



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, czerwiec 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 rekomendacji.

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. Skupiam się szczególnie na metodach opartych o semantyczną analizę tekstu.

Abstract

Spis treści

1	Wstęp	4
1.1	Rekomendacje artykułów tekstowych w Allegro	5
1.2	Struktura pracy	7
1.3	Uwagi	8
2	Przegląd wybranych metod	9
2.1	Systemy rekomendacji	9
2.1.1	Filtrowanie kolaboratywne (collaborative filtering)	10
2.1.2	Filtrowanie oparte na treści (content-based filtering)	10
2.2	Techniki przetwarzania języka naturalnego	11
2.2.1	Bag-of-words	11
2.2.2	Term frequency - inverted document frequency	12
2.2.3	Latent Semantic Analysis	13
2.2.4	Latent Dirichlet Allocation	14
2.2.5	Word embedding	14
2.2.6	Word2vec	17
2.2.7	Odległość między dokumentami	18
2.2.8	FastText	20
2.2.9	GloVe	22
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 istniejącego 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 stronie aktualnie odwiedzanej przez użytkownika.

W swojej pracy badam możliwość użycia istniejących metod semantycznej analizy tekstu w odniesieniu do opisanego problemu. Badane metody to Word2vec z wariantem Word Mover's Distance oraz bez niego. Jest to stosunkowo nowa

metoda (2013), z czego wynika moje zainteresowanie. Porównuję również działanie wspomnianych wariantów Word2vec z działaniem starszego algorytmu Latent Dirichlet Allocation.

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. Allegro pobiera prowizję za dokonanie sprzedaży za swoim pośrednictwem.

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 artykułów a nie wcześniejsze zachowanie użytkownika.



allegro

czego szukasz?

wszystkie działy

Elektronika Moda i uroda Dom i zdrowie Dziecko Kultura i rozrywka Sport i wypoczynek Motoryzacja Kolekcje i sztuka Firma Sztuka okazji

Allegro - Poradniki - Dom i zdrowie - Jaka farba dla alergika?

Jaka farba dla alergika?

autor: Ewelina Wojtunik, data publikacji: 23-04-2015

Za chwilę wiosna, a wraz z nią potrzeba porządków i odświeżenia ścian. Jak co roku będziemy sprzątać, wietrzyć i wymiatać zimę z kątów mieszkania. Zaraz po tym zaczną się pierwsze remonty.

Czy wiecie, że nawet do 30% populacji to alergicy? Uczulają nas pyłki, kurz, sierść zwierząt, pokarmy, ale przede wszystkim chemikalia i detergenty. Znajdujące się w nich alergeny mogą być powodem problemów zdrowotnych, a także nasilać objawy nadwrażliwości takie jak łzawienie oczu, zapalenie skóry czy kaszel astmatyczny.

Szkodliwe związki lotne

W styczniu 2010 roku Unia Europejska wprowadziła normę, która reguluje zawartość szkodliwych lotnych związków organicznych tak zwanych LZO (VOCs, ang. volatile organic compounds) w trafiających do sprzedaży farbach i lakierach. Warto wiedzieć, że lotne związki lubią pozostawiać aktywne pomimo wyschnięcia farby i starannego wywietrzenia mieszkania. Co więcej, mogą uwalniać się ze ścian całymi latami, nasilając objawy alergiczne i pogarszając samopoczucie mieszkańców. Im mniej ich w składzie, tym lepiej dla nas.

Ewelina Wojtunik

Zawodowo związana z Social Media, pisała m.in. do Aktivist.pl. Prywatnie pasjonatka projektowania wnętrz, zdrowego stylu życia i roślin doniczkowych. Podróże i kuchnie świata są dla niej inspiracją. W wolnym czasie spełnia się jako mama i uczy języków.

może Cię również zainteresować

Wnętrzarski hit – ściany ombre

Ombre stało się hitem w wizażu i modzie już kilka sezonów temu! Chętnie rozjaśniamy końcówki włosów, cieniujemy kolory na paznokciach, a także nosimy ubrania w przenikających si...

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 zobowiązuje się do nieujawniania żadnych danych, które otrzymałem. Stąd opisy danych, na których pracuję, zawarte w tej pracy nie wnikają w ich szczegóły i nieodciągają od informacji publicznie

dostępnych za pośrednictwem strony pod adresem <https://allegro.pl>.

Aktualnie w rzeczonym dziale serwisu Allegro istnieje system rekomendacyjny, który opiera się o wyszukiwanie podobnych artykułów tekstowych za pomocą silnika elasticsearch. Metoda ta opiera się na słowach kluczowych przypisanych do każdego artykułu jako metadane. 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, a nie na dołączonych metadanych.

1.2 Struktura pracy

W rozdziale 2 wprowadzam do zagadnienia rekomendacji i opisuję wybrane metody przetwarzania języka naturalnego służące do wyszukiwania podobieństw między 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 wstępne etapy przetwarzania ich, 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 opisuję proces zastosowania wybranych metod oraz porównanie efektów ich działania.

Ostatecznie w rozdziale 6 dokonuję podsumowania przeprowadzonych badań i rozważam kierunki dalszych prac w tej dziedzinie.

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

1.3 Uwagi

W celu uniknięcia nieporozumień pragnę 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 wybranych metod

W swojej pracy wykorzystuję i adaptuję do swoich potrzeb szereg metod i narzędzi umożliwiających przetwarzanie języka naturalnego, semantyczną analizę tekstu i wykrywanie podobieństwa pomiędzy tekstami. Część z nich (metoda tf-idf, bag-of-words, silnik Elasticsearch) jest od lat powszechnie wykorzystywana w zadaniu wyszukiwania tekstowego. Inne z kolei - korzystające z semantycznej analizy tekstu - nie są tak popularne z powodu swojej nowości, bądź trudności w zaaplikowaniu. Daje to pole do badań i ewentualnych usprawnień istniejących systemów opierających się o klasyczne metody. Wybrane metody stosuję, zgodnie z tematem pracy, w zadaniu generowania rekomendacji, stąd przegląd metod zaczynam właśnie od wprowadzenia do tego zagadnienia.

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 takiego 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 Techniki przetwarzania języka naturalnego

Temat niniejszej pracy skupia się na podobieństwie pomiędzy artykułami, czyli dokumentami tekstowymi. Ich treść zapisana jest w języku naturalnym, zrozumiałym dla człowieka, który jest niezrozumiały dla maszyny. 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 komputer jest w stanie określić podobieństwo pomiędzy dokumentami (według pewnej sformalizowanej miary). W poniższych paragrafach dokonuję przeglądu technik matematycznej reprezentacji dokumentów pisanych w języku naturalnym. Warto wspomnieć, iż dziedzina ta dynamicznie się rozwija a część z opisywanych metod zostało stworzonych na przestrzeni ostatnich kilku lat.

W celu formalizacji dalszych opisach stosowanych metod stosuję następujące oznaczenia:

- Korpus C : zbiór dokumentów d_i ,
- Dokument d : skończony ciąg zdań s_i ,
- Zdanie s : skończony ciąg słów w_i ,
- Słowo w : skończony ciąg znaków c_i ,
- W celu uproszczenia zapisu: $w \in d \equiv \exists_{s \in d} w \in s$,
- Słownik zbudowany na korpusie C : $V = \{w \mid \exists_{d \in C} w \in d\}$.

2.2.1 Bag-of-words

Bag-of-words (worek słów) jest metodą reprezentacji tekstu jako zbioru zawartych w nim słów niezachowującego kolejności słów w tekście, lecz liczbę ich

wystąpień. Metodę tę można opisać jako przekształcenie z korpusu w przestrzeń wektorów $bow : C \rightarrow \mathbb{R}^n$ 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 $i \in 1, 2, \dots, n$ wektor reprezentujący dokument $d_i \in C$

v_{ij} , gdzie $j \in 1, 2, \dots, m$: 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.

2.2.2 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 przypadku 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 frequency” to liczba dokumentów w korpusie podzielona przez liczbę dokumentów zawierających przynajmniej jedno wystąpienie słowa w_i ,

Każdy dokument reprezentowany jest tu przez wektor, składający się z wag słów występujących w tym dokumencie. TF-IDF informuje o częstości wystąpienia słów uwzględniając jednocześnie odpowiednie wyważenie znaczenia lokalnego słowa i jego znaczenia w kontekście pełnej kolekcji dokumentów.

W tej technice słowa występujące rzadko są premiowane względem słów

pospolitych.

2.2.3 Latent Semantic Analysis

LSA, znane również jako Latent Semantic Indexing (LSI) jest techniką przetwarzania języka naturalnego mającą na celu wydobycie powiązań pomiędzy zbiorem dokumentów. LSA zakłada, że istnieje korelacja pomiędzy rozmieszczeniem słów a ich znaczeniem.

LSA dokonuje transformacji każdego dokumentu w wektor dł. $|V|$ z liczbą wystąpień i -tego słowa na i -tej pozycji. W ten sposób tworzona jest macierz: kolumny reprezentują dokumenty a wiersze reprezentują unikalne słowa. W celu identyfikacji istotnych cech tej macierzy dokonuje się rozkład według wartości osobliwych (singular value decomposition, SVD), który jest techniką redukcji wymiarowości. Celem użycia SVD jest redukcja liczby wierszy macierzy 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. $\{(samochd), (ciarwka), (jezdnia)\} \rightarrow \{(1.3452 * car + 0.2828 * truck + 0.3 * jezdnia)\}$. Wymiar uzyskiwanej macierzy jest ustalany za pomocą hiperparametru, który oznacza liczbę tematów. Używając uzyskanej macierzy, podobieństwo pomiędzy kolumnami-dokumentami obliczane jest wykorzystując odległość kosinusową.

Rank lowering[edit] After the construction of the occurrence matrix, LSA finds a low-rank approximation[4] to the term-document matrix. There could be various reasons for these approximations:

The original term-document matrix is presumed too large for the computing resources; in this case, the approximated low rank matrix is interpreted as an approximation (a least and necessary evil"). The original term-document matrix is presumed noisy: for example, anecdotal instances of terms are to be eliminated. From this point of view, the approximated matrix is interpreted as a de-noisified

matrix (a better matrix than the original). The original term-document matrix is presumed overly sparse relative to the "true" term-document matrix. That is, the original matrix lists only the words actually in each document, whereas we might be interested in all words related to each document—generally a much larger set due to synonymy. The consequence of the rank lowering is that some dimensions are combined and depend on more than one term:

This mitigates the problem of identifying synonymy, as the rank lowering is expected to merge the dimensions associated with terms that have similar meanings. It also mitigates the problem with polysemy, since components of polysemous words that point in the "right" direction are added to the components of words that share a similar meaning. Conversely, components that point in other directions tend to either simply cancel out, or, at worst, to be smaller than components in the directions corresponding to the intended sense.

This means that if we have an LSI representation of a collection of documents, a new document not in the collection can be "folded in" to this representation using Equation 244. This allows us to incrementally add documents to an LSI representation. Of course, such incremental addition fails to capture the co-occurrences of the newly added documents (and even ignores any new terms they contain). As such, the quality of the LSI representation will degrade as more documents are added and will eventually require a recomputation of the LSI representation.

2.2.4 Latent Dirichlet Allocation

2.2.5 Word embedding

W ostatnim czasie nastąpił gwałtowny rozwój metod „word embedding” - osadzania słów w przestrzeni wektorowej. Słowo języka naturalnego samo w sobie jest tylko pewnym symbolem, tokenem - nie da się wprost zastosować dla niego żadnych matematycznych metod, które pozwoliłyby odwzorować semantyczne zależności pomiędzy słowami i wykorzystać je do

pewnych wyższych celów. Metody osadzania słów pozwalają na wektorową reprezentację słów, które są w stanie zachować semantyczne i syntaktyczne powiązania między słowami. Przykładem mogą być tu operacje arytmetyczne na wektorach odwzorowujące semantykę słów: $vector("krl") + vector("mczyzna") + vector("kobieta") \approx vector("krlowa")$

Wcześniejsze metody opierały się na głębokich sieciach neuronowych (Jednak istotną wadą tych rozwiązań pozostaje niska wydajność ich nauki, co jest szczególnie odczuwalne w przypadku ogromnych zbiorów 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, których czas nauki jest na tyle krótki, że sprawdzają się one do zastosowań komercyjnych.

Wyróżnia się dwa podstawowe podejścia generowania wektorowych reprezentacji słów:

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 konstruuje 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. Dwie z nich: metodę centroidu oraz Word Mover's Distance opisuję poniżej.

The two main model families for learning word vectors are: 1) global matrix factorization methods, such as latent semantic analysis (LSA) (Deerwester et al., 1990) and 2) local context window methods, such as the skip-gram model of Mikolov et al. (2013c). Currently, both families suffer significant drawbacks. While methods like LSA efficiently leverage statistical information, they do relatively poorly on the word analogy task, indicating a sub-optimal vector space structure. Methods like skip-gram may do better on the analogy task, but they poorly utilize the statistics of the corpus since they train on separate local context windows instead of on global co-occurrence counts.

Unlike the matrix factorization methods, the shallow window-based methods suffer from the disadvantage that they do not operate directly on the co-occurrence statistics of the corpus. Instead, these models scan context windows across the entire corpus, which fails to take advantage of the vast amount of repetition in the data.

Semantic vector space models of language represent each word with a real-valued vector. These vectors can be used as features in a variety of applications, such as information retrieval (Manning et al., 2008), document classification (Sebastiani, 2002), question answering (Tellex et al., 2003), named entity recognition (Turian et al., 2010), and parsing (Socher et al., 2013)

Learning continuous representations of words has a long history in natural language processing (Rumelhart et al., 1988).

These representations are typically derived from large unlabeled corpora using co-occurrence statistics (Deerwester et al., 1990; Schütze, 1992; Lund and Burgess, 1996).

In the neural network community, Collobert and Weston (2008) proposed to learn word embeddings using a feedforward neural network, by predicting a word based on the two words on the left and two words on

the right.

2.2.6 Word2vec

Word2vec jest stosunkowo nową (2013) metodą osadzania słów w przestrzeni wektorowej (word embedding), 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. 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 przewiduje N sąsiednich słów lub CBOW: na podstawie okna N sąsiednich słów sieć przewiduje słowo, którego z największym prawdopodobieństwem te N słów jest sąsiedztwem. 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 $j = 1, \dots, K$. 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 słów.

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 słów 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 $M \times N$, 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 używana funkcja aktywacji, 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.

2.2.7 Odległość między dokumentami

W celu wykorzystania omówionej metody Word2vec w obszarze tematyki pracy 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 wewnątrz 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: $P = \{f(p_1, w_{p_1}) \dots (p_m, w_{p_m})\}$, $Q = \{f(q_1, w_{q_1}) \dots (q_n, w_{q_n})\}$ 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 całosciowy koszt $Work(P, Q, F) = \sum_{i=1}^m \sum_{j=1}^n d_{ij} f_{ij}$ przy odpowiednich ograniczeniach[11]. 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 .

2.2.8 FastText

Most of these techniques represent each word of the vocabulary by a distinct vector, without parameters sharing. In particular, they ignore the internal structure of words, which is an important limitation for morphologically rich languages, such as Turkish or Finnish. These languages contain many words that occur rarely,

making it difficult to learn good word-level representations. In this paper, we propose to learn representations for character n-grams, and represent words as the sum of the n-gram vectors. Our main contribution is to introduce an extension of the continuous skip-gram model (Mikolov et al., 2013b), which takes into account subword information. We evaluate this model on five different languages, with various degree of morphology, showing the benefit of our approach.

Rozszerzeniem koncepcji word2vec jest opracowana w 2016 przez Facebook AI Research metoda fastText[?]. Główną innowacją tej metody jest wzięcie pod uwagę wewnętrznej struktury słów. Jest to szczególnie obiecujące w przypadku bogatych morfologicznie języków, mp. języka polskiego.

Metoda fastText w dużej mierze opiera się na zasadzie działania metody word2vec. Najistotniejszą różnicą jest sposób poziom szczegółowości analizy słów. Word2vec traktuje słowo jako niepodzielną jednostkę, ignorując wewnętrzną strukturę jego znaków. Zgodnie z opisem z powyższego paragrafu każde słowo przygotowywane jest do wejścia do sieci neuronowej w postaci unikalnego wektora wypełnionego zerami z wartością '1' na jednej pozycji („one-hot-vector”). FastText natomiast analizuje każde słowo pod kątem struktury jego znaków. Wykorzystuje do tego rozbięcie słowa na podsłowa - ciągi znaków o określonej długości „character n-grams”.

Słowo reprezentowane jest jako suma wektorowych reprezentacji jego n-gramów.

In this section, we thus propose a different scoring function s , in order to take into account this information. Given a word w , let us denote by $G_w = \{g_1, \dots, g_G\}$ the set of n-grams appearing in w . We associate a vector representation z_g to each n-gram g . We represent a word by the sum of the vector representations of its n-grams. We thus obtain the scoring function $s(w, c) = \sum_{g \in G_w} z_g \cdot v_c$. We always include the word w in the set of its ngrams, to also learn a vector representation for each word. The set of n-grams is thus a superset of the vocabulary. It should be noted that different vectors are assigned to a word and a n-gram sharing the same sequence of characters. For example, the word *as* and the bigram *as*, appearing in the word *paste*, will be assigned to different vectors. This simple model allows sharing the

representations across words, thus allowing to learn reliable representation for rare words. Dictionary of n-grams. The presented model is simple and leaves room for design choices in the definition of Gw. In this paper, we adopt a very simple scheme: we keep all the n-grams with a length greater or equal than 3 and smaller or equal than 6. Different sets of n-grams could be used, for example prefixes and suffixes. We also add a special character for the beginning and the end of the word, thus allowing to distinguish prefixes and suffixes. In order to bound the memory requirements of our model, we use a hashing function that maps n-grams to integers in 1 to K. In the following, we use K equal to 2 millions. In the end, a word is represented by its index in the word dictionary and the value of the hashes of its n-grams. To improve the efficiency of our model, we do not use n-grams to represent the P most frequent words in the vocabulary. There is a trade-off in the choice of P, as smaller values imply higher computational cost but better performance. When $P = W$, our model is the skip-gram model of Mikolov et al. (2013b).

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

2.2.9 GloVe

Algorytm GloVe różni się od word2vec w sposobie uzyskania wektorowej reprezentacji słów. Word2vec jest modelem predykcyjnym, natomiast GloVe opiera się na globalnej macierzy współwystąpień słów.

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:

$$f(X_{ij}) = \begin{cases} (\frac{X_{ij}}{x_{max}})^\alpha & \text{if } X_{ij} < XMAX \\ 1 & \text{otherwise} \end{cases}$$

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 podpisaniu 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 hierarchia 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 mechanizmie 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 konkretnego przypadku, zgodnie z intuicją.

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

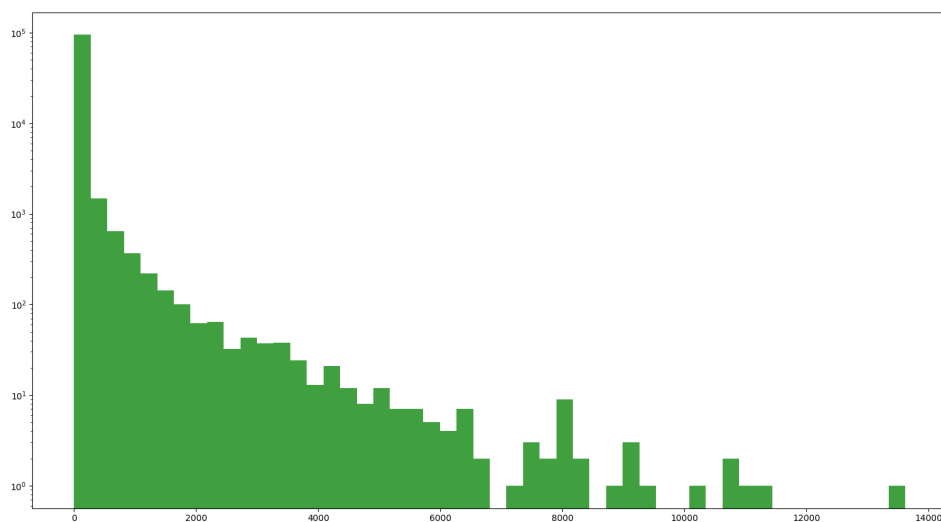
1. Oczyszczanie tekstu ze zbędnych, wspomnianych wcześniej znaczników. Z punktu widzenia semantycznej analizy tekstu są one bezużyteczne, czy wręcz szkodliwe (powodują pewne „zanieczyszczenie” tekstu). Stąd usuwam je wykorzystując odpowiednio skonstruowane wyrażenia regularne (ich postać jest szczegółem 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 stosuje narzędzie Morfologik[4].

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

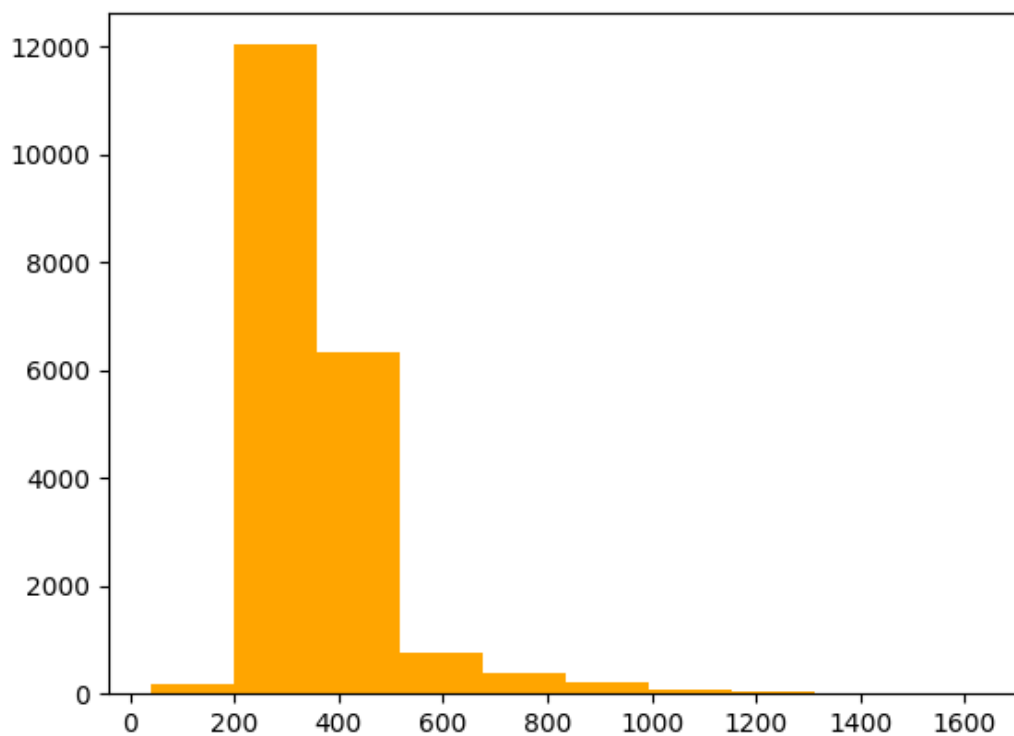
3.3 Opis danych po wstępnym przetwarzaniu

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 zależy jakość całej miary.

Działanie testowanych metod można sformalizować 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 artykułów 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 stosuje średnią ważoną podobieństwa kolejnych elementów wyjścia z wejściem. Stosowane wagi: $\frac{1}{i}$ dla $i = 1, \dots, N$, gdzie N to długość ciągu wyjściowego danej metody.

4.1 Miara 1: Dystans oparty na metadanych

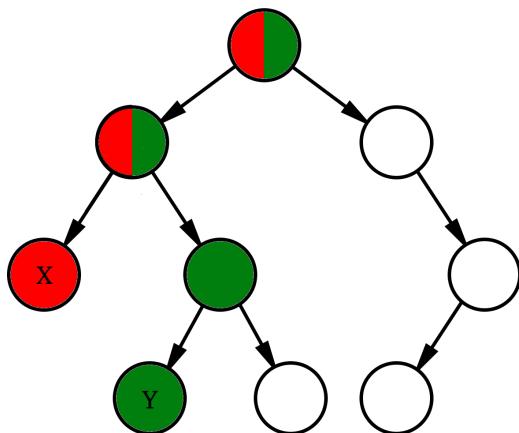
Jak wspomniałem wcześniej dane prócz treści artykułów zawierają również pewne metadane, a wśród nich umożliwiające tworzenie powiązań między artykułami. Skupiam się tu na polach: „słowa kluczowe” i „kategoria”.

4.1.1 Kategorie

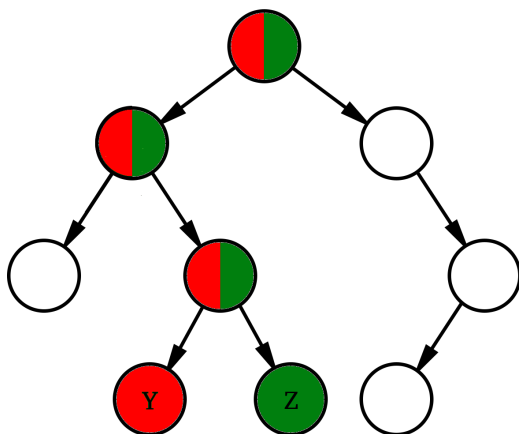
Pierwszą zastosowaną miarą, pozwalającą ocenić jakość dopasowania podobnych artykułów jest ich odległość we wcześniej wspomnianym drzewie kategorii. Zakładam tu, że im więcej wspólnych przodków w drzewie, tym bardziej podobne do siebie są artykuły reprezentowane przez węzły drzewa. Zaletą miary jest 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 odległymi 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 $d(X, Y) = \frac{1}{3}$, $d(X, Z) = \frac{2}{3}$. 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 operujących 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 odnośniki 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 Y_1 , a ile razy na rekomendowany artykuł Y_2 . Jeżeli liczba przejść na artykuł Y_1 jest większa niż na Y_2 , można wnioskować, iż Y_1 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 + podaje źródło

4.3.2 Adaptacja metody nDCG

Założmy, że dany algorytm A zwraca pewien ciąg artykułów $c_A = a_1, a_2, \dots, a_6$ podobnych do danego artykułu x , w kolejności od najbardziej adekwatnego. Założmy 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 . Założmy 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 częstość 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_cent] Word2Vec z modelem[16] uczonym na korpusie Słownosieci[17] (model opisany jest poniżej) oraz odległościami między dokumentami liczonymi na bazie centroidu dokumentu.

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

[w2v_art_cent] 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 oceną podobieństwa artykułów [random].

5.1.1 Modele Word2Vec

Model uczony na korpusie Słownosieci

Jako podstawowy model Word2vec użyłem gotowego modelu[16] stworzonego m.in. przez dr inż. M. Piaseckiego. Model ten był uczony na korpusie Słownosieci ver. 10. Dane przed uczeniem przeszły segmentację, lematyzację i ujednolicanie morfosyntaktyczne. Użyte parametry uczenia Word2Vec: metoda skip gram, wektory 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 stosunku 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 Słownosieci 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.

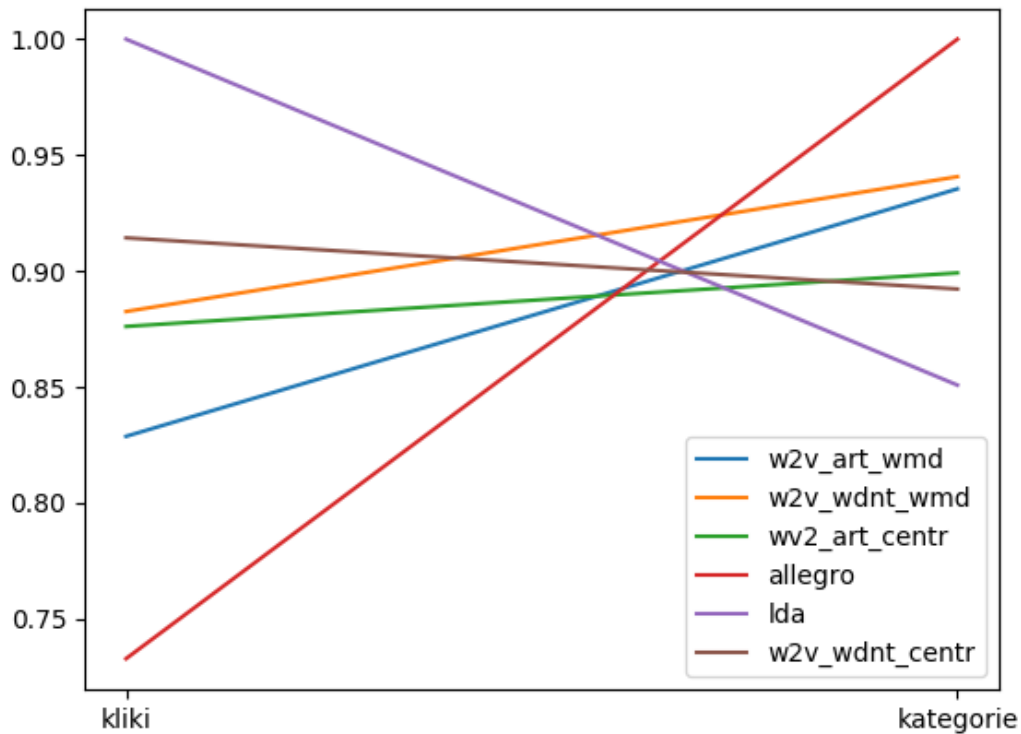
Do uczenia użyłem parametrów identycznych, jak w metodzie powyżej.

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_centra	1.37170817357	0.521784904438	
w2v_wdnt_wmd	1.3817891755	0.54587787496	
w2v_wdnt_centra	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. Złożoność czasowa metody wynosi: Dla przypadku: [w2v_art_wmd] obliczenia trwały 55 minut, co przy czasie <1sek dla centroidu jest wartością niedopuszczalną. Proporcjonalnie użycie tej metody dla całego korpusu trwałoby ok. 183 godzin.

Istotną zaletą dotychczasowego rozwiązania stosowanego w Allegro jest uniwersalność silnika elasticsearch oraz to, że pozwala edytować indeksowane 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 rekomendacje, tj. takie, które przynoszą przedsiębiorstwu największy zysk, wcale nie muszą być podobne do danego. Powszechnym zjawiskiem jest wzbogacanie rekomendacji 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 skryptów 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

- [11] Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas, *The Earth Mover's Distance as a Metric for Image Retrieval*, str. 8, 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)

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.

.....