POLITECHNIKA WARSZAWSKA
WYDZIAŁ MATEMATYKI
I NAUK INFORMACYJNYCH

PRACA DYPLOMOWA MAGISTERSKA

INFORMATYKA

Rekomendacje artykułów opisujących produkty w serwisach e-commerce

Content-based recommendations in e-commerce services

Autor:

Łukasz Dragan

Promotor: dr inż. Anna Wróblewska

Warszawa, wrzesień 2017

.....

podpis promotora

.....

podpis autora

Streszczenie

Tematyka niniejszej pracy skupia się wokół zagadnień określania podobieństwa semantycznego pomiędzy dokumentami tekstowymi i rekomendacji dokumentów podobnych do danego. Szczegółowy problem pochodzi z internetowego serwisu aukcyjnego Allegro, który posiada dział artykułów opisujących produkty dostępne w serwisie. W dziale tym funkcjonuje system rekomendacji podobnych artykułów tekstowych w oparciu o ich treść. Celem pracy jest zbadanie możliwości usprawnienia działania istniejącego systemu rekomandacji wykorzystując metody semantycznej analizy tekstu.

W niniejszej pracy adaptuję dostępne metody określania podobieństwa pomiędzy dokumentami tekstowymi do powyższego problemu, wprowadzam miary umożliwiające ocenę działania tych metod oraz dokonuję analizy możliwości ich wykorzystania w rzeczywistym systemie.

Abstract

The subject of this paper focuses on the issues of determining the semantic similarity between text documents and the recommendation of documents similar to a given. A detailed problem comes from the Allegro on-line auction site, which has a section of articles describing the products available on the site. This section offers a recommendation system for similar textual articles based on their content. The aim of this paper is to investigate the possibility of improving the existing recommendation system using semantic text analysis methods.

In this paper, I adapt the available methods for determining the similarity between text documents to the above problem, I introduce measures to evaluate the performance of these methods and analyze the possibilities of using them in the real system.

Spis treści

1 Wstęp **4**

1.1 Rekomendacje artykułów tekstowych w Allegro 5

1.2 Struktura pracy 7

1.3 Uwagi 8

2 Przegląd wiedzy z zakresu tematyki pracy **9**

2.1 Systemy rekomendacji 10

2.1.1 Filtrowanie kolaboratywne (collaborative filtering) 10

2.1.2 Filtrowanie oparte na treści (content-based filtering) 11

2.2 Techniki przetwarzania języka naturalnego 11

2.2.1 Bag-of-words 12

2.2.2 Term frequency - inverted document frequency 12

2.2.3 Distributional semantics 13

2.2.4 Latent Semantic Analysis 13

2.2.5 Latent Dirichlet Allocation 14

2.2.6 Word embeddings 14

2.2.7 Podejścia deep learningowe 16

2.2.8 Word2vec 17

2.2.9 FastText 20

2.2.10 GloVe 20

2.2.11 Odległość między dokumentami 21

3 Dane **24**

3.1 Opis danych 24

3.1.1 Treść artykułu 24

3.1.2 Kategoria 25

3.1.3 Słowa kluczowe 25

3.2 Wstępne przetwarzanie danych 26

3.3 Opis danych po wstępnym przetwarzaniu 27

4 Metody ewaluacji **30**

4.1 Miara 1: Dystans oparty na metadanych 31

4.1.1 Kategorie 31

4.1.2 Słowa kluczowe 33

4.2 Miara 2: Ocena przez użytkowników offline 33

4.3 Miara 3: Historyczna aktywność użytkowników serwisu 33

4.3.1 nDCG 34

4.3.2 Adaptacja metody nDCG 34

5 Opis i wyniki badań **36**

5.1 Przygotowanie eksperymentów 36

5.1.1 Modele Word2Vec 37

5.1.2 Model LDA 38

5.2 Wyniki badań 38

5.2.1 Efektywność czasowa 39

6 Podsumowanie **40**

A Technologie i narzędzie **41**

Rozdział 1

Wstęp

Systemy rekomendacji są powszechnym elementem wielu serwisów internetowych. Sprawdzają się na takich polach, jak polecanie produktów w sklepie czy rekomendacje ofert pracy. Dają użytkownikowi poczucie indywidualnego traktowania przez serwis internetowy dopasowujący niejako zawartość swoich stron do konkretnego użytkownika. Pozwala to użytkownikowi na bardziej efektywne korzystanie z serwisu oraz może prowadzić do większego zaangażowania ze strony użytkownika i przywiązania do serwisu. Systemy rekomendacji dają obopulną korzyść zarówno użytkownikowi jak i właścicielowi serwisu internetowego.

Celem niniejszej pracy magisterskiej jest analiza możliwości usprawnienia istniejacego systemu rekomendacji o oparciu o adaptację istniejących metod wyszukiwania semantycznego podobieństwa pomiędzy dokumentami tekstowymi. Rzeczony system rekomendacji istnieje w internetowym serwisie e-commerce Allegro w dziale artykułów tekstowych o tematyce związanej z produktami dostępnymi za pośrednictwem serwisu. System ma na celu zarekomendowanie użytkownikowi artykułów o tematyce podobnej do tego, który znajduje się na stornie aktualnie odwiedzanej przez uzytkownika.

W swojej pracy badam możliwość użycia istniejących metod semantycznej analizy tekstu w odniesieniu do opisanego problemu. Badane metody to: Latent Semantic Analysis, Latent Dirichlet Allocation, Word2vec, GloVe oraz FastText.

Jakość działania tych metod porównuję poprzez samodzielnie opracowane metody ewaluacji.

Podczas prowadzenia badań stworzyłem szereg skryptów przetwarzających dane i wykorzystujących implementacje opisywanych w tej pracy metod. Opis użytych narzędzi programistycznych i bibliotek zawarłem w dodatku A do niniejszej pracy.

1.1 Rekomendacje artykułów tekstowych w Allegro

Allegro jest największą[**?**] działającą na rynku polskim platformą aukcyjną on-line. Posiada ponad 20 mln zarejestrowanych klientów. Każdego dnia na Allegro sprzedaje się ponad 870 tysięcy przedmiotów. Zatrudnia 1300 pracowników.[3] Serwis umożliwia użytkownikom wystawianie na sprzedaż oraz kupno przedmiotów poprzez mechanizm licytacji lub natychmiastowego zakupu.

Oprócz głównej części serwisu odpowiedzialnej za transakcje Allegro posiada dział zajmujący się publikacją artykułów opisujących produkty wystawiane za pośrednictwem serwisu. Ma to na celu pomoc użytkownikom przy wyborze interesującego ich produktu.

Po to, aby zachęcić użytkowników do zapoznania się z treścią kolejnych artykułów, zastosowany został tu system rekomendacji przyporządkowujący danemu artykułowi listę powiązanych artykułów. Kryterium mówiącym, czy artykuły są powiązane jest tutaj jedynie treść samych artykułów, a nie wcześciejsze zachowanie użytkownika.

Rysunek 1.1: Widok strony internetowej zawierającej jeden z artykułów serwisu Allegro. [14]

Od serwisu Allegro otrzymałem zserializowaną kopię 20000 artykułów dostępnych na stronach serwisu. Pojedynczy artykuł składa się z głównej zawartości tekstowej oraz metadanych. W celu otrzymania wszelkich danych od firmy Allegro wynagane było, abym podpisał umowę, w której zobowiązuję się do nieujawniania żadnych danych, które otrzymałem. Stąd opisy danych, na których pracuję, zawarte w tej pracy nie wnikają w ich szczegóły i nieodbiegają od informacji

publicznie dostępnych za pośrednictwem strony pod adresem `https://allegro.pl/artykuly`.

Aktualnie w rzeczonym dziale serwisu Allegro istnieje system rekomendacyjny, który opiera się o wyszukiwanie podobnych artykułów tekstowych za pomocą silnika Elasticsearch[**?**]. Metoda ta wykorzystuje słowa kluczowe przypisane do każdego artykułu przez autora. W swojej pracy staram się porównać wyniki działania dotychczasowej metody z metodami semantycznej analizy tekstu, które potrafią wykryć podobieństwo pomiędzy artykułami bazując jedynie na ich treści, bez potrzeby dołączania żadnych metadanych. Pomyślna próba zastosowania metod semantycznych pozwoliłaby na dokładniejsze dopasowanie podobnych artykułów w oparciu być może o pewne ukryte cechy semantyczne nieosiągalne dla silnika wyszukiwania tekstowego, jakim jest Elasticsearch. Bardziej szczegółowego opisu silnika Elasticsearch dokonuję w kolejnym rozdziale.

1.2 Struktura pracy

W rozdziale 2 wprowadzam do zagadnienia rekomendacji oraz dokonuję przeglądu metod semantycznej analizy tekstu, które mogą zostać zastosowane w celu określenia podobieństwa pomiędzy dokumentami tekstowymi.

Następnie w rozdziale 3 dokonuję opisu konkretnego problemu, jakim jest generacja rekomendacji artykułów tekstowych w serwisie Allegro. Opisuję dane otrzymane z serwisu oraz kolejne etapy ich wstępnego przetwarzania, aby nadawały się do zaaplikowania do nich wybranych metod.

Dalej, w rozdziale 4 opisuję stworzone i zastosowane później metody ewaluacji wyników.

Następnie w rozdziale 5 dokonuję opisu testów: jakie metody i w jako sposób testuję.

W rozdziale 6 opisuję wyniki przeprowadzonych eksperymentów.

Ostatecznie w rozdziale 7 dokonuję podsumowania przeprowadzonych badań i rozważam kierunki dalszych prac w tej dziedzinie.

Załącznik A zawiera opis narzędzi programistycznych i bibliotek wykorzystanych przeze mnie podczas prowadzenia badań.

1.3 Uwagi

W celu uniknięcia nieporozumień należy podkreślić różnicę pomiędzy znaczeniami słowa „artykuł", które może oznaczać zarówno tekst publicystyczny, literacki lub naukowy jak i rzecz, która jest przedmiotem handlu.[2] W niniejszej pracy skupiam się na rekomendacjach artykułów tekstowych, stąd używam pierwszego znaczenia (chyba, że inne znaczenie jest wyraźnie zaznaczone).

Rozdział 2

Przegląd wiedzy z zakresu tematyki pracy

W swojej pracy dokonuję adaptacji metod przetwarzania języka naturalnego na potrzeby generowania rekomendacji artykułów tekstowych w oparciu o ich treść. W niniejszym rozdziale dokonuję przeglądu znanych metod z obszaru tematyki pracy dyplomowej, skupiając się szczególnie na nowo powstałych metodach wektorowej reprezentacji słów, które cieszą się obecnie dużym zainteresowaniem środowisk naukowych oraz firm komercyjnych.

Dokonuję krótkiego wprowadzenia do zagadnienia generowania rekomendacji, którego głęboka analiza nie jest konieczna z punktu widzenia niniejszej pracy. Następnie wykonuję chronologiczny przegląd metod ciągłej reprezentacji słów zaczynając od trywialnych metod zliczania słów (bag-of-words, tf-idf), przechodząc przez metody wykorzystujące koncepcję tematów (Latent Semantic Analysis, Latent Dirichlet Allocation) i kończąc na głośnych ostatnio metodach osadzania słów w przestrzeni wektorowej (Word2vec, GloVe, FastText). Przy zarysie historycznym opieram się w dużej mierze na artykule[25].

2.1 Systemy rekomendacji

Systemy rekomendacji to narzędzia i techniki mające na celu zasugerować użytkownikowi przedmioty. Sugestie te odnoszą się do różnych procesów podejmowania decyzji takich jak np. które artykuły kupić, jakiej muzyki słuchać czy też które wiadomości czytać. „Przedmiot" jest tutaj ogólnym pojęciem oznaczającym coś, co system poleca użytkownikowi. [1]

Przy wciąż wzrastającej ilości danych użytkownicy serwisów internetowych często nie są w stanie dotrzeć do informacji, która ich interesuje. Jest to pole do rozwoju zautomatyzowanych systemów rekomendacyjnych polecających użytkownikom treści, które mogą ich zainteresować. Działalność tekiego systemu daje zysk zarówno użytkownikowi, pozwalając mu dotrzeć do informacji, której mógłby samodzielnie nie odszukać, albo wręcz nie wiedzieć, iż taka informacja istnieje, jak i właścicielowi serwisu internetowego, któremu zależy, by przyciągnąć do siebie użytkowników, aby ci w jak największym stopniu korzystali z ich usług.

Sposoby działania systemów rekomendacji można podzielić na różne warianty, spośród których wyodrębnić można dwa najszerzej używane. Są to: filtrowanie kolaboratywne (collaborative filtering) i filtrowanie oparte na treści (content-based filtering).

2.1.1 Filtrowanie kolaboratywne (collaborative filtering)

Technika ta opiera się na spostrzeżeniu, iż użytkownicy o podobnych preferencjach zachowują się podobnie. Stąd jeżeli użytkownik zachowuje się podobnie do zaobserwowanej wcześniej grupy użytkowników, można przewidzieć jego preferencje na podstawie zachowań ów grupy. Istotną zaletą tej metody jest fakt, iż nie zależy ona od dziedziny, w której ulokowany jest system rekomendacji (w przeciwieństwie do rekomendacji opartych na treści), a jedynie od zachowań użytkowników.

2.1.2 Filtrowanie oparte na treści (content-based filtering)

W technice tej przedmioty polecane użytkownikowi zależą od innych przedmiotów, na temat których stwierdzono, że użytkownik się nimi interesuje. Mogą się one opierać np. na podobieństwie przedmiotów: jeżeli użytkownik „lubi" przedmiot A, który jest podobny do przedmiotu „B" to można spodziewać się, że również przedmiot B zainteresuje użytkownika. Technika ta jest mocno zależna od dziedziny rekomendowanych przedmiotów, gdyż wymaga wprowadzenia pewnej miary podobieństwa między nimi. Stąd jest trudniejsza do zastosowania, ale daje też możliwości nieosiągalne dla filtrowania kolaboratywnego.

Celem niniejszej pracy jest zbadanie metod sugerujących użytkownikowi artykuły podobne do aktualnie odwiedzanego, co wprost wiąże się z metodami używanymi w technice filtrowania opartego na treści.

2.2 Silnik Elasticsearch

Obecnie wykorzystywana przez Allegro metoda generowania rekomendacji artykułów opiera się o zapytanie do usługi Elasticsearch wykorzystujące słowa kluczowe dołączone do artykułów. Elasticsearch jest popularnym silnikiem wyszukiwania tekstu opartym o indeks Lucene[**?**]. Działa w architekturze rozproszonej a komunikacja z nim następuje poprzez protokół HTTP i format JSON. Umożliwia on efektywne przechowywanie dokumentów tekstowych oraz efektywne ich wyszukiwanie.

Apache Lucene jest biblioteką napisaną w języku Java służącą do wyszukiwania tekstu, która w tym celu wykorzystuje mechanizm odwróconego indeksu. Zasada działania biblioteki polega na stworzeniu słownika ze wszystkich (odpowiednio wstępnie przetworzonych) słów dokumentów przeznaczonych do wyszukiwania. Następnie na bazie ów słownika tworzony jest odwrócony indeks: każdemu ze słów przypisywana jest lista dokumentów, które zawierają to słowo. Pozwala to przyspieszyć proces wyszukiwania, gdyż w poszukiwaniu pojedynczego słowa biblioteka nie przeszukuje całego zbioru dokumentów, a jedynie słownik, który na

ogół jest wielokrotnie krótszy.

Zaletą silnika Elasticsearch są jego wydajność, skalowalność, niezawodność i prostota użytkowania, co przekłada się na jego dużą popularność wśród np. serwisów internetowych[**?**].

Wadą metody jest to, że ogranicza się ona do wyszukiwania tekstowego pomijając aspekt semantyczny. Stwarza to trudności przy wyszukiwaniu synonimów lub homonimów.

2.3 Techniki przetwarzania języka naturalnego

Oparcie rekomendacji jedynie na treści artykułu wymaga zagłębienia się w tematykę analizy i przetwarzania języka naturalnego, wszak właśnie w języku naturalnym, zrozumiałym dla człowieka (polskim) pisane są owe artykuły. Język naturalny z powodu swojego niskiego stopnia sformalizowania nie jest niestety wprost zrozumiały dla maszyn. W związku z tym koniecznym staje się tu użycie technik przetwarzania języka naturalnego (natural language processing), które to pozwalają wyodrębnić z tekstu pewne cechy, na bazie których maszyna obliczeniowa przy pomocy pewnych algorytmów jest w stanie określić podobieństwo pomiędzy dokumentami. W poniższych paragrafach dokonuję przeglądu technik matematycznej reprezentacji dokumentów pisanych w języku naturalnym. Warto wspomnieć, iż dziedzina ta bardzo dynamicznie się rozwija a część z opisywanych metod zostało stworzonych na przestrzeni ostatnich kilku lat, czy wręcz miesięcy.

W celu formalizacji w dalszych opisach stosowanych metod stosuję następujące oznaczenia:

- Korpus C: zbiór dokumentów d_i,
- Dokument d: skończony ciąg zdań s_i,
- Słowo w: skończony ciąg znaków c_i,
- Słownik zbudowany na korpusie C: $V = \{w \mid \exists_{d \in C}\, w \in d\}$.

2.3.1 Bag-of-words

Bag-of-words (worek słów)[20] jest jedną z pierwszych koncepcji reprezentacji tekstu jako zbioru zawartych w nim słów w postaci wektorów. Metoda nie zachowuje kolejności słów w tekście, lecz liczbę ich wystąpień. Istotną zaletą reprezentacji wektorowej dokumentów jest możliwość zdefiniowania miary odległości pomiędzy dokumentami (np. miara kosunusowa opisana później) odzwierciedlającej ich podobieństwo. Technikę tę można opisać jako przekształcenie z korpusu w przestrzeń wektorów $bow : C \to \mathbb{R}^n$ gdzie:
C: korpus
$m = |C|$: liczba dokumentów w korpusie C
V: słownik zbudowany na C
$n = |V|$: liczba słów w V
$v_i \in \mathbb{R}^n$, gdzie $i \in 1, 2, ..., n$ wektor reprezentujący dokument $d_i \in C$
v_{ij}, gdzie $j \in 1, 2, ..., m$: liczba wystąpień w dokumencie $d_i \in C$ słowa $w_j \in V$

Technika ta jest stosunkowo prosta, lecz jej wadą jest traktowanie każdego słowa z jednakową wagą. Pewne słowa (np. „i", „lub", „o") występują bardzo często, lecz ich wkład w znaczenie całego dokumentu jest marginalny. Stąd powstały bardziej zaawansowane techniki uwzględniające istotność słów dla znaczenia całego dokumentu. Mimo to metoda BOW jest często wykorzystywana w bardziej zawansowanych technikach NLP.

2.3.2 Term frequency - inverted document frequency

TF-IDF[**?**] (ważenie częstością termów - odwrotna częstość w dokumentach) jest metodą reprezentacji tekstu jako zbioru słów przy jednoczesnym uwzględnieniu wagi słów, która zależy od częstości występowania słowa w korpusie. Oznaczenia formalne takie same tak w przypdku BOW. $v_{ij} = tfidf_{ij} = tf_{ij} * idf_i$, gdzie:
$tf_{ij} = \frac{n_{ij}}{\sum_k n_{kj}}$, „term frequency" to liczba wystąpień słowa w_i w dokumencie d_j podzielona przez liczbę słów dokumentu d_j,

$idf_i = log\frac{|D|}{|d:w_i \in d|}$, „inversed document frequecy" to liczba dokumentów w korpusie podzielona przez liczbę dokumentów zawierających przynajmniej jedno wystąpienie słowa w_i,

Dokumenty reprezentowane są tu jako wektory, składające się z wag słów występujących w każdym z nich. TF-IDF przechowuje informację o częstotliwości występowania słów biorąc przy tym pod uwagę istotność znaczenia słowa lokalnego w stosunku do jego znaczenia w kontekście całego zbioru dokumentów. W tej technice słowa występujące rzadko są premiowane względem słów pospolitych. Wadą metody tej i poprzedniej jest postać wektorów reprezentujących słowa: są to na ogół rzadkie wektory dużej wymiarowości.

2.3.3 Distributional semantics

Kolejne, bardziej zaawansowane, omawiane tu metody opierają się na tzw. „distributional hypothesis" - hipotezie zakładającej, że słowa występujące w tym samym kontekście niosą ze sobą podobne znaczenie[20][21]. Sprzyja to zastosowaniu metod algebry liniowej jako narzędzia obliczeniowego oraz sposobu reprezentacji tekstu. Podstawowe podejście polega na zgromadzeniu informacji o rozkładzie słów w dokumentach w postaci wielowymiarowych wektorów a następnie wyodrębnieniu podobieństw pomiędzy tymi wektorami, które świadczyłyby o pewnych powiązaniach między reprezentowanymi słowami.

2.3.4 Latent Semantic Analysis

Analiza rozkładu słów w dokumentach tekstowych pozwala na wyodrębnienie podobieństw między słowami pod kątem: ich znaczenia (podobieństwo tematu słowa), ich osadzenia w stosunku do innych typów słów czy też ich struktury wewnętrznej. Dwie istotne metody: Latent Semantic Analysis[22] oraz Latent Dirichlet Allocation[23] zakładają istnienie abstrakcyjnych niejawnych (latent) tematów, do których można przydzielić słowa wchodzące w skład korpusu.

LSA (1990) [22], znane również jako Latent Semantic Indexing (LSI) dokonuje

transformacji każdego dokumentu w wektor dł. $|V|$ posiadający na i-tym miejscu wagę TF-IDF i-tego słowa ze słownika. W ten sposób tworzona jest rzadka macierz: kolumny reprezentują dokumenty a wiersze reprezentują unikalne słowa. W celu identyfikacji istotnych cech tej macierzy dokonuje się rozkładu według wartości osobliwych (Singular Value Decomposition[**?**], SVD), który jest techniką redukcji wymiarowości. Celem użycia SVD jest redukcja liczby wierszy macierzy dla wydajniejszych dalszych obliczeń numerycznych oraz pozbycie się szumów, utrzymując jednocześnie podobieństwa pomiędzy kolumnami. Ostatecznie uzyskuje się macierz przynależności tematów do dokumentów, gdzie wiersze odpowiadające tematom można interpretować jako kombinacje pierwotnych wierszy-słów o podobnym znaczeniu. Np. $\{(samochod), (ciagnik), (jezdnia)\} \rightarrow \{(1.3452 * samochod + 0.2828 * ciagnik + 0.3 * jezdnia)\}$. Wymiar uzyskiwanej macierzy jest ustalany za pomocą hiperparamtru, który oznacza liczbę tematów. Używając uzyskanej macierzy, podobieństwo pomiędzy kolumnami-dokumentami obliczane jest wykorzystując odległość kosinusową (opisaną później w tym rodziale). Metoda LSA łagodzi problem synonimów poprzez scalanie podobnych słów w jeden temat. Niweluje również problem homonimów, włączając je częściowo w skład różnych tematów. Niemniej jednak poprzez arbitralne ustalanie hiperparametru odpowiedzialnego za liczbę tematów część semantycznie odrębnych tematów może zostać wchłonięta przez inne lub też rozbicie na tematy może być zbyt „drobne" nie wykorzystując w pełni semantycznych powiązań.

2.3.5 Latent Dirichlet Allocation

LDA jest techniką automatycznego wykrywania tematów zawartych w dokumentach.

LDA reprezentuje dokumenty jako mieszanki tematów, które z kolei składają sie z mieszanki słów całego korpusu.

Wybierz mieszankę tematów dla dokumentu (zgodnie z rozkładem Dirichleta dla ustalonej liczby K tematów)

Wybierz temat zgodnie z rozkładem wielomianowym.

Użyj tematu do wybrania słowa zgodnie z rozkładem wielomianowym.

LDA iteruje po dokumentach starając się znaleźć mieszankę tematów, które z największym prawdopodobieństwem wygenerowały zbiór słów dokumentu.

Learning

So now suppose you have a set of documents. You've chosen some fixed number of K topics to discover, and want to use LDA to learn the topic representation of each document and the words associated to each topic. How do you do this? One way (known as collapsed Gibbs sampling*) is the following:

Go through each document, and randomly assign each word in the document to one of the K topics. Notice that this random assignment already gives you both topic representations of all the documents and word distributions of all the topics (albeit not very good ones). So to improve on them, for each document d... ....Go through each word w in d... .....And for each topic t, compute two things: 1) p(topic t | document d) = the proportion of words in document d that are currently assigned to topic t, and 2) p(word w | topic t) = the proportion of assignments to topic t over all documents that come from this word w. Reassign w a new topic, where you choose topic t with probability p(topic t | document d) * p(word w | topic t) (according to our generative model, this is essentially the probability that topic t generated word w, so it makes sense that we resample the current word's topic with this probability). (Also, I'm glossing over a couple of things here, such as the use of priors/pseudocounts in these probabilities.) .....In other words, in this step, we're assuming that all topic assignments except for the current word in question are correct, and then updating the assignment of the current word using our model of how documents are generated. After repeating the previous step a large number of times, you'll eventually reach a roughly steady state where your assignments are pretty good. So use these assignments to estimate the topic mixtures of each document (by counting the proportion of words assigned to each topic within that document) and the words associated to each topic (by counting the proportion of words assigned to each topic overall).

Jego ideą jest reprezentacja tekstów ja- ko mieszanki tematów, definiowanych

jako rozkłady prawdopodobieństwa na zbiorze słów. Z jednej strony jest to pewna metoda redukcji wymiaru danych, natomiast z drugiej strony LDA jest narzędziem, które dostarcza użytecznej reprezentacji danych wraz z często cenną interpretacją.

Model LDA jest przykładem probabilistycznego modelu graficznego, a konkretniej sieci bay- esowskiej, który został zaproponowany do modelowania danych tekstowych w [5]. Ideą LDA jest reprezentacja tekstu jako mieszanki tematów, definiowanych jako rozkłady prawdopo- dobieństwa na zbiorze słów. W modelu zakłada się, że każde słowo w dokumencie pochodzi z pewnego tematu i ostatecznie staramy się opisać dokument jako procentowy rozkład zawartych w nim tematów. LDA jest metodą uczenia bez nadzoru i można ją traktować jako pewien sposób redukcji wymiaru, gdyż staramy się przedstawić zbiór tekstów, przy pomo- cy pewnej (mniejszej niż liczba tekstów) liczby tematów. Ponadto z praktycznego punktu widzenia często ważna jest również interpretacja wyestymowanych tematów.

Przedstawimy ideę modelu LDA na przykładzie. Załóżmy, że naszymi dokumentami są trzy zdania: • Mam gorączkę i katar; • Graliśmy w siatkówkę; • Sport to zdrowie;2.3. Latent Dirichlet Allocation 19 gdzie pogrubienie oznacza, że tylko te słowa pozostały w tekstach po wstępnym przygoto- waniu danych. Załóżmy również, że mamy określone dwa tematy, z czego w pierwszym najczęstsze słowa to „przeziębienie", „gorączka", „grypa" i inne słowa związane ze zdrowiem, natomiast w drugim są to „piłka", „grać", „sport" i inne związane ze sportem. Wówczas jeżeli zakładamy, że słowa w dokumentach pochodzą z jakiegoś tematu, to można intuicyjnie ocenić, że pierwsze zdanie w 100w 100W niniejszej pracy opisujemy model LDA w zastosowaniu do modelowania danych tekstowych. Do formalnego opisu modelu, będzie potrzebne kilka pojęć, które teraz zdefiniuje- my: • Słowo – podstawowa jednostka danych. Zbiór wszystkich słów nazywamy „słownikiem", a jego liczność oznaczamy przez V . Formalnie słowo reprezentujemy jako liczbę ze zbioru 1, ..., V . • Dokument – skończony ciąg słów dowolnej długości. Formalnie jest ciąg liczbowy o war- tościach ze zbioru 1, ..., V , który można interpretować jako wektor. • Korpus – zbiór składający się ze

skończonej liczby dokumentów. • Temat – dyskretny rozkład prawdopodobieństwa wymiaru V , opisujący rozkład na zbio- rze słów. W pracy będziemy wymiennie używać sformułowań „rozkład słów w temacie", „rozkład tematu"

W modelu LDA występują trzy parametry: • K – liczba tematów, • ? – wektor dodatnich liczb rzeczywistych, sterujący rozkładami tematów w dokumen- tach, • ? – macierz wymiaru K $\times V, ktreji - tywierszopisujerozkadi - tegotematu.Jedynymparametrem, ktregowartotrzebazada, jestK.Oznaczato, emusimyapriorizaoy, ileter$

In natural language processing, latent Dirichlet allocation (LDA) is a generative statistical model that allows sets of observations to be explained by unobserved groups that explain why some parts of the data are similar. For example, if observations are words collected into documents, it posits that each document is a mixture of a small number of topics and that each word's creation is attributable to one of the document's topics. LDA is an example of a topic model and was first presented as a graphical model for topic discovery by David Blei, Andrew Ng, and Michael I. Jordan in 2003.[1] Essentially the same model was also proposed independently by J. K. Pritchard, M. Stephens, and P. Donnelly in the study of population genetics in 2000.[2] Both papers have been highly influential, with 16488 and 18170 citations respectively by December 2016.

In LDA, each document may be viewed as a mixture of various topics where each document is considered to have a set of topics that are assigned to it via LDA. This is identical to probabilistic latent semantic analysis (pLSA), except that in LDA the topic distribution is assumed to have a sparse Dirichlet prior. The sparse Dirichlet priors encode the intuition that documents cover only a small set of topics and that topics use only a small set of words frequently. In practice, this results in a better disambiguation of words and a more precise assignment of documents to topics. LDA is a generalisation of the pLSA model, which is equivalent to LDA under a uniform Dirichlet prior distribution.[5]

For example, an LDA model might have topics that can be classified as CAT$_r elatedandDOG_r elated.Atopichasprobabilitiesof generatingvariouswords, suchasmilk, meow, andkit$ $puppy, bark, andbonemighthavehighprobability.Wordswithoutspecialrelevance, suchasthe(seefunctionw$

$occurrence. A lexical word may occur in several topics with a different probability, however, with a differ$

Each document is assumed to be characterized by a particular set of topics. This is akin to the standard bag of words model assumption, and makes the individual words exchangeable.

LDA[23] jest modelem, w którym każdy dokument jest modelowany przez skończony zbiór tematów, z których się składa. Każdy temat z kolei

With plate notation, the dependencies among the many variables can be captured concisely. The boxes are "plates" representing replicates. The outer plate represents documents, while the inner plate represents the repeated choice of topics and words within a document. M denotes the number of documents, N the number of words in a document.

The generative process is as follows. Documents are represented as random mixtures over latent topics, where each topic is characterized by a distribution over words. LDA assumes the following generative process for a corpus $DD consisting of MM documents each of length N_i N_i$:

2.3.6 Word embeddings

Od 2013r., wraz z wprowadzeniem przez T. Mikolova metody Word2vec[5] nastąpił gwałtowny rozwój i niewątpliwy sukces metod „word embeddings". Określenie „word embeddings" oznacza osadzanie słów w przestrzei wektorowej przy pomocy uczenia nienadzorowanego i zostało po raz pierwszy użyte 2003r. w pracy Y. Bengio[24], gdzie wektory słów generowane są przez głęboką sieć neuronową. Ogół technik zaliczanych obecnie do „word embeddings" cechuje się usiłowaniem reprezentacji słów wraz z zależnościami pomiędzy nimi w postaci wektorów o stosunkowo niskiej wymiarowości. Dzieje się to w opozycji do wcześniejszych podejść podobnych do Bag of words - produkującego ogromne, rzadkie wektory, których wymiary równają się rozmiarowi słownika, o który oparty jest model (rzędu setek tysięcy). Ważną własnością metod osadzania słów jest zachowanie przez wektory semantycznych i syntaktycznych właściwości słów, co pozwala wykonywać na nich operacje artymetyczne na

wektorach odwzorowujące cechy tychże słów np. $vector(\text{"}king\text{"}) - vector(\text{"}man\text{"}) + vector(\text{"}woman\text{"}) \approx vector(\text{"}queen\text{"})$ Stosowane obecnie podejścia generowania wektorowych reprezentacji słów można podzielić na dwa typy:

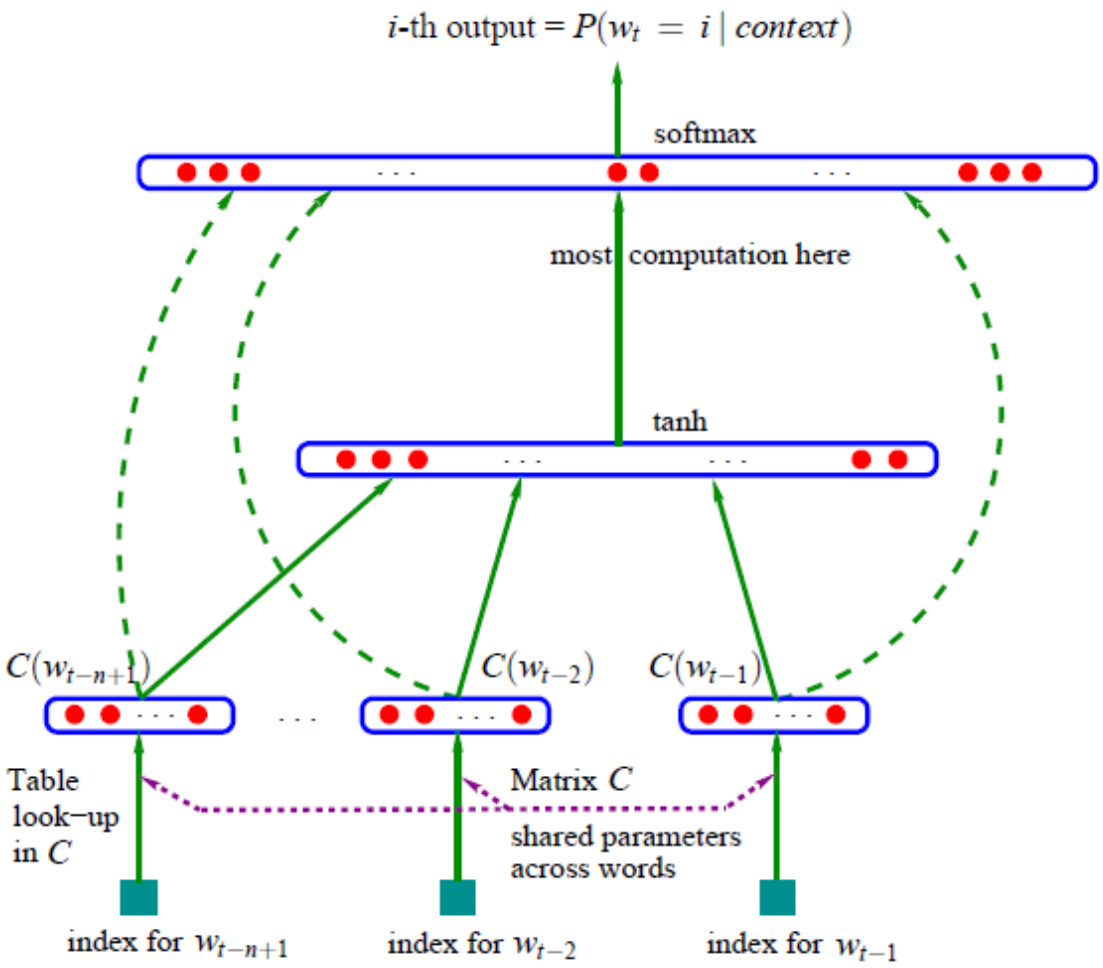
1. Modele predykcyjne: uczą się wektorowych reprezentacji słów poprzez zmniejszenie błędu predykcji słów należących do lokalnego kontekstu słowa i. Poniżej opisuję sztandarowy przykład takiego modelu - word2vec, gdzie sposobem na optymalizację funkcji celu jest zastosowanie płytkiej sieci neuronowej typu feed-forward optymalizowanej za pomocą metody stochastic gradient descent.

2. Metody oparte o zliczanie: generują wektory słów poprzez redukcję wymiarowości w globalnej macierzy współwystąpień słów. Jako pierwszy etap konstruują one ogromną (wymiar równa się liczbie słów w słowniku korpusu) macierz, która (podobnie, jak metodzie LSI) następnie ulega faktoryzacji, aby uzyskać macierz o mniejszym wymiarze, lecz nadal zachowującą powiązania pomiędzy słowami. Przykładem jest tu opisana poniżej metoda Global Vectors - GloVe.

Jedną z szerokiego wachlarza możliwości, jakie dają tego typu techniki jest określanie podobieństwa pomiędzy całymi dokumentami, wykorzystując dodatkowe metody pozwalające przenieść zależności między poszczególnymi słowami dokumentu na zależności między całymi zbiorami słów, co jest istotne z punktu widzenia tematu niniejszej pracy. Dwie z nich: metodę centroidu oraz Word Mover's Distance opisuję później w tym rozdziale.

2.3.7 Podejścia deep learningowe

Wspomniane podejście Bengio oparte jest o sieć neuronową typu feed-forward o jednej warstwie ukrytej zgodnie z architekturą z poniższego rysunku.



Rysunek 2.1: Neuronowy model języka. Źródło: [24].

Celem działania sieci jest maksymalizacja funkcji celu $J_\theta = \frac{1}{T}\sum_{t=1}^{T} \log f(w_t, w_{t-1}, \cdots, w_{t-n+1})$, gdzie $f(w_t, w_{t-1}, \cdots, w_{t-n+1})$ odpowiada prawdopodobieństwu $p(w_t \mid w_{t-1}, \cdots, w_{t-n+1})$ wystąpienia słowa w_t bezpośrednio po sekwencji słów $w_{t-1}, \cdots, w_{t-n+1}$. Wektorowa reprezentacja słowa uzyskiwana jest tu przez przemnożenie wejściowego wektora (wektor zer z jedynką na i-tym miejscu reprezentujący i-te słowo, „one-hot-vector") z macierzą wag pierwszej warstwy sieci.

Podejście to jak i kolejne ([?]) wykorzystujące głębokie sieci neuronowe nie znalazły zastosowań komercyjnych, ponieważ ich wydajność nauki jest na tyle niska, że niemożliwe jest użycie przy ogromnych zbiorach danych

wykorzystywanych w środowiskach produkcyjnych.

Rozwiązaniem tego problemu wydają się być nowe metody wektorowej reprezentacji słów powstałe na przestrzeni ostatnich lat. W odróżnieniu do metod deep learningowych opierają się one o metody szybkiej nauki, np. o płytkie sieci neuronowe, które uczą się na tyle krótko, że sprawdzają się one w zastosowaniach komercyjnych.

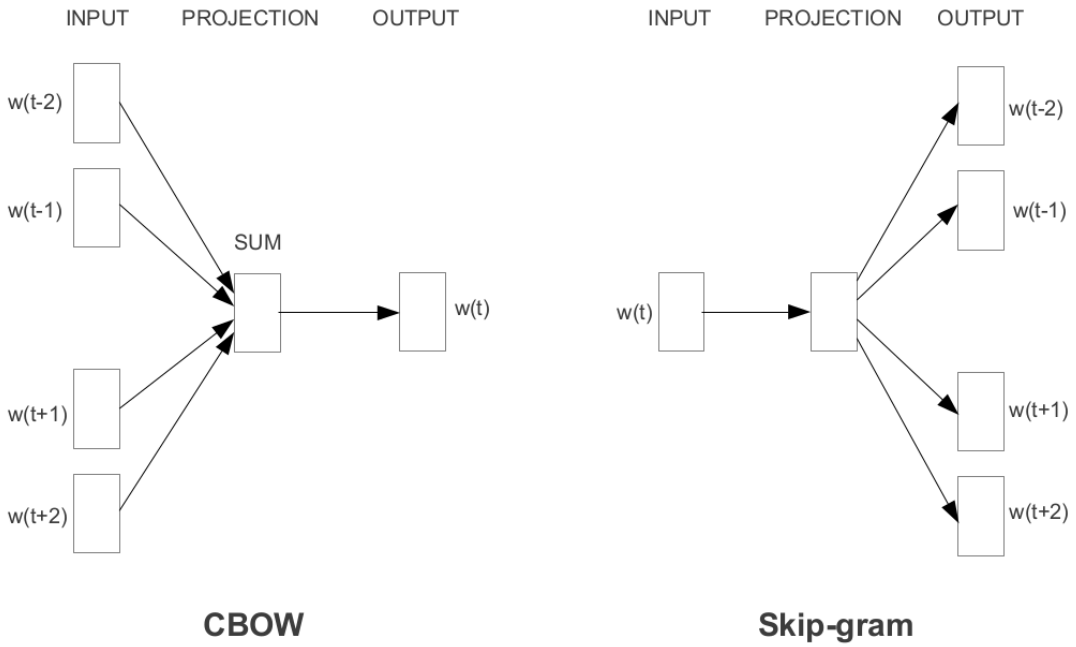
2.3.8 Word2vec

Word2vec jest stosunkowo nową (2013r.) predykcyjną metodą osadzania słów w przestrzeni wektorowej, opisaną w [5].

Autorzy metody proponują płytką, dwuwarstwową sieć neuronową, która ma za zadanie odtworzyć kontekst danego słowa i na tej podstawie dokonać reprezentacji słowa jako wektora liczb rzeczywistych. Jako wejście metoda otrzymuje słowa z korpusu, wyjściem metody są natomiast wektory z pewnej N wymiarowej przestrzeni odpowiadające słowom wejściowym. Użyta tu sieć neuronowa składa się z warstw: wejściowej, jednej warstwy ukrytej i warstwy wyjściowej. Wyróżnia się dwie architektury sieci:

- skip-gram: na podstawie słowa sieć dokonuje predykcji N sąsiednich słów. Zadaniem sieci neuronowej jest wtedy optymalizacja funkcji celu postaci $J_\theta = \frac{1}{T}\sum_{t=1}^{T} \log p(w_t \mid w_{t-n}, \cdots, w_{t-1}, w_{t+1}, \cdots, w_{t+n})$.

- CBOW (continuous bag of words): na postawie okna N sąsiednich słów sieć przewiduje słowo, którego z największym prawdopodobieństwem te N słów jest sąsiedztwem. W tym modelu funkcja celu przyjmuje postać $J_\theta = \frac{1}{T}\sum_{t=1}^{T} \sum_{-n \le j \le n, \neq 0} \log p(w_{t+j} \mid w_t)$.



Rysunek 2.2: Schemat sieci wykorzystującej podejście skip-gram i CBOW. Źródło: [5].

Wady i zalety obu podejść są wymienione w [6]. W celu szczegółowego opisu metody Word2vec wprowadzam pojęcie funkcji softmax.

Softmax jest generalizacją funkcji logistycznej, zamieniającą K-wymiarowy wektor z dowolnych liczb rzeczywistych na K-wymiarowy wektor liczb rzeczywistych z zakresu $(0, 1]$, które sumują się do 1[13]. Funkcja wyraża się wzorem $\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}\ dla\ j = 1,\ ..., K$. Wyjście funkcji można traktować jako pewien rozkład prawdopodobieństwa.

Używając tej stosunkowo prostej architektury można wykonać proces nauki używając milionów słów, których powiązania między sobą zostaną zachowane w systemie wag sieci neuronowej.

W metodzie Word2vec nauka polega na trenowaniu sieci neuronowej. Jednakże w odróżnieniu od innych metod wykorzystujących sieci neuronowe, Word2vec nie używa później wytrenowanej sieci jako takiej, a jedynie otrzymanych w wyniku nauki wag warstwy ukrytej sieci, które faktycznie są wynikowymi wektorami słow.

W dalszym opisie metody szczegółowo skupiam się na podejściu CBOW, lecz podejście skip-gram wygląda analogicznie.

Sieć neuronowa będąca wynikiem nauki przyjmuje na wejściu wektor binarny długości odpowiadającej liczbie słów w słowniku V zbudowanym na korpusie treningowym. Wektor ten wypełniony jest wartościami 0 oraz jedną wartością 1 na i-tej pozycji. Taki wektor odpowiada i-temu słowu ze słownika V. Wejściem sieci są kolejne słowa z korpusu w tej właśnie reprezentacji. Wyjściem sieci jest wektor tej samej długości o wartościach rzeczywistych z zakresu [0,1], w którym wartość na i-tej pozycji odpowiada prawdopodobieństwu, że i-te słowo ze słownika znajduje się w sąsiedztwie słowa wejściowego. Za „sąsiedztwo" wielkości x należy tu rozumieć zbiór złożony z x słów występujących przed danym słowem w korpusie i x słow położonych za danym słowem. Wartość x może być tu ograniczona przez początek/koniec zdania, które ograniczają kontekst danego słowa.

Jako efekt należy się spodziewać, że dla słowa wejściowego „Brytania" otrzymamy na wyjściu wysoką wartość prawdopodobieństwa dla słowa „Wielka", a niską np. dla słowa „skoroszyt".

Jednym z parametrów metody Word2vec jest wymiarowość przestrzeni, w której znajdują się otrzymane wektory odpowiadające słowom z korpusu. Liczba ta ma swoje źródło z wielkości warstwy ukrytej sieci neuronowej. Wagi warstwy ukrytej można interpretować jako macierz MxN, gdzie M to liczba słów słownika V - wielkość wektowa wejściowego, a N to liczba neuronów w warstwie ukrytej. Po przeprowadzeniu nauki i-ty wiersz tej macierzy odpowiada wektorowi długości N, który reprezentuje i-te słowo ze słownika V.

W sieci nie jest używa funkcja akywacji, ale prawdopodobieństwa na wyjściu są efektem działania funkcji softmax. Funkcja ta ma za zadanie sprowadzić wyjściowe wartości warstwy ukrytej do postaci rozkładu prawdopodobieństwa.

Isotną zaletą metody Word2vec jest fakt, iż pozwala ona ocenić „odległość" pomiędzy dwoma dokumentami nawet, jeżeli nie posiadają one wspólnych słów.

2.3.9 FastText

FastText[**?**] to biblioteka stworzona w 2016 w celu wydajnego uczenia wektorowej reprezentacji słów oraz klasyfikacji zdań. Od „klasycznego" word2vec różni się stopniem szczegółowości analizy słów. Word2vec traktuje słowo jaką najmniejszą, niepodzielną jednostkę, której wektorową reprezentację usiłuje wyznaczyć. FastText natomiast dokonuje analizy również wewnętrznej analizy słów. Wykorzystuje w tym celu rozbicie słowa na podsłowa - ciągi znaków o określonej długości n, „character n-grams". Np. słowo „pokój" składa się następujących 3-gramów: [pok], [okó], [kój]. Podejście takie daje szereg nowych możliwości. Pomaga wyznaczyć reprezentację wektorową rzadkich słów, które być może mają wspólny rdzeń (i znaczenie) z innymi, częściej występującymi słowami. Metoda pozwala również nadać wektorową reprezentację słowom, których w ogóle nie ma w słowniku, jako że ich podsłowa mogą należeć do słów w słowniku się znajdujących. Zalety te wydają się być szczególnie obiecujące w przypadku bogatych morfologicznie języków, np. języka polskiego, tureskiego, czy fińskiego.

Zasada działania metody bazuje na word2vec. Jednakże oprócz predykcji tylko całych słów następuje tu również predykcja n-gramów słowa a w otoczeniu słowa a. Ostatecznie słowu a zostaje przypisany wektor składający się ze średniej oryginalnej reprezentacji wektorowej słowa oraz reprezentacji jego n-gramów.

Jak pokazuje badanie[**?**] metoda ta sprawdza się lepiej od word2vec w wykrywaniu syntaktycznych podobieństw pomiędzy słowami.

2.3.10 GloVe

GloVe[19] (GLObal VEctors) jest kolejną wartą uwagi metodą word embedding powstałą na przestrzeni ostatnich lat. Algorytm GloVe różni się od word2vec w sposobie uzyskania wektorowej reprezentacji słów. Word2vec jest modelem predykcyjnym, natomiast trening w GloVe opiera się na globalnej macierzy współwystąpień słów. Ponadto w porównaniu do word2vec GloVe stara się wyznaczyć reprezentacje wektorowe wprost, podczas gdy w word2vec dzieje się

„przy okazji" - szkoli się sieć neuronową nie w celu jej dalszego wykorzystania w celu predykcji, a jedynie dla jej macierzy wag.

Algorytm GloVe składa się z następujących kroków[18]:

1. Zgromadź współwystąpienia słów w formie macierzy X. Każdy element X_{ij} takiej macierzy reprezentuje jak często słowo i występuje w pobliżu słowa j. Zazwyczaj macierz buduje się poprzez skanowanie bazowego korpusu oknem o ustalonej szerokości, w obrębie którego centralne słowo leży w kontekście słów je otaczających. Dodatkowo można tu wprowadzić wagi dla słów malejące wraz ze wzrostem dystansu od słowa centralnego.

2. Zdefiniuj ograniczenie dla każdej pary słów: $w_i^T w_j + b_i + b_j = log(X_{ij})$, gdzie w_i oznacza wektor głównego słowa, w_j słowa leżącego w pobliżu i, b_i i b_j to skalary.

3. Zdefiniuj funkcję kosztu $J = \sum_{i=1}^{V} \sum_{j=1}^{V} \ f(X_{ij})(w_i^T w_j + b_i + b_j - \log X_{ij})^2$, gdzie f jest funkcją ważącą, która pomaga zapobiec uczeniu tylko na podstawie najbardziej popularnych par słów. Autorzy proponują funkcję postaci: $f(X_{ij}) = \begin{cases} (\frac{X_{ij}}{x_{max}})^\alpha & \text{if } X_{ij} < XMAX \\ 1 & \text{otherwise} \end{cases}$ Celem funkcji optymalizacji funkcji kosztu jest minimalizacja różnicy pomiędzy iloczynami skalarnymi wektorów współwystępujących słów.

4. Dokonaj minimalizacji funkcji kosztu poprzez stopniową aktualizację wektorów w_i i w_j.

2.3.11 Odległość między dokumentami

W celu wykorzystania omówionych metod osadzania słów należy wybrać metodę obliczania odległości między całymi dokumentami, których słowa potrafimy reprezentować jako wektory. Zakładamy, że jeżeli dystans pomiędzy dokumentami jest mały, to ich tematyka jest podobna.

Centroid

Najprostszą i najbardziej intuicyjną metodą obliczenia odległości pomiędzy wektorową reprezentacją dokumentów jest wykonanie dwóch prostych kroków:

1. Uśrednienie wektorów wchodzących w skład każdego z dokumentów. Powstały w ten sposób wektor jest centroidem reprezentującym dokument w przestrzeni wektorowej.

2. Obliczenie dystansu między wektorami. Powszechnie przyjętą praktyką jest stosowanie tzw. odległości kosinusowej - znormalizowanego iloczynu skalarnego wektorów A i B. Jest to kosinus kąta pomiędzy dwoma wektorami reprezentującymi dokumenty. Zaletą tej metody jest natychmiastowa normalizacja wyniku do zakresu $(0, 1)$. Odległość $sim = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|_2 \|\mathbf{B}\|_2} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}}$, gdzie A_i i B_i są składowymi wektorów odpowiednio A i B

Wadą opisanej powyżej metody jest utrata potencjalnie użytecznych zależności pomiędzy poszczególnymi wektorami wchodzącymi w skład dokumentu.

W kontrze to tego prezentuję metodę obliczania dystansu między dokumentami uwzględniającą rozkład wektorów wewnątz dokumentu.

Word Mover's Distance

Word Mover's Distance[12] to rozwiązanie zwracające odległość między dokumentami tekstowymi. W tym celu adaptuje algorytm Earth Mover's Distance[10] oraz wektorową reprezentację słów dokumentu. WMD mierzy odległość między dokumentami jako minimalny dystans jaki wektory słów pierwszego dokumentu muszą „pokonać" aby osiągnąć wartości wektorów z drugiego dokumentu.

EMD jest metodą mierzenia odległości pomiędzy dwoma rozkładami, która opiera się na minimalnym koszcie, jaki musi zostać poniesiony, aby dokonać transformacji jednego rozkładu w drugi. Problem można sformalizować jako

problem programowania liniowego, gdzie: $P = \{f(p_1, w_{p_1})...(p_m, w_{p_m})\}$, $Q = \{f(q_1, w_{q_1})...(q_n, w_{q_n})\}$ są danymi rozkładami o m (odpowiednio n) klastrach p_i (q_j), a w_{p_i} (w_{q_j}) jest masą klastra. $D = [d_{ij}]$ jest macierzą odległości, w której d_{ij} reprezentuje odległość pomiędzy klastrami p_i i q_j. Celem jest znaleźć taki przepływ $F = [f_{ij}]$, gdzie f_{ij} to przepływ pomiędzy p_i i q_j, który minimalizuje całościowy koszt $Work(P, Q, F) = \sum_{i=1}^{m} \sum_{n}^{j=1} d_{ij} f_{ij}$ przy odpowiednich ogramiczeniach[10]. EMD jest to dobrze zbadanym problemem transportowym[10], dla którego powstały efektywne metody rozwiązania[9].

Przypuśćmy, że dzięki metodzie Word2vec dla słownika V o n słowach otrzymujemy macierz $X \in \mathbb{R}^{d \times n}$. i-ta kolumna tej macierzy reprezentuje i-te słowo ze słownika V. Odległości pomiędzy wektorami reprezentującymi semantycznie zbliżone słowa są relatywnie mniejsze od odległości dla słów niezwiązanych ze sobą. Celem WMD jest zawrzeć semantyczne podobieństwo pomiędzy poszczególnymi parami słów w dystans pomiędzy całymi dokumentami. Aby to osiągnąć metoda traktuje dokument jako rozkład, którego i-tym elementem jest liczba wystąpień i-tego słowa w tym dokumencie, a następnie stosuje metodę EMD do obliczenia dystansu między tymi rozkładami. Macierz odległości D używana w metodzie EMD jest zbudowana na bazie odległości między wektorami Word2vec reprezentującymi słowa dokumentów. $d_{ij} = ||x_i - x_j||$, gdzie i i j to indeksy słów ze słownika V a x_{ij} to element macierzy X[12]. Autorzy metody określają złożoność metody jako $O(p^3 \log p)$, gdzie p to wielkość słownika V.

Rozdział 3

Dane

Dane, na których testowane były opisywane w niniejszej pracy metody otrzymałem dzięki życzliwości serwisu Allegro. Jednak, by dane te otrzymać, zobowiązany zostałem po podpisania umowy o poufności. Stąd, w niniejszej pracy brak jakichkolwiek przykładów danych, a jedynie opisy metod użytych do ich przetwarzania i generowania rekomendacji.

3.1 Opis danych

Otrzymane dane to baza ok. 20000 artykułów tekstowych w formacie JSON. Są to te same artykuły, które są dostępne dla użytkowników poprzez serwis internetowy (stan na styczeń 2017). Pojedynczy rekord danych składa się z głównej treści artykułu oraz z metadanych, z których za istotne z punktu widzenia tematu pracy uznałem pola: id, kategoria i słowa kluczowe.

3.1.1 Treść artykułu

Treść każdego artykułu składa się z trzech pól: „zawartość", „nagłówek" i „tytuł". Średnia długość artykułu to 821 słów, w tym nagłówek to jednozdaniowy wstęp. Średnią tą estymuję na podstawie średniej liczby znaków artykułu i średniej długości słowa w języku polskim. Dokładne statystyki tekstu będą dostępne dopiero

po wstępnym przetwarzaniu.

Wszystkie artykuły napisane są w języku polskim, w nielicznych przypadkach wykryłem błędy, tzw. „literówki". Jako, że artykuły ze zbioru dotyczą produktów sprzedawanych za pośrednictwem serwisu Allegro, w skład słownika zbudowanego na ich bazie wchodzi wiele słów specyficznych dla różnych branż. Są to m.in. nazwy modeli aparatów (np. „Sony Alpha 77 II"), samochodów, gier komputerowych, a także nazwy techniczne: „sprężarka", „hipertoniczny", „autofocus".

Artykuły posiadają w swej treści wiele znaczników interpretowanych przez system, na podstawie których wzbogacana jest warstwa wizualna strony internetowej zawierającej artykuł, np. obrazki czy łącza do ofert związanych z tematem artykułu.

Spójność danych oceniam na wysoką, tj. każde pole zawarte w strukturze dokumentu jest zawsze wypełnione - brak jest wartości typu NULL.

3.1.2 Kategoria

Każdy artykuł został przez autora przydzielony do pewnej kategorii, która odpowiada tematyce artykułu, np. „Aparaty cyfrowe" czy „Przyprawy i zioła". W skład pola „kategoria" wchodzi również lista kategorii nadrzędnych, a cała hierarcha kategorii ma strukturę drzewiastą. Np. kategoria nadrzędna dla kat. „Przyprawy i zioła" to „Delikatesy", a dla kat. „Delikatesy" to „Dom i zdrowie". Każdy artykuł należy do tylko jednej kategorii będącej dowolnym węzłem w drzewie (nie tylko liściem).

3.1.3 Słowa kluczowe

Do każdego artykułu dołączone są słowa kluczowe charakteryzujące jego zawartość, np. „aparaty", „aparaty cyfrowe", „lustrzanki", „sony". Pole to jest wykorzystywane w dotychczasowym mechaniźmie generowania rekomendacji - artykuły podobne do danego są wyszukiwane na podstawie jego słów kluczowych.

3.2 Wstępne przetwarzanie danych

W celu zwiększenia skuteczności metod analizy tekstu stosuje się wstępne przetwarzanie danych. Ma ono na celu takie przygotowanie tekstu, aby zmaksymalizować jakość wyników operujących na nim później algorytmy. Techniki wstępnego przetwarzania tekstu nie wchodzą w skład żadnego standardu - dobieram je indywidualnie do konkrtenego przypadku, zgodnie z intuicją.

Niżej opisuję kolejne kroki wstępnego przetwarzania tekstu, które wykonuję na posiadanym zbiorze artykułów.

1. Oczyszczanie tekstu ze zbędnych, wspomnianych wcześniej znaczników. Z punktu widzenia semantycznej analizy tekstu są one bezużyteczne, czy wręcz szkodliwe (powodują pewne „zanieczyszczenie" tekstu). Stąd usuwam je wykorzystując odpowiednio skonstruowane wyrażenia regularne (ich postać jest szczegółem nieistotnym z punktu widzenia tematyki niniejszej pracy).

2. Usunięcie „słów stopu"(ang. stopwords) - na ogół krótkich słów nie wnoszących nic do znaczenia całości artykułu. Są to np. „w", „z", „ponieważ". Ich usunięcie zmniejsza liczbę słów dokumentu skracając tym samym czas jego przetwarzania. Jako że słowa te występują często, usunięcie ich daje możliwość uwypuklenia znaczenia innych słów mających wpływ na rzeczywiste znaczenie całego artykułu. Zbiór słów stopu czerpię z [15].

3. Sprowadzenie wszystkich słów dokumentu do małych liter. Pomaga to ujednolicić postać części słów o tym samym znaczeniu, wśród których jedno występuje na początku zdania a inne w środku.

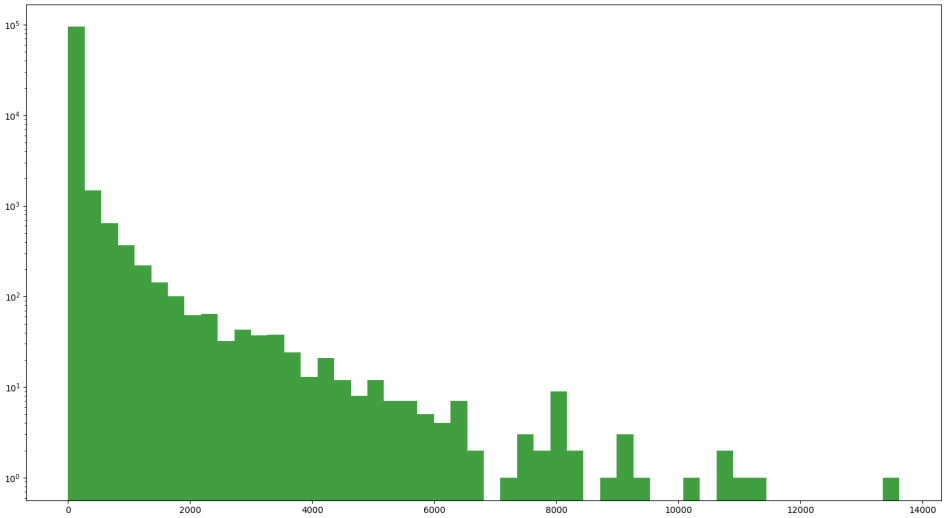
4. Rozbicie słów połączonych myślnikiem. Doświadczenie w późniejszym etapie (tokenizacji) pokazuje, że narzędzie jej dokonujące nie radzi sobie z tego typu słowami (np. „biało-czerwony") i zostania je w niezmienionej postaci gramatycznej (np. „biało-czerwonego"). Stąd konieczność ręcznego wykoania mechanizmu rozbijającego takie słowa do postaci kompatybilnej z tokenizerem.

5. Tokenizacja. Jest to najistotniejszy element całego procesu. Polega na sprowadzaniu słów o tym samym znaczeniu, a różnej formie gramatycznej do tej samej postaci. Sporym utrudnieniem jest tutaj stopień skomplikowania języka polskiego oraz liczba wyjątków, jaką ten język posiada. Za przykład może posłużyć słowo „mieć", którego jedna z form to „ma", kolejna to „miej". Celem etapu jest sprowadzenie każdego z tych wyrazów do formy podstawowej „mieć". Do przeprowadzenia tej operacji stosuję narzędzie Morfologik[4].

Użycie wymienionych technik nie jest jedynym standardem a wynikiem analizy przetwarzanych danych i techniki te zostały dobrane dla tego konkretnego przypadku

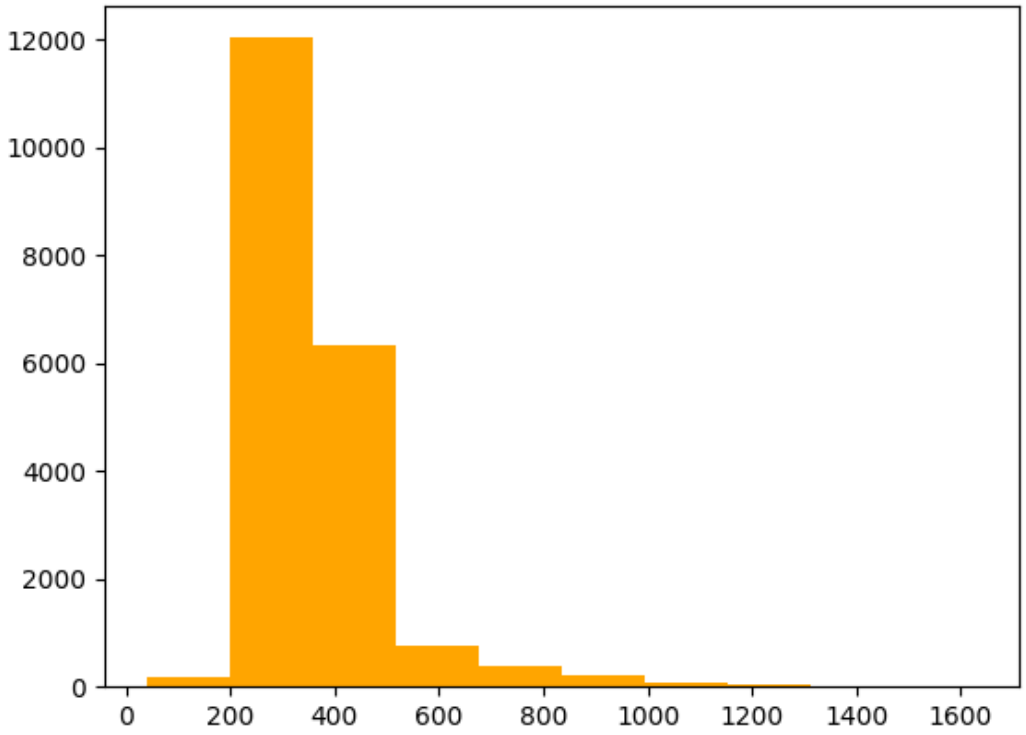
3.3 Opis danych po wstępnym przetwarzaniu

Powyższe kroki doprowadzają dane do stanu, w którym można zastosować techniki semantycznej analizy tekstu. Słownik zbudowany na wstępnie przetworzonym korpusie zawiera 98174 unikalnych słów, oraz 7409145 wszystkich słów (z powtórzeniami).



Rysunek 3.1: Histogram liczby wystąpień słów w korpusie w skali logarytmicznej.

Większość artykułów okazała się być podobnej długości, średnia długość artykułu to 370 słów.



Rysunek 3.2: Histogram długości artykułów.

Rozdział 4

Metody ewaluacji

W celu porównania stosowanych metod wyznaczania podobieństwa między artykułami konieczna jest formalizacja pewnych miar tego podobieństwa.

Ewaluacja rankingu, w którym trafność wyników zależy od oceny użytkowników jest zadaniem nietrywialnym. Podobieństwo artykułów napisanych w języku naturalnym jest rzeczą subiektywną. W sytuacji idealnej dysponowalibyśmy obiektywną miarą podobieństwa pomiędzy parami N artykułów (np. wyznaczoną wcześniej przez miarodajną grupę użytkowników), które to N artykułów stanowiłoby zbiór testowy. Uzyskanie takich danych wiąże się jednak z dużymi kosztami i leży poza moimi możliwościami.

Praktyką umożliwiającą obiektywną ocenę, wykorzystywaną w działających systemach są tzw. testy A/B polegające na podziale użytkowników na grupy i zaaplikowaniu każdej grupie innego rozwiązania. Następnie mierzone są pewne wskaźniki wśród każdej grupy (w naszym przypadku np. liczba „kliknięć" prawdziwych użytkowników w artykuły rekomendowane) i spośród zgromadzonych wyników wybierane jest rozwiązanie najlepsze.

Z powodu braku możliwości wykorzystania rzeczywistych użytkowników do ewaluacji rozwiązań jestem zmuszony wprowadzić własne miary oparte na dostępnych danych. Należy tu zaznaczyć niedoskonałość wprowadzanych miar, ponieważ każda z nich opiera się na pewnych założeniach, od których prawdziwości zlaeży jakość całej miary.

Działanie testowanych metod można sformalizwać w postaci pewnej funkcji $S_n : C \rightarrow \{a_i\}_{i<n}$, gdzie $a_i \in C$, a n to liczba elementów zwracanego ciągu. Funkcja S przyjmuje artykuł tekstowy (bądź jego identyfikator) i zwraca skończony ciąg artykłów do niego podobnych zgodnie ze stopniem dopasowania (najlepsze na początku). Celem działania niżej opisanych miar jest każdej parze postaci: wyjście-wejście funkcji S reprezentującej testowaną metodę przypisać ocenę jakości zwróconego wyjścia dla danego wejścia. Oceny dla konkretnej metody, dla ustalonej próby artykułów są następnie uśredniane.

Opisane poniżej miary 1 i 2 dokonują porównania podobieństwa dla pary artykułów. W celu rozszerzenia działania tych miar do pary wejście-wyjście metody stosuję średnią ważoną podobieństwa kolejnych elementów wyjścia z wejściem. Stosowane wagi: $\frac{1}{i}$ dla $i = 1, \ ..., \ N$, gdzie N to długość ciągu wyjściowego danej metody.

4.1 Miara 1: Dystans oparty na metadanych

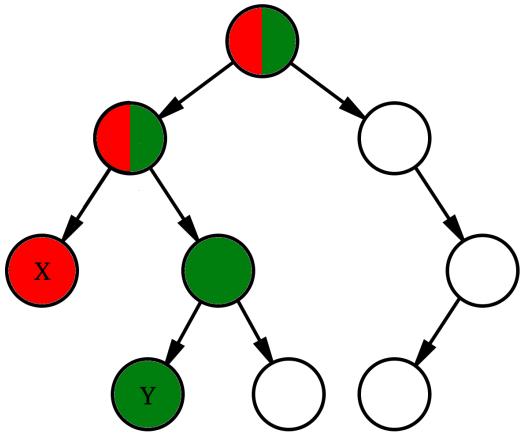
Jak wspomniałem wcześniej dane prócz treści artykułów zawierają również pewne metadane, a wśród nich umożliwiające tworzenie powiązań między artykułami. Skupiam się tu na polach: „słowa kluczowe" i „kategoria".

4.1.1 Kategorie

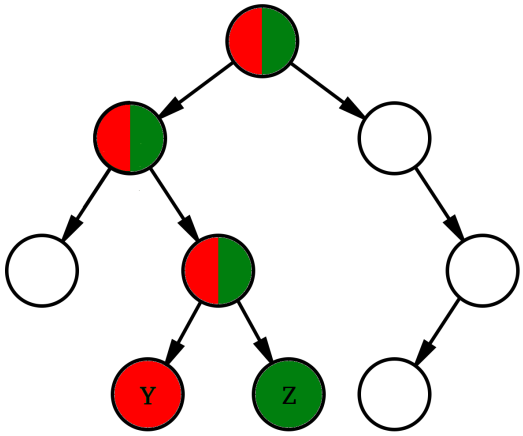
Pierwszą zastosowaną miarą, pozwalającą ocenić jakość dopasowania podobnych artykułów jest ich odległość we wcześniej wspomnianym drzewie kategorii. Zakładam tu, że im więcej wspólnych przodków w drzewie, tym bardziej podobe do siebie są artykuły reprezentowane przez węzły drzewa. Zaletą miary jest fakt, iż przypisanie artykułu do kategorii zostało wykonane przez autora, którego można określić ekspertem w dziedzinie tematyki artykułu. Stąd przynależność artykułu do danej kategorii jest mocno uzasadniona. Kolejną zaletą tej miary jest fakt, iż można ją zastosować automatycznie - wiedza ekspercka jest już zapisana w danych artykułów. Należy zaznaczyć tu jednak, że miara nie jest idealna - każdy

artykuł należy do tylko jednego liścia drzewa kategorii. Stąd artykuł poruszający zagadnienia z różnych obszarów, który można by przypisać dwóm stosunkowo odległym kategoriom A i B, zostanie przypisany tylko do jednej kategorii, np. A. Miara pokaże wtedy dużą odległość od artykułów z kategorii B, co nie jest prawdą.

Formalnie miarę można zapisać jako: $d(a_1, a_2) = \frac{w(a_1, a_2)}{D}$, gdzie d to dystans między artykułami a_1 i a_2, $w(x, y)$ to długość części wspólnej ścieżek od korzenia drzewa kategorii do węzłów reprezentujących artykuły x i y, a D to głębokość całego drzewa (wprowadzone w celu normalizacji). Im wyższy wynik, tym większe podobieństwo artykułów.



Rysunek 4.1: Drzewo kategorii dla przykładu 1.



Rysunek 4.2: Drzewo kategorii dla przykładu 2.

W powyższych przykładowych drzewach $w(X,Y) = 1$, $w(Y,Z) = 2$, $D = 3$, stąd $d(X,Y) = \frac{1}{3}$, $d(X,Y) = \frac{2}{3}$. Miara wskazuje, że artykuły X i Y są do siebie mniej podobne, niż artykuły Y i Z.

4.1.2 Słowa kluczowe

4.2 Miara 2: Ocena przez użytkowników offline

Kolejną wypracowaną miarą jest subiektywna ocena ekspercka. W celu obiektywizacji oceny, ewaluacja powinna być dokonana przez grupę osób operującycha na tych samych danych. Wadą tej metody jest jej powolność i potrzeba zaangażowania dodatkowych osób dokonujących ewaluacji. Niemożliwym wydaje się przeprowadzenie badania dla wszystkich artykułów, stąd konieczny jest wybór losowej próby artykułów, które parami poddane zostaną ocenie pod kątem podobieństwa. Skala ocen to 1-10: 1, gdy artykuły nie są do siebie podobne, 10, gdy podobieństwo jest całkowite.

4.3 Miara 3: Historyczna aktywność użytkowników serwisu

Zbieranie a następnie przechowywanie informacji o aktywności użytkownika w ramach serwisu internetowego jest powszechną praktyką. Proces ten pozwala na analizę zachowania użytkowników co może doprowadzić do wniosków, jakie usprawnienia należy przedsięwziąć, aby spełnić cele biznesowe. Jednym z przykładów aktywności użytkownika zapisywanej przez serwis Allego są kliknięcia w linki znajdujące się na stronie internetowej. Informacja ta pozwala sporządzić jeszcze jedną miarę jakości dopasowania podobnych do siebie artykułów. Postać danych, jakie udało mi się uzyskać z serwisu to tabela o polach: adres strony, na której nastąpiło kliknięcie, adres strony, na którą prowadzi link, data kliknięcia.

Zaletą metody jest, iż można ją zastosować automatycznie, lecz jest zależna od

danych analitycznych pochodzących z serwisu, które są niedoskonałe.

Jak już zostało opisane powyżej strona z artykułem tekstowym zawiera odnośiki do innych artykułów poruszających tematykę podobną do danego. Skoro zapisywana jest informacja o przejściach pomiędzy podstronami serwisu, to można policzyć ile razy z artykułu X dokonano przejścia na rekomendowany do niego artykuł Y1, a ile razy na rekomendowany artykuł Y2. Jeżeli liczba przejść na artykuł Y1 jest większa niż na Y2, można wnioskować, iż Y1 wydaje się być bardziej adekwatną rekomendacją dla artykułu X.

Posługując się powyższym założeniem, można zaproponować miarę jakości rekomendacji generowanych przez testowane metody w odniesieniu do popularności rzeczywistych rekomendacji wyekstrahowanej z danych serwisu o aktywności użytkowników.

W tym celu dokonuję adaptacji miary nDCG (Normalized Discounted Cumulative Gain). Miara ta służy do oceny jakości uszeregowania przedmiotów, np. wyników zwracanych przez silniki wyszukiwania.

4.3.1 nDCG

TUTAJ OPISUJĘ MIARĘ + podaję źródło

4.3.2 Adaptacja metody nDCG

Załóżmy, że dany algorytm A zwraca pewien ciąg artykułów $c_A = a1,\ a2,\ ...,\ a6$ podobnych do danego artykułu x, w kolejności od najbardziej adekwatnego. Załóżmy również, część elementów ciągu c_B artykułów rekomandowanych w serwisie dla x (używaną dotychczas w serwisie metodą B) znajduje się również w ciągu c_A, tj. np. ISTNIEJĄ TAKIE a_i, a_j (i, j to indeksy w ciągu c_A), że należą do c_A i c_B. Załóżmy ponadto, że z danych o kliknięciach użytkowników w linki w ramach serwisu wiadomo, że przejście z x na a_i jest bardziej popularne niż przejście z x na a_j. Stąd jeżeli $i < j$ ($i > j$), to jakość działania metody A jest dobra (zła), bo metoda ta generuje podobne artykuły w kolejności zgodnej ze stopniem podobieństwa z

artykułem bazowym, opartym o częstośc przejść użytkowników między artykułami.

Za wagi metody nDCG przyjmuję liczby przejść pomiędzy artykułami, a samą metodę stosuję tylko do przecięcia zbioru artykułów podobnych do danego generowanych przez daną metodę ze zborem artykułów rekomendowanych do danego przez dotychczasową metodę działającą w serwisie.

Rozdział 5

Opis i wyniki badań

5.1 Przygotowanie eksperymentów

Metodami, które aplikuję do problemu rekomendacji artykułów są:

[w2v_wdnt_centr] Word2Vec z modelem[16] uczonym na korpusie Słowosieci[17] (model opisany jest poniżej) oraz odległościami między dokumentami liczonymi na bazie centroidu dokumentu.

[w2v_wdnt_wmd] Word2Vec z modelem[16] uczonym na korpusie Słowosieci oraz odległościami między dokumentami liczonymi metodą Word Mover's Distance

[w2v_art_centr] Word2Vec z modelem uczonym na korpusie oraz odległościami między dokumentami liczonymi na bazie centroidu dokumentu.

[w2v_art_wmd] Word2Vec z modelem uczonym na korpusie oraz odległościami między dokumentami liczonymi metodą Word Mover's Distance

[lda] Latent Dirichlet Allocation

Ponadto wyniki zastosowania powyższych metod porównuję z dotychczasową

metodą wykorzystywaną w serwisie allegro [allegro] oraz z losową ocena podobieństwa artykułów [random].

5.1.1 Modele Word2Vec

Model uczony na korpusie Słowosieci

Jako podstawowy model Word2vec użyłem gotowego modelu[16] stworzonego m.in. przez dr inż. M. Piaseckiego. Model ten był uczony na korpusie Słowosieci wer. 10. Dane przed uczeniem przeszły segmentację, lematyzację i ujednoznacznianie morfosyntaktyczne. Użyte parametry uczenia Word2Vec: metoda skip gram, wekroty długości 100, okno kontekstu wielkości 5.

Model ten zawiera 73875 spośród 98174 (75%) unikalnych słów oraz 7313915 z 7409145 (99%) wszystkich słów korpusu artykułów. Wskazuje to, iż słowa nieobecne w modelu są bardzo mało popularne w korpusie artykułów (stanowią ok 1% całości). Po samodzielnym sprawdzeniu stwierdzam, że słowa nieobecne w modelu to: „literówki" lub słowa niepoprawnie stokenizowane (np. „urządzeia"), symbole marek produktów (np. „ux305fa", „i7-4700qm"), żargon branżowy (np. „bootsów"), złożenia wyrazów (np. „kurzoodporne"), wyrazy obce lub ich spolszczenia (np. „thermoprotect"). Uważam, iż mimo niewielkiej liczby tych słów w stosuku mogą mieć one znaczący wpływ na semantykę artykułów.

Model uczony na korpusie artykułów

W związku z powyższym stwierdzeniem wykonuję naukę modelu Word2vec na korpusie artykułów w celu zawarcia brakujących w poprzednim modelu słów. Najrozsądniejszym postępowaniem byłoby tutaj rozszerzenie modelu opartego na korpusie Słowosieci również o brakujące słowa, jednak metoda Word2vec nie pozwala na dodanie nowych słów do słownika istniejącego modelu, a jedynie na dalszą naukę w oparciu o słowa już istniejące w słowniku.

Siłą rzeczy model ten zawiera wszystkie słowa zawarte w korpusie artykułów.

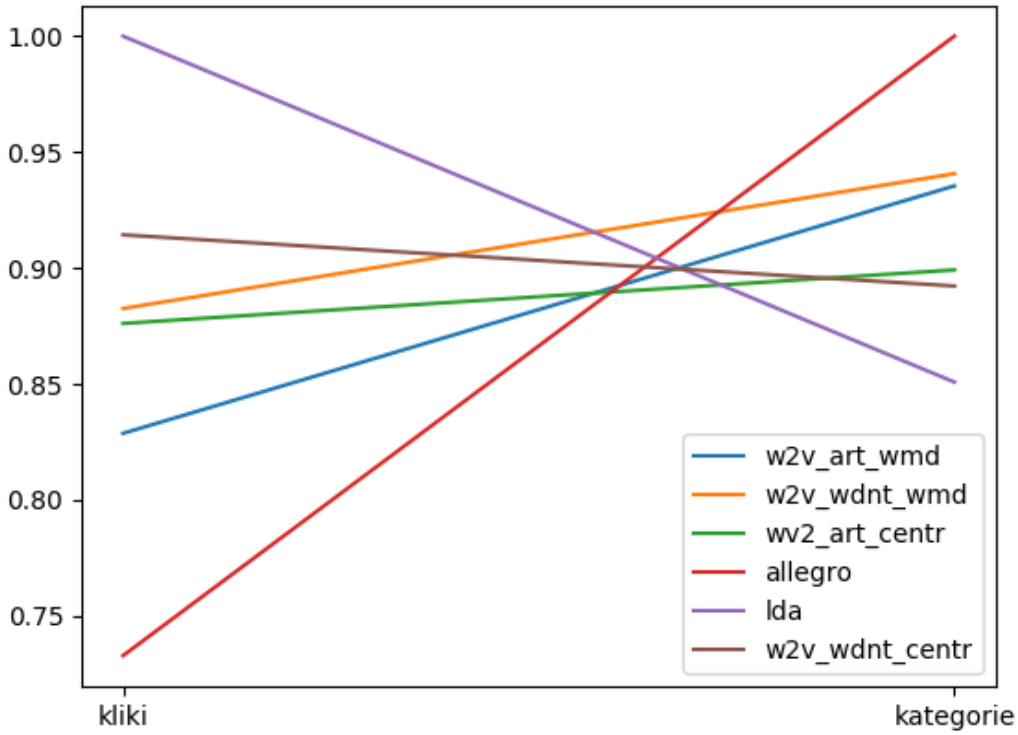
Do uczenia użyłem parametrów identycznych, jak w metodzie powyżej.

5.1.2 Model LDA

5.2 Wyniki badań

Do każdej z wymienionych powyżej metod stosuję każdą z trzech opracowanych przez mnie, opisanych wcześniej miar : opartą na kategoriach [categories], ocenach użytkowników offline [users] i kliknięciach prawdziwych użytkowników serwisu [clicks]. Wyniki zestawiam w tabelce.

Alias metody	clicks	categories	users
random	-	0.145630871396	
w2v_art_wmd	1.29748226281	0.542820699708	
wv2_art_centr	1.37170817357	0.521784904438	
w2v_wdnt_wmd	1.3817891755	0.54587787496	
w2v_wdnt_centr	1.43155208802	0.517755911889	
lda	1.56573051337	0.493703433754	
allegro	1.14732593575	0.580296404276	



Rysunek 5.1: Porównanie znormalizowanych wyników.

5.2.1 Efektywność czasowa

Wadą metody WMD wykluczającą ją z użycia w tym przypadku jest jej powolność. Złżoność czasowa metody wynosi: Dla przypadku: [w2v_art_wmd] obliczenia trwały 55 minut, co przy czasie <1sek dla centroidu jest wartościa niedopuszczalną. Proporcjonalnie użycie tej metody dla całego korpusu trwałoby ok. 183 godzin.

Istotną zaletą dotychczasowego rozwiązainia stosowanego w Allegro jest uniwersalność silnika elasticsearch oraz to, że pozwala edytować inteksowane dane w locie, bez konieczności przebudowy systemu. Metody LDA oraz Word2vec potrzebują przebudowania modelu przy każdej zmianie korpusu, na którym się opierają.

Rozdział 6

Podsumowanie

Dalsze badania.

Niniejsza praca nie wyczerpuje sposobów wyboru artykułów podobnych.

Nie wszystkie pola zawarte w strukturze zostały wykorzystane. Pozostają np. „autor”.

Przed zastosowaniem metod wyznaczania podobieństwa wykonałem przetwarzanie wstępne dokumentów, które można przeprowadzić również na inne sposoby. Jest to temat osobnych badań.

Zdaję sobie sprawę z niedoskonałości zastosowanych miar.

Tematem niniejszej pracy jest przypisanie danemu artykułowi artykułów najbardziej podobnych. Warto tutaj zaznaczyć różnicę pomiędzy tematyką pracy a komercyjnym zagadnieniem najlepszych rekomendacji. Artykuły, które można uznać za dobre rekomandacje, tj. takie, które przynoszą przedsiębiorstwu największy zysk, wcale nie musż być podobne do danego. Powszechnym zjawiskiem jest wzbogacanie rekomandacji o przedmioty niepodobne do danego, a pozwalające użytkownikowi na poznanie osobnej kategorii przedmiotów, która może go zainteresować a tym samym przyciągnąć do serwisu.

Dodatek A

Technologie i narzędzie

Analizę danych, ich wstępne przetworzenie a następnie przeprowadzenie docelowych eksperymentów wykonałem korzystając głównie z języka Python i szeregu skrytpów napisanych w nim własnoręcznie, wykorzystujących istniejące specjalistyczne biblioteki posiadające interfejs w tymże języku.

Wykorzystane narzędzia:

- Elasticsearch - silnik wyszukiwania tekstowego. Używam go do przechowywania bazy artykułów oraz ich przetworzonych wersji.
- MongoDB - nierelacyjna baza danych, której używam do przechowywania wyników generowanych przez testowane algorytmy.

Wykorzystane biblioteki języka Python:

- Gensim - rozbudowana biblioteka służąca do przetwarzania języka naturalnego. Zawiera implementację metod Word2Vec, LDA, TF-IDF i inne.
- Morfologik - tokenizer języka polskiego
- Numpy - pozwala wydajnie wykonywać obliczenia numeryczne
- Pyemd - implementacja algorytmu Earth Mover's Distance

- Elasticsearch - ułatwia wykonywanie zapytań do silnika Elasticsearch wprost z kodu Pythona
- Matplotlib - biblioteka służąca do wykonywania wykresów
- Pymongo - umożliwia wykonywanie zapytań do bazy MongoDB wprost z kodu Pythona

Bibliografia

[1] Francesco Ricci, Lior Rokach, Bracha Shapira, *Introduction to Recommender Systems Handbook*, Springer, 2011

[2] Słownik Języka Polskiego PWN `http://sjp.pwn.pl/sjp/artykul;2441396.html` (07.05.2017)

[3] `https://magazyn.allegro.pl/3333-serwis-allegro-to-nasz-sposob-na-wasze-szybkie-i-wygodne-zakupy-przez-internet` (07.05.2017)

[4] `http://morfologik.blogspot.com/` (07.05.2017)

[5] Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean, *Efficient Estimation of Word Representations in Vector Space*, International Conference on Machine Learning (ICML), 2013

[6] `https://code.google.com/archive/p/word2vec/` (26.05.2017)

[7] `http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/` (26.05.2017)

[9] Ofir Pele, Michael Werman, *Fast and robust earth mover's distances*, ICCV, 2009

[10] Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas, *The Earth Mover's Distance as a Metric for Image Retrieval* str. 1, Computer Science Department, Stanford University, 2000

[12] Matt J. Kusner, Yu Sun, Nicholas I. Kolkin, Kilian Q. Weinberger, *From Word Embeddings To Document Distances*, International Conference on Machine Learning (ICML), 2015

[13] `https://en.wikipedia.org/wiki/Softmax_function/` (11.06.2017)

[14] `https://allegro.pl/artykul/jaka-farba-dla-alergika-55917/` (26.06.2017)

[15] `https://pl.wikipedia.org/wiki/Wikipedia:Stopwords` (15.04.2017)

[16] Paweł Kędzia, Gabriela Czachor, Maciej Piasecki, Jan Kocoń *Vector representations of polish words (Word2Vec method)* Wrocław University of Technology 2016 `https://clarin-pl.eu/dspace/handle/11321/327` (26.06.2017)

[17] `http://plwordnet.pwr.wroc.pl/wordnet/` (28.06.2017)

[18] `https://cran.r-project.org/web/packages/text2vec/vignettes/glove.html` (30.08.2017)

[19] Jeffrey Pennington, Richard Socher, Christopher D. Manning *GloVe: Global Vectors for Word Representation* Computer Science Department, Stanford University, Stanford, CA 94305 2014

[20] Zellig Harris *Distributional Structure* WORD, tom 10, num. 2-3 1954

[21] J.R. Firth. *A synopsis of linguistic theory 1930-1955* Oxford: Philological Society 1957

[22] Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, Richard Harshman *Indexing by latent semantic analysis* Journal of the American Society for Information Science, tom 41, num. 6 1990

[23] David M. Blei, Andrew Y. Ng, Michael I. Jordan *Latent Dirichlet Allocation* Journal of Machine Learning Research, tom 3 num. 4–5 2003

[24] Bengio

[25] `http://blog.aylien.com/overview-word-embeddings-history-word2vec-cbow-glove/` (30.08.2017)

[26] `http://gadzetomania.pl/11824,zakupy-w-sieci-porownanie-najwiekszych-polskich-serwisow-aukcyjnych-2` (09.08.17)

[25] `https://www.elastic.co/use-cases` (10.08.17)

[26] G. Salton and M. McGill, *Introduction to modern information retrieval*, McGraw-Hill, 1983

[27] G. H. Golub, W. Kahan, *Calculating the singular values and pseudo-inverse of a matrix*, Journal of the Society for Industrial and Applied Mathematics: Series B, Numerical Analysis. 2 (2), 1965

[28] R. Collobert, J. Weston, *A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning*, NEC Labs America, 2008

[29] A. Joulin, E. Grave, P. Bojanowski T. Mikolov, *Bag of Tricks for Efficient Text Classification*, Facebook AI Research, 2016

Warszawa, dnia .....

Oświadczenie

Oświadczam, że pracę magisterską pod tytułem: „Rekomendacje artykułów opisujących produkty w serwisach e-commerce”, której promotorem jest dr inż. Anna Wróblewska, wykonałem samodzielnie, co poświadczam własnoręcznym podpisem.

.....