0	0	0	0	0	0

Rysunek 20 - Przykładowa reprezentacja piosenek w modelu Bag of words (niebinarnym)

	0	1	2	3	4	5	6	7	8	9	...
2018-12-02	-0.121121	-0.059556	0.082484	-0.045753	0.116072	0.151930	0.177413	-0.006329	-0.021609	-0.100500	...
2018-12-03	-0.105592	-0.029397	0.084049	-0.038691	0.111987	0.162478	0.181121	-0.011368	-0.022192	-0.087183	...
2018-12-04	-0.107072	-0.028963	0.083541	-0.050785	0.107085	0.159416	0.191822	-0.007065	-0.025459	-0.088049	...
2018-12-06	-0.113271	-0.033094	0.088512	-0.033768	0.110105	0.165157	0.178141	-0.002147	-0.031010	-0.097023	...
2018-12-09	-0.108156	-0.041641	0.077921	-0.029807	0.111689	0.155726	0.169806	-0.005059	-0.024685	-0.091935	...

5 rows × 300 columns

Rysunek 21 - Przykładowa reprezentacja czytań z użyciem modelu word2vec (300 cech)

Dane ewaluacyjne również sprowadzono do reprezentacji w postaci wektorów i zapisano w ramce danych. Indeksem są tytuły piosenek (po tokenizacji i lematyzacji) a cechami (kolumnami) dni, w których piosenki były używane.

	2018-12-02	2018-12-03	2018-12-04	2018-12-06	2018-12-09	2018-12-10	2018-12-11	2018-12-12	2018-12-13
święte imię jezus	0	0	0	0	0	0	0	0	0
święty nadchodzi święty	1	0	0	0	0	0	0	0	0

Rysunek 22 - Przykładowa reprezentacja rekomendacji ewaluacyjnych

8.4. Obliczanie podobieństw

Podobieństwa między piosenkami a czytaniami obliczono dla każdego modelu reprezentacji przy użyciu miary podobieństwa kosinusowego. Do tego wykorzystano moduł *metrics* z biblioteki *sckit-learn.*

	2018-12-02	2018-12-03	2018-12-04	2018-12-06	2018-12-09	2018-12-10	2018-12-11	2018-12-12	2018-12-13	2018-12-16	..
abba ojcze	0.000000	0.000000	0.000000	0.0	0.0	0.0	0.0	0.0	0.00000	0.000000	..
alleluja niech zabrzmi panu	0.019375	0.025833	0.025144	0.0	0.0	0.0	0.0	0.0	0.08437	0.000000	..

Rysunek 23 - Ramka danych z wyliczonym podobieństwem kosinusowym między reprezentacjami piosenek i czytań w postaci bigramów

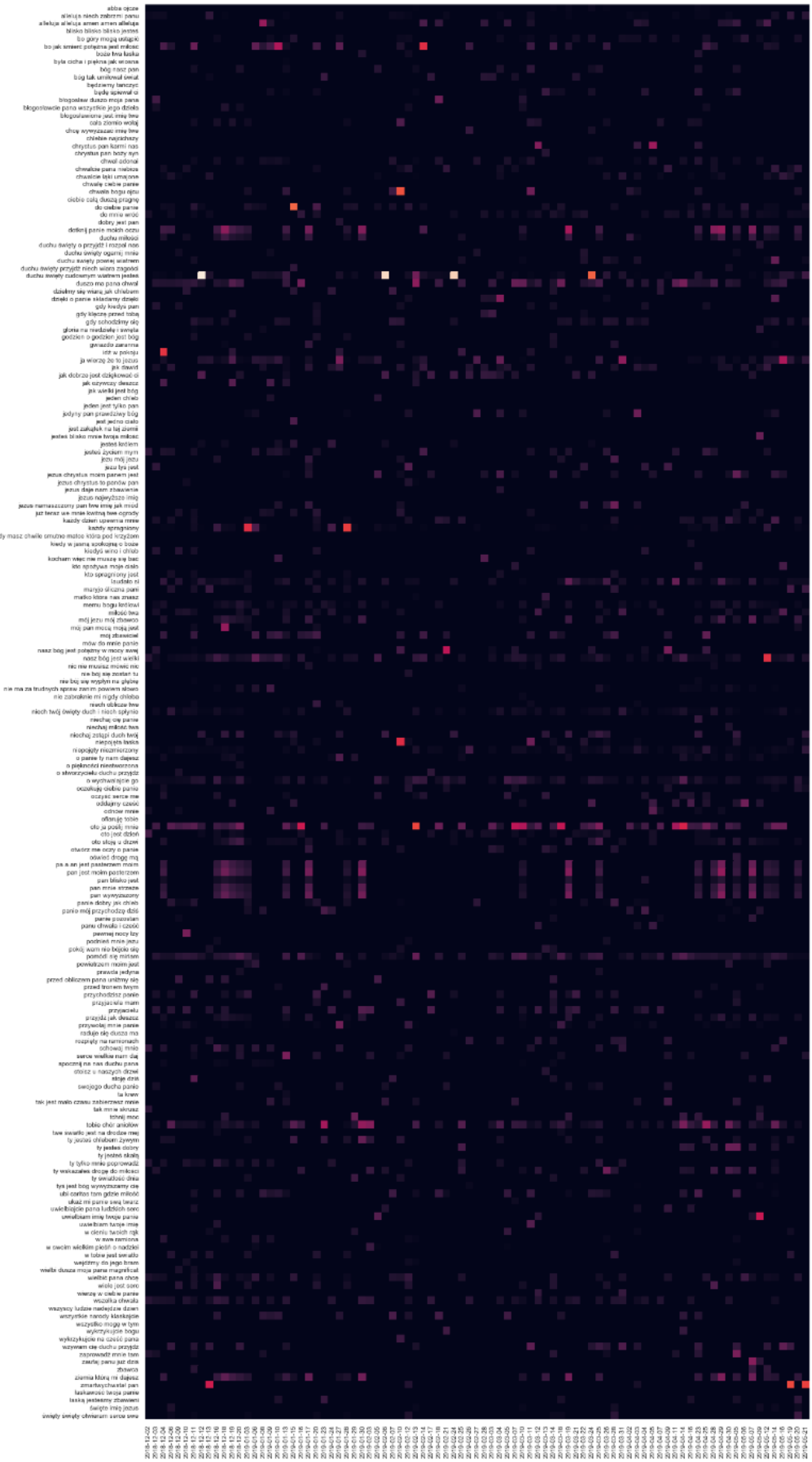
W celu łatwiejszej interpretacji przedstawiono podobieństwa jako tzw. „mapę ciepła”. Przykład przedstawia Rys.24.

8.5. Ewaluacja

Ewaluacji dokonano zgodnie z planem zaprezentowanym w rozdziale 7. Na początku obliczono liczbę *k* piosenek rekomendowanych danego dnia w zbiorze rekomendacji ewaluacyjnych tworząc wektor. Następnie dla każdej reprezentacji na podstawie wyników podobieństwa kosinusowego w ramce danych dla danego dnia, wybrano *k* najbardziej podobnych do czytań piosenek. Jeśli rekomendowany przez program tytuł był tożsamy z rekomendowanym tytułem w zbiorze rekomendacji ewaluacyjnych – elementowi w wektorze o indeksie danego dnia zwiększano wartość o 1. Na koniec sumowano wszystkie wartości wektora zwracając ogólną liczbę poprawnych rekomendacji dla danej metody reprezentacji.

Algorytm ewaluacji przedstawić można w postaci pseudokodu:

```
rozpocznij algorytm
zainicjalizuj pustą tablicę x
dla każdego dnia:
   wybierz k najbardziej podobnych piosenek do czytania z dnia
   zainicjalizuj zmienną licznik
   dla każdej z k wybranych piosenek:
       jeśli tytuł wybranej piosenki znajduje się w rekomendacjach
              ewaluacyjnych tego dnia
       zwiększ licznik o 1
   przypisz licznik elementowi x o indeksie dnia
zwróć sumę elementów tablicy x
zakończ algorytm
```

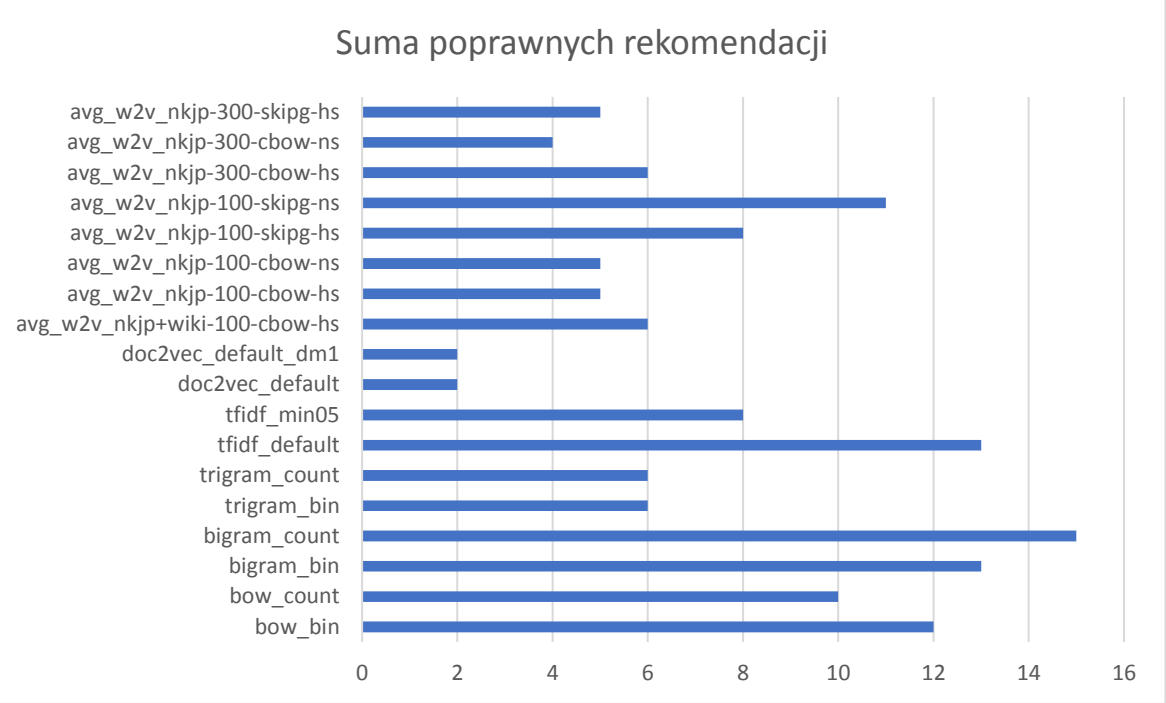


Rysunek 24 - "Mapa ciepła" podobieństw między piosenkami a czytaniami jako reprezentacja z użyciem bigramów

9. Omówienie wyników badań

9.1. Przedstawienie wyników badań

Opisany w podrozdziale 8.5 algorytm ewaluacji zastosowany do wymienionych w rozdziale 7. metod reprezentacji daje następujące rezultaty.

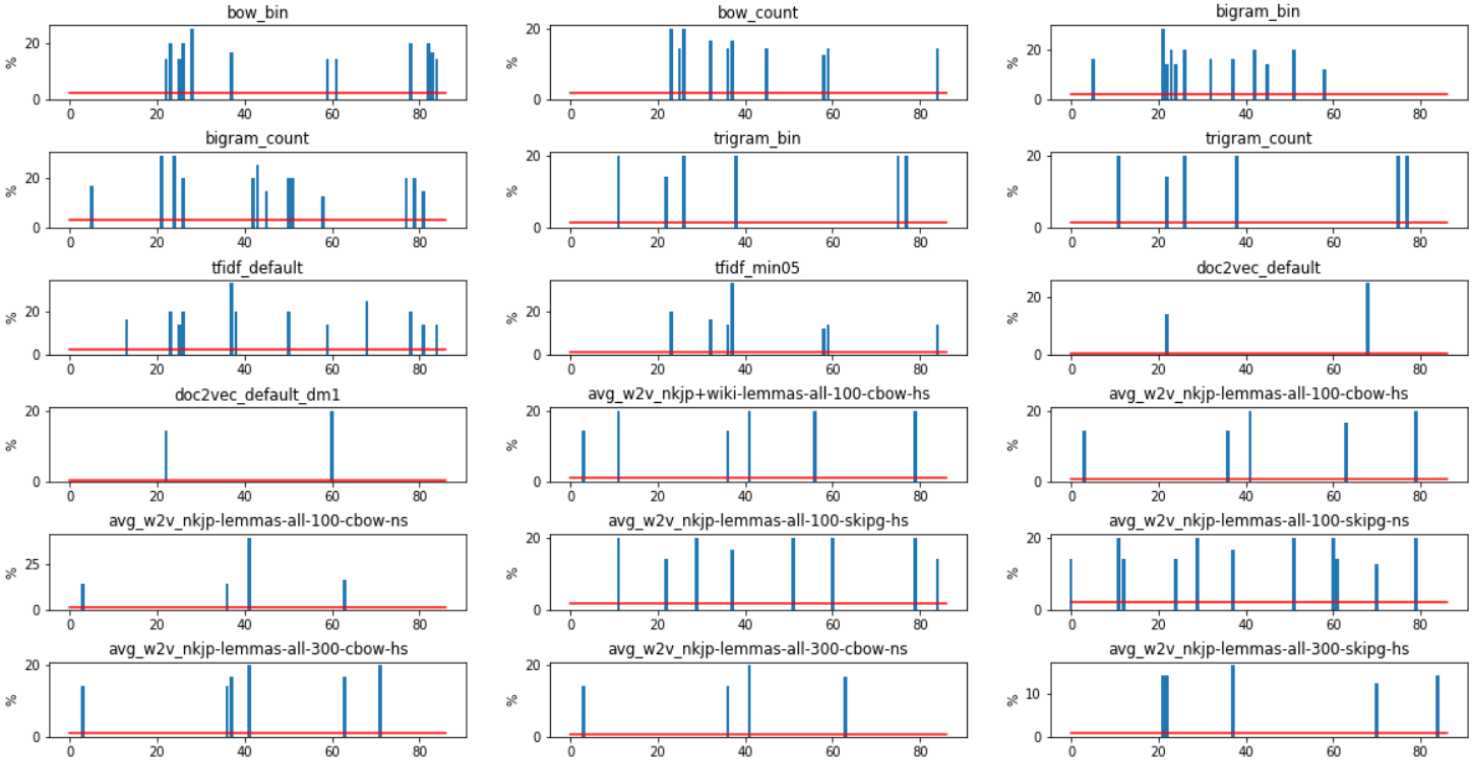


Wykres 1 - Suma poprawnych rekomendacji z użyciem poszczególnych reprezentacji znaczenia

Procentowy udział poprawnych rekomendacji w liczbie wszystkich czytań (dni) w rozróżnieniu na poszczególne modele przedstawia się następująco.

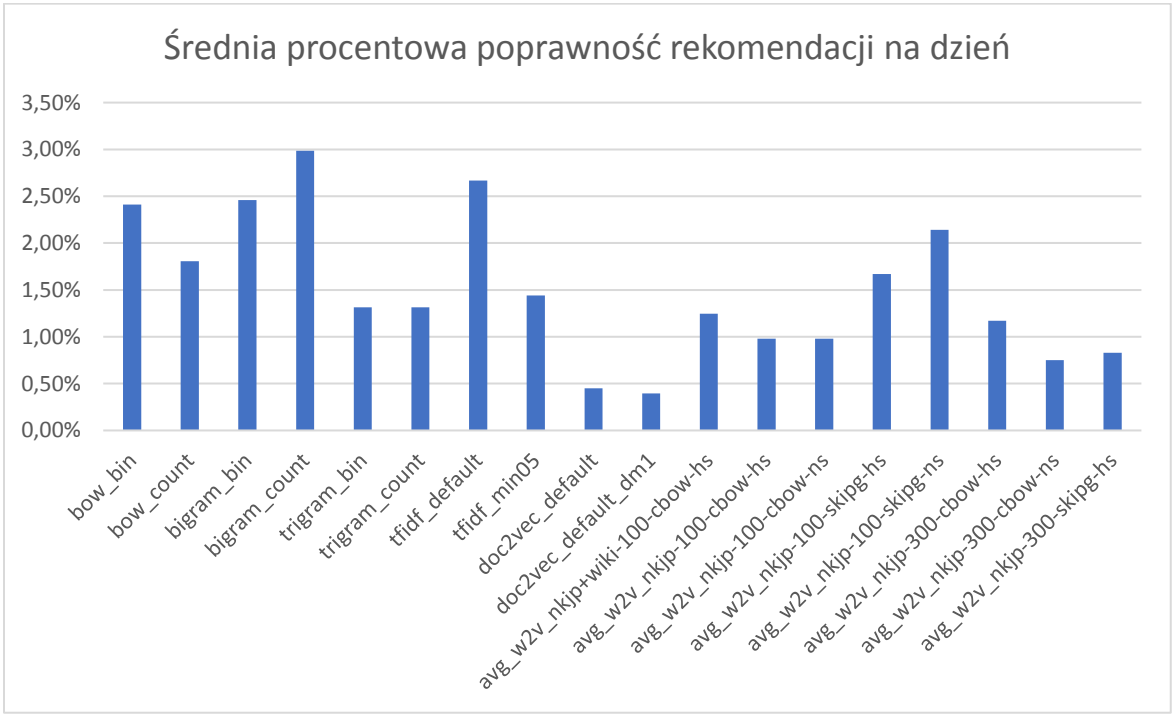
Wykres 2 - Procentowy udział poprawnych rekomendacji w liczbie dni

Procentowy udział poprawnych rekomendacji w poszczególne dni z rozróżnieniem na modele przedstawia się następująco. Czerwona linia prezentuje średnią.



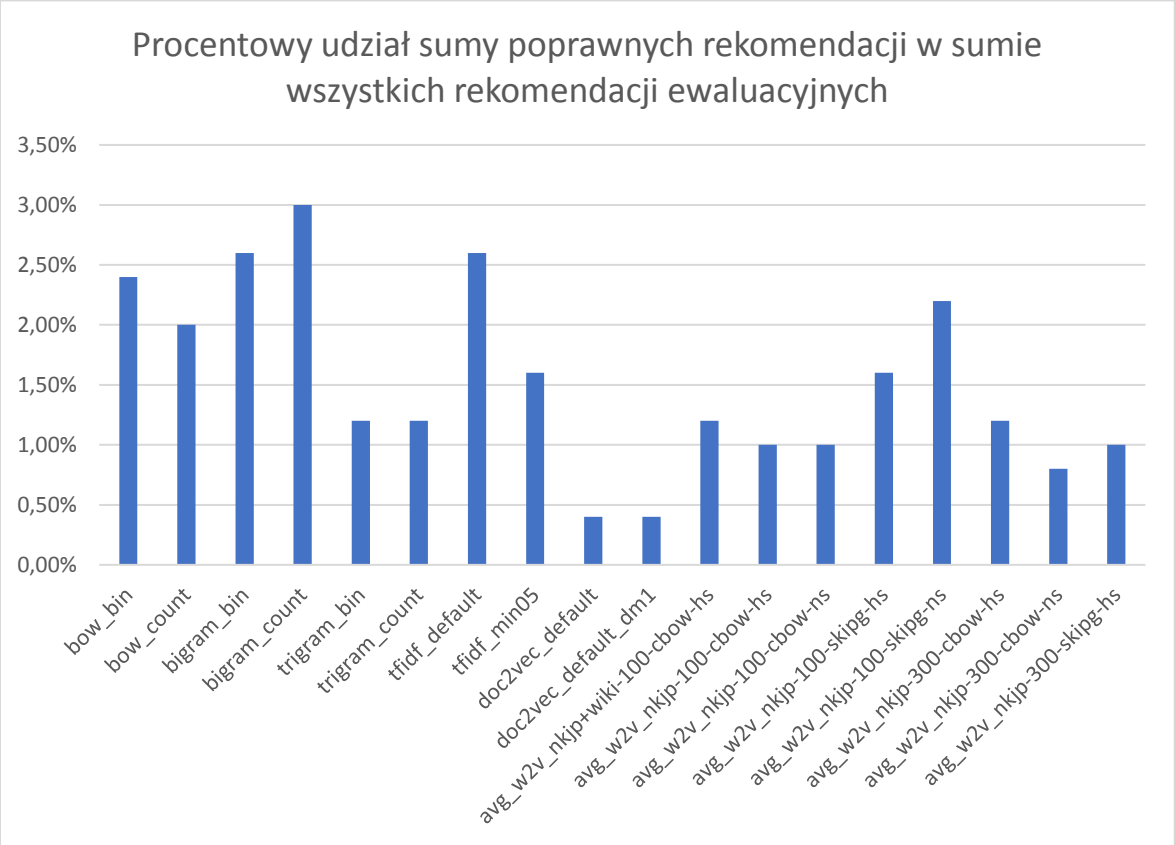
Rysunek 25 – Wykresy przedstawiające procentowy udział poprawnych rekomendacji w poszczególne dni dla każdego modelu wraz ze średnią

Powyższe średnie dla każdego modelu przedstawia Wykres 3.



Wykres 3 - Średnia procentowa poprawność rekomendacji na dzień

Dodatkowo, dla każdego modelu policzono procentowy udział sumy poprawnych rekomendacji w sumie wszystkich rekomendacji ewaluacyjnych. Wyniki prezentuje poniższy wykres.



Wykres 4 - Procentowy udział sumy poprawnych rekomendacji w sumie wszystkich rekomendacji ewaluacyjnych

9.2. Interpretacja wyników

Najlepszym rozwiązaniem okazał się model reprezentacji budowany w oparciu o bigramy w wersji niebinarnej, to znaczy takiej, w której elementy poszczególnych wektorów przedstawiały sumę występowania danych par słów w tekstach. Model ten uzyskał najlepszy wynik zarówno w sumie poprawnych rekomendacji, jak i średniej procentowej poprawności rekomendacji na dzień.

Niski procentowy udział sumy poprawnych rekomendacji w sumie rekomendacji ewaluacyjnych może być spowodowany faktem, iż tylko niektóre piosenki ze zbioru rekomendacji ewaluacyjnych znajdują się w śpiewniku. Ponadto, tytuły piosenek proponowane przez muzyków, na podstawie których zbudowano zbiór rekomendacji ewaluacyjnych nie są spójne – różne osoby inaczej nazywały te same piosenki a dodatkowo często odmiennie niż nazwa zapisana w śpiewniku.

Najgorszymi modelami okazały się metody doc2vec – reprezentujące w postaci wektorów całe dokumenty. Może to być spowodowane niedużą wielkością zbioru danych.

10. Wykorzystanie wyników badań

Wyznaczoną w wyniku badań najlepszą metodę do wydobywania znaczenia i wyszukiwania tekstów o zbliżonym znaczeniu wykorzystano do zbudowania systemu rekomendacyjnego piosenek o nazwie *PiosenKatoR – Rekomendator Piosenek Katolickich* zaproponowanego w podrozdziale 3.4.

System powstał w oparciu o rozwiązania implementacyjne opisane w rozdziale 8. Na początku narzędzie pobiera aktualne na dany dzień czytania i łączy w jednolity tekst. Następnie przy użyciu interfejsu programistycznego udostępnionego przez konsorcjum CLARIN, tekst zostaje poddany procesowi lematyzacji. Stokenizowany i wyczyszczony tekst jest gotowy do użycia przez najlepszą znalezioną w wyniku badań metodę reprezentacji znaczenia. Modelowanie odbywa się dwustopniowo: najpierw tworząc wektory piosenek, które za indeksy przyjmują poszczególne bigramy, a następnie wektor czytania używając tych samych cech. Następnie liczone jest podobieństwo kosinusowe pomiędzy wektorem czytań a wektorem każdej piosenki. Program zwraca czytanie z danego dnia i 5 tytułów piosenek, które najbardziej pasują do niego pod względem znaczeniowym.

```
python .\piosenkator_app.py
Paweł powiedział do starszych Kościoła efeskiego: „Uważajcie na samych siebie i na całą trzodę, na
d którą Duch Święty ustanowił was biskupami, abyście kierowali Kościołem Boga, który On nabył włas
ną krwią. Wiem, że po moim odejściu wejdą między was wilki drapieżne, nie oszczędzając trzody. Tak
że spośród was samych powstaną ludzie, którzy głosić będą przewrotne nauki, aby pociągnąć za sobą
uczniów. Dlatego czuwajcie, pamiętając, że przez trzy lata we dnie i w nocy nie przestawałem ze łz
ami upominać każdego z was. A teraz polecam was Bogu i słowu Jego łaski władnemu zbudować i dać dz
iedzictwo z wszystkimi świętymi. Nie pożądałem srebra ani złota, ani szaty niczyjej. Sami wiecie,
że te ręce zarabiały na potrzeby moje i moich towarzyszy. We wszystkim pokazałem wam, że tak pracu
jąc trzeba wspierać słabych i pamiętać o słowach Pana Jezusa, który powiedział: "Więcej szczęścia
jest w dawaniu aniżeli w braniu". Po tych słowach upadł na kolana i modlił się razem ze wszystkimi
. Wtedy wszyscy wybuchnęli wielkim płaczem. Rzucali się Pawłowi na szyję i całowali go, smucąc się
 najbardziej z tego, co powiedział: że już nigdy go nie zobaczą. Potem odprowadzili go na statek.
 Śpiewajcie Bogu wszystkie ludy ziemi O Boże, okaż swą potęgę, potęgę Bożą, której dla nas użyłeś.
 W Twej świątyni nad Jeruzalem, niech królowie złożą Tobie dary!śpiewajcie Bogu królestwa ziemi, z
agrajcie Panu, który przemierza odwieczne niebiosa. Oto wydał głos swój, głos potężny: „Uznajcie m
oc Bożą!”Jego majestat jest nad Izraelem, a Jego potęga w obłokach. On sam swojemu ludowi daje pot
ęgę i siłę. Niech będzie Bóg błogosławiony.   Słowo Twoje, Panie jest prawdą, uświęć ich w prawdzi
e W czasie ostatniej wieczerzy Jezus podniósłszy oczy ku niebu, modlił się tymi słowami: „Ojcze Św
ięty, zachowaj ich w Twoim imieniu, które Mi dałeś, aby tak jak My stanowili jedno. Dopóki z nimi
byłem, zachowywałem ich w Twoim imieniu, które Mi dałeś, i ustrzegłem ich, a nikt z nich nie zginą
ł z wyjątkiem syna zatracenia, aby się spełniło Pismo. Ale teraz idę do Ciebie i tak mówię, będąc
jeszcze na świecie, aby moją radość mieli w sobie w całej pełni. Ja im przekazałem Twoje słowo, a
świat ich znienawidził za to, że nie są ze świata, jak i Ja nie jestem ze świata. Nie proszę, abyś
 ich zabrał ze świata, ale byś ich ustrzegł od złego. Oni nie są ze świata, jak i Ja nie jestem ze
 świata. Uświęć ich w prawdzie. Słowo Twoje jest prawdą. Jak Ty Mnie posłałeś na świat, tak i Ja i
ch na świat posłałem. A za nich Ja poświęcam w ofierze samego siebie, aby i oni byli uświęceni w p
rawdzie”.

Pan – jest moim pasterzem
Dotknij Panie moich oczu
Pan wywyższony
Ziemia, którą mi dajesz
Pa-a-an jest pasterzem moim
```

Rysunek 26 - Wynik działania systemu PiosenKatoR

11. Zakończenie

11.1. Podsumowanie

Osiągnięto cel pracy polegający na zbadaniu skuteczności i porównaniu wybranych metod wydobywania znaczenia z zadanego tekstu i wyszukiwania tekstów o podobnym znaczeniu. Jako tekst zadany wykorzystano fragmenty biblii, a tekstami, wśród których odbywało się wyszukiwanie tekstów podobnych był zbiór piosenek o tematyce religijnej.

Pobrane teksty zostały poddane procesowi tokenizacji i czyszczenia, w którego skład wchodziły następujące kroki: usuwanie szumu i pozbycie się tzw. „stop words", ujednolicenie wielkości słów, lematyzacja. Następnie, używając rozmaitych metod, utworzono reprezentację znaczenia tekstów jako wektorów cech.

Przeprowadzono badania porównawcze i zidentyfikowano metody, które są skuteczniejsze niż inne w reprezentacji znaczenia i porównywaniu jego podobieństwa. W tym celu posłużono się zbiorem rekomendacji ewaluacyjnych a do obliczenia podobieństwa między reprezentacjami znaczenia wykorzystano podobieństwo kosinusowe.

Najlepsze metody to przede wszystkim metody opierające się o wykorzystanie jako cech bigramów oraz współczynnika TF-IDF. W związku z powyższym można wyciągnąć wniosek, iż teksty najbardziej do siebie podobne pod względem znaczeniowym to takie, wśród których występują te same słowa lub zbitki słów.

Wyniki badań wykorzystano do budowy systemu rekomendacyjnego *PiosenKatoR – Rekomendatora Piosenek Katolickich*, w którym użyto zidentyfikowaną w trakcie badań najlepszą z metod – opierającą się o wykorzystanie bigramów.

11.2. Kierunki dalszych prac

Przeprowadzone badania można poszerzyć o zastosowanie tłumaczenia tekstów na język angielski, innej miary podobieństwa tekstów czy innego sposobu reprezentacji znaczenia.

Dodatkowo, samo narzędzie wspomagające muzyków kościelnych podczas doboru repertuaru *PiosenKatoR – Rekomendator Piosenek Katolickich* może być rozbudowane o graficzny interfejs użytkownika (ang. graphical user interface, GUI) czy dalej w aplikację webową. Nowoczesne trendy wskazywałyby również opracowanie aplikacji mobilnej na najbardziej popularne systemy operacyjne.

Literatura

[1] F. Provost, T. Fawcett, *Data Science for Business. What You Need to Know About Data Mining and Data-Analytic Thinking* , O'Reilly Media, Inc, 2013

[2] D. Sarkar, *Text Analytics with Python: A Practical Real-World Approach to Gaining Actionable Insights from Your Data*, Apress, 2016

[3] V. Mayer – Schönberger, K. Cukier, *Big Data. Rewolucja, która zmieni nasze myślenie, pracę i życie* , MT Biznes, 2014

[4] Y. Lin, J. Jiang, S. Lee, *A Similarity Measure for Text Classification and Clustering*, in IEEE Transactions on Knowledge and Data Engineering, vol. 26, no. 7, pp. 1575-1590, lipiec 2014.

[5] Z. Hyung, K. Lee, K. Lee, *Music recommendation using text analysis on song requests to radio stations, Expert Systems with Applications*, Volume 41, Issue 5, Pages 2608-2618, 2014

[6] G. Salton, A. Wong, C. Yang, *A Vector Space Model for Automatic Indexing*. Commun. ACM. 18. 613-617, 1975

[7] A. Figiel, *Tekst jako wzorzec informacyjny – automatyczna ocena podobieństwa tematycznego tekstów za pomocą Latent Semantic Analysis* w pracy zbiorowej *Słowniki komputerowe i automatyczna ekstrakcja informacji z tekstu* pod redakcją W. Lubaszewskiego, Wydawnictwo AGH, 2009

[8] Q. Le, T. Mikolov, *Distributed representations of sentences and documents*. In Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32 (ICML'14), E. Xing , T. Jebara (Eds.), Vol. 32., 2014

[9] Z. Harris, *Distributional Structure*, WORD, 10:2-3, 146-162, 1954

[10] K. Kowsari, K. Jafari Meimandi, M. Heidarysafa, S. Mendu, L. Barnes, D. Brown, *Text Classification Algorithms: A Survey*. Information, *10*(4), 150., 2019

[11] T. Mikolov, et al., *Efficient estimation of word representations in vector space*, 2013

[12] T. Mikolov, et al., *Distributed representations of words and phrases and their compositionality.* Advances in neural information processing systems, 2013

[13] J. Pennington, R. Socher, C. Manning, *Glove: Global vectors for word representation*. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) (pp. 1532-1543), 2014

[14] P. Bojanowski, E. Grave, A. Joulin, T. Mikolov, *Enriching word vectors with subword information*, Transactions of the Association for Computational Linguistics, *5*, 135-146, 2017

[15] D. Raj, *How to obtain Sentence Vectors?*, Medium, 12 kwietnia 2018 [Dostęp: 6 czerwca 2019], https://medium.com/explorations-in-language-and-learning/how-to-obtain-sentence-vectors-2a6d88bd3c8b

[16] J. Lau, T. Baldwin, *An empirical evaluation of doc2vec with practical insights into document embedding generation*, 2016

[17] J. Flisar, V. Podgorelec, *Document Enrichment using DBPedia Ontology for Short Text Classification*. In Proceedings of the 8th International Conference on Web Intelligence, Mining and Semantics (p. 8). ACM, czerwiec 2018

[18] Z. Hyung, M. Lee, K. Lee, *Music recommendation based on text mining. In* IMMM 2012, the second international conference on advances in information mining and management (pp. 129–134), 2012

[19] S. Deerwester, S. Dumais, G. Furnas, T. Landauer, R. Harshman, *Indexing by latent semantic analysis*. Journal of the American society for information science, 41(6), 391-407, 1990

[20] T. Hofmann, *Probabilistic latent semantic analysis*. In Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence (pp. 289-296). Morgan Kaufmann Publishers Inc., July 1999

[21] F. Isinkaye, Y. Folajimi, B. Ojokoh, *Recommendation systems: Principles, methods and evaluation.* Egyptian Informatics Journal, 16(3), 261-273, 2015

[22] G. Reynolds, et al. *Interacting with large music collections: Towards the use of environmental metadata*. 2008 IEEE International Conference on Multimedia and Expo. IEEE, 2008.

[23] J. Yan, *Text Representation*, Microsoft Research, 163, 2009

[24] D. Cielen, A. Meysman, M. Ali. *Introducing Data Science: Big Data, Machine Learning, and More, Using Python Tools* (1st ed.). Manning Publications Co., Greenwich, CT, USA, 2016.

[25] B. Habert, et al. *Towards tokenization evaluation*. Proceedings of LREC. Vol. 98. 1998

[26] S. Bird, E. Klein, E. Loper. *Natural Language Processing with Python* (1st ed.). O'Reilly Media, Inc, 2009

[27] J. Singh, V. Gupta. *Text Stemming: Approaches, Applications, and Challenges*. *ACM* Comput. Surv. 49, 3, Article 45 (September 2016), 46 pages, 2016

[28] P. Joël, N. Lavrac, D. Mladenic. *A rule based approach to word lemmatization*. Proceedings of IS-2004: 83-86, 2004

[29] J. Brownlee, *A Gentle Introduction to the Bag-of-Words Model*, Machine Learning Mastery, 9 października 2017 [Dostęp: 6 czerwca 2019], https://machinelearningmastery.com/gentle-introduction-bag-words-model/

[30] J. Martin, D. Jurafsky. *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition*. Upper Saddle River: Pearson/Prentice Hall, 2009.

[31] J. Gilyadov, *Word2Vec Explained*, Hacker's Blog, 23 marca 2017 [Dostęp: 6 czerwca 2019], https://israelg99.github.io/2017-03-23-Word2Vec-Explained/

[32] S. Dimitris, N. Passalis, A.Tefas. *Interactive dimensionality reduction using similarity projections*. Knowledge-Based Systems 165: 77-91, 2019

[33] A. Dai, C. Olah, Q. Le. *Document embedding with paragraph vectors*., 2014

[34] A. Budhiraja, *Simple introduction to doc2vec* , Medium, 14 maja 2018 [Dostęp: 6 czerwca 2019], https://medium.com/@amarbudhiraja/understanding-document-embeddings-of-doc2vec-bfe7237a26da

[35] G. Shperber, *A gentle introduction to Doc2Vec*, Medium, 26 lipca 2017 [Dostęp: 6 czerwca 2019], https://medium.com/scaleabout/a-gentle-introduction-to-doc2vec-db3e8c0cce5e

[36] C. Emmery, *Euclidean vs. Cosine Distance,* Chris Emmery Blog, 25 marca 2017 [Dostęp: 6 czerwca 2019], https://cmry.github.io/notes/euclidean-v-cosine

[37] O. Yetty, *Implementing similarity measures in python: Cosine Similarity versus Jaccard Similarity,* TechInPink, 4 sierpnia 2017 [Dostęp 6 czerwca 2019], http://techinpink.com/2017/08/04/implementing-similarity-measures-cosine-similarity-versus-jaccard-similarity/

[38] A. Sieg, *Text similarity measures*, Medium, 5 lipca 2018 [Dostęp: 6 czerwca 2019], https://medium.com/@adriensieg/text-similarities-da019229c894