

Uniwersytet Wrocławski

Wydział Matematyki i Informatyki
Instytut Informatyki

Borys Dyczko

**System odpowiadający na pytania
zadawane Wikipedii**

Wikipedia question answering system

Praca magisterska

Praca wykonana pod kierunkiem
dr Pawła Rychlikowskiego

Wrocław 2020

Streszczenie

Celem niniejszej pracy jest zbudowanie i rzetelne opisanie systemu odpowiadającego na pytania w języku naturalnym zadawane polskojęzycznej Wikipedii. Wynikiem zapytania jest artykuł, którego treść zawiera możliwie szczegółową odpowiedź na pytanie. Badane modele wykorzystują metody statystyczne oparte o liczbę wystąpień słów oraz sieci neuronowe. Użyto również algorytmu genetycznego w celu zagregowania dotychczasowych metod. Efektem końcowym pracy jest aplikacja umożliwiająca uzyskanie listy powiązanych tematycznie artykułów dla wpisanego przez użytkownika zapytania.

Abstract

The purpose of this study is to build and reliably describe a system that answers questions asked Polish-language Wikipedia in natural language. As a result of a query is an article which has the most detailed answer as it is possible. Models used in the research use statistical methods which are based on the number of words occurrences and neural networks. In order to aggregate earlier methods, the genetic algorithm was used. The result of this document is an application that allows acquiring the list of connected articles in terms of subject and query provided by the user.

Spis treści

1	Wstęp	7
1.1	Wprowadzenie	7
1.2	Zakres pracy	8
1.3	Opis systemu	9
1.4	Struktura pracy	10
2	Dane, przetwarzanie danych, miary wyników	11
2.1	Artykuły	11
2.2	Słowa	15
2.2.1	Formy bazowe słów	16
2.2.2	Unifikacja słów	17
2.2.3	Stop words	18
2.2.4	Pozycje słów	18
2.3	Pytania	18
2.3.1	Czywieszki łatwe	20
2.3.2	Czywieszki średnie	20
2.3.3	Czywieszki trudne	20
2.4	Ranking artykułów dla pytania	21
2.5	Miary wyników modeli	21
2.5.1	Miary p_1, p_{10}, p_{100}	21
2.5.2	Miara MRR	22
3	Modele statystyczne	24
3.1	Użyte metody	24
3.1.1	Miara TF-IDF	24
3.1.2	Waga artykułu dla pytania	25
3.2	Model oparty o liczbę wystąpień słów	25
3.2.1	Waga słowa, waga artykułu dla pytania	26
3.2.2	Wyniki modelu opartego o liczbę wystąpień słów	26
3.3	Model oparty o liczbę wystąpień bigramów	27
3.3.1	Waga bigramu, waga artykułu dla pytania	27
3.3.2	Wyniki modelu opartego o liczbę wystąpień bigramów	27
3.4	Modele kontekstowe	28
3.4.1	Model kontekstowy oparty o wystąpienia słów	28
3.4.2	Model kontekstowy oparty o wystąpienia bigramów	28
3.4.3	Wyniki modeli kontekstowych	29
3.5	Model odległości wektorów rzadkich	31
3.5.1	Wektor pytania	32

3.5.2	Wektor artykułu	32
3.5.3	Miary podobieństwa	33
3.5.4	Wyniki modeli wektorów rzadkich	34
4	Modele neuronowe	39
4.1	Word2vec	40
4.1.1	Model bazowy	42
4.1.2	Model odpowiadający na pytania	42
4.1.3	Wyniki modelu word2vec	43
4.2	Sieci neuronowe	44
4.2.1	Dane dla sieci neuronowych	44
4.2.2	Konwolucyjna sieć neuronowa	45
4.2.3	Głęboka sieć uśredniająca	46
4.2.4	Poprawiony zbiór danych	47
4.2.5	Wyniki sieci neuronowych	48
4.2.6	Uproszczenie sieci neuronowych	48
5	Model agregujący	52
5.1	Algorytm genetyczny	52
5.1.1	Dane dla algorytmu ewolucyjnego	53
5.1.2	Osobnik	54
5.1.3	Przebieg algorytmu	54
5.1.4	Wyniki modelu opartego o algorytm genetyczny	54
5.1.5	Wyniki algorytmu ewolucyjnego z pominięciem wybranych mo- deli	57
6	Podsumowanie	58
6.1	Wyniki zbiorcze wybranych modeli	58
6.2	Dalszy rozwój	63
6.3	SQAD	63
6.4	Użycie innych języków	63
6.5	Wyszukiwarka	63
6.6	Wybrane problemy natury technicznej	68
Dodatek A	Szczegóły techniczne	70
A.1	Implementacja	70
A.2	Dane wejściowe	71
A.3	Obliczenia	71
A.4	Kod źródłowy	71

Rozdział 1

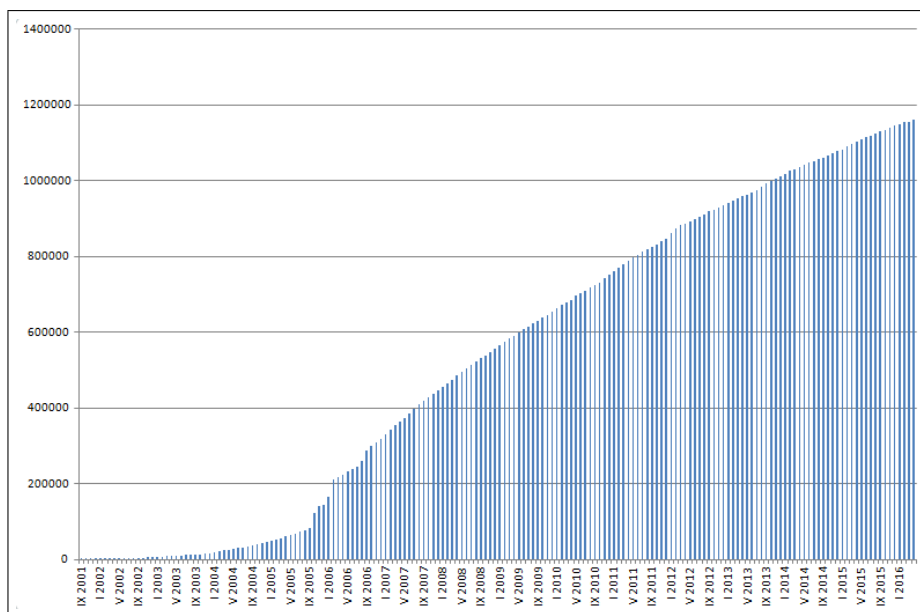
Wstęp

1.1 Wprowadzenie

W erze gigabajtowych danych tekstowych, gdy Internet posiada wiele milionów stron internetowych dostęp do informacji ściśle związanych z tematem jest sprawą kluczową. Użytkownik szukający odpowiednich treści oczekuje kilku wyników, które jest w stanie szybko sprawdzić pod kątem zawierania danej informacji.

Serwis **Wikipedia.pl** na dzień 03.11.2019 posiada w bazie 1 665 102 artykułów. Dla przeciętnego człowieka przejrzenie takiej ilości tekstu, choćby pobieżne, jest niemożliwe do zrealizowania. Dla pytań, na które spodziewana odpowiedź zawiera się, np. w kategorii „*Ludzie związani z Krakowem*” (co jednak nie zawsze jest łatwe do ustalenia) można mocno zawęzić liczbę potencjalnych odpowiedzi. Jednak nawet w tym przypadku, przeczytanie 1026 artykułów przez człowieka jest prawie niemożliwe.

Celem tej pracy jest stworzenie systemu, który ułatwi użytkownikowi przeglądanie dużych zasobów Wikipedii. O ogromnej przydatności i potrzebie istnienia systemów wyszukiwania informacji może świadczyć popularność wszelkich wyszukiwarek, które w ostatnich czasach mają coraz większe znaczenie w wielu dziedzinach życia oraz tempo wzrostu artykułów Wikipedii. Wzrost liczby artykułów został przedstawiony na rysunku 1.



Rysunek 1: Wzrost liczby artykułów polskojęzycznej Wikipedii w latach 2001–2016¹.

1.2 Zakres pracy

Tematyka tej pracy zawiera w sobie kilka dziedzin informatyki.

Przetwarzanie języka naturalnego

Przetwarzanie języka naturalnego (ang. natural language processing, NLP) jest dziedziną łączącą sztuczną inteligencję i językoznawstwo. Zajmuje się automatycznym przetwarzaniem, rozumieniem, tłumaczeniem i generowaniem języka naturalnego przez komputer. Język naturalny to język zrozumiały dla człowieka (np. polski, angielski lub hiszpański). Jest to ostatnio bardzo popularna i wygodna metoda komunikacji człowieka z komputerem. Rozumienie języka naturalnego przez komputer to zadanie bardzo trudne, uważane za jeden z problemów sztucznej inteligencji. Alan Turing uznał kompetencję językową za podstawowy wyznacznik inteligentnego zachowania się komputera i stworzył test Turinga, szczegółowo opisany w artykule [13]. Test polega na rozmowie² w języku naturalnym między sędzią, a pozostałymi stronami, z których jedna jest komputerem. Test uznaje się za zaliczony, jeśli sędzia nie jest w stanie odróżnić maszyny od człowieka. Aby komputer mógł w pełni rozumieć znaczenie słów w tekstach, musiałby rozumieć świat i język ludzki. Istnieje wiele problemów z tym związanych: niejednoznaczność słów (w kontekście), syntaktyczna niejednoznaczność, nieprawidłowe dane (błędy ludzkie) czy akcent. Jednym ze sposobów radzenia sobie z niejednoznacznością, to użycie modeli korzystających z prawdopodobieństwa, statystyki, uczenia maszynowego.

¹Rysunek pobrany ze strony https://pl.wikipedia.org/wiki/Wikipedia:Liczba_artyku%C5%82%C3%B3w_w_polskiej%C4%99zycznej_Wikipedii.

²Rozmowa odbywa się za pośrednictwem komputerowego komunikatora tekstowego w celu zmniejszenia różnic pomiędzy komputerem, a człowiekiem.

Wyszukiwanie informacji

Wyszukiwanie informacji (ang. information retrieval, IR) to dziedzina zajmująca się przetwarzaniem dużych zbiorów danych i wyszukiwaniem w nich informacji. Głównym zadaniem jest optymalne indeksowanie danych (np. dokumentów, zdjęć), aby możliwie najszybciej i najtrafniej udzielać odpowiedzi na zapytania. Większość systemów wyszukiwania liczy wartość numeryczną (wagę) dla obiektów znajdujących się w bazie, a następnie tworzy ranking. Najwyższe pozycje są zwracane użytkownikowi. Naturalnym i najbardziej popularnym obszarem zastosowań dla IR są wyszukiwarki internetowe.

Odpowiadanie na pytania

Odpowiadanie na pytania (ang. question answering, QA) to dziedzina wykorzystująca NLP oraz IR. Zajmuje się automatycznym odpowiadaniem na pytania zadane w języku naturalnym. System udziela odpowiedzi na podstawie wcześniej przygotowanej bazy wiedzy lub w oparciu o nieuporządkowany i nieustrukturalizowany zbiór dokumentów w języku naturalnym. System powinien sobie radzić z szeroką gamą typów pytań, np. : „*jak?*”, „*czemu?*”, „*gdzie?*”, poszukiwaniem pytań na stawiane hipotezy oraz definiowaniem określonych zagadnień.

1.3 Opis systemu

System odpowiadający na pytania zadawane Wikipedii, będący celem tej pracy, jest szczególnym systemem QA, dla którego źródłem informacji są artykuły serwisu Wikipedia. Artykuły będące podstawą systemu są napisane w języku naturalnym i zawierają ogromną liczbę informacji. Wiele z nich jest zbędnych, mało znaczących, a kluczowa informacja świadcząca o przydatności dokumentu jest często bardzo krótka (często jest to jedno zdanie).

Liczba wystąpień słów we wszystkich artykułach Wikipedii to ponad $6 \cdot 10^{12}$. Przetwarzanie tak dużych danych w czasie rzeczywistym jest bardzo trudne do zrealizowania na standardowym komputerze. Nawet przy założeniu optymistycznej złożoności $O(n)$, czas potrzebny na przetworzenie takiej liczby danych to co najmniej kilka godzin. Dlatego też kluczowym zadaniem jest odpowiednie wstępne przetworzenie danych i selektywne lub zagregowane ich przetwarzanie w celu znalezienia artykułów zawierających tylko istotne informacje dotyczące pytania.

Trudności związane z językiem naturalnym

Aby system był możliwie prosty i intuicyjny w korzystaniu, dane na których operuje wyszukiwarka muszą być zapisane w języku naturalnym dla człowieka. Językiem przyjętym w tej pracy jest język polski. Stwarza to pewne trudności podczas przetwarzania.

Język polski jest językiem słowiańskim. Jest on bardzo bogaty i trudny, co czyni system bardzo złożonym. Znaczna część obcokrajowców, po wielu latach nauki nadal ma problemy z poprawną odmianą słów, wymową, ortografią. Z punktu widzenia tej pracy, najistotniejszym faktem jest odmiana przez liczby, rodzaje oraz przypadki. Jest wiele reguł, którymi należy się posługiwać, aby poprawnie odmienić słowo. Istnieje duża liczba wyjątków dotyczących słów pochodzenia zagranicznego. Słowa

obecnie używane w tekstach polskich są różnie traktowane (często niepoprawnie), nieraz są odmieniane, a nieraz nie. To wszystko sprawia, że połączenie dwóch słów o tym samym znaczeniu jest niekiedy bardzo trudne. Przykładem jest np. „*Friedrich Nietzsche*” i jego odpowiednik w dopełniaczu „*Friedricha Nietzschego*”. Powiązanie ze sobą wielu różnych odmian jednego słowa jest warunkiem koniecznym do uzyskania dobrych wyników. Traktowanie dwóch odmian jednego słowa jako różnych, znacząco obniży skuteczność, ponieważ może pominąć wystąpienia słów być może istotnych.

Problemem jest również to, że prawie wszystkie słowa posiadają synonimy, które nieraz bardzo ciężko jest ze sobą powiązać. Słowa „*auto*” i „*samochód*” mają to samo znaczenie w każdym kontekście. Natomiast „*droga*” i „*jezdnia*” często oznaczają to samo, ale nie zawsze, bo jezdnia jest częścią drogi. Zwykła zamiana słów, często używana przez Wikipedystów, aby wzbogacić tekst i unikać powtórzeń, jest dla komputera stosunkowo ciężkim zadaniem.

Teksty pisane w języku naturalnym przez ludzi często zawierają błędy językowe i błędy literowe. Zdarza się, że słowa dwuczłonowe są pisane z łącznikiem lub bez. Z artykułami Wikipedii jest trochę inaczej, choć te błędy zdarzają się. Z racji otwartości Wikipedii, wszystkie artykuły są sprawdzane wielokrotnie przez moderatorów oraz samych użytkowników, którzy zgłaszają błędy.

1.4 Struktura pracy

Materiał zawarty w pracy podzielono na tematycznie powiązane rozdziały.

Rozdział 2 opisuje dane, zbiory tekstowe użyte do odpowiadania na pytania. Zawiera również sposób przetwarzania danych, szczegółowy opis pozyskiwania istotnych informacji z punktu widzenia problemu danych, jak i sposób ich przechowywania oraz opis oceny użytych metod.

Rozdziały 3, 4 oraz 5 są szczegółowym opisem systemu, użytych algorytmów jak i również napotkanych trudności. Wszystkie metody zostały zaprezentowane w kolejnych rozdziałach. Jest to stopniowe ulepszanie systemu oraz próba połączenia kilku metod w celu polepszenia wyników.

Ostatni rozdział zawiera podsumowanie zaprezentowanych metod, wnioski oraz wyniki.

Dodatek A zawiera kilka technicznych zagadnień dotyczących implementacji przedstawionych modeli.

Niniejsza praca przedstawia jedno z możliwych rozwiązań problemów niejednoznaczności słów, odmian słów, przetwarzania dużej liczby danych, wyszukiwania informacji i stopniowe ulepszanie systemu.

Rozdział 2

Dane, przetwarzanie danych, miary wyników

Do działania systemu odpowiadania na pytania potrzebna jest baza wiedzy oraz baza pytań wraz z odpowiedziami. Baza wiedzy to najczęściej duży zbiór tekstów, artykułów, stron internetowych, a nawet haseł encyklopedycznych. Baza pytań stanowiąca dane uczące zawiera listę pytań oraz odpowiedzi, które powinno móc się wydedukować na podstawie bazy wiedzy. Zdecydowanie cięższym zbiorem do zdobycia jest dobra baza pytań i odpowiedzi. W języku angielskim takie zbiory są dostępne w Internecie, jednak w języku polskim takie zbiory są trudno dostępne. Prawdopodobnie, wynika to z faktu, że język polski jest mało popularny wśród naukowców zajmujących się szeroko pojętym przetwarzaniem tekstu, ponadto ogólnie przyjętym językiem w świecie nauki jest język angielski. W celu zdobycia bazy wiedzy oraz pytań postanowiono skorzystać z zasobów Wikipedii.

2.1 Artykuły

Artykuły są najważniejszą częścią systemu. Wikipedia udostępnia zrzut bazy danych³ zawierający, m. in. pełną listę artykułów wraz z ich treścią. Jest to plik w formacie XML o regularnej i sformalizowanej budowie. Struktura opisująca przykładowy artykuł została pokazana na rysunku 2. Każdy artykuł posiada szereg informacji z nim związanych: tytuł, link do serwisu Wikipedia, komentarz, ostatnią datę zmian, nazwę użytkownika współautora, identyfikator. Na potrzeby systemu, skorzystano jedynie z tytułu artykułu i treści.

³Baza danych dostępna pod adresem <http://dumps.wikimedia.org/plwiki/>.

```

▼<html>
  ▼<xmp>
    ▼<page>
      <title>AWK</title>
      <ns>0</ns>
      <id>2</id>
      ▼<revision>
        <id>55942769</id>
        <parentid>55942492</parentid>
        <timestamp>2019-02-17T20:15:09Z</timestamp>
        ▼<contributor>
          <username>Daroooo</username>
          <id>434122</id>
        </contributor>
        <minor/>
        <comment>/* Samodzielne skrypty AWK */ drobne merytoryczne</comment>
        <model>wikitext</model>
        <format>text/x-wiki</format>
        <text xml:space="preserve">TREŚĆ ARTYKUŁU</text>
        <shal>h8umnoy607kw1w1lea0xz6me79oveax</shal>
      </revision>
    </page>
  </xmp>
</html>

```

Rysunek 2: Struktura opisująca przykładowy artykuł.

Niektóre artykuły posiadają dodatkowy znacznik #REDIRECT, przykładowy artykuł z takim polem znajduje się na rysunku 3. Artykuły takie nie posiadają żadnej treści, a jedynie wskazują na inny artykuł. Informacje te zostały uwzględnione podczas projektowania systemu. Wszystkie artykuły, które są tylko przekierowaniami, są utożsamiane jako jeden, z tym na który wskazują. Liczba takich artykułów to 276 127 co stanowi 17% wszystkich artykułów. Wybrane przekierowania artykułów znajdują się w tabeli 1.

```

▼<page>
  <title>Ac</title>
  <ns>0</ns>
  <id>314</id>
  <redirect title="Aktyn"/>
  ▼<revision>
    <id>1070722</id>
    <parentid>925139</parentid>
    <timestamp>2004-10-31T21:00:29Z</timestamp>
    ▼<contributor>
      <username>Kbsc</username>
      <id>7097</id>
    </contributor>
    <model>wikitext</model>
    <format>text/x-wiki</format>
    <text xml:space="preserve">#REDIRECT [[Aktyn]]</text>
    <shal>huhvd576n4zi0i0dowqqaaznlmz8xg2</shal>
  </revision>
</page>

```

Rysunek 3: Struktura opisująca przykładowy artykuł z przekierowaniem.

Tablica 1: Przykładowe przekierowania artykułów.

Artykuł bazowy	Artykuł do którego przekierowuje artykuł bazowy
cm	centymetr
bitwa Warszawska 1656	bitwa pod Warszawą (1656)
bit	bit (ujednoznacznienie)
cyklotrial	trial rowerowy
całka Newtona-Leibniza	całka Riemanna
fale myriametrowe	skrajnie niska częstotliwość
język galisyjski	język galicyjski
ksywa	pseudonim
protokoły internetowe	protokół internetowy
przekątna macierzy	macierz
rpg-2	granatnik rpg-2
Theodosius II	Teodozjusz II (cesarz bizantyński)

Treść artykułu jest bardzo ustrukturalizowana. Przykład treści artykułu znajduje się na rysunku 5. Znaki specjalne (np. „<”, „&”) są zapisane w kodzie HTML (odpowiednio „<”, „&”). Niektóre nazwy własne (wyróżnione) są zapisane w cudzysłowie np. „*Acre Municipal Stadium*”. Odnośniki do innych artykułów są postaci `[[tytuł artykułu do którego prowadzi odnośnik|nazwa odnośnika]]`, np. `[[język hebrajski|hebr.]]`. Jeśli tytuł artykułu i nazwa odnośnika są takie same to występuje tylko jedna strona, np. `[[Hapoel Akka]]`. Tabele mają specjalny format, przykładowa tabela znajduje się na rysunku 4. Nazwy poszczególnych działów są wyróżnione znakiem „=”, np. `== Informacje ogólne ==`. Na końcu każdego artykułu znajdują się informacje o kategorii artykułu. Mają one format `[[Kategoria:nazwa kategorii]]`, np. `[[Kategoria:Specjalności lekarskie]]`.

```

{{Stadion infobox
|nazwa                = Olimpico
|pełna_nazwa          = Stadio Olimpico
|przydomek            =
|zdjęcie               = Stadio Olimpico 2008.JPG
|podpis_zdjęcia       =
|lata_budowy           = 1928-1937
|data_otwarcia         = [[1937]]
|koszt_budowy          =
|poprzednia_nazwa     =
|kraj                  = ITA
|miejsowość           =
|adres                 = [[Foro Italico]]&lt;br /&gt;00194 Roma
|współrzędne          =
|data_zamknięcia      =
|właściciel           =
|operator              =
|klub                  = [[S.S. Lazio]]&lt;br /&gt;[[AS Roma]]
|inauguracja          =
|pojemność_stadionu   = 70 634
|rekordowa_frekwencja =
|wymiary_boiska       = 105&nbsp;×&nbsp;68&nbsp;m
|oświetlenie          =
|żużel                 =
|wyposażenie          = [[bieżnia]]
|nawierzchnia         = [[Trawnik|trawa]]
|imdb                 =
|commons               = Category:Stadio Olimpico (Rome)
|kod_mapy              = Rzym
|stopniN              = 41
|minutN               = 56
|sekundN              = 01.99
|stopniE              = 12
|minutE               = 27
|sekundE              = 17.23
}}

```

Rysunek 4: Struktura przykładowej tabeli.

```

{{Zwierzę infobox |nazwa łacińska = Heteronemertea |TSN = 57438 |zoolog = |okres
istnienia = |grafika = Lineus longissimus Grevelingen.jpg |opis grafiki = ''[[Lineus
longissimus]]'' |typ = [[wstężnice]] |gromada = [[Anopla]] |rząd = Heteronemertea
|synonimy = * Heteronemertini |wikispecies = Heteronemertea |commons =
Category:Heteronemertea }} ''Heteronemertea'' - [[rząd (biologia)|rząd]]
[[wstężnice|wstężnic]] obejmujący [[gatunek (biologia)|gatunki]] o trójwarstwowej
strukturze wra powłokowo-mięśniowego. Pomiędzy dwiema warstwami mięśni podłużnych leży
warstwa mięśni okrężnych. Pnie nerwowe znajdują się pomiędzy zewnętrzną warstwą mięśni
podłużnych a środkową warstwą mięśni okrężnych. W układzie krwionośnym oprócz bocznych
naczyń krwionośnych zawsze występują naczynia grzbietowe i - zwykle - łączące,
poprzeczne. W większości są to [[Gatunek (biologia)|gatunki]] morskie, kilka występuje
w wodach słodkich. Przechodzą [[rozwój złożony]], w którym występują różne typy
[[larwa|larw]]: [[pilidium]], [[larwa Desora]] i [[larwa Schmidta]]. Rząd
Heteronemertea obejmuje rodziny{{r|Tholleson}}: * [[Lineidae]], * [[Riseriellidae]]. ==
Przypisy == {{Przypisy|<ref name="Tholleson">{{cytuj
pismo|nazwisko=Tholleson|imię=M.|nazwisko2=Norenborg|imię2=J. |tytuł=Ribbon worm
relationships: a phylogeny of the phylum Nemertea|czasopismo=Proceedings of the Royal
Society of London. Series B, Biological Sciences|wolumin=270|strony=407-415|język=en
|data=2002|doi=10.1098/rspb.2002.2254}}</ref> }} == Bibliografia == * {{Cytuj książkę |
nazwisko= Jura | imię=Czesław | tytuł= Bezkręgowce : podstawy morfologii funkcjonalnej,
systematyki i filogenezy | data=2007 | wydawca=[[Wydawnictwo Naukowe PWN]]|
miejsce=Warszawa | isbn=978-83-01-14595-8}} * {{Cytuj książkę | tytuł = Zoologia :
bezkęgowce. T. 1 | inni = Red. nauk. Czesław Błaszak | data = 2009 | wydawca =
[[Wydawnictwo Naukowe PWN]] | miejsce = Warszawa | isbn = 978-83-01-16108-8 | strony =
252-253}} [[Kategoria:Wstężnice]]

```

Rysunek 5: Treść artykułu „*Heteronemertea*”.

Dane zawarte w tabelach najczęściej zawierają dane liczbowe bądź informacje wyrwane z kontekstu. W związku z tym, że system jest tworzony tak, aby korzystać z danych tekstowych (spójnych), zostały one pominięte. Podobnie postąpiono z odnośnikami. Zachowano ich nazwy, ale usunięto nazwy artykułów, do których prowadziły. Mogłyby one wprowadzać fałszywe dane (tytuł artykułu słabo powiązany z nazwą odnośnika) i zakłócenia (podwójne zliczanie). Odnośniki do zewnętrznych stron zawarte w „*[*” również zostały usunięte.

2.2 Słowa

Artykuły zawierają ogromną liczbę informacji, a ich przetwarzanie w czasie rzeczywistym wymagałoby dużych zasobów. Aby łatwiej i szybciej odpowiadać na pytania, wykonano preprocessing danych, a w bazie danych zapisano wyłącznie potrzebne informacje.

Słowo to podstawowa jednostka na jakiej system wykonuje operacje. Słowo jest niepustym ciągiem złożonym z liter alfabetu łacińskiego, znaków diakrytycznych języków, które takie posiadają (np. ö, ó), cyfr, przecinka i kropki (specjalne przypadki) oraz znaków języków nieeuropejskich (hebrajski, chiński). Każdy ciąg wyżej wymienionych znaków w tekście jest traktowany jako jedno słowo. Wyjątkiem jest przecinek i kropka. Jeśli znaki te występują pomiędzy cyframi, to ciąg jest słowem (tworzy wtedy liczbę rzeczywistą). W każdym innym wypadku rozdzielają ciąg znaków na dwa osobne słowa.

Specjalnego traktowania wymagają słowa zawierające litery i cyfry np. „*ABC123*”. W żadnym języku takie słowo nie istnieje, więc pozornie może się wydawać, że jest to niepotrzebne, a nawet błędne. W praktyce jednak często używa się kombinacji liter i cyfr do różnego rodzaju oznaczeń modeli, wersji, nazw. Przykładem niech będzie silnik widlasty o dwunastu cylindrach, który w skrócie oznacza się „*V12*”. Potraktowanie tego jako dwa osobne słowa, bardzo straci na wartości, ponieważ „*V12*” jest słowem rzadkim, natomiast „*V*” i „*12*”, już nie.

Podobnie można również potraktować dwa ciągi połączone łącznikiem, np. „*ABC-123*”, „*ABC-ABC*”. Okazało się jednak, że bardzo rzadko używa się myślnika w przypadku oznaczeń modeli, nazw, wersji tak jak wyżej. Często natomiast używa

się myślnika do pisania wyrazów dwuczłonowych np. „*biało-czerwona*”, „*polsko-włoski*”. Niestety, często zdarza się, że pisownia tych wyrazów jest błędna, bez myślnika. Dzięki interpretowaniu myślnika jako odstępu, zarówno pisownia „*biało-czerwona*”, jak i „*biało czerwona*” będzie zinterpretowana w ten sam sposób, jako dwa osobne słowa.

2.2.1 Formy bazowe słów

Forma bazowa to słowo sprowadzone do podstawowej, nieodmienionej formy. Dla słów „*szkoła*”, „*szkoly*”, „*szkole*” forma bazowa jest taka sama - „*szkoła*”. Połączenie słów o tej samej formie bazowej jest bardzo istotne, gdyż w innym wypadku są one traktowane jako dwa różne słowa i wiele wystąpień słów zostanie pominiętych. Znacząco wpływa to na niższą pozycję w wynikach wyszukiwania.

Formy bazowe uzyskane na podstawie artykułów Wikipedii

Dzięki ustrukturalizowanej budowie artykułów, pozyskano wiele form bazowych dla bardzo rzadkich słów i nazw własnych.

Niektóre odnośniki zawarte w treści artykułu, oprócz informacji o tytule artykułu, zawierają również formy bazowe, np. `[[Hapoel Akka|Hapoelu Akka]]`. Wszystkie odnośniki po sprawdzeniu, czy liczba słów po obu stronach jest taka sama oraz czy prefiks 3 literowy odpowiednich słów pokrywa się, zostały zapisane w bazie danych. Przykładowe zaakceptowane formy bazowe znajdują się w tabeli 3, natomiast odrzucone formy bazowe znajdują się w tabeli 4.

Podobnie postąpiono z odnośnikami, dla których tytuł artykułu i nazwa odnośnika są takie same. W niektórych przypadkach po odnośniku występuje końcówka, tzn. `[[tytuł artykułu]]końcówka`. Te informacje również zostały zapisane. Przykładowe pozyskane formy bazowe znajdują się w tabeli 2.

W ten sposób zostało pozyskanych 399 698 nowych odmian słów.

Tablica 2: Przykładowe skrócone odnośniki artykułów z których uzyskano formy bazowe.

<code>[[stadion]]em</code>
<code>[[podział]]y</code>
<code>[[robot]]ami</code>

Tablica 3: Przykładowe odnośniki artykułów z których uzyskano formy bazowe.

<code>[[Noteć Noteci]]</code>
<code>[[Grupa kapitałowa grup kapitałowych]]</code>
<code>[[Średniowiecze średniowiecza]]</code>
<code>[[Magistrala węglowa magistrali węglowej]]</code>

Tablica 4: Przykładowe odnośniki artykułów odrzucone w procesie pozyskiwania form bazowych. Odnośniki zawierają słowa bliskoznaczne, ale nie są to formy bazowe.

[[Reakcja benzoilowania benzoilowania]]
[[Makrum Pomorskie Zakłady Budowy Maszyn „Zremb-Makrum”]]
[[Polska Rzeczpospolita Ludowa PRL]]
[[Młyny Kentzera Słoneczny Młyn]]

Formy bazowe uzyskane na podstawie zbioru morfologik

Morfologik⁴ to słownik morfo-syntaktyczny dla języka polskiego. Zawiera on potężną bazę słów, ich odmian oraz informacje na temat formy gramatycznej (liczba, rodzaj, część mowy, przypadek). Na potrzeby tej pracy skorzystano jedynie z części danych mówiących o formie bazowej słowa. Uzyskano w ten sposób 4 800 433 odmian słów. Przykładowe dane znajdują się na rysunku 6.

Łącznie uzyskano w ten sposób 5 101 900 odmian słów, z czego zbiór morfologik stanowi prawie 93%. Pomimo małego udziału form bazowych uzyskanych z artykułów, słowa te często są bardzo rzadkie i w wielu przypadkach nie wyglądają na proste, popularne (np. „*leukergii*” i „*leukergia*”, „*beyerem*” i „*beyer*”).

bajkarzami	bajkarz	subst:pl:inst:m1
dewaloryzował	dewaloryzować	verb:praet:sg:m1.m2.m3:ter:imperf:refl.nonrefl
niebezpieportowościami	niebezpieportowość	subst:pl:inst:f
Dobrowójach	Dobrowój	subst:pl:loc:m1
promilach	promil	subst:pl:loc:m3

Rysunek 6: Przykładowe dane ze zbioru morfologik.

2.2.2 Unifikacja słów

W języku polskim dwa zdania o tym samym znaczeniu, mogą być zapisane w innej, ale podobnej formie. Przykład takich zdań znajduje się w tabeli 5. Aby system mógł poznać podobieństwo słów oraz zdań zapisanych w innej formie, należy sprowadzić je do formy bazowej. W tym celu wszystkie słowa występujące w pytaniach i artykułach są utożsamiane z formą bazową opisaną w poprzednim rozdziale, jeśli taka dla danego słowa istnieje. Niech $allForms(w)$ oznacza zbiór wszystkich odmian słowa w . Przykładowo: $allForms(„dom”) = \{ „dom”, „domu”, „domach”, „domie”, „domach”, „domy”, \dots \}$. Forma słowa nie ma wpływu na końcowy wynik algorytmu, a jedynie fakt wystąpienia słowa. Jeśli dla danego słowa nie znaleziono formy bazowej, słowo jest używane w formie oryginalnej. Dotyczy to zwłaszcza nazw własnych, nazwisk, słów, dla których znalezienie odmian jest zadaniem ciężkim. Przykładowe słowa, dla których nie znaleziono form bazowych to: „*wilde’ami*”, „*Szyndrowska*”,

⁴Zbiór morfologik znajduje się na stronie <http://morfologik.blogspot.com/>.

„krajoznawczościom”. Funkcja $allForms(w)$ jest obliczana wg wzorów poniżej.

$allForms(w) = \{x \in W : baseForms(x) \cap baseForms(w) \neq \emptyset\}$,
 $baseForms(w)$ - zbiór form bazowych słowa w ,
 W - zbiór wszystkich słów.

Tablica 5: Przykładowe zdania mające to samo znaczenie, ale inaczej zredagowane.

zdanie	forma bazowa
„Najwyższy budynek w Polsce to Pałac Kultury i Nauki w Warszawie.”	„Najwyższy”, „budynek”, „w”, „Polska”, „to”, „Pałac”, „Kultura”, „i”, „Nauka”, „w”, „Warszawa”
„Najwyższym polskim budynkiem jest Pałac Kultury i Nauki znajdujący się w Warszawie.”	„Najwyższy” „polski” „budynek” „jest” „Pałac” „Kultura” „i” „Nauka” „znajdować” „się” „w” „Warszawa”

2.2.3 Stop words

Wiele słów występujących w artykułach i pytaniach nie niesie ze sobą żadnych istotnych informacji. Są to często występujące słowa niezwiązane z kontekstem. Lista słów stop words została pobrana z serwisu Wikipedia⁵. Zawiera on 375 takich słów. Lista niektórych słów stop words znajduje się w tabeli 6. Jeśli słowo występujące w artykule bądź pytaniu należy do zbioru stop words, jest ono pomijane i nie ma wpływu na wyniki.

2.2.4 Pozycje słów

Pozycja wystąpienia słowa w artykule to indeks danego słowa na liście wystąpień wszystkich słów artykułu. Koniec zdania jest traktowany jako separator. Dzięki temu ostatnie słowo zdania i pierwsze słowo następnego zdania nie mają kolejnej pozycji i nie są traktowane jako słowa sąsiadujące.

Dla każdego artykułu przechowywane są wszystkie słowa wraz z ich pozycją występowania (zarówno w samym tekście, jak i w tytule) oraz liczba wszystkich słów występujących w artykule.

Początkowo, w bazie przechowywane były formy bazowe słów (o ile takowe były znane), jednak później okazało się, że powodowało to dużo problemów i błędów. Słowa, które posiadają więcej niż jedną formę bazową były wielokrotnie liczone i przez to uzyskiwały zawyżoną ocenę. Słowa są przechowywane w formie oryginalnej, a forma bazowa jest uzyskiwana dopiero w późniejszym etapie.

2.3 Pytania

Wikipedia udostępnia wiele ciekawostek⁶ na temat swoich artykułów. Przykładowe ciekawostki znajdują się na obrazku 7. Strona zawierająca ciekawostki jest stale

⁵Zbiór słów stop words znajduje się na stronie <http://pl.wikipedia.org/wiki/Wikipedia:Stopwords>.

⁶Lista haseł „Czy wiesz, że” dostępna jest na stronie https://pl.wikipedia.org/wiki/Wikipedia:Wikireadery/Czy_wiesz,_%C5%BCe.

rozwijana przed Wikipedystów. Uzyskano w ten sposób 10 668 pytań, które odnosiły się do 12 094 artykułów. Są to pojedyncze zdania, zaczerpnięte bezpośrednio z artykułów i częściowo przeredagowane bądź napisane ręcznie przez Wikipedystów. Zdania te są zwane nieformalnie **czywieszkami**⁷, ponieważ większość zdań zaczyna się od słów „Czy wiesz, że”. Czwieszki zawierają odnośnik (nieraz kilka) do artykułu w serwisie Wikipedia, w którym można przeczytać o ciekawostce zawartej w zdaniu. W ten sposób uzyskano dane testowe. Zdanie jest traktowane jako pytanie, natomiast artykuł zawarty w odnośniku jako odpowiedź. Czwieszki dotyczą bardzo szerokiej gamy artykułów, nie skupiają się na jednej tematyce. Dotyczą osób, państw, zwierząt, dat, rzeczy, budynków, imprez sportowych i wielu innych. Przykładowe pytania i odpowiedzi znajdują się w tabelach 7, 8 oraz 9.



Rysunek 7: Przykładowe czwieszki znajdujące się na stronie polskojęzycznej Wikipedii w dniu 02.02.2020. Odnośniki do artykułów zaznaczone są kolorem niebieskim. Artykuły wyróżnione pogrubioną czcionką zawierają szczegółową odpowiedź na pytanie, natomiast zwykłą czcionką są oznaczone artykuły powiązane z pytaniem. W poniższej pracy skorzystano jedynie z artykułów zawierających szczegółową odpowiedź na pytanie, ponieważ pozostałe artykuły często były powiązane jedynie tematycznie z ciekawostką.

Po przeanalizowaniu wyników wyróżniono 3 grupy pytań w zależności od wyniku. Pytania łatwe to takie, które nie sprawiały trudności większości modeli. Pytania o średniej trudności to takie, dla których istnieje kilka modeli, które zwracały dobry wynik. Ostatnia grupa - pytania trudne to takie, dla których większości modeli, a w niektórych przypadkach wszystkie, nie były w stanie zwrócić dobrych rezultatów. Analiza grup pytań pokazała, że trudność pytania jest powiązana między innymi z liczbą słów kluczowych, nazw własnych zawartych w pytaniu. Im pytanie zawiera

⁷W kolejnych rozdziałach pracy, słowo czwieszki jest rozumiane jako pytania.

więcej takich słów, tym lepszy wynik osiągnął system.

2.3.1 Czywieszki łatwe

Pytania łatwe to takie, które w treści zawierają mało popularne słowa kluczowe. Przykład pytania to „*czy wiesz, że pierwszym superkomputerem był CDC 6600?*”. Łatwo zauważyć, że słowa „*superkomputer*”, „*CDC*”, „*6600*” są słowami, których występowanie w dużej mierze ogranicza się do artykułów ściśle związanych z tą tematyką. Pytania takie raczej nie sprawiały trudności, ponieważ sama liczba artykułów zawierających te słowa jest mocno ograniczona.

2.3.2 Czywieszki średnie

Pytania średnie są to te pytania, które w treści zawierają jedno lub kilka mało popularnych słów kluczowych, ale zawierają również słowa bardzo popularne, które mogą utrudniać wskazanie poprawnej odpowiedzi. Przykład pytania to „*czy wiesz, że ciężarówka PZInż 703 została wyposażona w hydrauliczny układ hamulcowy z pneumatycznym wspomaganiem?*” Oprócz słów kluczowych „*PZInż*”, „*703*”, wszystkie inne występują w wielu artykułach dotyczących motoryzacji i techniki.

2.3.3 Czywieszki trudne

Pytania trudne to te, które zawierają wyłącznie bardzo popularne słowa używane w wielu artykułach. Przykład pytania to „*czy wiesz, że mierzący ponad 20 mln km² obszar Oceanu Południowego został formalnie zatwierdzony w 2000 roku?*”. Słowa użyte w tym pytaniu nie są silnie powiązane z pytaniem i należą do wielu artykułów znajdujących się w Wikipedii. Podobna sytuacja ma miejsce dla ostatniego pytania z tabeli 9 „*Kto władał trzydziestoma trzema językami?*”. Zawiera ono 4 słowa istotne z punktu widzenia poszukiwania odpowiedzi, są to: „*władał*”, „*trzydziestoma*”, „*trzema*”, „*językami*”. Jednak każde z nich jest wykorzystywane przy pisaniu wielu innych artykułów. Jest to pewna trudność, aby odpowiedzieć na pytanie zawierające wyłącznie popularne słowa. Analiza wyników pokazała, że takie pytania sprawiały największą trudność w niniejszej pracy.

Tablica 7: Przykładowe łatwe pytania i artykuły.

Pytanie	Artykuł
„ <i>Kogo i kiedy rozstrzelano w chojnickiej 'Dolinie śmierci'?</i> ”	„ <i>Egzekucje w chojnickiej 'Dolinie śmierci'</i> ”
„ <i>Czy wiesz, że 1 grudnia to światowy dzień walki z aids a czerwona wstążka oznacza ludzi solidaryzujących się z chorymi?</i> ”	„ <i>Światowy dzień aids</i> ”
„ <i>Czy wiesz skąd Zamek Runkel miał wziąć swoją nazwę?</i> ”	„ <i>Zamek Runkel</i> ”
„ <i>Czy wiesz, że pierwszym superkomputerem był CDC 6600</i> ”	„ <i>CDC 6600</i> ”

Tablica 8: Przykładowe średnie pytania i artykuły.

Pytanie	Artykuł
„Czy wiesz, że w czasie powstania warszawskiego działała <i>Harcerska Poczta Polowa</i> ?”	„ <i>Harcerska Poczta Polowa</i> ”
„Czy wiesz gdzie swoje spotkania obecnie rozgrywają piłkarze <i>Hapoelu Akka</i> i dlaczego musieli opuścić swój dawny obiekt?”	„ <i>Stadion miejski w Akce</i> ”, „ <i>Stadion im. Napoleona</i> ”
„Czy wiesz, że ciężarówka <i>PZInż 703</i> została wyposażona w hydrauliczny układ hamulcowy z pneumatycznym wspomaganiem?”	„ <i>PZInż 703</i> ”

Tablica 9: Przykładowe ciężkie pytania i artykuły.

Pytanie	Artykuł
„ <i>Ogon jakiego gatunku zwierzęcia stanowi 60% długości jego ciała?</i> ”	„ <i>Allactaga euphratica</i> ”
„ <i>Z jaką piosenką wystąpiła Candy Girl na festiwalu w Opolu w 2009 roku?</i> ”	„ <i>Barbara Hetmańska</i> ”
„ <i>Czy wiesz skąd pochodzi woda pitna dla mieszkańców Tallinna?</i> ”	„ <i>Ülemiste</i> ”
„ <i>Kto władał trzydziestoma trzema językami?</i> ”	„ <i>Zhao Yuanren</i> ”

2.4 Ranking artykułów dla pytania

Dla każdej pary <model, pytanie> jest tworzony ranking artykułów. Pozycja artykułu w rankingu jest wyznaczana na podstawie jego wagi zwracanej przez dany model dla danego pytania, wg kolejności malejącej. Im wyższa waga artykułu, tym wyższa pozycja w rankingu. Stworzony ranking artykułów jest wykorzystywany do oceny wyników modeli przez poszczególne miary opisane w kolejnym rozdziale.

2.5 Miary wyników modeli

Na potrzeby niniejszej pracy wykorzystano kilka miar ocen wyników: p_1 , p_{10} , p_{100} oraz MRR (ang. mean reciprocal rank).

2.5.1 Miary p_1 , p_{10} , p_{100}

Wartość p_1 oznacza procent pytań, dla których poprawny artykuł znalazł się na pierwszym miejscu. Jest to miara najbardziej surowa, ale jednocześnie najbardziej praktyczna dla potencjalnych użytkowników, ponieważ dostają oni poprawny wynik na pierwszym miejscu. Miara p_{10} oznacza procent pytań, dla których poprawny artykuł znalazł się w pierwszej dziesiątce, co w praktyce przekłada się na znalezienie artykułu na pierwszej stronie przez użytkownika w wyszukiwarce. Miara p_{100} dla normalnego użytkownika jest mało miarodajna, ponieważ przejrzanie 100 artykułów jest ponad możliwości zwykłego człowieka. Może natomiast stanowić dobrą bazę

danych dla bardziej złożonych systemów, które mogą wykorzystać pierwsze 100 artykułów do stworzenia jeszcze lepszego rankingu, z pominięciem już odrzuconych artykułów. Miary skuteczności p_i są liczone wg wzorów poniżej.

$$p_i = \frac{\text{liczba odpowiedzi na miejscu co najwyżej } i}{\text{liczba wszystkich odpowiedzi}}.$$

2.5.2 Miara MRR

Miara MRR to średnia odwrócona pozycja artykułu, która obliczana jest jako odwrotność średniej harmonicznnej pozycji poprawnych artykułów w rankingu. Miara faworyzuje wysokie wyniki w rankingu, ale jednocześnie daje pogląd na wszystkie wyniki za pomocą jednej wartości. Dalsze pozycje w rankingu mają odpowiednio mniejsze znaczenie na końcowy wynik. Miara MRR jest dobrym przybliżeniem średniej wartości miar typu p co widać w tabelce 10. Przykład obliczania MRR znajduje się na rysunku 8. Miara skuteczności jest liczona wg wzorów poniżej.

$$\text{MRR} = \frac{1}{|Q|} \cdot \sum_{q \in Q} \frac{1}{\text{rank}(q)},$$

rank(q) - pozycja poprawnego artykułu w rankingu pytania q ,
 Q - zbiór wszystkich pytań.

ID	Correct Outcome	Predicted Ranks in Order	Rank	Reciprocal Rank
1	Hip-Hop	Electronic, Hip-Hop, Instrumental	2	1/2
2	Electronic	Electronic, Experimental, International	1	1/1
3	Experimental	Hip-Hop, Electronic, Experimental	3	1/3
4	Hip-Hop	Instrumental, Electronic, Hip-Hop	2	1/2
5	Instrumental	Hip-Hop, Instrumental, Experimental	2	1/2
			MRR	0.57

Rysunek 8: Przykład obliczania wartości MRR⁸.

Tablica 10: Przykładowe wartości miar dla różnych danych. Dla pokazania różnic między miarami dodatkowo pokazano miarę p_2 .

Dane	Miara				
	p_1	p_2	p_{10}	p_{100}	MRR
1, 2	0.50	1.00	1.00	1.00	0.75
2	0.00	1.00	1.00	1.00	0.50
2, 11	0.00	0.50	0.50	1.00	0.30
11	0.00	0.00	0.00	1.00	0.09

⁸Rysunek pobrany ze strony <https://www.oreilly.com/library/view/c-machine-learning/9781788996402/21d965bc-3e1c-4af0-8508-a218a7f87687.xhtml>.

Tablica 6: Wybrane słowa z listy stop words.

a	dla	jakkolwiek	mało	niemu	również	u
aby	dłaczego	jako	mam	nigdy	sam	w
ach	dlatego	jakoś	mi	nim	sama	wam
acz	do	je	mimo	nimi	są	wami
aczkolwiek	dobrze	jeden	między	niż	się	was
aj	dokąd	jedna	mną	no	skąd	wasz
albo	dość	jedno	mnie	o	sobie	wasza
ale	dużo	jednak	mogą	obok	sobą	wasze
ależ	dwa	jednakże	moi	od	sposób	we
ani	dwaj	jego	moim	około	swoje	według
aż	dwie	jej	moja	on	ta	wiele
bardziej	dwoje	jemu	moje	ona	tak	wielu
bardzo	dziś	jest	może	one	taka	więc
bo	dzisiaj	jestem	możliwe	oni	taki	więcej
bowiem	gdy	jeszcze	można	ono	takie	wszyscy
by	gdyby	jeśli	mój	oraz	także	wszystkich
byli	gdyż	jeżeli	mu	oto	tam	wszystkie
bynajmniej	gdzie	już	musi	owszem	te	wszystkim
być	gdziekolwiek	ją	my	pan	tego	wszystko
był	gdzieś	każdy	na	pana	tej	wtedy
była	go	kiedy	nad	pani	ten	wy
było	i	kilka	nam	po	teraz	właśnie
były	ich	kimś	nami	pod	też	z
będzie	ile	kto	nas	podczas	to	za
będą	im	ktokolwiek	nasi	pomimo	tobą	zapewne
cali	inna	ktoś	nasz	ponad	tobie	zawsze
cała	inne	która	nasza	ponieważ	toteż	ze
cały	inny	które	nasze	powinien	trzeba	zł
ci	innych	którego	naszego	powinna	tu	znowu
cię	iż	której	naszych	powinni	tutaj	znów
ciebie	ja	który	natomiast	powinno	twoi	został
co	ją	których	natychmiast	poza	twoim	żaden
cokolwiek	jak	którym	nawet	prawie	twoja	żadna
coś	jakaś	którzy	nią	przecież	twoje	żadne
czasami	jakby	ku	nic	przed	twym	żadnych
czasem	jaki	lat	nich	przede	twój	że
czemu	jakichś	lecz	nie	przedtem	ty	żeby
czy	jakie	lub	niech	przez	tych	
czyli	jakiś	ma	niego	przy	tylko	
daleko	jakiż	mają	niej	roku	tym	

Rozdział 3

Modele statystyczne

Wszystkie modele przedstawione w poniższym rozdziale bazują jedynie na liczbie i miejscach wystąpień słów w artykule. Każdy kolejny model jest pewną modyfikacją i próbą ulepszenia poprzednich modeli.

3.1 Użyte metody

Wszystkie rozpatrywane w poniższym rozdziale modele bazują na opisanych poniżej metodach.

3.1.1 Miara TF-IDF

Podstawową opisaną poniżej modeli jest miara TF-IDF. Miara **TF-IDF** to iloczyn dwóch czynników: TF oraz IDF. Wartość **TF** (ang. term frequency) liczona dla pary <słowo, artykuł> oznacza częstotliwość słowa⁹ dla danego artykułu. Wartość ta jest liczona jako iloraz wystąpień danego słowa w danym artykule przez liczbę wystąpień danego słowa we wszystkich artykułach. Wartość **IDF** (ang. inverse document frequency) liczona dla każdego słowa osobno, oznacza odwróconą częstotliwość artykułów. Wartość ta jest liczona jako logarytm naturalny z ilorazu liczby wszystkich artykułów przez liczbę artykułów zawierających dane słowo. Przykładowe wartości TF-IDF znajdują się na rysunku 9. Wartość TF-IDF dla słowa w i artykułu a oznaczona jako $word\text{-}tf\text{-}idf(w, a)$ jest liczona wg wzorów poniżej.

$$\begin{aligned} word\text{-}tf\text{-}idf(w, a) &= word\text{-}tf(w, a) \cdot word\text{-}idf(w), \\ word\text{-}tf(w, a) &= \frac{word\text{-}count(w, a)}{\sum_{v \in W} word\text{-}count(v, a)} \text{ (term frequency),} \\ word\text{-}idf(w) &= \log \frac{|A|}{|\{a \in A : w \in a\}|} \text{ (inverse document frequency),} \\ word\text{-}count(w, a) &\text{ - liczba wystąpień słowa } w \text{ w artykule } a, \\ W &\text{ - zbiór wszystkich słów,} \\ A &\text{ - zbiór wszystkich artykułów.} \end{aligned}$$

Wynika z tego, że na wagę słowa mają wpływ: liczba wystąpień słów w artykule, liczba wszystkich słów w artykule, liczba artykułów zawierających dane słowo. Wy-

⁹Częstotliwość słowa w artykule jest rozumiana jako jego istotność. Im wyższa częstotliwość, tym słowo jest istotniejsze.

soką wagę może mieć zarówno słowo, które wiele razy występuje w artykule i jest popularne, jak i takie które występuje rzadko, co jest zgodne z oczekiwaniami.

Sentence 1 : The car is driven on the road.

Sentence 2: The truck is driven on the highway.

Word	TF		IDF	TF*IDF	
	A	B		A	B
The	1/7	1/7	$\log(2/2) = 0$	0	0
Car	1/7	0	$\log(2/1) = 0.3$	0.043	0
Truck	0	1/7	$\log(2/1) = 0.3$	0	0.043
Is	1/7	1/7	$\log(2/2) = 0$	0	0
Driven	1/7	1/7	$\log(2/2) = 0$	0	0
On	1/7	1/7	$\log(2/2) = 0$	0	0
The	1/7	1/7	$\log(2/2) = 0$	0	0
Road	1/7	0	$\log(2/1) = 0.3$	0.043	0
Highway	0	1/7	$\log(2/1) = 0.3$	0	0.043

Rysunek 9: Wartości TF-IDF dla 2 przykładowych zdań¹⁰.

3.1.2 Waga artykułu dla pytania

Waga artykułu dla pytania to suma wartości TF-IDF wszystkich słów występujących w pytaniu (liczona dla każdego artykułu osobno) pomnożona przez liczbę wspólnych słów pytania i artykułu do potęgi 3 (wartość dobrana eksperymentalnie, tak aby promować artykuły silniej powiązane z pytaniem). Waga artykułu a dla pytania q oznaczona jako $word-weight(a, q)$ jest liczona wg wzorów poniżej.

$$word-weight(a, q) = |words(a) \cap words(q)|^3 \cdot \sum_{w \in words(q)} word-tf-idf(w, a), \quad (3.1)$$

$words(a)$ - zbiór słów występujących w artykule a ,
 $words(q)$ - zbiór słów występujących w pytaniu q .

3.2 Model oparty o liczbę wystąpień słów

Wstępna analiza kilku pytań pokazała, że wiele słów kluczowych zawartych w pytaniach występuje w artykułach w różnych częściach. Często zdarza się, że odpowiedź na pytanie zawarta jest w dwóch lub więcej zdaniach, które są od siebie mocno oddalone. Dało to podstawy do stworzenia pierwszego modelu.

¹⁰Rysunek pobrany ze strony <https://medium.com/@imamun/creating-a-tf-idf-in-python-e43f05e4d424>.

Pierwsze podejście do problemu opiera się na założeniu, że właściwy artykuł posiada więcej słów zawartych w pytaniu, niezależnie od tego na jakiej pozycji i w jakim otoczeniu. Oznacza to, że na wagę i tym samym pozycję artykułu ma wpływ wyłącznie liczba wystąpień słów. Model mocno faworyzuje artykuły, które są powiązane z pytaniem większą liczbą słów.

3.2.1 Waga słowa, waga artykułu dla pytania

Model oparty o wystąpienia słów to podstawowy model bazujący bezpośrednio na mierze TF-IDF, bez żadnych modyfikacji – z tego powodu liczenie poszczególnych wag jest najprostsze. Waga słowa jest liczona analogicznie jak waga TF-IDF dla słowa w rozdziale 3.1.1. Waga artykułu jest obliczana tak samo, jak w rozdziale 3.1.2.

3.2.2 Wyniki modelu opartego o liczbę wystąpień słów

Testy zostały przeprowadzone osobno dla treści i tytułów artykułów. Wyniki znajdują się w tabeli 11.

Tablica 11: Wyniki modelu TF-IDF.

dane	p_1	p_{10}	p_{100}	MRR
tytuł	0.3729	0.5797	0.6825	0.4477
treść	0.4601	0.7500	0.8818	0.5628

Wyniki dla tytułów artykułów

Dla około 37% model zwraca poprawny artykuł na pierwszym miejscu. Może się wydawać, że nie jest to wynik wysoki, jednak należy wziąć pod uwagę, że jest to bardzo surowe kryterium oceny. Biorąc pod uwagę pierwsze 10 odpowiedzi modelu, skuteczność wzrasta już do ponad 57%. Jest to miarodajne kryterium, gdyż przejrzanie 10 artykułów jest wykonalne przez człowieka w krótkim czasie.

Wyniki dla treści artykułów

Liczba poprawnych artykułów na pierwszym miejscu wzrosła o niecałe 10 punktów procentowych, a pozostałe kryteria nawet więcej. Prawdopodobnie wynika to z faktu, że tytuł artykułu jest bardzo mocno związany z jego treścią, ale zawiera bardzo mało słów. Poprawna odpowiedź będzie wysoko tylko wtedy, jeśli pytanie zadawane jest na temat szczegółów, które są wymienione w treści. Np. na pytanie „z którymi miastami związany był artysta Simon De Vlieger?” i odpowiedź „Simon de Vlieger”.

Oba modele w zależności od użytych danych artykułu, zwracają lepsze wyniki dla innych pytań. Dobrym pomysłem wydaje się być połączenie tych dwóch modeli w jeden w celu uzyskania lepszych rezultatów. Idea ta została rozwinięta na większą liczbę modeli w późniejszym rozdziale.

3.3 Model oparty o liczbę wystąpień bigramów

Bigram jest to sekwencja dwóch słów. Ich kolejność jest istotna, tzn. bigram („duży”, „dom”) jest różny od („dom”, „duży”). Unifikacja bigramów została wykonana analogicznie, jak w rozdziale 2.2.2, jednak na poziomie pojedynczych słów. Formy bazowe bigramu to iloczyn kartezjański form bazowych poszczególnych słów. Dla bigramów: („duży”, „dom”), („dużego”, „domu”), („dużym”, „domem”) forma bazowa to („duży”, „dom”). Formy bazowe dla bigramu (*word1*, *word2*) oznaczone jako *bigramBaseForm*(*word1*, *word2*) są liczone wg wzoru poniżej.

$$\text{bigramBaseForm}(\text{word1}, \text{word2}) = \text{baseForm}(\text{word1}) \times \text{baseForm}(\text{word2}),$$

gdzie \times oznacza iloczyn kartezjański.

Wadą poprzedniego modelu było to, że słowa zawarte w pytaniu często występowały w artykule w zupełnie innym otoczeniu, kontekście. Dla pytania związanego, np. z Powstaniem Warszawskim wiele artykułów zawierających słowa kluczowe („Warszawa” lub „Powstanie”) otrzymywało wysoką pozycję. Bardzo często było to działanie niepożądane, ponieważ występowanie jednego z tych 2 słów w wielu przypadkach nie jest związane z tematem. W celu usunięcia powyższej wady, postanowiono zmienić podstawową jednostkę, jaką było słowo, na bigram, czyli dwa sąsiadujące ze sobą słowa.

3.3.1 Waga bigramu, waga artykułu dla pytania

Miara **TF-IDF** bigramu jest liczona analogicznie jak waga TF-IDF dla słowa w rozdziale 3.1.1. Wartości TF, IDF są liczone tak samo jednak z uwzględnieniem bigramów zamiast słów.

Waga artykułu jest obliczana podobnie jak w rozdziale 3.1.2. Waga artykułu dla pytania to suma wartości TF-IDF wszystkich bigramów występujących w pytaniu (liczona dla każdego artykułu osobno) razy liczba wspólnych bigramów pytania i artykułu do potęgi 3 (ta sama wartość jaka została wybrana w poprzednim modelu).

3.3.2 Wyniki modelu opartego o liczbę wystąpień bigramów

Jednym z głównych powodów powstania modelu bigramowego były słabe wyniki poprzedniego modelu dla trudnych pytań. Spodziewane wyniki to poprawna skuteczności dla pytań trudnych. Wyniki znajdują się w tabeli 12.

Tablica 12: Wyniki modelu bigramowego. Dla porównania na dole wyniki modelu opartego o słowa.

model	dane	p_1	p_{10}	p_{100}	MRR
bigramowy	tytuł	0.3018	0.4804	0.5108	0.3681
	treść	0.3919	0.6816	0.8196	0.4923
słowny	tytuł	0.3729	0.5797	0.6825	0.4477
	treść	0.4601	0.7500	0.8818	0.5628

Wyniki są zauważalnie gorsze dla każdego z kryterium, w porównaniu do poprzedniego modelu. Podczas dokładniejszej analizy pytań z różnych pozycji poprzedniego

modelu okazało się, że model bigramowy dla części pytań trudnych (są to pytania których poprawne artykuły znalazły się na dalekich pozycjach) uzyskał lepszy wynik niż model TF-IDF na bazie słów. Poprawa wyniku jednak odbyła się kosztem pytań łatwych, których było na tyle dużo, że całkowita skuteczność modelu okazała się niższa niż klasycznego modelu TF-IDF.

3.4 Modele kontekstowe

Klasyczne liczenie wag bierze pod uwagę cały artykuł. Ma to jednak dużo niedoskonałości. Jeśli artykuł posiada słowa zawarte w pytaniu, ale jednocześnie jest on bardzo długi to jego pozycja w rankingu nie będzie wysoka. Postanowiono więc stworzyć model, który będzie liczył wagi tylko na podstawie fragmentów artykułów, a następnie wybierał te o maksymalnej wartości. Model w dużej mierze bazuje na modelach TF-IDF oraz bigramowym, zmieniony jest jedynie zakres artykułu.

3.4.1 Model kontekstowy oparty o wystąpienia słów

Parametrem tego modelu jest liczba n , oznaczająca długość przetwarzanego fragmentu artykułu. Fragment o długości n jest to n sąsiadujących ze sobą słów. Dla każdego fragmentu artykułu o długości n jest liczona niezależna waga i wybierana jest maksymalna wartość. Waga fragmentu dla danego artykułu jest to miara TF-IDF, liczona analogicznie jak w rozdziale 3.1.1, z uwzględnieniem tylko części artykułu, który należy do danego fragmentu. Waga artykułu a dla pytania q , dla fragmentów o długości n opartych o pojedyncze słowa, oznaczona jako *chunks-word-weight*(a, q, n) jest liczona wg wzorów poniżej.

$$\begin{aligned} \text{chunks-word-weight}(a, q, n) &= \max_{0 \leq p \leq \text{len}(a) - n} \text{chunks-word-weight}(a, q, p, p+n-1), \\ \text{chunks-word-weight}(a, q, p_1, p_2) &= \left(\sum_{w \in \text{words}(q)} \text{word-tf-idf}(w, a, p_1, p_2) \right) \cdot \\ &|\text{words}(a, p_1, p_2) \cap \text{words}(q)|^3, \\ \text{chunks-word-tf-idf}(w, a, p_1, p_2) &= \text{chunks-word-tf}(w, a, p_1, p_2) \cdot \text{chunks-word-idf}(w), \\ \text{chunks-word-tf}(w, a, p_1, p_2) &= \frac{\text{chunks-word-count}(w, a, p_1, p_2)}{\sum_{v \in W} \text{word-count}(v, a, p_1, p_2)}, \\ \text{chunks-word-idf}(w) &= \log \frac{|A|}{|\{a \in A : w \in a\}|}, \\ \text{chunks-word-count}(w, a, p_1, p_2) &- \text{liczba wystąpień słowa } w \text{ w artykule } a \text{ na pozy-} \\ &\text{cjach } p_1 \dots p_2, \\ \text{words}(a, p_1, p_2) &- \text{zbiór słów występujących w artykule } a \text{ na pozycjach } p_1 \dots p_2, \\ \text{words}(q) &- \text{zbiór słów występujących w pytaniu } q, \\ \text{len}(a) &- \text{liczba słów artykułu } a, \\ W &- \text{zbiór wszystkich słów}, \\ A &- \text{zbiór wszystkich artykułów}. \end{aligned}$$

3.4.2 Model kontekstowy oparty o wystąpienia bigramów

Waga artykułu jest liczona podobnie jak dla modelu kontekstowego opartego o wystąpienia pojedynczych słów, jednak z uwzględnieniem par słów. Waga artykułu a dla pytania q , dla fragmentów o długości n opartych o bigramy, oznaczona jako

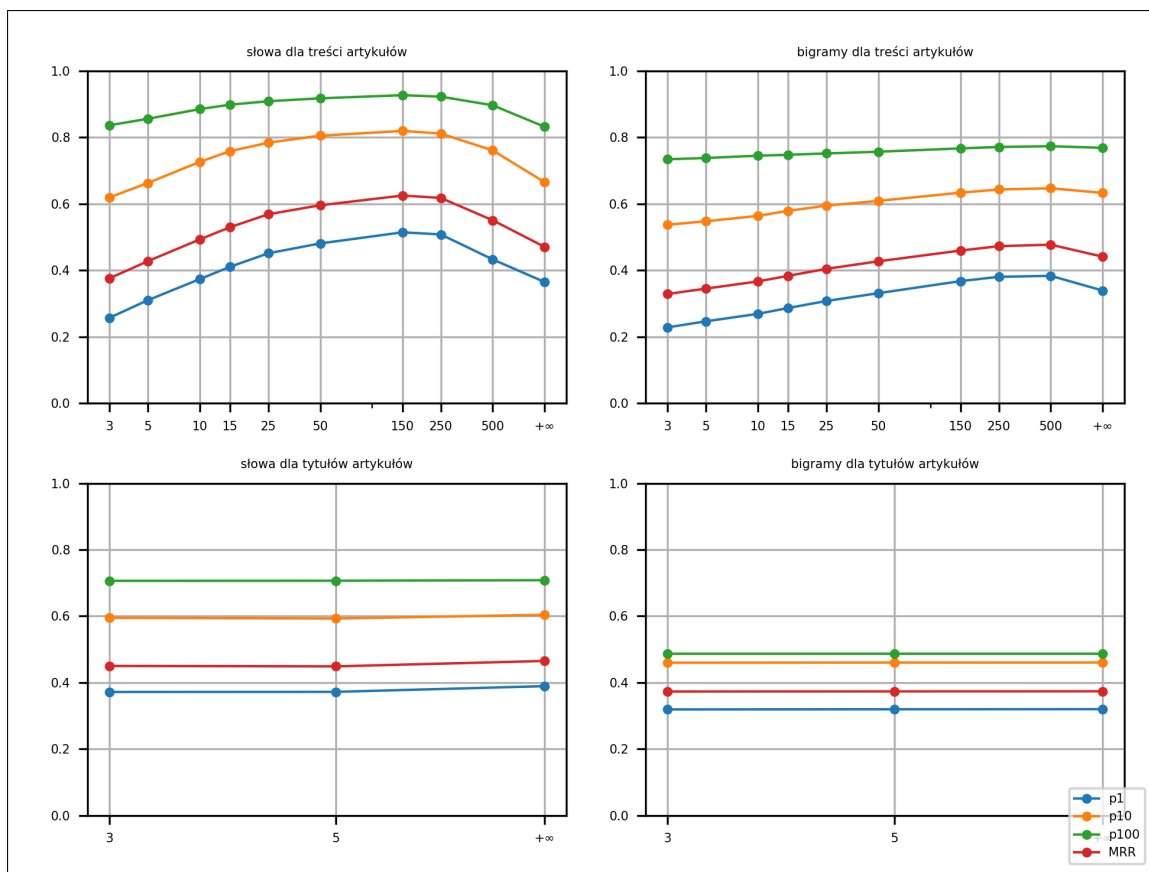
$chunks\text{-}bigram\text{-}weight(a, q, n)$ jest liczona wg wzorów poniżej.

$$\begin{aligned}
&chunks\text{-}bigram\text{-}weight(a, q, n) = \max_{0 \leq p \leq \text{len}(a) - n} chunks\text{-}bigram\text{-}weight(a, q, p, p + n - 1), \\
&chunks\text{-}bigram\text{-}weight(a, q, p_1, p_2) = \left(\sum_{b \in \text{bigrams}(q)} chunks\text{-}bigram\text{-}tf\text{-}idf(b, a, p_1, p_2) \right) \cdot \\
&|\text{bigrams}(a, p_1, p_2) \cap \text{bigrams}(q)|^3, \\
&chunks\text{-}bigram\text{-}tf\text{-}idf(b, a, p_1, p_2) = chunks\text{-}bigram\text{-}tf(b, a, p_1, p_2) \cdot chunks\text{-}bigram\text{-}idf(w), \\
&chunks\text{-}bigram\text{-}tf(b, a, p_1, p_2) = \frac{\text{bigram-count}(b, a, p_1, p_2)}{\sum_{d \in B} \text{bigram-count}(d, a, p_1, p_2)}, \\
&chunks\text{-}bigram\text{-}idf(b) = \log \frac{|A|}{|\{a \in A : b \in a\}|}, \\
&chunks\text{-}bigram\text{-}count(b, a, p_1, p_2) - \text{liczba wystąpień bigramu } b \text{ w artykule } a \text{ na pozycjach } p_1 \dots p_2, \\
&\text{bigrams}(a, p_1, p_2) - \text{zbiór bigramów występujących w artykule } a \text{ na pozycjach } p_1 \dots p_2, \\
&\text{bigrams}(q) - \text{zbiór bigramów występujących w pytaniu } q, \\
&\text{len}(a) - \text{liczba słów artykułu } a, \\
&B - \text{zbiór wszystkich bigramów}, \\
&A - \text{zbiór wszystkich artykułów}.
\end{aligned}$$

3.4.3 Wyniki modeli kontekstowych

Wyniki znajdują się na rysunku 10 oraz w tabelach 13, 14¹¹. Wraz ze wzrostem parametru n , który odpowiada za wielkość przetwarzanego pojedynczo fragmentu wyniki są coraz lepsze. Dopiero przy wielkości $n = 500$ wyniki się pogarszają. Najlepszy wynik osiągnął model oparty o liczbę wystąpień słów w treści artykułów dla $n = 250$. Wynik jest znacznie lepszy od pierwotnego modelu TF-IDF.

¹¹Wartości dla $n = +\infty$ to wyniki modeli z poprzednich rozdziałów, dla których fragmentem jest cały artykuł.



Rysunek 10: Wyniki modeli kontekstowych. Na osi X znajdują się szerokości przetwarzanego fragmentu. Na osi Y znajdują się wyniki skuteczności poszczególnych miar. Opis poszczególnych miar znajduje się w prawym dolnym rogu. Wyniki modeli opartych na treściach artykułów są różne w zależności od parametru n . Najlepsze wyniki są w okolicach $n = 250$. Inaczej wygląda sytuacja w przypadku tytułów artykułów, gdzie parametr n nie ma znaczącego wpływu na wyniki – stąd prawie linia prosta na wykresie wyników modeli opartych o tytuły artykułów.

Tablica 13: Wyniki modeli kontekstowych na podstawie treści artykułów. Dla porównania na dole znajdują się wyniki najlepszego dotychczas modelu.

model	dane	n	p_1	p_{10}	p_{100}	MRR
kontekstowy	słowa	3	0.2575	0.6197	0.8368	0.3763
	słowa	5	0.3103	0.6630	0.8562	0.4276
	słowa	10	0.3737	0.7265	0.8853	0.4932
	słowa	15	0.4109	0.7587	0.8986	0.5300
	słowa	25	0.4515	0.7844	0.9090	0.5690
	słowa	50	0.4811	0.8054	0.9178	0.5960
	słowa	150	0.5146	0.8197	0.9272	0.6252
	słowa	250	0.5083	0.8118	0.9229	0.6181
	słowa	500	0.4331	0.7612	0.8969	0.5509
	słowa	$+\infty$	0.3651	0.6655	0.8322	0.4708
	bigramy	3	0.2285	0.5374	0.7344	0.3291
	bigramy	5	0.2468	0.5479	0.7382	0.3452
	bigramy	10	0.2692	0.5640	0.7452	0.3668
	bigramy	15	0.2868	0.5793	0.7479	0.3836
	bigramy	25	0.3080	0.5948	0.7519	0.4047
	bigramy	50	0.3314	0.6092	0.7572	0.4273
	bigramy	150	0.3678	0.6339	0.7671	0.4598
	bigramy	250	0.3807	0.6435	0.7715	0.4729
	bigramy	500	0.3837	0.6471	0.7737	0.4774
	bigramy	$+\infty$	0.3393	0.6333	0.7688	0.4413
słowny TF-IDF	tytuł		0.3729	0.5797	0.6825	0.4477
	treść		0.4601	0.7500	0.8818	0.5628

Tablica 14: Wyniki modeli kontekstowych na podstawie tytułów artykułów. Dla porównania na dole znajdują się wyniki najlepszego dotychczas modelu.

model	dane	n	p_1	p_{10}	p_{100}	MRR
kontekstowy	słowa	3	0.3718	0.5950	0.7062	0.4501
	słowa	5	0.3722	0.5929	0.7065	0.4489
	słowa	$+\infty$	0.3893	0.6039	0.7082	0.4651
	bigramy	3	0.3192	0.4599	0.4869	0.3731
	bigramy	5	0.3197	0.4603	0.4869	0.3736
	bigramy	$+\infty$	0.3201	0.4605	0.4869	0.3739
słowny TF-IDF	tytuł		0.3729	0.5797	0.6825	0.4477
	treść		0.4601	0.7500	0.8818	0.5628

3.5 Model odległości wektorów rzadkich

Wektor rzadki to wektor, którego przeważająca większość wartości jest równa 0. Przykład wektora rzadkiego to $[0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 3, 0, 0]$. Postawiono skorzystać z wektorów do reprezentacji tekstu¹². Każda wartość takiego wektora odpowiada za inną cechę tekstu (w tym przypadku za istotność wybranego

¹²W tym przypadku tekstem będzie pytanie lub artykuł.

słowa w tekście). Dla każdego pytania większość artykułów nie zawiera żadnych słów użytych w pytaniu. Większość wektorów opisujących artykuły jest wypełniona w całości lub przeważającej mierze zerami. Tylko artykuły silnie powiązane z pytaniem mają wektor niezawierający w ogóle lub bardzo mało zer. Rozmiar wektorów w poniższym modelu jest zmienny w zależności od przetwarzanego pytania i jest równy liczbie unikalnych słów zawartych w pytaniu (z pominięciem słów należących do zbioru stop words). Przykłady wektorów rzadkich znajdują się w tabeli 17.

Wszystkie rozważane do tej pory modele używały prostego sumowania wag poszczególnych słów lub bigramów w celu otrzymania wagi artykułu. Zgodnie ze wzorem 3.1, na końcową ocenę artykułu wpływ ma suma wag słów oraz liczba wspólnych słów pomiędzy pytaniem, a artykułem. Jest to bardzo prosta metoda, która do tej pory dała satysfakcjonujące wyniki. Postanowiono jednak przetestować inne rozwiązania. Podstawową różnicą między nowym modelem, a poprzednimi jest to, że w nowym podejściu oceną istotności artykułu dla pytania jest odległość lub podobieństwo wektorowe pomiędzy wektorem artykułu, a wektorem pytania.

3.5.1 Wektor pytania

Wektor pytania to wektor wag poszczególnych słów pytania. Waga pojedynczego słowa w pytaniu to miara TF-IDF liczona w oparciu o zbiór wszystkich pytań. W odróżnieniu od poprzednich modeli opartych o tę miarę, model ten nie traci informacji o poszczególnych słowach. W tym celu zapamiętuje wszystkie wagi w wektorze, który później jest porównywany z wektorem artykułu. Wektor przykładowego pytania znajduje się w tabeli 15. Wektor wag pytania q oznaczony jako *question-vector*(q) jest liczony wg wzorów poniżej.

$\text{question-vector}(q) = [\text{question-tf-idf}(q, q_w^1), \dots, \text{question-tf-idf}(q, q_w^n)],$ $\text{question-tf-idf}(q, w) = \frac{\text{question-count}(q, w)}{\sum_{w' \in W} \text{question-count}(q, w')} \cdot \log \frac{ Q }{ \{q' \in Q : w \in q'\} },$ $\text{question-count}(q, w) - \text{liczba wystąpień słowa } w \text{ w pytaniu } q,$ $q_w^i - i\text{-te słowo pytania } q,$ $W - \text{zbiór wszystkich słów},$ $Q - \text{zbiór wszystkich pytań}.$
--

Tablica 15: Wektor wag słów pytania „Kto od 11 października 2012 roku jest premierem Jordanii?” z pominięciem stop words.

słowo	11	października	2012	premierem	Jordanii
waga	0.4447	0.4494	0.5197	0.7310	1.0743

3.5.2 Wektor artykułu

Wektor artykułu dla pytania jest liczony podobnie jak w rozdziale 3.5.1, jednak na podstawie słów artykułów. Dodatkowo, model ten jest rozszerzeniem modelu kontekstowego, tzn. wagi liczone są dla poszczególnych fragmentów o długości n , podobnie jak w rozdziale 3.4. Wektor zawiera tylko wagi liczone dla słów występujących w pytaniu. Wektory przykładowych artykułów dla pytania znajdują się w tabeli 16.

Tablica 16: Wektor wag słów 5 najwyżej ocenionych artykułów (dla $n = 250$) dla pytania „*Kto od 11 października 2012 roku jest premierem Jordanii?*” z pominięciem stop words.

artykuł	waga słowa				
	11	października	2012	premierem	Jordanii
„ <i>Alicja Bachleda-Curus</i> ”	0.6670	0.4494	0.2598	0.7310	1.0743
„ <i>Abd Allah An-Nusur</i> ” ¹³	0.2021	0.4085	0.4725	0.9968	1.4649
„ <i>August Burns Red</i> ”	0.3705	0.7490	0.4331	0.6091	0.8952
„ <i>Jordan Pickford</i> ”	0.3176	0.3210	0.3712	1.0443	1.5347
„ <i>Ranbir Kapoor</i> ”	0.0000	0.5618	0.6497	0.9137	1.3428

Tablica 17: Przykłady wektorów rzadkich na podstawie pytania „*Czy wiesz, że podróże w I Rzeczypospolitej pomimo bardzo złego stanu dróg i mostów były częste, a Polaków nazywano nawet 'największymi podróżnikami Europy'?*”

dane	wektor
pytanie z opisu tabeli	[0.53, 0.31, 0.35, 0.17, 0.26, 0.38, 0.18, 0.37, 0.29, 0.19, 0.67, 0.26]
artykuł „ <i>Stefan Szolc-Rogoziński</i> ”	[0.91, 0.00, 0.00, 0.00, 0.45, 1.30, 0.00, 1.27, 0.00, 0.00, 1.16, 0.00]
artykuł „ <i>transport i podróże w czasach i Rzeczypospolitej</i> ” ¹⁴	[0.18, 0.64, 0.49, 0.24, 0.27, 0.52, 0.38, 0.38, 0.00, 0.20, 0.23, 0.00]
artykuł „ <i>Aleksander Branicki</i> ”	[2.12, 0.00, 0.00, 0.00, 1.05, 0.00, 0.00, 0.00, 0.00, 0.00, 2.70, 0.00]
artykuł „ <i>Katedra św. Krzysztofa w Roermond</i> ”	[0.00, 0.00, 0.00, 0.00, 0.00, 2.28, 0.00, 0.00, 0.00, 0.00, 4.05, 0.00]
artykuł „ <i>politologia</i> ”	[0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 2.29, 0.00, 0.00]
artykuł „ <i>arytmetyka</i> ”	[0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 1.76, 1.15, 0.00, 0.00]

3.5.3 Miary podobieństwa

Oceną artykułu jest odległość między wektorem reprezentującym artykuł, a wektorem reprezentującym pytanie. Im odległość mniejsza, tym artykuł jest mocniej skorelowany z pytaniem. Postanowiono użyć 3 najpopularniejszych i najbardziej obiecujących funkcji odległości:

- odległość absolutna - $\text{cityblock-distance}(v, w) = \sum_{i=1}^n |v_i - w_i|$;
- odległość euklidesowa - $\text{euclidean-distance}(v, w) = \sqrt{\sum_{i=1}^n (v_i - w_i)^2}$;
- podobieństwo cosinusowe - $\text{cosine-similarity}(v, w) = \frac{v \cdot w}{\|v\| \cdot \|w\|} = \frac{\sum_{i=1}^n v_i \cdot w_i}{\sqrt{\sum_{i=1}^n v_i^2} \cdot \sqrt{\sum_{i=1}^n w_i^2}}$.

¹³Artykuł będący poprawną odpowiedzią na pytanie.

¹⁴Artykuł będący poprawną odpowiedzią na pytanie.

Warto zauważyć różnicę pomiędzy odległością, a podobieństwem. Odległość między wektorami jest tym mniejsza, im wektory są sobie bliższe. W przypadku podobieństwa, miara jest odwrotna. Podobieństwo jest tym większe, im wektory są bliższe.

3.5.4 Wyniki modeli wektorów rzadkich

Dla każdej z metryk z rozdziału 3.5.3 stworzono odpowiadający jej model. Wyniki znajdują się na rysunku 11 oraz w tabelach 19, 20, 21, 22¹⁵.

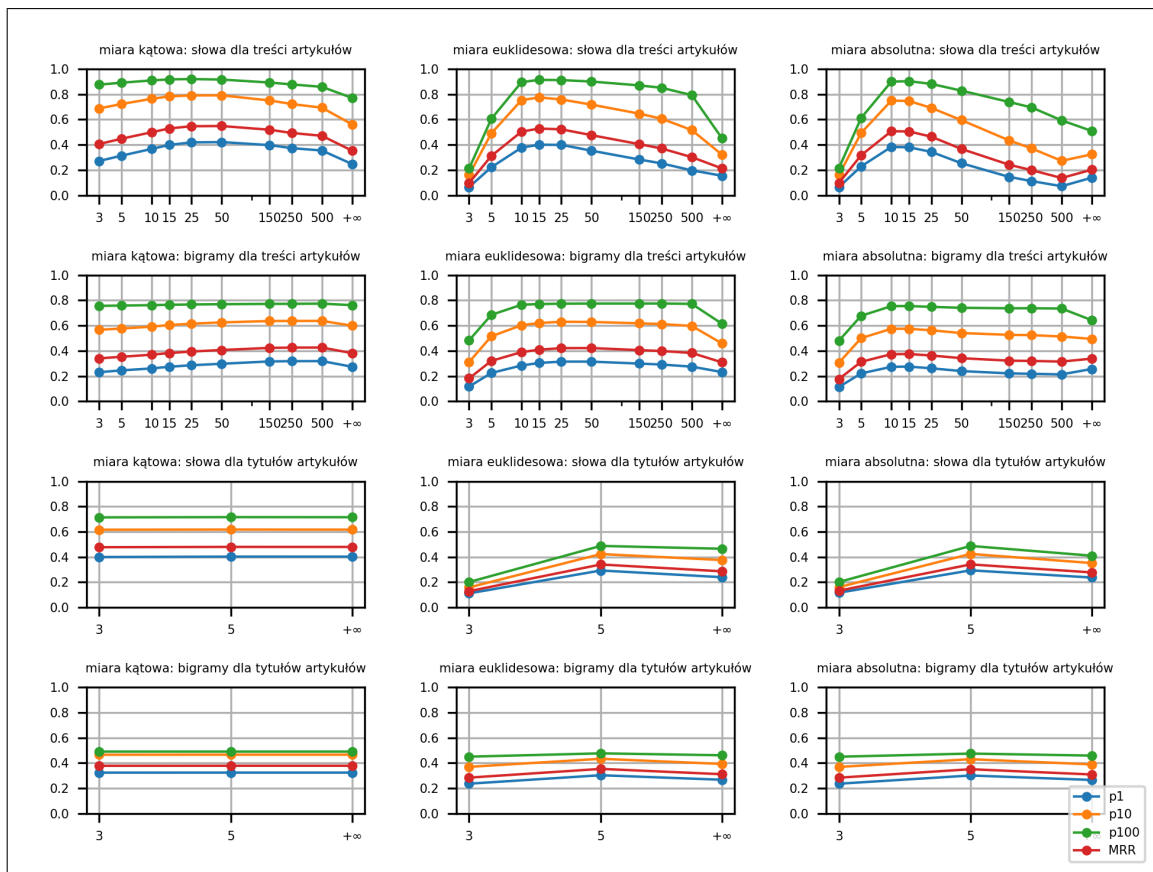
Wyniki są gorsze, niż te dla modeli bazujących na prostym sumowaniu wag poszczególnych słów. Jest to nieoczekiwany rezultat, ponieważ metoda wektorowa ze względu na swoją budowę i fakt rozpatrywania wag poszczególnych słów osobno, powinna lepiej sobie radzić ze słowami nieniosącymi istotnych informacji. Przykładowo, dla pytania „*Która biblioteka jest w posiadaniu najstarszej w Polsce gęsi?*” wagi poszczególnych słów nienależących do stop words znajdują się w tabeli 18.

Słowo „*gęsi*” ma znacznie większą wagę niż reszta słów. Modele korzystające z prostego sumowania nie biorą pod uwagę zróżnicowania wag, a jedynie ich sumę oraz liczbę. Ignorują tym samym informację o tym, czy słowo jest kluczowe (ma dużą wagę) czy nie (ma małą wagę). Model wektorowy jest tej wady pozbawiony, ponieważ metryki odległości z rozdziału 3.5.3 bazują na różnicy poszczególnych elementów wektora, czyli różnicy wag słów między pytaniem, a artykułem. Pomimo tej teoretycznej zalety wyniki tego nie potwierdzają. Prawdopodobną przyczyną takiego stanu rzeczy może być mała liczba pytań tego typu. Problemem mogą być również źle dobrane metryki. Pomimo wcale nie najlepszych wyników, model wektorowy ma bardzo duże możliwości. W odróżnieniu od poprzednich modeli daje sposobność porównywania artykułów i pytań na podstawie poszczególnych wag, a nie pojedynczej wartości będącej wypadkową wszystkich wag. Należałoby jednak starannie dobrać metrykę odległości pod kątem zbiorów tekstowych lub zaproponować inny sposób porównywania wektorów.

Tablica 18: Wagi słów dla pytania „*Która biblioteka jest w posiadaniu najstarszej w Polsce gęsi?*” z pominięciem stop words.

słowo	biblioteka	posiadaniu	najstarszej	Polsce	gęsi
waga	0.83	0.51	0.48	0.23	1.33

¹⁵Wartości dla $n = +\infty$ to wyniki modeli dla których fragmentem jest cały artykuł.



Rysunek 11: Wyniki modeli odległości wektorów. Pierwsza kolumna wykresów zawiera modele oparte o miarę kątową. Druga kolumna wykresów zawiera modele oparte o miarę euklidesową. Trzecia kolumna zawiera modele oparte o miarę absolutną. Na osi X znajdują się szerokości przetwarzanego fragmentu. Na osi Y znajdują się wyniki skuteczności poszczególnych miar. Opis poszczególnych miar znajduje się w prawym dolnym rogu. Duży wpływ parametru n na wyniki widać na wykresach modelu opartego o miarę euklidesową. W przypadku miary kątowej wyniki są prawie takie same dla różnych parametrów n . Ciekawostką są wyniki modelu opartego o miarę absolutną w przypadku której wykres wyników przypomina wielomian trzeciego stopnia.

Tablica 19: Wyniki modeli wektorowych na podstawie tytułów artykułów opartych o słowa. Dla porównania na dole znajdują się wyniki najlepszego dotychczas modelu.

model	miara	n	p_1	p_{10}	p_{100}	MRR
wektorowy	absolutna	3	0.1164	0.1609	0.2016	0.1329
	absolutna	5	0.2939	0.4241	0.4872	0.3408
	absolutna	$+\infty$	0.2369	0.3517	0.4098	0.2774
	euklidesowa	3	0.1114	0.1586	0.2013	0.1290
	euklidesowa	5	0.2927	0.4230	0.4874	0.3402
	euklidesowa	$+\infty$	0.2395	0.3753	0.4648	0.2862
	cosinusowa	3	0.3995	0.6157	0.7136	0.4772
	cosinusowa	5	0.4025	0.6169	0.7154	0.4795
	cosinusowa	$+\infty$	0.4024	0.6163	0.7148	0.4794
słowny kontekstowy TF-IDF		150	0.5146	0.8197	0.9272	0.6252

Tablica 20: Wyniki modeli wektorowych na podstawie tytułów artykułów opartych o bigramy. Dla porównania na dole znajdują się wyniki najlepszego dotychczas modelu.

model	miara	n	p_1	p_{10}	p_{100}	MRR
wektorowy	absolutna	3	0.2362	0.3691	0.4501	0.2841
	absolutna	5	0.3014	0.4304	0.4749	0.3510
	absolutna	$+\infty$	0.2661	0.3893	0.4592	0.3094
	euklidesowa	3	0.2365	0.3698	0.4506	0.2845
	euklidesowa	5	0.3035	0.4337	0.4767	0.3536
	euklidesowa	$+\infty$	0.2676	0.3928	0.4616	0.3115
	cosinusowa	3	0.3240	0.4658	0.4924	0.3783
	cosinusowa	5	0.3242	0.4663	0.4924	0.3786
	cosinusowa	$+\infty$	0.3241	0.4661	0.4924	0.3786
słowny kontekstowy TF-IDF		150	0.5146	0.8197	0.9272	0.6252

Tablica 21: Wyniki modeli wektorowych na podstawie treści artykułów opartych o słowa. Warto zauważyć, że dla miary absolutnej oraz cosinusowej najlepsze wyniki dla poszczególnych miar są dla różnych parametrów n . Może to wynikać z faktu, że miara p_{100} wynik na pierwszym oraz setnym miejscu rankingu zlicza tak samo, natomiast w przypadku miary MRR, coraz dalsze miejsce w rankingu ma coraz mniejszy wpływ na wynik. Dla porównania na dole znajdują się wyniki najlepszego dotychczas modelu.

model	miara	n	p_1	p_{10}	p_{100}	MRR
wektorowy	absolutna	3	0.0651	0.1607	0.2147	0.0975
	absolutna	5	0.2284	0.4931	0.6122	0.3167
	absolutna	10	0.3828	0.7497	0.9000	0.5071
	absolutna	15	0.3797	0.7466	0.9035	0.5052
	absolutna	25	0.3436	0.6921	0.8821	0.4641
	absolutna	50	0.2536	0.5952	0.8286	0.3667
	absolutna	150	0.1450	0.4341	0.7385	0.2416
	absolutna	250	0.1124	0.3713	0.6960	0.1990
	absolutna	500	0.0714	0.2725	0.5933	0.1366
	absolutna	$+\infty$	0.1391	0.3238	0.5089	0.2023
	euklidesowa	3	0.0644	0.1604	0.2118	0.0968
	euklidesowa	5	0.2215	0.4902	0.6049	0.3110
	euklidesowa	10	0.3770	0.7533	0.8959	0.5038
	euklidesowa	15	0.4005	0.7748	0.9140	0.5281
	euklidesowa	25	0.3988	0.7586	0.9123	0.5222
	euklidesowa	50	0.3528	0.7179	0.9005	0.4758
	euklidesowa	150	0.2828	0.6464	0.8701	0.4045
	euklidesowa	250	0.2522	0.6062	0.8514	0.3704
	euklidesowa	500	0.1970	0.5160	0.7929	0.3018
	euklidesowa	$+\infty$	0.1545	0.3219	0.4516	0.2125
	cosinusowa	3	0.2713	0.6879	0.8761	0.4059
	cosinusowa	5	0.3135	0.7226	0.8914	0.4475
	cosinusowa	10	0.3668	0.7640	0.9095	0.4991
	cosinusowa	15	0.3988	0.7830	0.9180	0.5291
	cosinusowa	25	0.4186	0.7908	0.9203	0.5467
	cosinusowa	50	0.4210	0.7912	0.9164	0.5483
	cosinusowa	150	0.3971	0.7509	0.8928	0.5188
	cosinusowa	250	0.3730	0.7226	0.8778	0.4933
	cosinusowa	500	0.3543	0.6935	0.8588	0.4707
	cosinusowa	$+\infty$	0.2469	0.5612	0.7709	0.3528
słowny kontekstowy TF-IDF		150	0.5146	0.8197	0.9272	0.6252

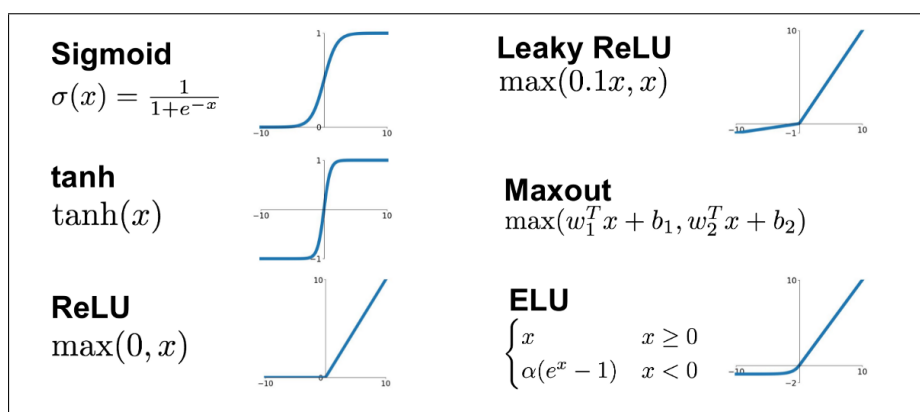
Tablica 22: Wyniki modeli wektorowych na podstawie treści artykułów opartych o bigramy. Dla porównania na dole znajdują się wyniki najlepszego dotychczas modelu.

model	miara	n	p_1	p_{10}	p_{100}	MRR
wektorowy	absolutna	3	0.1134	0.3052	0.4802	0.1770
	absolutna	5	0.2213	0.5008	0.6778	0.3131
	absolutna	10	0.2735	0.5747	0.7542	0.3736
	absolutna	15	0.2750	0.5750	0.7551	0.3746
	absolutna	25	0.2616	0.5626	0.7485	0.3633
	absolutna	50	0.2397	0.5410	0.7409	0.3417
	absolutna	150	0.2213	0.5266	0.7373	0.3221
	absolutna	250	0.2180	0.5246	0.7373	0.3201
	absolutna	500	0.2134	0.5136	0.7352	0.3149
	absolutna	$+\infty$	0.2563	0.4944	0.6429	0.3382
	euklidesowa	3	0.1169	0.3111	0.4831	0.1818
	euklidesowa	5	0.2247	0.5149	0.6850	0.3202
	euklidesowa	10	0.2839	0.6030	0.7653	0.3898
	euklidesowa	15	0.3040	0.6206	0.7707	0.4091
	euklidesowa	25	0.3149	0.6304	0.7735	0.4214
	euklidesowa	50	0.3145	0.6287	0.7748	0.4221
	euklidesowa	150	0.2991	0.6181	0.7747	0.4056
	euklidesowa	250	0.2915	0.6120	0.7750	0.3985
	euklidesowa	500	0.2748	0.5975	0.7721	0.3834
	euklidesowa	$+\infty$	0.2319	0.4587	0.6147	0.3088
	cosinusowa	3	0.2310	0.5673	0.7566	0.3405
	cosinusowa	5	0.2450	0.5766	0.7586	0.3530
	cosinusowa	10	0.2608	0.5915	0.7619	0.3694
	cosinusowa	15	0.2739	0.6044	0.7648	0.3827
	cosinusowa	25	0.2860	0.6150	0.7673	0.3946
	cosinusowa	50	0.2975	0.6254	0.7695	0.4065
	cosinusowa	150	0.3162	0.6361	0.7720	0.4234
	cosinusowa	250	0.3183	0.6367	0.7724	0.4257
	cosinusowa	500	0.3187	0.6373	0.7739	0.4260
	cosinusowa	$+\infty$	0.2740	0.5992	0.7619	0.3818
słowny kontekstowy TF-IDF		150	0.5146	0.8197	0.9272	0.6252

Rozdział 4

Modele neuronowe

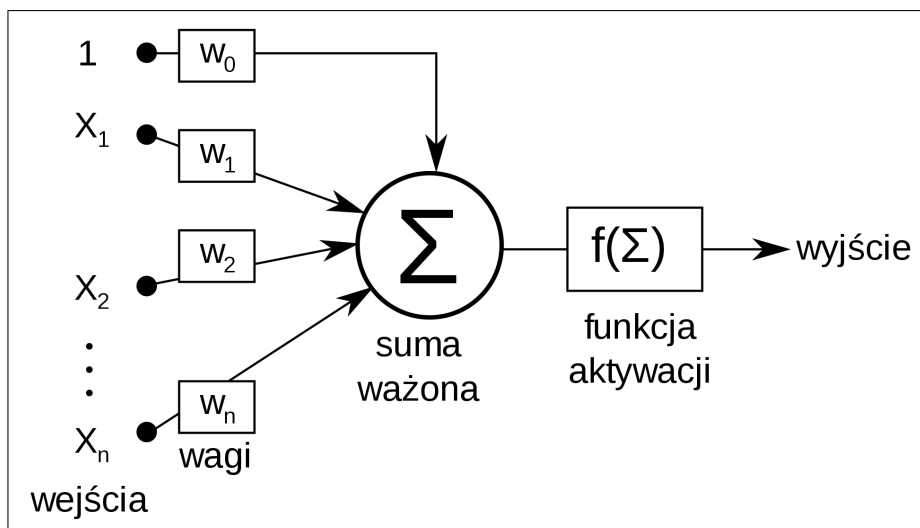
Sieci neuronowe to narzędzie zdobywające coraz większą popularność w rozwiązywaniu problemów, poszukiwaniu przybliżonych rozwiązań. Sztuczna sieć neuronowa inspirowana prawdziwymi neuronami i połączeniami jest stosowana w wielu zagadnieniach, m.in. do przetwarzania tekstu, rozpoznawania obrazów, klasyfikowania danych. Sieć neuronowa składa się z co najmniej kilku warstw. Pierwsza warstwa to dane wejściowe, ostatnia warstwa to dane wyjściowe. Warstwy pośrednie to warstwy ukryte, które odpowiadają za przetwarzanie danych. Połączenia pomiędzy neuronami są zależne od zastosowania projektowanej sieci neuronowej. Wartość każdego neuronu to suma iloczynów neuronów wchodzących w skład danego neuronu i odpowiadających im wag, po nałożeniu funkcji aktywacji (przykładowe funkcje znajdują się na rysunku 12), co dokładnie widać na rysunku 13. Istotną częścią sieci neuronowych jest to, aby odpowiednio dobrać wagi każdego z neuronów. Dobieranie tych wag nazywa się uczeniem¹⁶ sieci neuronowej i odbywa się, np. za pomocą algorytmu propagacji wstecznej (ang. backpropagation). Przykład sieci neuronowej znajduje się na rysunku 14. Szczegółowe informacje na temat budowy sieci neuronowych znajdują się w książce [5].



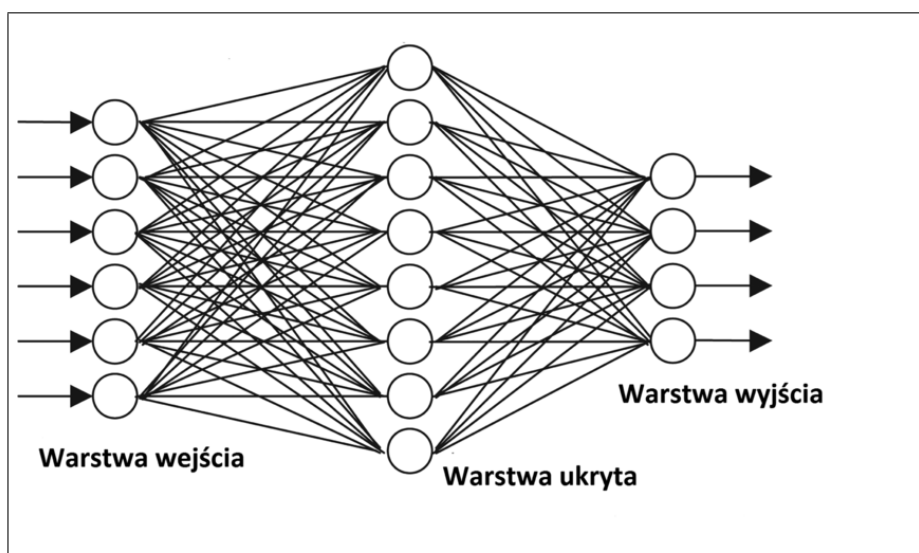
Rysunek 12: Przykładowe funkcje aktywacji¹⁷.

¹⁶Znane również jako trenowanie sieci neuronowej. Nazwa pochodzi od nazwy zbioru danych na którym wykonuje się uczenie – zbiór treningowy.

¹⁷Rysunek pobrany ze strony <https://towardsdatascience.com/complete-guide-of-activation-functions-34076e95d044>.



Rysunek 13: Schemat neuronu.¹⁸



Rysunek 14: Przykład klasycznej sieci neuronowej typu przekaz dalej (ang. feedforward neural network)¹⁹.

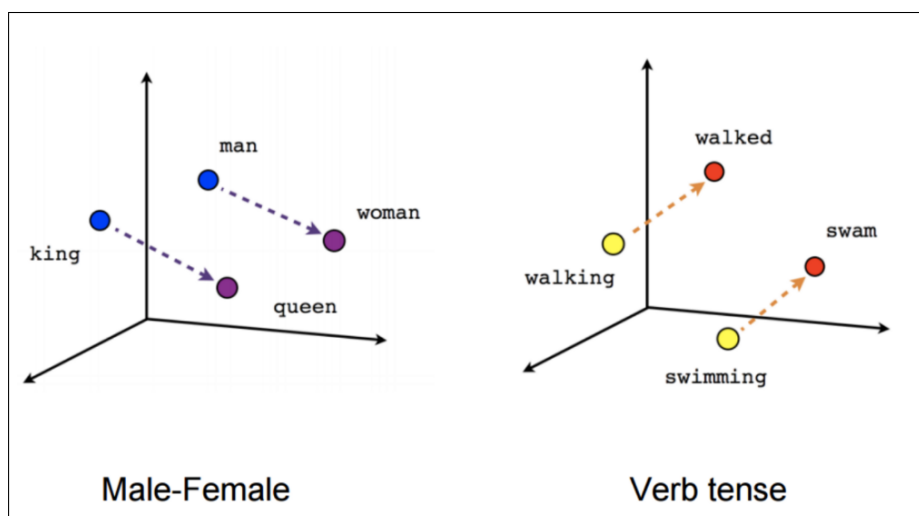
4.1 Word2vec

Word2vec to zbiór metod umożliwiający reprezentowanie słów z języka naturalnego za pomocą gęstych wektorów liczb rzeczywistych, najczęściej rzędu kilkuset. Różne cechy słowa są reprezentowane przez różne wartości wektora. Pojedyncza wartość wektora jest składową kilku cech słów. Wektor zawiera, m.in. informacje o znaczeniu słowa, części mowy, rodzaju, czasie. Wynikają stąd różne własności wektorów. Dla 2 słów o podobnym znaczeniu, odległość cosinusowa (opisana w

¹⁸Rysunek pobrany ze strony https://pl.wikipedia.org/wiki/Neuron_McCullocha-Pittsa.

¹⁹Rysunek pobrany ze strony <https://mlodytechnik.pl/technika/29211-czego-sie-ai-nie-nauczy-tego-terminator-nie-bedzie-umial>.

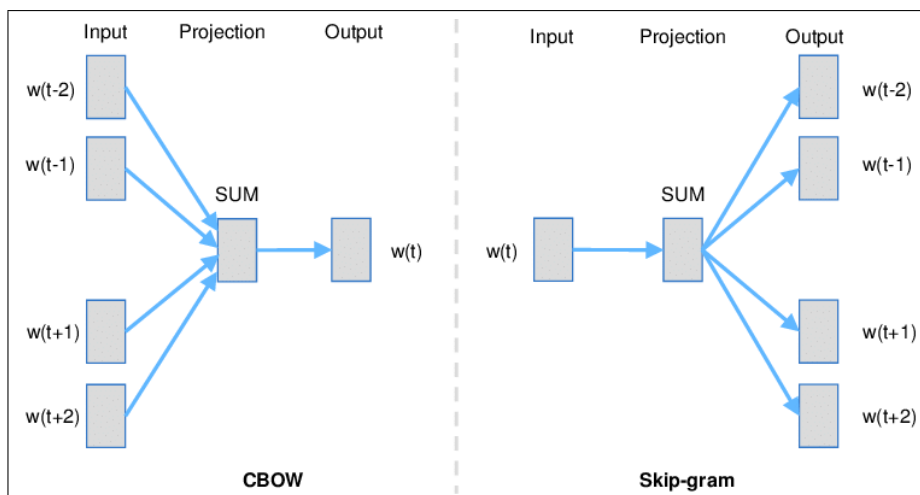
rozdziale 3.5.3) pomiędzy odpowiadającymi słowom wektorami jest mała. Wektory spełniają zależności, np.: kobieta - mężczyzna = królowa - król, oraz przeszedł - chodzenie = przepłynął - pływanie. Co pokazuje rysunek 15.



Rysunek 15: Przykładowe operacje na wektorach word2vec²⁰.

Uczenie modelu word2vec odbywa się za pomocą dwuwarstwowej sieci neuronowej. Wejściem jest jak największy korpus tekstowy, natomiast wyjściem przestrzeń wektorowa, najczęściej o rozmiarze kilkuset, w której każde słowo posiada swój unikalny wektor. Do projektowania sieci neuronowej wykorzystuje się 2 metody: CBOW (ang. continuous bag-of-words) oraz SG (ang. skip-gram). Obie metody przetwarzają tekst za pomocą przesuwne okna o zadanej szerokości, zwanego dalej kontekstem. Różnica między tymi dwiema metodami polega na kierunku przewidywania. Pierwsza z nich przewiduje słowo na podstawie kontekstu, natomiast druga przewiduje kontekst na podstawie słowa. Weźmy przykładowy kontekst zawierający słowa: „Ala”, „ma”, „dwa”, „czarne”, „koty”. Dla każdego ze słów „Ala”, „ma”, „czarne”, „koty” z osobna, CBOW próbuje przewidzieć słowo „dwa”. Natomiast SG na podstawie słowa „dwa” próbuje przewidzieć każde ze słów „Ala”, „ma”, „czarne”, „koty” osobno. Kolejność słów w kontekście nie ma znaczenia. Schemat tych metod znajduje się na rysunku 16. Szczegółowy opis modelu word2vec znajduje się w artykule [11].

²⁰Rysunek pobrany ze strony <https://www.urdunlp.com/2019/08/how-to-build-word-2-vector-for-urdu.html>.



Rysunek 16: Schemat sieci neuronowych wykorzystujących odpowiednio CBOW oraz SG do uczenia wag word2vec²¹.

4.1.1 Model bazowy

W niniejszej pracy został użyty gotowy, wytrenowany model **word2vec**²². Model był uczony na podstawie polskiej Wikipedii, książek oraz artykułów (łącznie około 1,5 miliarda słów). Rozmiar wektorów użytych w modelu to 100. Model został użyty przy pomocy biblioteki **gensim**²³.

4.1.2 Model odpowiadający na pytania

W celu łatwej reprezentacji tekstu w formie czytelnej dla komputera skorzystano z reprezentacji tekstu jako multizbiór wektorów (ang. bag of vectors, BOV). W tym celu, dla każdego słowa w tekście²⁴ (z wyłączeniem słów ze zbioru stop words) wzięto, będącą również słowem, jego formę bazową, następnie dla każdej formy bazowej wzięto jego wektor z modelu **word2vec** z poprzedniego akapitu. W celu oceny istotności artykułu względem pytania postanowiono przekształcić odpowiadające im multizbiory wektorów w pojedynczy wektor. Jest on liczony, jako średnia arytmetyczna wektorów multizbioru. Formalnie, przekształcenie tekstu na wektor można zapisać wzorami poniżej.

$$\text{vector}(\text{text}) = \frac{\sum \text{bov}(\text{normalise}(\text{text}))}{|\text{normalise}(\text{text})|}$$

$$\text{normalise}(\text{text}) = \text{baseForms}(\text{text}) \setminus \text{SW},$$

$$\text{bov}(\text{text}) = \{\text{word} \in \text{text} : \text{w2v}(\text{word})\},$$

$\text{w2v}(\text{word})$ - wektor słowa word z modelu bazowego **word2vec**,
 SW - zbiór słów stop words.

²¹Rysunek pobrany ze strony https://www.researchgate.net/figure/Continuous-Bag-of-words-CBOW-CB-and-Skip-gram-SG-training-model-illustrations_fig1_326588219.

²²Model **word2vec** znajduje się na stronie <https://github.com/sdadas/polish-nlp-resources/>.

²³Bibliotekę **gensim** można znaleźć na stronie <https://radimrehurek.com/gensim/>.

²⁴Tekstem w tym przypadku może być artykuł lub pytanie.

Warianty wyboru słów artykułu dla pytania

Model został przetestowany na kilka sposobów. Każdy z nich spełniał inne kryterium wyboru słów z artykułów:

- tylko słowa, które występują w pytaniu;
- dla każdego słowa w pytaniu wybrano 10 najbliższych słów (na podstawie odległości cosinusowej) i wybierano tylko te słowa;
- wszystkie słowa.

Pierwsze 2 opisane wyżej kryteria biorą pod uwagę tylko 500 pierwszych słów artykułu, natomiast 3 kryterium liczone jest dla pierwszych 100 słów. Założenie to zostało przyjęte w celu uproszczenia obliczeń, ale również aby ograniczyć szum spowodowany zbyt dużą liczbą wektorów. Pierwsze 500 słów artykułu bardzo często dobrze przybliża tematykę artykułu.

Ranking artykułów dla pytania

Do oceny istotności artykułów w kontekście pytań użyto funkcji podobieństwa cosinusowego opisaną w rozdziale 3.5.3. Na tej podstawie stworzono ranking artykułów dla pytania. Im wektor artykułu jest bliższy wektorowi pytania, tym wyższą ma pozycję w rankingu. Podobieństwo artykułu a do pytania q oznaczone jako $\text{similarity}(a, q)$ jest liczone wg wzoru poniżej.

$$\text{similarity}(a, q) = \text{cosine-similarity}(\text{vector}(a), \text{vector}(q)).$$

4.1.3 Wyniki modelu word2vec

Najlepszym z modeli okazał się ten, który brał pod uwagę tylko słowa artykułu, które wystąpiły w pytaniu. Dużo gorszy okazał się ten, który brał pod uwagę 10 najbliższych słów. Wyniki modelu uśredniającego wszystkie pierwsze 100 słów artykułu są zdecydowanie gorsze niż reszta modeli. Może to wynikać z faktu, że uśredniona wartość 100 wektorów nie zawiera już cech charakterystycznych artykułu, z powodu zbyt dużej liczby uśrednianych wektorów. Niestety, żaden z modeli nie uzyskał wyniku p_1 lepszego niż poprzednie modele. Wyniki najlepszego z nich są bardzo bliskie modelowi TF-IDF, ale nie dorównują modelowi opartemu o wektory rzadkie z rozdziału 3.5. Prawdopodobnie, model uzyskałby lepsze wyniki wykorzystując bazowy model word2vec wytrenowany na większym korpusie tekstowym ze zwiększoną wielkością wektorów do kilkuset. Pełne wyniki znajdują się w tabelach 23, 24 oraz 25.

Tablica 23: Wyniki modelu word2vec przy użyciu tylko tych słów z pierwszych 500 słów artykułu które znajdują się w pytaniu.

dane	p_1	p_{10}	p_{100}	MRR
tytuł	0.1189	0.2070	0.3300	0.1517
treść	0.2918	0.4840	0.6375	0.3593

Tablica 24: Wyniki modelu `word2vec` przy użyciu najbliższych 10 słów dla pierwszych 500 słów artykułu które znajdują się w pytaniu.

dane	p_1	p_{10}	p_{100}	MRR
tytuł	0.1025	0.1789	0.2770	0.1301
treść	0.1854	0.3560	0.5338	0.2432

Tablica 25: Wyniki modelu `word2vec` przy użyciu pierwszych 100 słów artykułu.

dane	p_1	p_{10}	p_{100}	MRR
tytuł	0.0838	0.1593	0.2572	0.1104
treść	0.0390	0.1110	0.2446	0.0638

4.2 Sieci neuronowe

Sieć neuronowa jest w stanie nauczyć się reprezentacji wektorowej słów, co pokazano w poprzednim rozdziale. Postanowiono jednak wykorzystać te wektory do zbudowania większej sieci neuronowej, która będzie korzystać z wektorów `word2vec`. Do implementacji sieci neuronowych wykorzystano bibliotekę `Tensorflow`²⁵.

4.2.1 Dane dla sieci neuronowych

Na dane wejściowe składają się pary <pytanie, artykuł>, ponieważ każdy artykuł należy rozpatrywać pod kątem danego pytania. Dla każdego pytania wzięto wektory `word2vec` pierwszych 20 słów. Dla każdego artykułu wzięto wektory `word2vec` pierwszych 20 słów tytułu oraz wektory `word2vec` pierwszych 500 słów treści. Uzyskano w ten sposób dane wejściowe składające się z 3 bloków wektorów: słów pytania, słów tytułu artykułu oraz słów treści artykułu.

Dane wyjściowe to waga artykułu. Waga artykułu który jest poprawną odpowiedzią na pytanie wynosi 1.0. Aby sieć miała możliwość rozróżnienia artykułów względem pytania, dane muszą zawierać również próbki niepoprawne (ang. *negative sampling*), tzn. artykuły które nie są odpowiedzią na pytanie. Artykuły niepoprawne zostały wybrane w sposób losowy z całego zbioru artykułów i mają wagę 0.0.

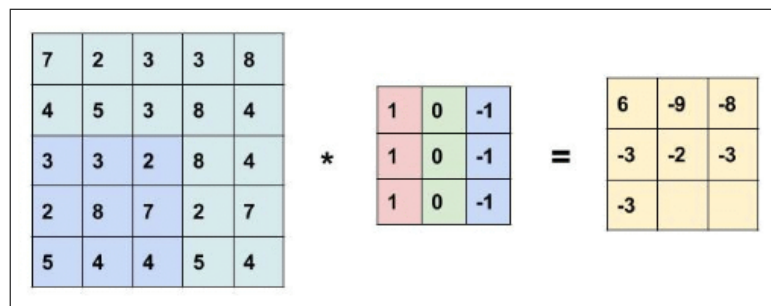
Aby zachować równowagę danych, liczba poprawnych i niepoprawnych odpowiedzi na pytania jest równa.

Uzyskano w ten sposób 24 188 próbek (co po pomnożeniu przez rozmiar pojedynczej próbki daje 1 306 152 000 danych). Dane zostały losowo podzielone w proporcjach 60:20:20, odpowiednio na zbiory treningowy, walidujący i testowy.

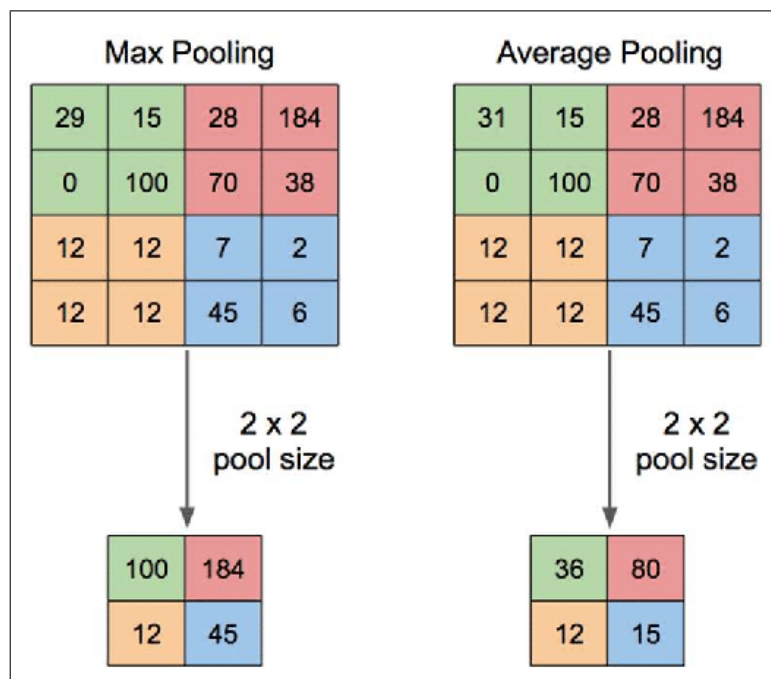
²⁵Biblioteka `Tensorflow` dostępna jest pod adresem <https://www.tensorflow.org/>.

4.2.2 Konwolucyjna sieć neuronowa

Konwolucyjna sieć neuronowa²⁶ (ang. convolutional neural network, CNN) to sieć złożona z warstw konwolucyjnych (ang. convolutional layer) oraz warstw łączących (ang. pooling layer). Warstwy konwolucyjne odpowiadają za ekstrakcję cech z poprzedniej warstwy za pomocą przesuwanego filtra. Warstwa łącząca służy do zmniejszenia rozmiaru cech oraz złożoności sieci. Zasada działania warstw jest pokazana na rysunkach 17 raz 18.



Rysunek 17: Przykładowa warstwa konwolucyjna sieci neuronowej. Widoczne na rysunku macierze od lewej strony to odpowiednio: dane wejściowe, filtr, dane wyjściowe (mapa cech, ang. feature map). Macierz filtra zawiera wagi które są dobierane w trakcie uczenia się sieci neuronowej²⁷.



Rysunek 18: Przykładowa warstwa łącząca. Po lewej łączenie przy użyciu maksimum, po prawej łączenie przy użyciu średniej²⁸.

²⁶W polskiej literaturze znana również jako splotowa sieć neuronowa.

²⁷Rysunek pobrany ze strony <https://mc.ai/understanding-backpropagation-in-convolution-layer-in-convnets/>.

²⁸Rysunek pobrany ze strony https://www.researchgate.net/figure/Illustration-of-Max-Pooling-and-Average-Pooling-Figure-2-above-shows-an-example-of-max_fig2_

Dla każdego bloku danych zostały użyte po 2 warstwy konwolucyjne złożone z 64 filtrów o szerokości filtrów 2 oraz 3. Istotne jest to, że warstwy użyte w każdym z tych bloków współdzielią ze sobą te same wagi²⁹. Ma to na celu nauczenie się tych samych cech tekstu, niezależnie z którego bloku pochodzą. Uzyskane w ten sposób dane są łączone przez kolejną warstwę. Następnie za pomocą 3 w pełni połączonych warstw (ang. dense layer) dane są redukowane kolejno do 32, 16 i ostatecznie pojedynczej wartości odpowiadającej wadze artykułu. Szczegółowy schemat sieci znajduje się na rysunku 19.

Sieć była uczona do momentu braku lepszych wyników dla danych walidujących (zajęło to około 100 epok co trwało 30 minut przy użyciu komputera opisanego w rozdziale A.3). Skuteczność sieci dla danych testowych to nieco ponad 90%. Dla porównania skuteczność losowego klasyfikatora przyporządkowującego danym jedną z dwóch klas to 50%. Jest więc to wynik bardzo dobry, trzeba jednak zaznaczyć, że jest to wynik na bardzo ograniczonej liczbie artykułów i pytań. Wyniki dla pozostałych zbiorów znajdują się w tabeli 26.

Wyniki na pełnych danych (użyte wszystkie artykuły, pytania) oznaczone w poprzednich rozdziałach jako: p_1 , p_{10} , p_{100} okazały się praktycznie zerowe nawet dla najłagodniejszej metryki p_{100} . Po przeanalizowaniu wyników okazało się, że sieć w przeważającej mierze zwracała wagi bliskie 1.0 i 0.0. Oznacza to, że sieć nauczyła się klasyfikować pary <pytanie, artykuł> do dwóch kategorii. Wynika to prawdopodobnie z faktu, że dostarczone dane zawierają wagi artykułów wyłącznie 0.0 i 1.0. Prawdopodobnie sieć zwróciłaby dużo lepsze wyniki, gdyby w danych wejściowych znalazły się artykuły z wagami pomiędzy 0.0 i 1.0. Niestety, zebranie takich danych jest bardzo czasochłonne i wykracza poza niniejszą pracę. Dane uzyskane przez zbudowaną sieć mogą posłużyć jako wstępna selekcja artykułów, ale budowanie rankingu artykułów na ich podstawie jest bezcelowe.

Tablica 26: Wyniki konwolucyjnej sieci neuronowej.

dane	skuteczność
treningowe	0.9966
walidujące	0.9033
testowe	0.9010

4.2.3 Głęboka sieć uśredniająca

W odróżnieniu od sieci konwolucyjnej, sieć uśredniająca (ang. deep averaging network, DAN) jest dużo prostsza. Ideą sieci jest uśrednienie wektorów wejściowych do jednego pojedynczego wektora. Każdy z trzech bloków wejściowych jest za pomocą osobnej warstwy zmniejszany do jednego wektora. Następnie wektory zostają połączone, co daje warstwę o rozmiarze 300. Następnie za pomocą 2 w pełni połączonych warstw (ang. dense layer) dane są redukowane kolejno do 100 i ostatecznie pojedynczej wartości odpowiadającej wadze artykułu. Szczegółowy schemat sieci znajduje się na rysunku 20.

Wyniki sieci uśredniającej dla zbioru testowego są lepsze o niecałe 4 punkty procentowe względem sieci z poprzedniego rozdziału. Wyniki dla pozostałych zbiorów

333593451.

²⁹W uproszczeniu oznacza to, że dane wejściowe każdego bloku są przetwarzane w ten sam sposób

znajdują się w tabeli 27.

Podobnie jak dla konwolucyjnej sieci neuronowej, głęboka sieć uśredniająca nauczyła się klasyfikować pary <pytanie, artykuł> do dwóch kategorii, dlatego pełny ranking jak i metryki p_1 , p_{10} , p_{100} nie mają praktycznego zastosowania również i w tym przypadku.

Tablica 27: Wyniki głębokiej sieci uśredniającej.

dane	skuteczność
treningowe	0.9910
walidujące	0.9426
testowe	0.9388

4.2.4 Poprawiony zbiór danych

Po przeanalizowaniu użytych danych do przedstawionych wyżej sieci neuronowych, zastanowiono się, czy można te dane wybrać lepiej. Dla każdego pytania były wybierane 2 artykuły, z czego jeden był poprawną odpowiedzią, natomiast drugi był wybierany losowo z całego zbioru artykułów. Pierwszy artykuł był silnie powiązany z pytaniem, natomiast drugi najczęściej był zupełnie niezwiązany z pytaniem. Odróżnienie, który spośród tych 2 artykułów jest bliżej związany z pytaniem wydaje się być rzeczą prostą. Postanowiono więc lepiej wybrać niepoprawne artykuły, tak aby różnica pomiędzy poprawnym, a niepoprawnym artykułem była możliwie mała. W tym celu wykorzystano ranking jednego z wcześniejszych modeli³⁰. Przygotowano nowy zbiór danych, ale tym razem dla każdego pytania zamiast losowego artykułu, wybrano **najwyżej oceniony artykuł**³¹ z rankingu artykułów danego pytania wybranego modelu. Tak wybrany artykuł bardzo często jest mocno powiązany z pytaniem, ale jednocześnie nie zawiera odpowiedzi na zadane pytanie. Przykładowe próbki danych znajdują się w tabeli 28.

Tablica 28: Przykładowe próbki danych sieci neuronowej.

pytanie	poprawny artykuł z wagą 1.0	niepoprawny artykuł z wagą 0.0
„gdzie powstaje FAST, największy na świecie radioteleskop”	„Five hundred meter Aperture Spherical Telescope”	„Obserwatorium Arcibo”
„w jakich krajach obchodzone są święta Boxing Day i Małe Boże Narodzenie?”	„Little Christmas”	„kultura Kanady”
„kogo w kronice opery nazywano muzycznym mózgiem festiwalu w Glyndebourne”	„Fritz Busch”	„Malena Ernman”
„kto zbudował pierwszą kuchenkę elektryczną i zatrudniał Gabriela Narutowicza”	„Friedrich Wilhelm Schindler”	„Przemysław Podgórski”

³⁰Wybrano model TF-IDF zastosowany dla treści artykułów.

³¹Z pominięciem poprawnego artykułu jeżeli znalazł się na pierwszym miejscu w rankingu.

4.2.5 Wyniki sieci neuronowych

Uczenie przebiegało tak samo, jak w przypadku poprzedniego zbioru danych. Wyniki znajdują się w tabelach 29 oraz 30. Tak jak się spodziewano, odróżnienie dwóch blisko związanych ze sobą artykułów jest zadaniem znacznie trudniejszym niż odróżnienie dwóch niepowiązanych ze sobą artykułów. Problem jest na tyle trudny, że żadna z wyżej przedstawionych sieci neuronowych nie była w stanie się tego nauczyć. Obie sieci uzyskały dokładność na poziomie 0.7 i pomimo długiego czasu uczenia się, tego wyniku nie udało się poprawić. Duża różnica wyników pomiędzy zbiorem testowym, a treningowym świadczy o przeuczeniu (ang. overfitting) sieci neuronowych³².

Tablica 29: Wyniki konwolucyjnej sieci neuronowej dla poprawionego zbioru danych.

dane	skuteczność
treningowe	0.9890
walidujące	0.6922
testowe	0.6845

Tablica 30: Wyniki głębokiej sieci uśredniającej dla poprawionego zbioru danych.

dane	skuteczność
treningowe	0.9923
walidujące	0.7312
testowe	0.7104

4.2.6 Uproszczenie sieci neuronowych

W celu zniwelowania zjawiska przeuczenia się sieci neuronowych z poprzedniego rozdziału postanowiono uprościć i zmniejszyć ich rozmiar. W tym celu nieznacznie przeprojektowano sieci neuronowe.

Konwolucyjna sieć neuronowa

W przypadku sieci konwolucyjnej dla każdego bloku przetwarzającego pytanie lub artykuł zmniejszono rozmiar filtrów pierwszej warstwy konwolucyjnej z 64 na 4. Rozmiar filtrów kolejnych warstw konwolucyjnych zmieniono z 64 do 32. Dodatkowo usunięto pierwszą pełną warstwę o rozmiarze 32 w końcowym bloku agregującym wyniki 3 bloków danych wejściowych. Tym sposobem liczba parametrów uczących konwolucyjnej sieci neuronowej zmniejszyła się z blisko 40 000 do 4213.

Głęboka sieć uśredniająca

Głęboka sieć uśredniająca jest dużo łatwiejsza w budowie niż konwolucyjna sieć neuronowa, dlatego zmiany były dużo prostsze. Jedyną zmianą było dodanie war-

³²Zjawisko przeuczenia sieci neuronowej zachodzi wtedy, gdy model posiada zbyt dużo parametrów względem liczby i wielkości próbek. Zjawisko to objawia się bardzo dobrymi wynikami na danych treningowych, które nie przekładają się na dobre wyniki na danych z którymi model do tej pory nie miał styczności.

stwy dropout³³ przed pierwszą pełną warstwą.

Wyniki uproszczonych sieci neuronowych

Pomimo lepszej skuteczności wyników sieci neuronowych, wyniki miar p_1 , p_{10} , p_{100} oraz MRR nie są zadowalające. Wyniki niektórych miar są nieznacznie lepsze niż w przypadku poprzednich sieci neuronowych, ale warto przypomnieć że wyniki miar dla poprzednich modeli sieci neuronowych były bardzo bliskie, a nawet równe 0. Jedyne zaobserwowane wyniki które są lepsze niż poprzednie to miara p_{100} dla konwolucyjnej sieci neuronowej wynosząca blisko 0.032 oraz miary p_1 , MRR wynoszące odpowiednio 0.011 oraz 0.013 dla głębokiej sieci uśredniającej. Oznacza to, że modele nauczyły się poprawnie odpowiadać przynajmniej na kilka pytań.

Niestety nauka na ograniczonej liczbie artykułów bardzo słabo przekłada się na cały korpus zawierający ponad 1 600 000 artykułów. Prawdopodobną przyczyną takiego wyniku, jest niewystarczający zbiór danych. Korpus zawiera zaledwie 10 668 pytań, z czego po odjęciu zbioru walidującego i testowego zostaje ich jedynie 6401. Biorąc pod uwagę fakt, że jedna próbka danych składa się z $20 + 20 + 500 = 540$ wektorów o rozmiarze 100 co daje 54 000 wartości, nauczanie prawidłowości na tak dużej liczbie wartości za pomocą zaledwie 6401 pytań wydaje się być skazane na niepowodzenie. Wyniki uproszczonych sieci neuronowych znajdują się w tabelach 31 oraz 32.

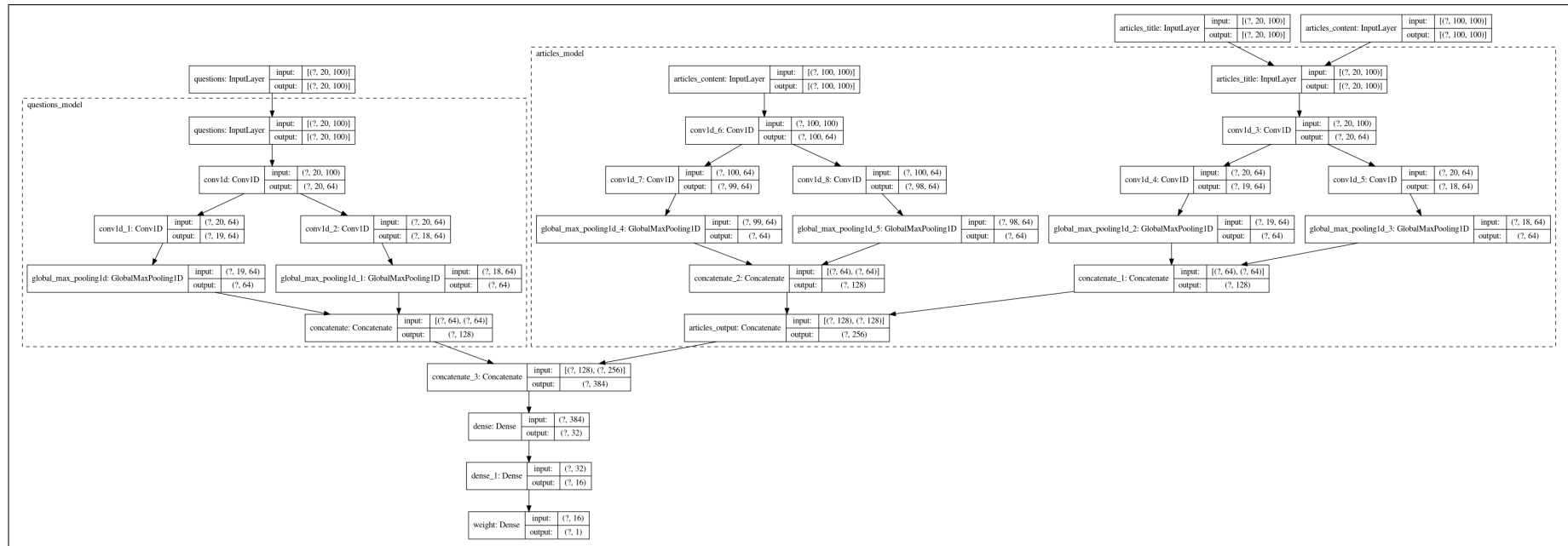
Tablica 31: Wyniki uproszczonej konwolucyjnej sieci neuronowej dla poprawionego zbioru danych.

dane	skuteczność
treningowe	0.8089
walidujące	0.7616
testowe	0.7548

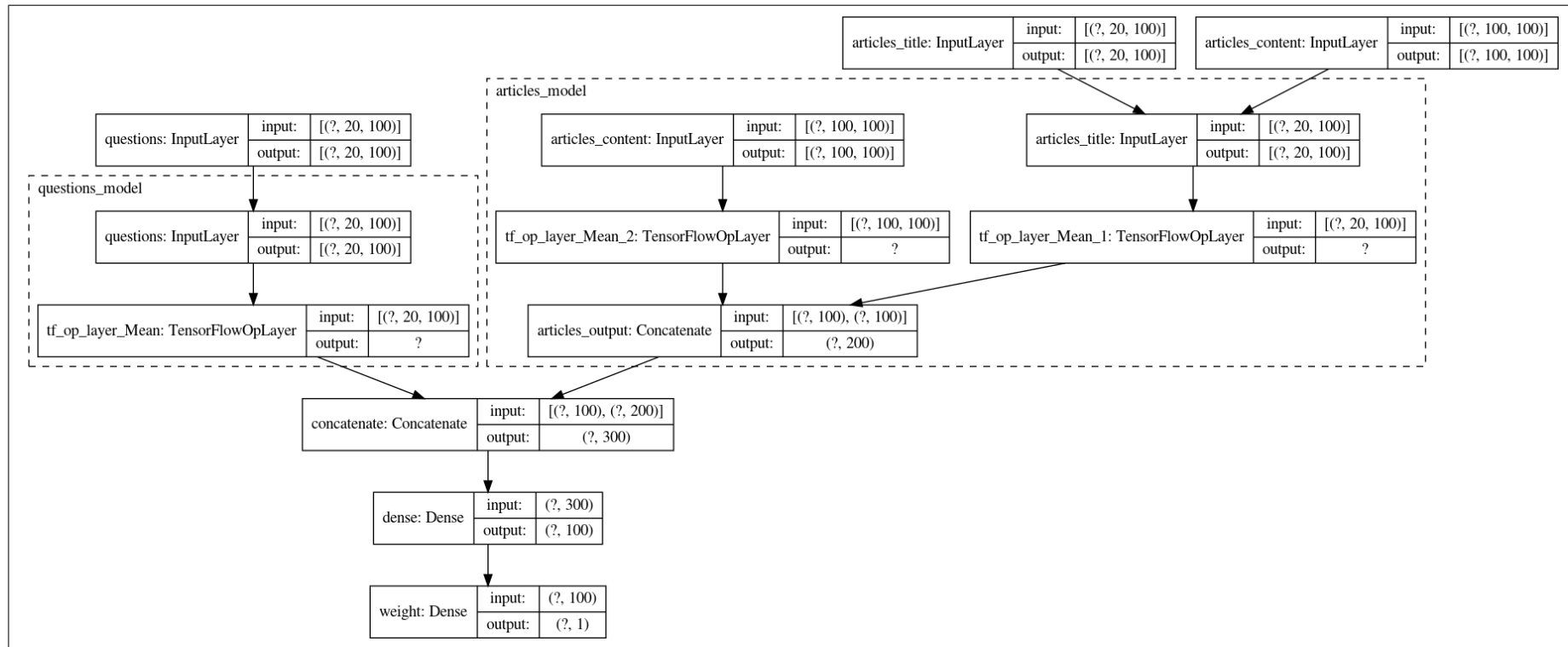
Tablica 32: Wyniki uproszczonej głębokiej sieci uśredniającej dla poprawionego zbioru danych.

dane	skuteczność
treningowe	0.8667
walidujące	0.8194
testowe	0.7978

³³Dropout to warstwa usuwająca połączenia między niektórymi neuronami w celu zapobiegania przeuczeniu się sieci neuronowej.



Rysunek 19: Schemat użytej konwolucyjnej sieci neuronowej.



Rysunek 20: Schemat głębokiej sieci uśredniającej.

Rozdział 5

Model agregujący

Każdy z poprzednich modeli daje lepsze wyniki dla innych pytań. Jeden dla pytań krótkich, inny dla pytań mających dużo słów kluczowych albo zawierających dwuczłonowe nazwy własne. Aby uzyskać możliwie najlepszy wynik końcowy, podjęto próbę połączenia poprzednich modeli za pomocą jednego nadrzędnego.

5.1 Algorytm genetyczny

Algorytm genetyczny to rodzaj heurystycznego algorytmu przeszukującego zbiór rozwiązań w celu znalezienia możliwie najlepszego rozwiązania. Nazwa pochodzi od genetyki³⁴, na której wzorowane były niektóre mechanizmy użyte w algorytmie genetycznym. Sposób działania algorytmu genetycznego bardzo przypomina naturalne zjawisko ewolucji występujące w przyrodzie.

Bazą algorytmu genetycznego jest zbiór osobników nazywany populacją. Każdy osobnik³⁵ jest złożony z genów które odpowiadają za część rozwiązania. Każdy osobnik to potencjalne rozwiązanie. Za pomocą funkcji oceny algorytm ocenia przystosowanie osobnika i tym samym jakość rozwiązania. Przy pomocy mutacji oraz krzyżowania algorytm modyfikuje populację³⁶, aby znaleźć możliwie najlepszego osobnika, dla którego funkcja oceny będzie możliwie minimalna lub maksymalna w zależności od wariantu. Schemat działania algorytmu genetycznego znajduje się na rysunku 21 oraz 22.

Bardzo istotną kwestią podczas projektowania algorytmu genetycznego jest odpowiednie dobranie genów osobnika oraz funkcji oceny osobnika. Dobre dobranie genów wpływa pozytywnie na jakość i szybkość poszukiwania rozwiązania.

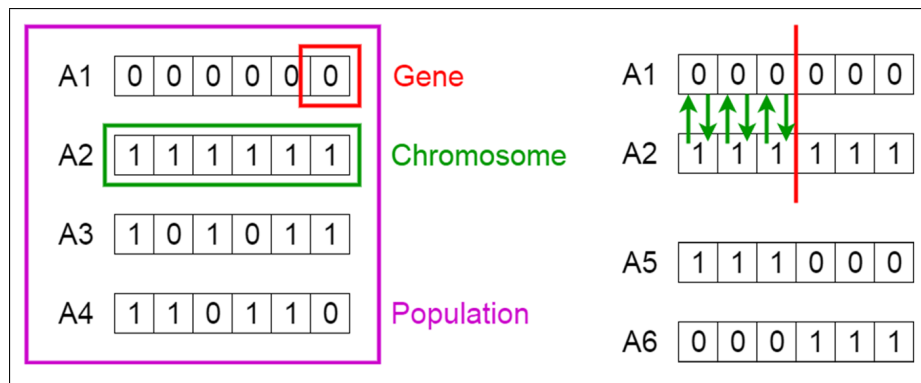
Do implementacji algorytmu genetycznego w poniższej pracy użyto biblioteki `deap`³⁷.

³⁴Genetyka to dziedzina nauki zajmująca się dziedzicznością i zmiennością organizmów.

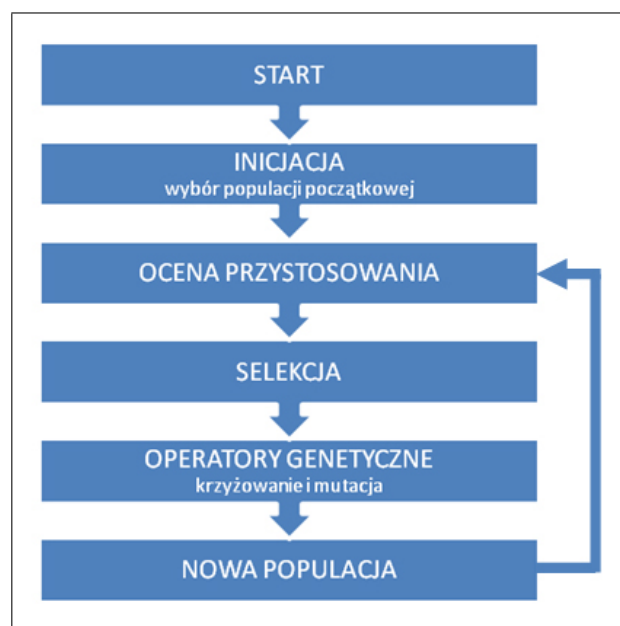
³⁵Osobnik jest nazywany również chromosomem. Jest to pojęcie zaczerpnięte z genetyki.

³⁶Jest to element przeszukiwania zbioru rozwiązań.

³⁷Biblioteka `deap` dostępna jest pod adresem <https://github.com/DEAP/deap>.



Rysunek 21: Po lewej stronie przykład genów, chromosomów, populacji. Po prawej stronie przykład krzyżowania osobników³⁸.



Rysunek 22: Schemat działania algorytmu genetycznego³⁹.

5.1.1 Dane dla algorytmu ewolucyjnego

Jako dane wejściowe wybrano wyniki poszczególnych modeli:

- TF-IDF dla tytułów artykułów zarówno dla słów jak i bigramów (2);
- TF-IDF dla treści artykułów zarówno dla słów jak i bigramów (2);
- kontekstowe dla tytułów artykułów zarówno dla słów jak i bigramów (z użyciem modeli TF-IDF oraz wektorów z miarą cosinusową) dla $n = 3$ ($2 \cdot 2 = 4$);

³⁸Rysunek pobrany ze strony <https://towardsdatascience.com/introduction-to-genetic-algorithms-including-example-code-e396e98d8bf3>.

³⁹Rysunek pobrany ze strony http://home.agh.edu.pl/~pernach/wyklady/index.php?action=wyklad10_5.

- kontekstowe dla treści artykułów zarówno dla słów jak i bigramów (z użyciem modeli TF-IDF oraz wektorów z miarą cosinusową) dla $n = 15, 250$ ($2 \cdot 2 \cdot 2 = 8$);
- wszystkie `word2vec` (6);
- konwolucyjna sieć neuronowa (1);
- głęboka sieć uśredniająca (1).

Łącznie dało to $2 + 2 + 4 + 8 + 6 + 1 + 1 = 24$ modele. Wyniki wszystkich użytych modeli zostały znormalizowane⁴⁰, tak aby wagi artykułów mieściły się w zakresie $[0, 1]$ przy czym artykuły bliżej związane z pytaniem miały wyższą wagę. W tym celu dla każdego z modeli osobno przeprowadzono transformację danych zgodnie ze wzorem $X' = \frac{X - X_{min}}{X_{max} - X_{min}}$, gdzie X to wektor wag artykułów dla pytań.

Pytania

W celu zapewnienia tego samego zbioru pytań uczących, walidujących, testowych dla wszystkich modeli w niniejszej pracy, przyjęto te same zbiory pytań treningowych, walidujących, testowych, które były użyte w sieciach neuronowych.

5.1.2 Osobnik

Pojedynczy osobnik, to wektor 24 wartości rzeczywistych z przedziału $[0, 1]$, które odpowiadają każdemu z wybranych modeli wejściowych.

Ostateczna waga artykułu dla pytania q dla osobnika i to suma iloczynów wag artykułu poszczególnych modeli i odpowiadających im wagom osobnika i . Na podstawie tych wag jest tworzony ranking artykułów dla każdego z pytań. Ocena osobnika to miara MRR pozycji poprawnych artykułów w rankingu dla każdego z pytań.

5.1.3 Przebieg algorytmu

Populacja zawiera 200 osobników, których cechy na początku zostały wylosowane. Liczba generacji to 100. Mutacja jest wykonywana z prawdopodobieństwem 0.1, podczas której zmieniane są 3 losowe wagi osobnika. Krzyżowanie osobników wykonywane jest z prawdopodobieństwem 0.5. Selekcja osobników do krzyżowania jest wybierana w sposób turniejowy o wielkości 5. Algorytm dąży do maksymalizacji funkcji oceny osobnika.

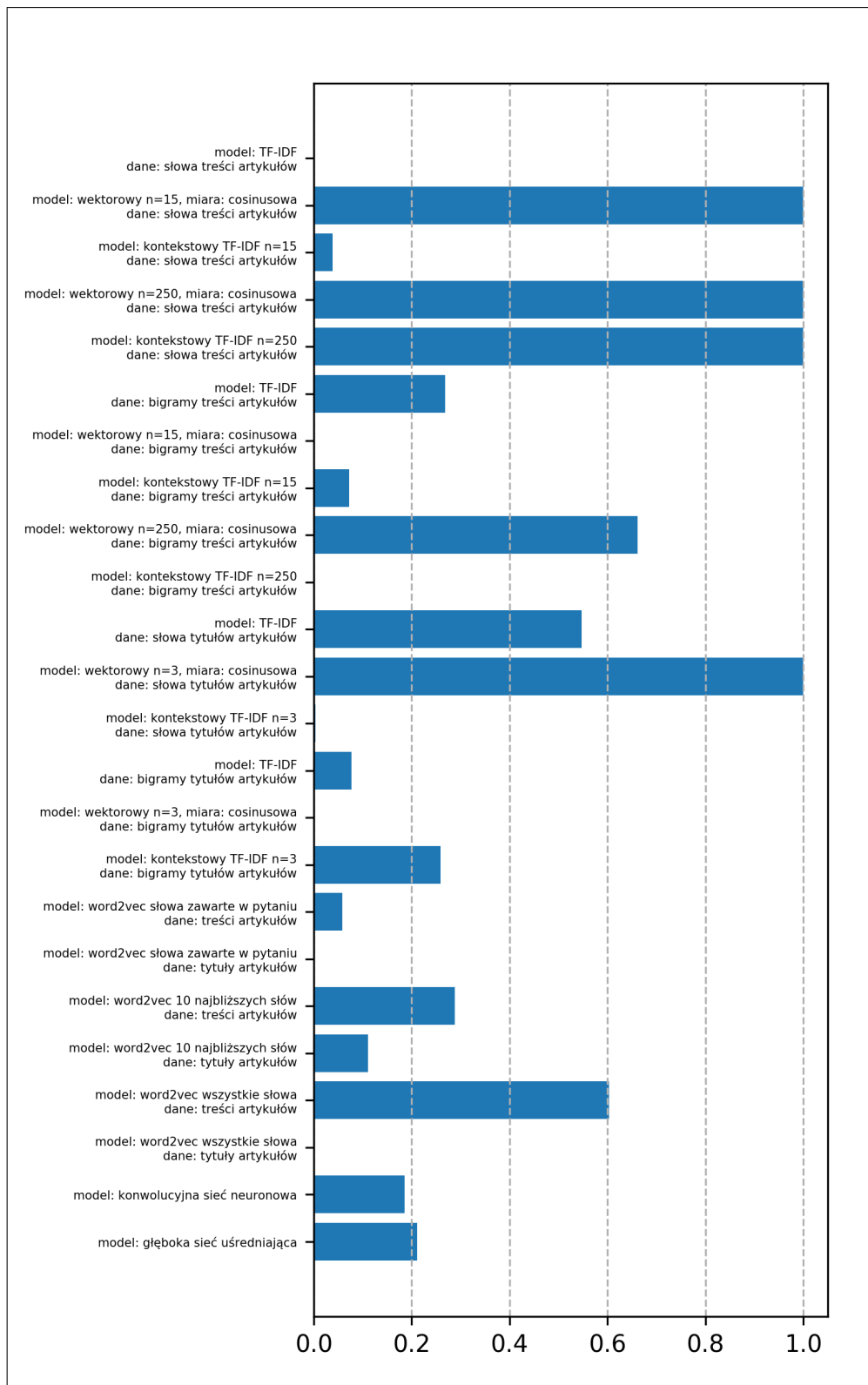
5.1.4 Wyniki modelu opartego o algorytm genetyczny

Wyniki zostały przeprowadzone dla najlepszego z osobników, jego cechy są pokazane na rysunku 23. Zgodnie z oczekiwaniami model zintegrowany wykazał poprawę względem poprzednich modeli. Skuteczność p_1 dla danych testowych to lekko ponad 60%. Jest to najlepszy wynik z dotychczasowych modeli, wyższy o około 10 punktów procentowych. Oznacza to, że niektóre z użytych modeli dają lepsze wyniki dla innych pytań i odpowiednie ich dobranie w jeden system pozwala uzyskać lepsze rezultaty. Szczegółowe wyniki znajdują się w tabeli 33.

⁴⁰Normalizacja której użyto w tym przypadku jest znana jako skalowanie min-max (ang. min-max Feature scaling).

Tablica 33: Wyniki algorytmu genetycznego. Dla porównania na dole znajdują się wyniki najlepszego dotychczas modelu.

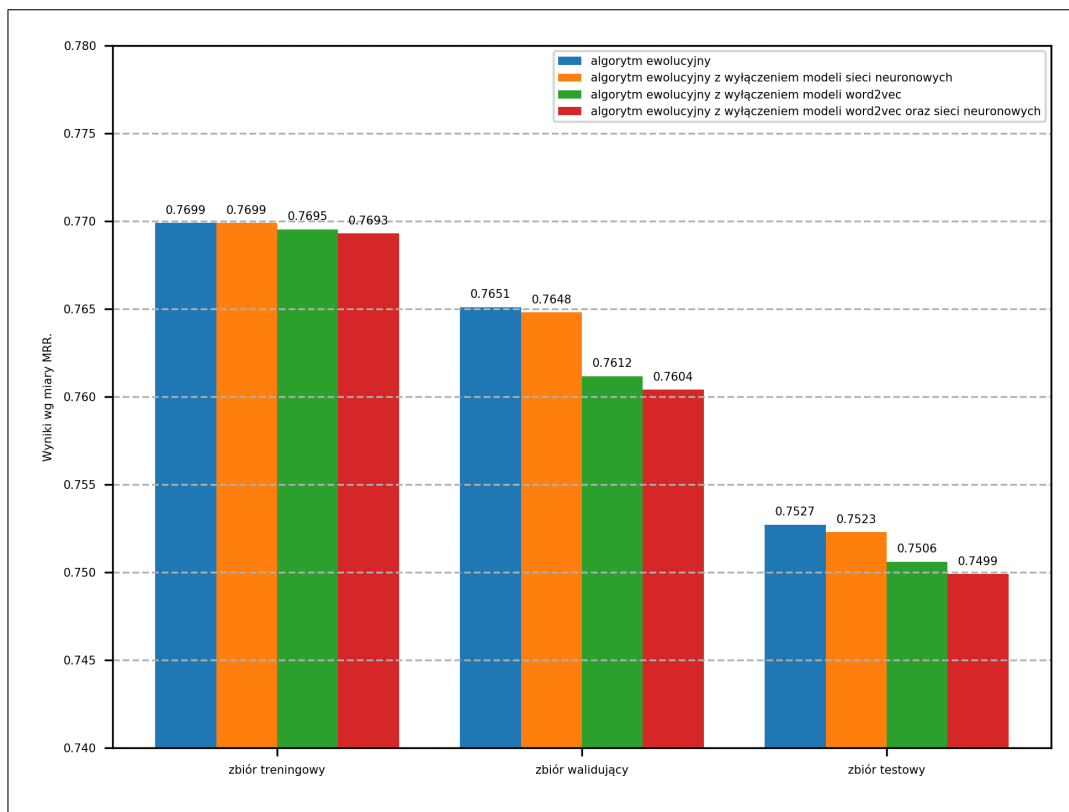
model	zbiór	p_1	p_{10}	p_{100}	MRR
genetyczny	treningowy	0.7004	0.9258	0.9692	0.7842
	walidujący	0.6721	0.8869	0.9487	0.7541
	testowy	0.6561	0.8777	0.9429	0.7385
słowny kontekstowy TF-IDF dla $n = 150$		0.5146	0.8197	0.9272	0.6252



Rysunek 23: Cechy najlepszego osobnika populacji oznaczające wagę odpowiedniego modelu. Po wartościach na osi X widać które modele mają większy wpływ na wynik, a które mniejszy. Cztery modele mają wartość bliską lub równą 1.0, trzy modele mają wartość w okolicach 0.5, natomiast pozostałe modele w okolicach 0.2 lub niżej.

5.1.5 Wyniki algorytmu ewolucyjnego z pominięciem wybranych modeli

W celu sprawdzenia wpływu modeli opartych o word2vec oraz sieci neuronowe na końcowy wynik, postanowiono przeprowadzić testy algorytmu ewolucyjnego z pominięciem wyżej wymienionych modeli. Wyniki znajdują się na rysunku 24. Użycie modeli bazujących na sieciach neuronowych dało nieznacznie lepsze rezultaty o około 0,1%. Z kolei użycie modeli bazujących na modelach word2vec dało lepsze rezultaty o około 0,5%. Nie są to duże różnice, jednak zauważalne, co oznacza, że modele sieci neuronowych oraz word2vec mają pozytywny wpływ na wynik końcowy.



Rysunek 24: Wyniki algorytmu ewolucyjnego z pominięciem wybranych modeli. Na osi X znajdują się poszczególne modele pogrupowane ze względu na zbiór danych. Na osi Y znajdują się wyniki poszczególnych modeli w oparciu o miarę MRR.

Rozdział 6

Podsumowanie

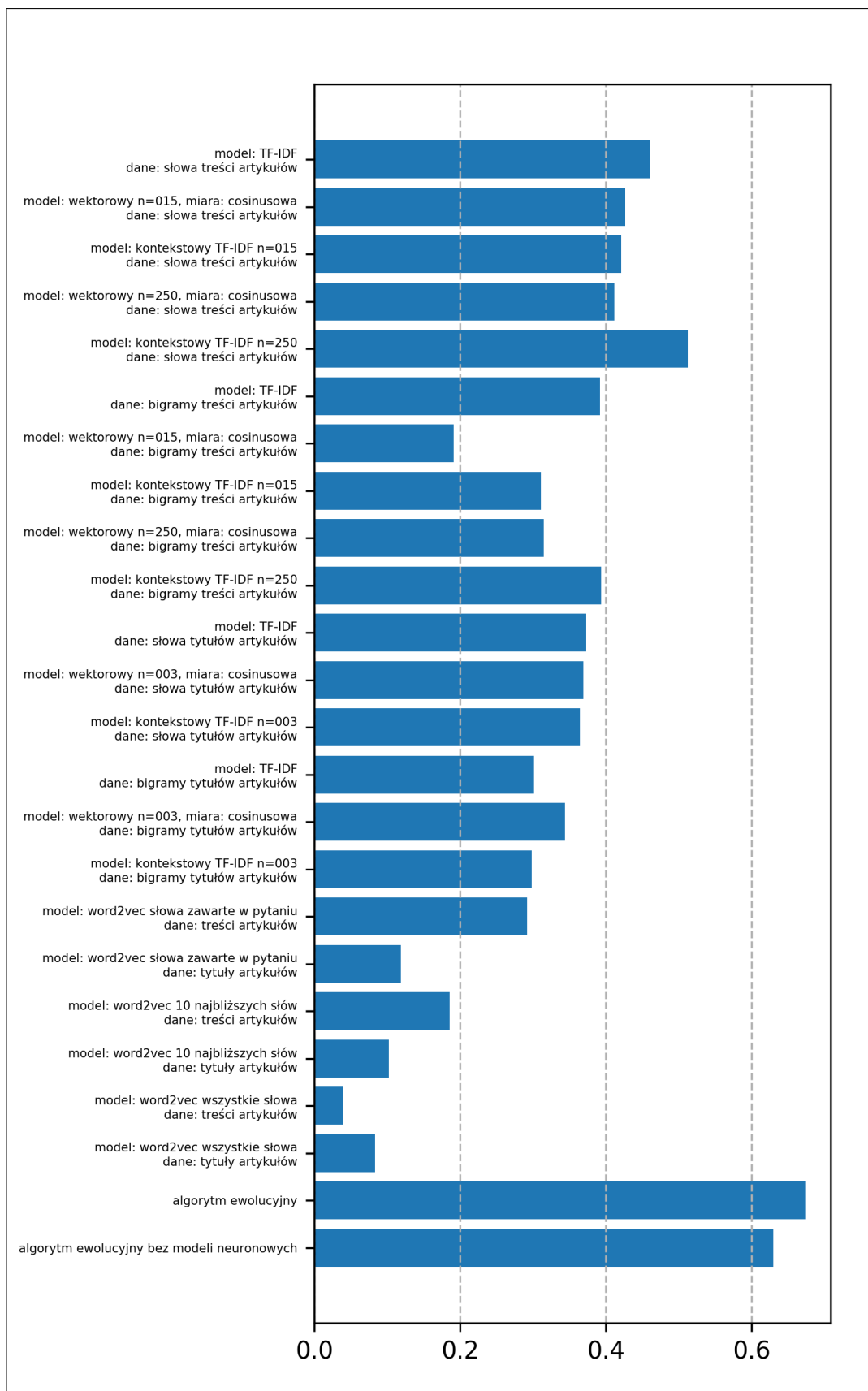
W powyższej pracy zostały przedstawione i opisane trzy typy modeli rozwiązujące zagadnienie odpowiadania na pytania zadawane Wikipedii. Kolejne modele to rozwinięcie lub połączenie rozwiązań z poprzednich rozdziałów.

6.1 Wyniki zbiorcze wybranych modeli

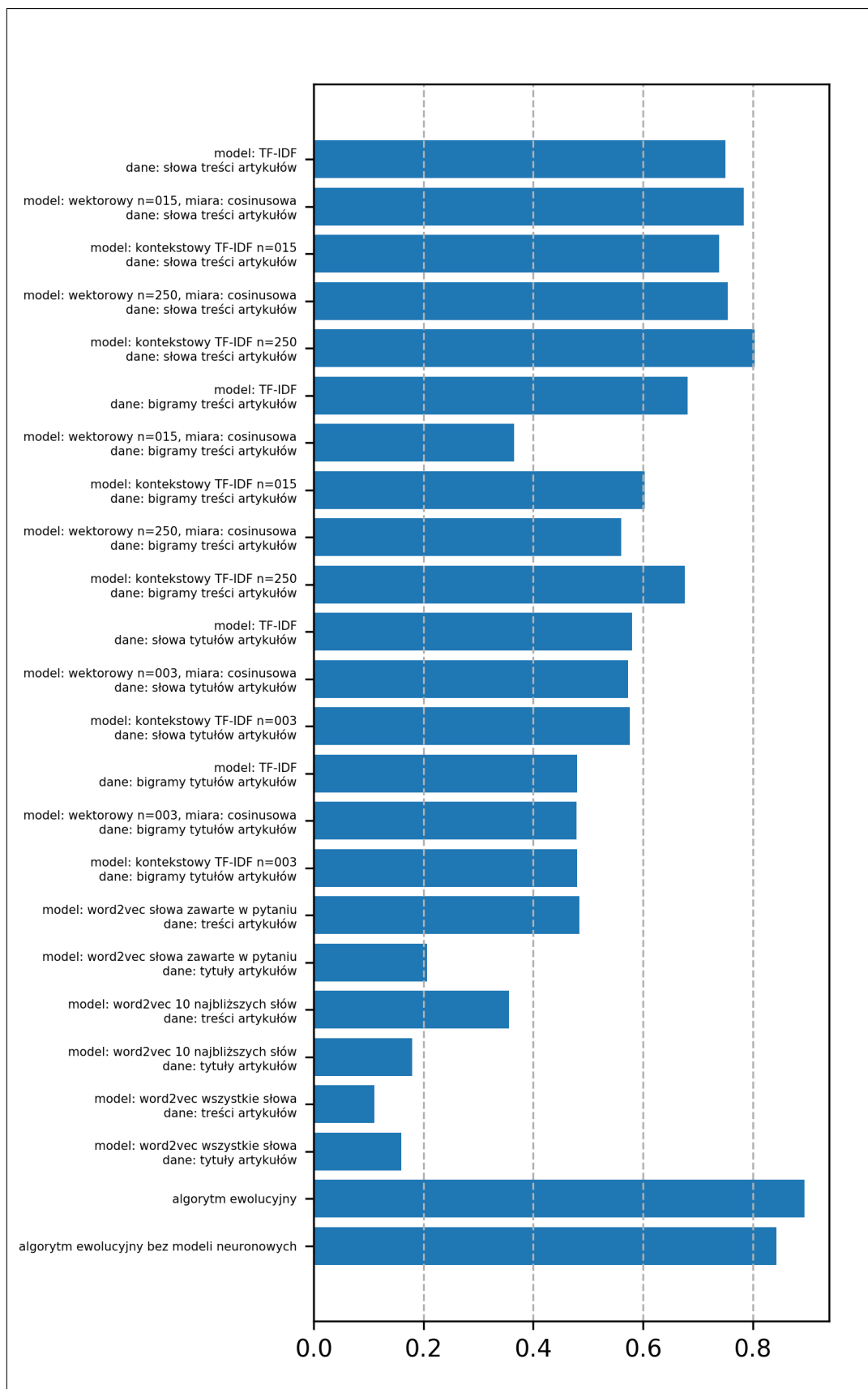
Modelem, który osiągnął najlepsze rezultaty okazał się model zagregowany oparty o algorytm genetyczny. Jest to wynik zgodny z oczekiwaniami, ponieważ model ten dysponował wszystkim wynikami wcześniejszych modeli. Wynik miary p_1 dla algorytmu genetycznego to ponad 65%. Miara p_{10} , którą można rozumieć, jako pierwszą stronę wyników wyszukiwania artykułów osiągnęła wynik prawie 90%. Po uwzględnieniu miary p_1 i p_{10} wynika, że dla 65 na 100 pytań, poprawny artykuł znajduje się na pierwszym miejscu, natomiast dla kolejnych 25 na 100 pytań poprawny artykuł znajduje się na miejscach od 2 do 10. Jest to wynik, który można uznać za satysfakcjonujący z punktu widzenia potencjalnego użytkownika.

Warto przyrzeć się wynikom modeli bazowych, które odpowiadają za dobre wyniki modelu zagregowanego. Najlepszy z modeli podstawowych okazał się model kontekstowy, oparty o treści artykułów dla parametru $n = 250$. Osiągnął on wynik p_1 w okolicach 50%, natomiast p_{10} w granicach 80%. Najlepszy model bazujący na tytułach artykułów to klasyczny model TF-IDF, który osiągnął wynik p_1 równy 37%, zaś p_{10} równy blisko 58%.

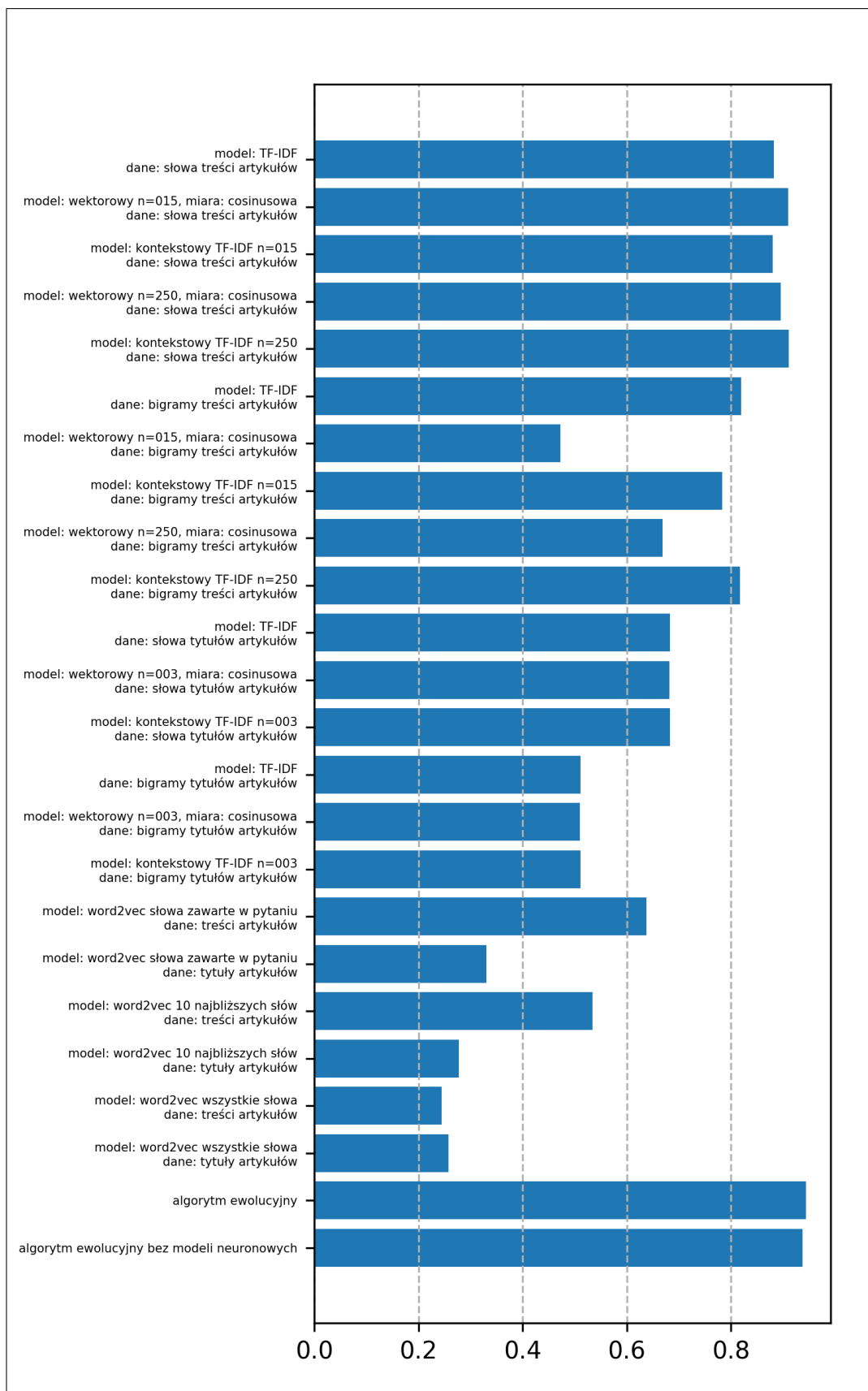
Wyniki zbiorcze wybranych modeli dla miar typu p znajdują się w tabelach 25, 26, 27. Wyniki zbiorcze wybranych modeli dla miary MRR znajdują się w tabeli 28.



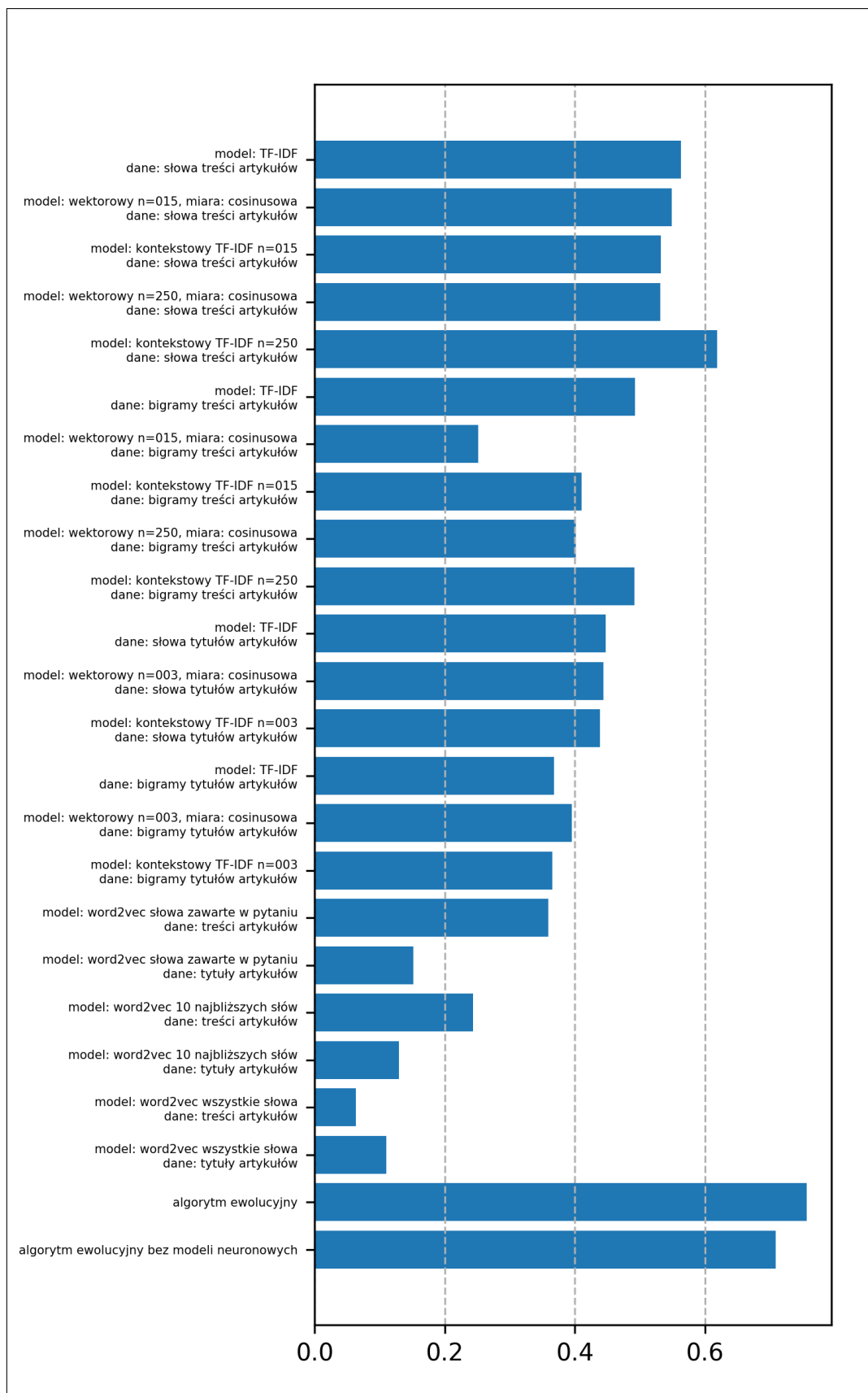
Rysunek 25: Zbiorne wyniki wybranych modeli dla miary p_1 .



Rysunek 26: Zbiorcze wyniki wybranych modeli dla miary p_{10} .



Rysunek 27: Zbiorcze wyniki wybranych modeli dla miary p_{100} .



Rysunek 28: Zbiorne wyniki wybranych modeli dla miary MRR.

6.2 Dalszy rozwój

Przedstawione w powyższej pracy rozwiązania mają dużo możliwości rozwoju. Wiele z parametrów modeli może zostać zmienionych. Zamiast miary TF-IDF, która została użyta jako podstawowa technika oceny relewantności artykułów można spróbować użyć innej, bardziej złożonej metody. Modele wektorowe używały bardziej złożonych metod do porównania wag, zamiast sumowania. Dało to nieznaczłą poprawę wyników. Jest to miejsce do przetestowania innych, bardziej złożonych metod porównywania wektorów i być może uzyskania jeszcze lepszych wyników. Sieci neuronowe użyte w powyższej pracy można znacznie rozbudować i spróbować lepiej dobrać dane uczące.

6.3 SQAD

Zbiór SQAD (ang. Stanford Question Answering Dataset)⁴¹, szczegółowo opisany w artykule [12], to zbiór ponad 100 000 artykułów, pytań do artykułów oraz odpowiedzi na pytania. W odróżnieniu od rozważanego w niniejszej pracy problemu, odpowiedzią w zbiorze SQAD jest pojedyncze słowo lub fraza. W wersji 2.0 zbiór SQAD został rozszerzony o brak odpowiedzi, tzn. dla wskazanego artykułu jest możliwe, że nie zawiera on odpowiedzi na dane pytanie.

Możliwe jest użycie zbioru SQAD do rozszerzenia opisanych wyżej modeli, tak aby oprócz zwrócenia poprawnego artykułu, model zwracał również konkretne słowo, frazę lub fragment, w którym jest odpowiedź na żądane pytanie. Dokładniejsza analiza problemu SQAD wykracza poza zakres poniższej pracy⁴², jednak połączenie modelu rozwiązującego problem odpowiadania na pytania zadawane Wikipedii oraz modelu bazującego na zbiorze SQAD mogłoby dać ciekawe rezultaty. Problem może być to, że SQAD dostępny jest w momencie publikacji pracy w języku angielskim, jednak opisane wyżej modele mogą być użyte również dla języka angielskiego.

6.4 Użycie innych języków

Wszystkie opisane powyżej modele zostały użyte dla języka polskiego, jednak nic nie stoi na przeszkodzie, aby zastosować opisane modele dla dowolnych innych języków. Po dostarczeniu korpusu pytań, artykułów oraz form bazowych (o ile takie w danym języku występują) wyniki powinny być podobne.

6.5 Wyszukiwarka

Przedstawione w powyższej pracy modele mogą zostać użyte jako wyszukiwarka artykułów Wikipedii. Zapytaniem może być fraza zadana przez użytkownika, natomiast odpowiedzią ranking artykułów, w którym na pierwszym miejscu znajduje się artykuł oceniony jako najbardziej związany z zapytaniem. Przykładowe zapytania⁴³

⁴¹Zbiór dostępny jest na stronie <https://rajpurkar.github.io/SQuAD-explorer/>.

⁴²Przykładowy model rozwiązujący problem odpowiadania na pytania na podstawie zbioru SQAD został szczegółowo opisany w pracy [9].

⁴³Zapytania zostały stworzone ręcznie przez autora pracy.

oraz ranking 10 najwyżżej ocenionych artykułów⁴⁴ znajdują się w tabelach 34, 36, 38, 40 oraz 42. Dla porównania, pod każdą tabelą z pytaniem znajduje się druga tabela z wynikami popularnej wyszukiwarki firmy Google, zawężonymi do artykułów Wikipedii. Dla pytania „*W którym roku wybuchło Powstanie Warszawskie?*”, dwa artykuły znajdujące się na miejscach 4 i 8 zawierają odpowiedź na pytanie. Obszerny artykuł pod tytułem „*Powstanie warszawskie*” który zawiera bardzo dokładny opis pytania jak i odpowiedź na nie, nie znalazł się nawet w pierwszej dziesiątce. Powodem niskiej pozycji (340) pomimo wielu słów kluczowych jest bardzo długa treść artykułu (16 425 słów). Dla pytania „*Jaki jest najwyższy szczyt Pienin?*” aż 6 artykułów z pierwszej dziesiątki zawiera poprawną odpowiedź na pytanie. Na uwagę zasługuje fakt, że jest to wynik lepszy niż wynik wyszukiwarki firmy Google, co widać w tabeli 38 oraz 39.

Tablica 34: Wyniki zapytania „*Ile trwa ciąża słonia?*”.⁴⁵

Pozycja	Artykuł	Czy odpowiedź na pytanie znajduje się w artykule?
1	Ciąża	nie
2	Słoń indyjski	tak
3	Ciąża człowieka	nie
4	Ciąża jajnikowa	nie
5	Mirunga północna ⁴⁶	tak
6	Góralkowce	nie
7	Ninio	nie
8	Tyreotoksykoza ciążowa	nie
9	Ciąża po terminie	nie
10	Słoń leśny	tak

Tablica 35: Wyniki zapytania „*Ile trwa ciąża słonia?*” w wyszukiwarce firmy Google.

Pozycja	Artykuł	Czy odpowiedź na pytanie znajduje się w artykule?
1	Słoń afrykański	nie
2	Słoń indyjski	tak
3	Ciąża	nie
4	Słoń leśny	tak
5	Mirunga	tak
6	Mirunga północna	tak
7	Mamut włochaty	nie
8	Lew afrykański	nie
9	Ssaki	nie
10	Hipopotam nilowy	nie

⁴⁴W celu stworzenia rankingu artykułów użyto modelu opartego o algorytm genetyczny.

⁴⁵Ciąża słonia trwa około 21 miesięcy.

⁴⁶Mirunga północna jest to słoń morski północny.

Tablica 36: Wyniki zapytania „*Jaki jest język urzędowy w Kanadzie?*”⁴⁷.

Pozycja	Artykuł	Czy odpowiedź na pytanie znajduje się w artykule?
1	Język urzędowy	nie
2	Rozprzestrzenienie języków urzędowych	tak
3	Języki urzędowe Indii	nie
4	Rozmieszczenie geograficzne języka hiszpańskiego	nie
5	BBC World Service	nie
6	Inuktitut	nie
7	Romanofonia	nie
8	Język chorwacki	nie
9	Języki urzędowe Rosji	nie
10	Najczęściej używane języki świata	nie

Tablica 37: Wyniki zapytania „*Jaki jest język urzędowy w Kanadzie?*” w wyszukiwarce firmy Google.

Pozycja	Artykuł	Czy odpowiedź na pytanie znajduje się w artykule?
1	Kanada	tak
2	Monarcha Kanady	nie
3	Justin Trudeau	nie
4	Ottawa	nie
5	Julie Payette	nie
6	Kanadyjczycy	nie
7	Kultura Kanady	nie
8	Quebec	nie
9	Języki urzędowe państw świata	tak
10	Ontario	nie

⁴⁷Języki urzędowe Kanady to angielski oraz francuski.

Tablica 38: Wyniki zapytania „*Jaki jest najwyższy szczyt Pienin?*”⁴⁸.

Pozycja	Artykuł	Czy odpowiedź na pytanie znajduje się w artykule?
1	Pieniny	tak
2	Pieniny właściwe	nie
3	Pieniński pas skałkowy	tak
4	Małe Pieniny	tak
5	Pieniny Spiskie	nie
6	Pieniński Park Narodowy (Słowacja)	tak
7	Wysokie Skałki	tak
8	Smerekowa	nie
9	Trzy Korony (szczyt)	nie
10	Korona Gór Polski	tak

Tablica 39: Wyniki zapytania „*Jaki jest najwyższy szczyt Pienin?*” w wyszukiwarce firmy Google.

Pozycja	Artykuł	Czy odpowiedź na pytanie znajduje się w artykule?
1	Pieniny	tak
2	Pieniny Właściwe	nie
3	Trzy Korony (szczyt)	nie
4	Sokolica (Pieniny)	nie
5	Facimiech (Pieniny)	nie
6	Wysokie Skałki	tak
7	Pieniński Park Narodowy	nie
8	Palenica (Pieniny)	nie
9	Masyw Trzech Koron	nie
10	Małe Pieniny	tak

⁴⁸Najwyższy szczyt Pienin to Wysoka, często mylony ze szczytem Trzech Koron.

Tablica 40: Wyniki zapytania „*W którym roku wybuchło Powstanie Warszawskie?*”⁴⁹.

Pozycja	Artykuł	Czy odpowiedź na pytanie znajduje się w artykule?
1	Kalendarium historii Irlandii	nie
2	1831	nie
3	1863	nie
4	Skutki Powstania Warszawskiego	tak
5	1648	nie
6	Powstanie Ostranicy	nie
7	Jesień Ludów	nie
8	1 sierpnia	tak
9	Stanisław Ciesielczuk	nie
10	Edward Warchałowski	nie

Tablica 41: Wyniki zapytania „*W którym roku wybuchło Powstanie Warszawskie?*” w wyszukiwarce firmy Google.

Pozycja	Artykuł	Czy odpowiedź na pytanie znajduje się w artykule?
1	Powstanie warszawskie	tak
2	Siły polskie w powstaniu warszawskim	tak
3	Godzina „W”	tak
4	Antoni Chruściel	nie
5	Zbrodnie niemieckie w powstaniu warszawskim	nie
6	Kalendarium powstania warszawskiego – 1 sierpnia	tak
7	Skutki powstania warszawskiego	nie
8	Powstanie w getcie warszawskim	nie
9	Powstanie warszawskie w dyplomacji	nie
10	Powstanie warszawskie w kulturze popularnej	nie

⁴⁹Powstanie warszawskie wybuchło 1 sierpnia 1944.

Tablica 42: Wyniki zapytania „*Kto był pierwszym królem Polski?*”⁵⁰.

Pozycja	Artykuł	Czy odpowiedź na pytanie znajduje się w artykule?
1	Polscy pretendenci i samozwańcy	nie
2	Królestwo Polskie (kongresowe)	nie
3	Król Polski	tak
4	Król Lear	nie
5	Władcy polski	tak
6	Najświętsza Maryja Panna Królowa Polski	nie
7	Ludwik Węgierski	nie
8	Kaliningrad	nie
9	4 kwietnia	nie
10	Polskie królowe	tak

Tablica 43: Wyniki zapytania „*Kto był pierwszym królem Polski?*” w wyszukiwarce firmy Google.

Pozycja	Artykuł	Czy odpowiedź na pytanie znajduje się w artykule?
1	Król Polski	tak
2	Władcy Polski	tak
3	Koronacja na króla Polski	tak
4	Polskie królowe	tak
5	Wratysław II	nie
6	Bolesław I Chrobry	tak
7	Król	nie
8	Drzewo genealogiczne królów Polski	nie
9	1025	nie
10	Piastowie	nie

6.6 Wybrane problemy natury technicznej

Podczas pracy nad powyższymi modelami napotkano kilka problemów natury technicznej. Bardzo szybko okazało się, że liczba danych jest ogromna, a zasoby domowej stacji roboczej są mocno ograniczone. Danych jest na tyle dużo, że podczas implementacji wielokrotnie zabrakło pamięci RAM, co w najlepszym przypadku skutkowało zawieszeniem się programu. Podczas implementacji bardzo dużą uwagę trzeba było zwrócić na oszczędne użycie pamięci RAM oraz zapisy/odczyty dysku twardego.

Po implementacji kilku pierwszych modeli okazało się, że liczenie poszczególnych modeli za pomocą aplikacji jednowątkowej będzie trwało bardzo długo. Postanowiono przepisać aplikację tak, aby korzystała z wielu wątków jednocześnie. Dużym wyzwaniem okazało się takie dobranie i przetworzenie danych, aby były wystarczające do obliczeń, ale jednocześnie były w trybie tylko do odczytu, aby uniknąć czasochłonnej i zużywającej cenny czas procesora synchronizacji.

⁵⁰Pierwszym królem Polski był Bolesław Chrobry.

W trakcie wielogodzinnego testowania poszczególnych modeli, wielokrotnie okazywało się, że w implementacji modelu był błąd i żmudne, wielogodzinne obliczenia należy powtórzyć od nowa.

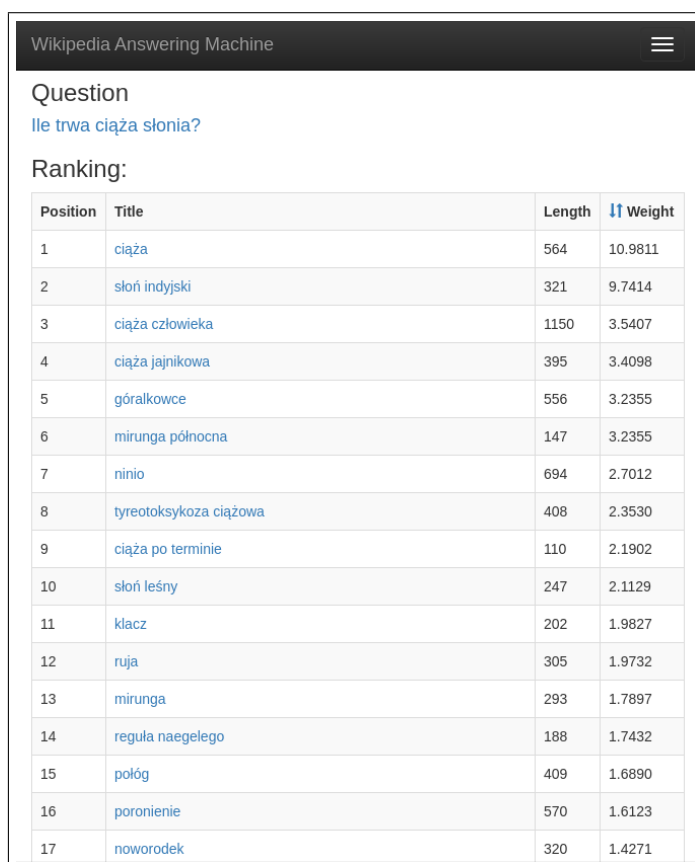
Z racji dużej ilości czasu potrzebnego do obliczeń, często w ich trakcie był problem z dostępnością maszyny z różnych powodów (restart maszyny, przypadkowe wyłączenie, brak prądu, wyłączenie przez innego użytkownika, brak dostępnych zasobów). Po kilku utratach wielogodzinnych obliczeń, postanowiłem dodać szybkie przerywanie i wznowianie obliczeń w dowolnym momencie.

Dodatek A

Szczegóły techniczne

A.1 Implementacja

Wszystkie opisane wyżej metody zostały napisane i przetestowane w języku Python. W celu łatwego zarządzania danymi postanowiono skorzystać z gotowej bazy danych biblioteki Django. Jest to popularne narzędzie do tworzenia stron internetowych, ale moduł bazy danych oraz wewnętrznych modeli świetnie się sprawdził w odwzorowaniu zależności między poszczególnymi elementami systemu. Szczegółowy schemat bazy danych znajduje się na rysunku 30. Zrzut ekranu prezentujący działanie systemu odpowiadania na pytania znajduje się na rysunku 29.



The screenshot shows a web interface titled "Wikipedia Answering Machine". It displays a question "Ile trwa ciąża słonia?" and a ranked list of 17 answers. Each answer entry includes its position, title, length, and weight. The weights are sorted in descending order.

Position	Title	Length	Weight
1	ciąża	564	10.9811
2	słoń indyjski	321	9.7414
3	ciąża człowieka	1150	3.5407
4	ciąża jajnikowa	395	3.4098
5	góralkowce	556	3.2355
6	mirunga północna	147	3.2355
7	ninio	694	2.7012
8	tyreotoksykoza ciążowa	408	2.3530
9	ciąża po terminie	110	2.1902
10	słoń leśny	247	2.1129
11	klacz	202	1.9827
12	ruja	305	1.9732
13	mirunga	293	1.7897
14	regula naegelego	188	1.7432
15	połóg	409	1.6890
16	poronienie	570	1.6123
17	noworodek	320	1.4271

Rysunek 29: Zrzut ekranu prezentujący działanie systemu odpowiadania na pytania zadawane Wikipedii.

A.2 Dane wejściowe

Zasoby polskiej Wikipedii stanowiące podstawę niniejszej pracy są znacznej wielkości. Sam plik zawierający artykuły zajmuje około 8 GB. Wielkość danych pokazuje tabela 44.

Tablica 44: Liczba poszczególnych danych.

dane	liczba danych
słowa	8 450 424
artykuły	1 651 186
wystąpienia słów w artykułach	290 054 355
pytania	10 668
wystąpienia słów w pytaniach	121 186
odpowiedzi na pytania	12 094

A.3 Obliczenia

Tak duża liczba danych wymaga znacznych zasobów mocy obliczeniowej.

Domowa stacja robocza

Początkowe testy były przeprowadzone na domowym komputerze wyposażonym w 8-rdzeniowy procesor AMD Ryzen 5 2400G, 32 GB pamięci RAM oraz 256 GB dysk SSD. Dostępna moc pozwoliła na preprocessing surowych danych do wygodnej bazy danych w około 12 godzin. Policzenie wyników modeli opisanych w rozdziałach 3.2, 3.3 i 5.1 zajęło około 36 godzin.

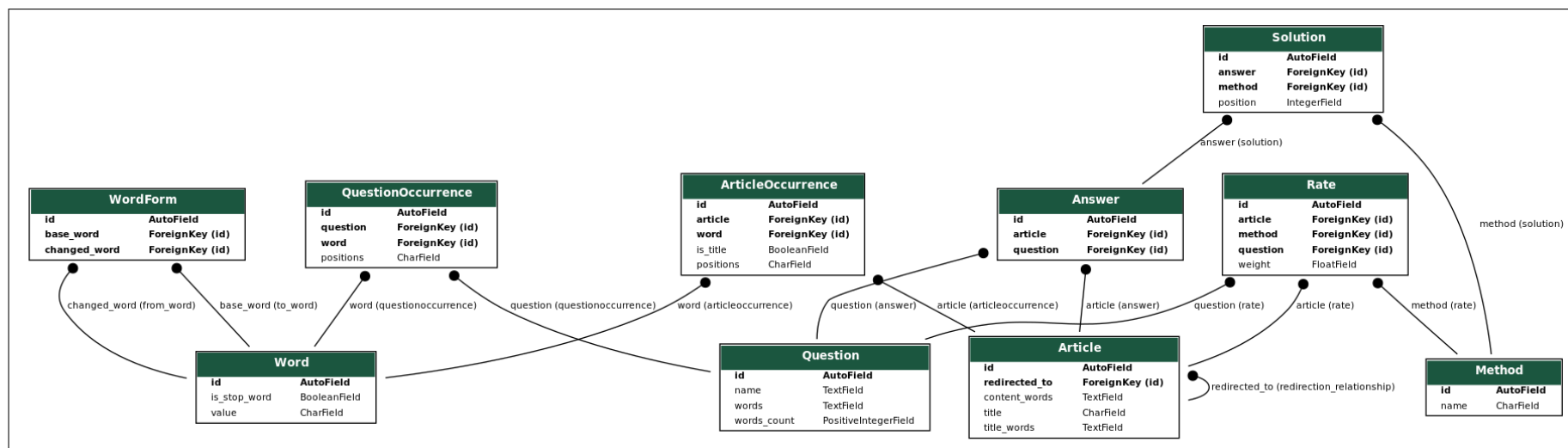
Compute Engine - Google Cloud

Zaczynając od modelu opisanego w rozdziale 3.4 moc obliczeniowa domowej stacji roboczej przestała wystarczać. Postanowiono skorzystać z usługi **Compute Engine** od firmy **Google**. Uzyskano tymczasowy, bezpłatny dostęp do maszyny wirtualnej wyposażonej w 64 rdzenie, 256 GB pamięci RAM oraz 100 GB dysk SSD. Za jej pomocą udało się przeprowadzić wymagane obliczenia do reszty modeli w czasie około 96 godzin.

A.4 Kod źródłowy

Źródła wszystkich programów pozwalających odtworzyć wyniki powyższej pracy znajdują się na repozytorium⁵¹. Projekt został opublikowany z możliwością dalszego rozwoju, do czego niniejszym autor zachęca.

⁵¹Repozytorium dostępne pod adresem <https://github.com/shajen/Wikipedia-Answering-Machine>.



Rysunek 30: Schemat bazy danych.

Bibliografia

- [1] Cormen T.: *Introduction to algorithms*, MIT Press, Cambridge, Mass, 2009.
- [2] Goldberg D.: *Genetic algorithms in search, optimization, and machine learning*, Addison-Wesley Publishing Company, Reading, Mass, 1989.
- [3] Goodfellow I., Bengio Y., Courville A.: *Deep Learning*, MIT Press, 2016.
- [4] Grama A.: *Introduction to parallel computing*, Addison-Wesley, Harlow, England New York, 2003.
- [5] Hagan M.: *Neural network design*, 2014.
- [6] Iyyer M., Manjunatha V., Boyd-Graber J., Daumé III H.: *Deep unordered composition rivals syntactic methods for text classification*, <https://www.aclweb.org/anthology/P15-1162>, 2015.
- [7] Jacovi A., Sar Shalom O., Goldberg Y.: *Understanding convolutional neural networks for text classification*, <https://www.aclweb.org/anthology/W18-5408>, 2018.
- [8] Jurafsky D.: *Speech and language processing : an introduction to natural language processing, computational linguistics, and speech recognition*, Pearson Prentice Hall, Upper Saddle River, N.J, 2009.
- [9] Lan Z., Chen M., Goodman S., Gimpel K., Sharma P., Soricut R.: *Albert: A lite bert for self-supervised learning of language representations*, in *International Conference on Learning Representations*, 2019.
- [10] Manning C.: *Introduction to information retrieval*, Cambridge University Press, New York, 2008.
- [11] Mikolov T., Chen K., Corrado G., Dean J.: *Efficient estimation of word representations in vector space*, <https://arxiv.org/abs/1301.3781>, 2013.
- [12] Rajpurkar P., Jia R., Liang P.: *Know what you don't know: Unanswerable questions for SQuAD*, <https://www.aclweb.org/anthology/P18-2124>, 2018.
- [13] Turing A.M.: *Computing machinery and intelligence*, <http://cogprints.org/499/>, 1950.
- [14] Voorhees E.M.: *Overview of the trec-9 question answering track*, in *In Proceedings of the Ninth Text REtrieval Conference (TREC-9*, pages 71–80, 2001.