Automatyczna klasyfikacja i ekstrakcja tematu krótkich notatek w języku polskim

Paweł Obrok
pod kierunkiem dr. Michała Korzyckiego

29 września 2012

Oświadczenie prawnoautorskie

Strona tytułowa po angielsku

Spis treści

1 Wstęp

Poniższy rozdział przybliża temtykę pracy oraz opisuje jej zawartość.

1.1 Wprowadzenie

Metody automatycznego przetwarzania tekstu nabierają w dzisiejszym świecie coraz większego znaczenia — jesteśmy zalewani przez powodź informacji, a większość z nich ma właśnie postać tekstową. Aby poradzić sobie z tym ogromem napływających bodźców próbujemy uciekać się do mechanizmów, które pozwolą je przeanalizować albo przynajmniej wstępnie przetowrzyć. Technologie w rodzaju wyszukiwarek internetowych, agregatorów wiadomości czy baz filmów lub książek potrafiących wskazać dzieła podobne, do tych, które nam się podobały stają się coraz popularniejsze i często koniecznie do znalezienia porządanej informacji.

Wspomniane mechanizmy zajmują się głównie segregowaniem czy przeszukiwaniem danych, ale wyobrażalne są również zadania na wyższym poziomie koncepcyjnym. Chcielibyśmy, aby odpowiednio zaprogramowane maszyny były w stanie rozumieć tekst tak jak ludzie i przetwarzać go nie tyle jako ciągi znaków ile jako idee, opisy zdarzeń, itp. wyrażone tymi znakami. Zadania takie jak automatyczne tłumaczenie, generowanie streszczeń lub parafrazowanie tekstów wydają się na tym etapie wymagać takiego właśnie zrozumienia i nadal ich sprawne wykonywanie pozostaje wyłączną domeną człowieka, mimo postępujących prac w tych dziedzinach.

Naczelnym problemem, który należy przezwyciężyć w tego typu zagadnieniach jest fakt, że słowo pisane, a ogólniej język odnoszą się do pewnej rzeczywistości zewnętrznej, a wręcz jest to ich zasadniczą funkcją. Maszyny, które przetwarzają tekst nie mają zwykle bezpośredniej styczności z tą rzeczywistością, starając się operować jedynie na ciągach symboli. Ujawnia się tutaj przewaga w tym polu ludzi, którzy mają wiele lat na zbieranie informacji o świecie i o tym w jaki sposób odnosi się do niego język, którym się posługują.

Istnieje wiele kierunków badań mających rzucić więcej światła na rozwiązanie tego problemu. Najbardziaj ambitne zakładają, że konieczne jest tutaj stworzenie uniwersalnej sztucznej inteligencji czyli maszyny o możliwościach podobnych ludzkiemu umysłowi, która miałaby dostęp do bodźców zmysłowych dokładnie tak jak człowiek i byłaby w stanie w jakimś sensie w stanie rozumieć język i rzeczywistość, którą opisuje.

Nieco mniej radykalne podejście sugeruje budowę ontologii, czyli sformalizowanego opisu znaczeń wyrazów i powiązań między nimi wraz z typami tych powiązań. Przykładowo mogłaby ona zawierać wpis „tryb part-of zegar", reprezentujący fakt, że zegary często zawierają tryby jako części. Taka baza wiedzy opisuje znaczenie słów niejako przez przykłady

kontekstów, w których mogą występować i pozwala dokonywać rozumowania na temat tekstu w rodzaju wykrywania synekdochy (wykorzystywania całości na oznaczenie części lub odwrotnie, np. „Hollywood" zamiast „Amerykański przemysł filmowy") co umożliwi odkrycie jego prawdziwego znaczenia.

Jeszcze inne podejście, któremy poświęcona jest ta praca, polega na próbie wykrywania powiązań w samym analizowanym tekście. U jej podstaw leży obserwacja, że wyrazy będą pojawiać się częściej w kontekście wyrazów związanych z nimi znaczeniowo, przykładowo „żaba" może pojawić się blisko „skakać" czy „pływać", rzadziej „biegać". Znów daje nam to zestaw kontekstów dla danego wyrazu, tym razem jednak należy go wyekstrahować z tekstu, a częściej zbioru tekstów, natomiast konteksty dla słów, które w takim korpusie się nie pojawią nie są w ogóle dostępne. Metody tego rodzaju mają jednak tę zaletę, że operują jedynie na tekście, nie potrzebują dodatkowych informacji jak przedstawione wyżej podejścia.

Najprostszą wersją tego podejścia jest Vector Space Modelling - metoda polegająca na reprezentowaniu dokumentów jako wektorów, których współrzędne odpowiadają w jakiś sposób częstościom występowania w nich pewnych wyrazów. Pozwala to na porównywanie dokumentów metodami algebry liniowej, na przykład ocenianie podobieństwa dokumentów przez pomiar kąta między odpowiadającymi im wektorami.

VSM wykazuje niestety znaczne wady, przykładowo jest całkowicie nieczułe na istnienie synonimów albo możliwość występowania słowa w więcej niż jednym znaczeniu. Co więcej duża część wyrazów w dokumentach, to z semantycznego punktu widzenia szum, przykładowo znaki przestankowe, chociaż występują dość często nie są zwykle kluczowe do zrozumienia tekstu. Aby uporać się z tymi wadami proponuje się różne udoskonalenia tej metody, ta praca zajmuje się porównaniem dwóch z nich — Latent Semantic Indexing oraz Latent Dirichlet Analysis.

LSI to metoda pochodząca wprost z algebry liniowej. Sprowadza się ona do wprowadzenia nowej, mniejszej bazy do przestrzeni wektorów odpowiadających dokumentom, w taki jednak sposób, aby zachować jak najwięcej informacji zawartych w oryginalnej reprezentacji. Ma to wyeliminować szum, a także wykryć związki między wyrazami przez ich wspólne występowanie w wektorach nowej bazy.

LDA stawia sobie za cel przedstawienie przynajmniej częściowego probabilistycznego modelu powstawania dokumentów i założywszy poprawność tego modelu znalezienie jego parametrów dla danego korpusu. Ze względu na specyfikę tego modelu, konkretnie założenie o tym, że na każdy tekst składa się jeden lub więcej tematów i pojawienie się w nim każdego wyrazu można przypisać jednemu z tych tematów, również pozwala on wykrywać zależności między słowami. W tym wypadku zależności te będą zadane przez występowanie słów w tym samym temacie.

Zarówno LSI jak i LDA mogą być wykorzystane w szeregu zastosowań związanych z automatycznym przetwarzaniem tekstu, takich jak wyszukiwanie dokumentów w korpusie, automatyczna ekstrakcja słów kluczowych czy ocena podobieństwa dokumentów. Praca ta stawia sobie za zadanie porównanie działania tych metod w tego typu zadaniach.

Istnieją już porównania LSI i LDA, jak choćby [6], jednak traktują one głównie o języku angielskim, który ma względnie prostą fleksję. Wyniki te nie zawsze przenoszą się na inne języki, jak choćby polski — wykorzystany do testów w tej pracy — gdzie każdy wyraz występuja w bardzo wielu formach. Aby rozwiązać ten problem zastosowano wraz z metodami LSI i LDA słownik fleksyjny języka polskiego. Poddano także porównaniu wyniki osiągane z i bez jego zastosowania.

1.2 Zawartość pracy

Rozdział 2 poświęcony jest teoretycznemu opisowi porównywanych metod jak i metryk stosowanych do ich oceny. W rozdziale 3 opisany został zbiór danych użytych do testów i wygenerowany przy jego użyciu testowy problem, natomiast w rozdziale 4 opisane są mechanizmy użyte do jego rozwiązania. Rozdział 5 zawiera uzyskane wyniki numeryczne wraz z ich omówieniem, a rozdział 6 podsumowuje pracę i proponuje dalsze ścieżki jej rozwoju.

Dodatek A tej pracy omawia najczęściej pojawiające się w niej terminy i oznaczenia. Ponadto w dodatku B podano sposób wykorzystania kodu użytego do uzyskania wyników przedtstawionych w tej pracy. W dodatku C zawarto pełną tabelę tematów wygenerowanych dla przykładowych danych.

2 Podstawy teoretyczne

W tym rozdziale opisana jest ogólnie koncepcja wektorowego modelowania dokumentów oraz metody, które poddano porównaniu w tej pracy, a które od tej koncepcji pochodzą. W dalszej części rozdziału zawarto opis metryk perpexity, dokładności i kompletności wykorzystywanych przy ocenie działania modeli językowych i systemów wyszukiwania informacji.

2.1 Vector Space Model

Model przestrzeni wektorowej (Vector Space Model — VSM) to metoda modelowania języka polegająca na reprezentowaniu dokumentów jako wektorów co umożliwia na przykład porównywanie ich metodami algebry liniowej. Wektor dla i-tego dokumentu ma postać:

$$d_i = (w_0^i, w_1^i, \ldots, w_N^i) \tag{1}$$

gdzie wymiar w_k^i odpowiada k-temu tokenowi z słownika. Współczynniki te nazywane są także wagami i mają oddawać znaczenie danego tokenu w danym dokumencie — powinny być tym wyższe im bardziej kluczowy dla sensu i-tego dokumentu jest k-ty token. Przyjmuje się zwykle, że wymiary odpowiadające tokenom nie występującym w danym dokumencie są zerowe.

2.1.1 Schemat wagowy

Najbardziej ogólnie wagi w_k^i uzyskuje się przez pomnożenie wagi globalnej g_k, która ma odzwierciedlać popularność tokenu w całym korpusie i wagi lokalnej l_k^i, która opisuje stricte znaczenie tokenu w danym dokumencie. Tabele 1 i 2 podsumowują popularnie stosowane wagi globalne i lokalne. Ich drobiazgowe porównanie dla języka polskiego znaleźć można w [8]. Dobór tych dwóch wag nazywamy schematem wagowym.

Konkretny schemat wagowy dobiera się empirycznie, często pod jakąś klasę problemów. Dla zadań typu information retrieval dobrze sprawdzają się lokalne wagi wygładzone i entropia jako waga globalna z czego można wywnioskować, że obserwacja o mniejszym znaczeniu kolejnych wystąpień tokenu w tym samym dokumencie jest trafna.

Brak wagi globalnej daje z reguły znacznie gorsze wyniki niż pozostałe możliwości. Jej zastosowanie pozwala automatycznie wykryć wyrazy takie jak spójniki, które występują bardzo często we wszystkich tekstach, a nie mają dużego ładunku semantycznego. W korpusach dotyczących jednej dziedziny mogą się pojawić inne wyrazy o takiej charakterystyce,

Tablica 1: Możliwe wagi globalne g_k

1	Brak wagi globalnej — częstość występowania tokenów w całym korpusie jest ignorowana
$\frac{1}{\sum_j (l_k^j)^2}$	Normalizacja — tutaj przez sumę kwadratów wag lokalnych tokenu w całym korpusie
$\frac{gf_k}{df_k}$	gf-idf — waga tym większa im więcej razy występuje dany token i tym mniejsza im więcej dokumentów go zawiera
$log(\frac{n}{df_k}) + 1$	idf — logarytm z liczby dokumentów w których występuje dany token znormalizowanej przez liczbę wszystkich dokumentów
$1 - \sum_{i/0}^{n} \frac{tf_k^i log(tf_k^i)}{log(n)}$	Entropia — waga pochodząca z teorii informacji, ilość informacji którą daje pojawienie się danego tokenu o numerze tekstu, w którym to nastąpiło

Tablica 2: Możliwe wagi lokalne l_k^i

1	Waga binarna — zachowana zostaje jedynie informacja o fakcie wystąpienia tokenu, liczba wystąpień jest ignorowana
t_k^i	Częstość występowania — najprostsza waga, można powiedzieć, że inne są próbą jej usprawnienia przez odpowiednid wygładzenie
$\sqrt{t_k^i}$	Wygładzona częstość występowania — aby oddać fakt, kolejne wystąpienia tego samego wyrazu dają stosunkowo mniejszą wskazówkę co do jego znaczenia dla danego dokumentu można wygładzić częstość występowania pierwiatskiem lub logarytmem

przykładowo wyraz „wyraz" występuje bardzo często w tej pracy i najprawdopodobniej nie niesie wiele informacji o tematyce poszczególnych jej akapitów, w których się pojawia.

Podobnie zastosowanie binarnej wagi lokalnej negatywnie wpływa na jakość otrzymywanych wyników. Utrata informacji na temat dokumentu jest w tym wypadku zbyt duża, dodatkowe wystąpienia tokenu mimo iż nie tak znaczące jak pierwsze są jednak wskazówką co do większego znaczenia tego tokenu w tekście.

2.1.2 Podobieństwo dokumentów

Aby ocenić podobieństwo dwóch dokumentów w modelu przestrzeni wektorowej stosuje się miarę kosinusową — kosinus kąta między wektorami reprezentującymi te dokumnety. Dla wektorów dokumentów d^i

i d^j wynosić ona będzie:

$$d(d^i, d^j) = \langle \frac{d_i}{||d^i||}, \frac{d^j}{||d^j}\rangle \tag{2}$$

Gdzie $\langle \bullet, \bullet \rangle$ oznacza iloczyn skalarny, a $||d|| = \{d, d\}$. Daje to liczbę z przedziału $[-1, 1]$, gdzie -1 oznacza dokumenty skrajnie różne, a większe wartości coraz większe podobieństwo. Wartość 1 osiągnięta zostanie dla dwóch dokumentów, w których względna częstość występowania słów (ewentualnie znormalizowana) jest taka sama.

2.1.3 Wady

Zgodnie z przedstawionym mechanizmem działania wszystkie wyrazy, które mają różne postaci w zapisie traktowane są jako całkowicie osobne wymiary przestrzeni wektorowej, pomijana jest możliwość występowania wyrazów w wielu formach (fleksja), możliwość oznaczenia tego samego pojęcia przez wiele różnych ciągów symboli (synonimy) oraz możliwość, że dany ciąg symboli może opisywać kilka różnych pojęć (polisemia).

Aby mitygować te niedociągnięcia stosowane są różne rozszerzenia modelu przestrzeni wektorowej. Opisany w rozdziale 4.2.2 słownik fleksyjny pozwala sprowadzić wyrazy do formy podstawowej co „spłaszczy" je na powrót w jeden wymiar. Opisane w dalszych dwóch rozdziałach metody redukcji wymiarowości macierzy pozwalając wyekstrahować najczęściej pojawiające się kombinacje wyrazów i częściowo zredukować problem synonimów. Aby radzić sobie z polisemią możliwe jest zastosowanie mechanizmów kontekstowego ujednoznaczniania wyrazów.

2.2 Latent Semantic Indexing

Metoda LSI to rozszerzenie modelu przestrzeni wektorowej mające na celu uzyskanie lepszych wyników dzięki automatycznej identyfikacji powiązań między wyrazami występującymi w dokumencie przez redukcję wymiaru macierzy term-dokument. Rozwiązuje to wspomniany wcześniej problem synonimów i do pewnego stopnia polisemii. Istnieje przy tym nadzieja, że przez odpowiednie dobranie mechanizmu redukcji wymiaru macierzy usunięty zostanie szum przy jednoczesnym zachowaniu zawartej w niej informacji.

2.2.1 Intuicja

Dzięki temu, że traktujemy dokumenty jako elementy przestrzeni wektorowej możliwe staje się wprowadzenie dowolnej bazy w tej przestrzeni i reprezentowanie wektorów przy jej użyciu. Jedną ze stosowanych do tego metod jest SVD. SVD stara się znaleźć taką bazę przestrzeni, żeby jej

kolejne współrzędne posorotwane były w kolejności malejącej wariancji wektorów z danego zbioru (w tym wypadku korpusu) w ich kierunku. Następnie po przeniesieniu wektorów do tej bazy możliwe jest odcięcie współrzędnych dalszych niż n-ta przy utracie minimalnej ilości informacji.

W przypadku zastosowania SVD do przestrzeni dokumentów wektory nowej bazy można traktować właśnie jako tematy dokumentów z korpusu. Każdy z nich będzie opisywał kierunek w starej przestrzeni, wzdłuż którego dokumenty są jak najbardziej różne, a przez to wskazuje on na powiązanie między swoimi składowymi przez ich skorelowanie z tą właśnie różnicą.

2.2.2 Singular Value Decomposition

SVD to sposób rozkładu macierzy na iloczyn macierzy o pewnych szczególnych własnościach, które okazują się przydatne w zastosowaniach związanych na przykład z przetwarzaniem sygnałów. Każdą macierz M o wartościach będących liczbami zespolonymi można przedstawić jako:

$$M = U\Sigma V^* \tag{3}$$

Przy czym macierz Σ jest macierzą diagonalną, która na przekątnej zawiera wartości osobliwe macierzy M, macierz U zawiera lewe wektory osobliwe M, a macierz V^* jest macierzą sprzężoną do macierzy zawierającej prawe wektory osobliwe M.

Tak zdefiniowany rozkład pozwala na redukcję rzędu macierzy zgodnie z twierdzeniem Eckharta-Younga, które głosi, że macierz o rzędzie k, najlepiej przybliżająca M w sensie normy Frobeniusa czyli pierwiastka sumy kwadratów różnic odpowiadających sobie elementów macierzy jest zadana przez:

$$M \approx \tilde{M} = U\tilde{\Sigma}V^* \tag{4}$$

Gdzie macierz $\tilde{\Sigma}$ uzyskuje się poprzez wybranie k największych wartości osobliwych i zastąpienie pozostałych zerami w macierzy Σ.

2.3 Latent Dirichlet Allocation

Podczas gdy LSI ma u swoich podstaw metodę algebraiczną redukcji wymiarowości macierzy przy utracie jak najmniejszej ilości zawartej w tej macierzy informacji intuicja stojąca za LDA jest znacznie bliżej spokrewniona z wyobrażeniami na temat powstawania dokumentów tekstowych. Jak zostanie jednak przedstawione w tym rozdziale ostatecznie LDA również sprowadza się do redukcji wymiaru macierzy term-dokument przez

wprowadzenie nowej przestrzeni, w której reprezentowane będą dokumenty.

Metoda LDA została zaproponowana przez Davida Blei et al. w 2003 roku w [6].

2.3.1 Intuicja

LDA to probabilistyczny model generatywny opisujący powstawanie dokumentów. Zakłada, że dokumenty są zbiorami wyrazów i kolejność ich występowania nie ma znaczenia (założenie bag-of-words). Zakłada następnie, że każdy dokument traktuje o małej liczbie tematów z pewnego zbioru tematów, których liczba jest znana *a priori*.

Proces generowania dokumentów przebiega przez powtarzanie dwuetapowego losowania: najpierw z dystrybucji tematów dla danego dokumentu losowany jest temat, a następnie z dystrybucji wyrazów w tym temacie losowany jest kolejny wyraz.

2.3.2 Dobieranie parametrów modelu

Something here

2.4 Perplexity

Współczynnik perplexity, który zaproponowany został do oceny modeli języka [11], ma oddawać niejako poziom "zaskoczenia" niewidzianymi dotąd danymi. Dla rozkładu prawdopodobieństwa p definiujemy go jako eksponentę entropii p, czyli:

$$2^{H(p)} = 2^{-\sum_x p(x) log_2 p(x)} \tag{5}$$

Dla dyskretnej dystrybucji o n równoprawdopodonych możliwych wydarzeniach współczynnik perplexity wyniesie n — daje to możliwość porównania innego rodzaju dytrybucji (rónież ciągłych — zamieniwszy sumę we wzorze 5 na całkę) pod względem ich „rozstrzelenia", a co więcej daje dodatkową intuicję, że dana dystrybucja będzie generować dane, które będą podobnie „zaskakujące" jak rzut n-ścienną kostką.

Jeżli chcemy wykorzystać perplexity do oceny modelu jakiegoś zjawiska q, który przypisuje zdarzeniom prawdopodobieństwa $q(x)$, to możemy obliczyć znormalizowany współczynnik perplexity dla pewnego zbioru obserwacji tego zjawiska $\{x_1, x_2, \ldots, x_N\}$ następująco:

$$2^{H(q)} = 2^{-\frac{1}{N}\sum_i^N log_2 p(x_i)} \tag{6}$$

Zaletą tego podejścia jest możliwość całkowicie automatycznej oceny jakości modelu — modele lepiej opisujące dane zjawisko będą przypo-

rządkowywać wyższe prawdopodobieństwa wydarzeniom, które rzeczywiście zachodzą i co za tym idzie będą miały niższy współczynnik perplexity. Zwykle taka ewaluacja stosowana jest do upewnienia się, że model nie jest przeuczony, że jest w stanie uogólniać swoje przewidywania na sytuacje inne niż dane treningowe. W takim wypadku alalizuje się współczynnik perplexity obliczony dla dodatkowego zbioru danych, które nie były wykorzystane do treningu i przykładowo zatrzymuje interacyjny proces uczenia modelu, kiedy zaczyna on wzrastać.

W modelach języka jako pojedyncze zdarzenia stosuje się wyrazy lub zdania. Przykładowo [7] podaje współczynnik perplexity 247 dla korpusu Browna, zawierającego zróżnicowane teksty w języku algielskim osiągnięty przez model trigramowy. Jest to równoważne przypadkowi, gdy każde słowo jest wybierane niezależnie z równym prawdopodobieństwem spośród 247 możliwości.

2.5 Dokładność i kompletność

Aby ocenić i porównywać wyniki działania algorytmów wyszukujących dokumenty w korpusach należy wprowadzić formalne metryki opsisujące jakość wyników przez nie uzyskiwanych. Należy przy tym zauważyć, że znaczenie ma nie tylko fakt zwrócenia wielu czy nawet wszystkich dokumentów skojarzonych z zapytaniem, ale także liczba zwróconych dokumentów nieskojarzonych, gdyż stanowią one szum i wyniki zaweierające wiele takich dokumentów wymagają dalszego, ręcznego przetwarzania. Aby uchwycić te dwie interesujące nas cechy stosuje się zazwyczaj dwie metryki: dokładność i kompletność.

Kompletność ma wyrażać jak dużo interesujących wyników zostało odnalezione przez algorytm, obliczana jest według wzoru:

$$K = \frac{TP}{TP + FN} \tag{7}$$

W tym wzorze licznik, to liczba poprawnie zwrócocnych dokumentów, a mianownik to po prostu liczba wszyskich dokumentów skojarzonych, które występują w danym korpusie.

Dokładność opisuje udział skojarzonych dokumentów w zwróconych wynikach, formalnie opisuje ją wzór:

$$D = \frac{TP}{TP + FP} \tag{8}$$

Te dwie metryki stoją do siebie w pewnym sensie w opozycji — w wielu systemach możliwe jest sterowanie progiem detekcji i zwiększenie kompletności kosztem dokładności lub odwrotnie. Z tego też powodu analizuje się je zwykle w tandemie i interesujące są nie tyle ich wartości jako takie, co zależność między nimi.

3 Opis danych

Działanie algorytmów LDA i LSI analizowane było na zbiorze 51574 notatek Polskiej Agencji Prasowej. Notatki te to krótkie wiadomości tekstowe (średnio mające 409 znaków), z których większość dotyczy pojedynczego wydarzenia. Dotyczą one różnych dziedzin życia takich jak sport, polityka, gospodarka, sprawy obyczajowe, etc., co nadaje temu zbiorowi dodatkową różnorodność i pozwala oczekiwać, że osiągnięte wyniki będą miarodajne dla efektywności algorytmów w prawdziwych zastosowaniach.

3.1 Słownik

W korpusie znajduje sie w sumie 2846997 tokenów - wyrazów, liczb, znaków interpunkcyjnych i innych, wśród nich 443935 różnych form (pozostałe są powtórzeniami). Po sprowadzeniu wyrazów od form podstawowych oraz zastosowaniu podstawień w rodzaju użycia tokenu „NUMBER" zamiast konkretnych liczb pozostaje 70536 różnych form. Z tych odrzucono występujące tylko w jednym dokumencie, występujące w więcej niż w 70% dokumentów, oraz tokeny mające małe znaczenie semantyczne takie jak spójniki i znaki interpunkcyjne, co dało ostateczną liczbę 39007 tokenów. W przypadku eksperymentów bez zastosowania słownika fleksyjnego po tych operacjach pozostaje 131155 różnych tokenów.

3.2 Przykładowy problem

Aby przetestować działanie obu algorytmów przygotowane zostało przykładowe zapytanie do systemu wyszukiwania informacji dla zbioru notatek prasowych PAP. Problem polega na znalezieniu dokumentów podobnych do pojedynczej wybranej notatki na temat bliźniaczek syjamski. Została przygotowana modelowa odpowiedź systemu dla porównania z faktycznymi odpowiedziami.

Zapytanie:

> ***** #000424 *****
> W sobotę polskie bliźniaczki syjamskie Weronika i Wiktoria przylecą do Polski rejsem Newark-Kraków - poinformował PAP oddział LOT-u na nowojorskim lotnisku. Bliźniaczki syjamskie Weronika i Wiktora urodziły się 26 maja 1999 w szpitalu Akademii Medycznej w Lublinie. Od 16 sierpnia przebywają w Filadelfii. W listopadzie przeszły udaną operację rozdzielenia. Pierwszy termin wypisania dziewczynek wyznaczono na 11 lutego. Został jednak przesunięty o tydzień z uwagi na gorączkę Weroniki.

Najbardziej podobne dokumenty wybrane ręcznie:

***** #000516 *****
Wiktoria i Weronika, siostry syjamskie rozdzielone w listopadzie w Filadelfii, przyleciały z mamą w sobotę rano do Krakowa. Na lotnisku w Balicach żonę i córki przywitał ojciec dziewczynek, Edward Paleń. Byli też trzej bracia dziewczynek. "Dziewczynki przyjechały w dobrym stanie, są zdrowe- powiedziała na lotnisku w Balicach mama rozdzielonych bliźniaczek pani Krystyna Paleń. Łekarze amerykańscy twierdzili, że gdyby dziewczynki miały po powrocie do kraju trafić do polskiego szpitala, to oni by je przytrzymali dłużej u siebie. Wypisali dziewczynki w takim stanie, że mogą iść do domu- powiedziała dziennikarzom. Dodała, że Wiktoria i Weronika będą znajdowały się pod opieką lekarza pediatry w Stalowej Woli.

***** #009928 *****
W klasztorze Ojców Kapucynów w Stalowej Woli (Podkarpacie) ochrzczone zostały w niedzielę syjamskie bliźniaczki - Weronika i Wiktora Paleniówny, które urodziły się 26 maja ub. roku częściowo zrośnięte klatkami piersiowymi i brzuszkami. Pomoc w rozdzieleniu dzieci zaoferował Szpital Dziecięcy w Filadelfii w USA. Rozdzielenia dokonano 3 listopada, po kilkunastogodzinnej operacji, w której udział wzięło 45 lekarzy.

***** #011189 *****
Powoli stabilizuje się stan zdrowia 9-miesięcznego Kamila, jednego z rozdzielonych w Krakowie braci syjamskich. Drugi z braci - Patryk - nadal jest w ciężkim stanie - poinformował w poniedziałek PAP opiekujący się braćmi docent Adam Bysiek. Stan zdrowia Kamila docent Bysiek określił jako żokujący nadzieję". Obaj bracia nadal przebywają na oddziale intensywnej terapii Polsko-Amerykańskiego Instytutu Pediatrii Uniwersytetu Jagiellońskiego w Krakowie Prokocimiu. Bracia zostali rozdzieleni tydzień temu. O terminie operacji zadecydowało nagłe pogorszenie się stanu zdrowia jednego z nich. Początkowo operacja rozdzielenia była planowana na wrzesień. Na początku czerwca braciom wszczepiono ekspandery, mające za zadanie namnożyć tkankę skórną potrzebną po operacji. Operacja trwała 8 godzin, uczestniczyło w niej 14 lekarzy oraz zespół anestezjologów i pielęgniarek. Jej pierwszą część zajęło rozdzielenie braci, w drugiej lekarze zajęli się rekonstrukcją rozdzielonych narządów. Bracia byli zrośnięci powłokami

piersiowo-brzusznymi, mieli wspólną przeponę, worek osierdziowy i wątrobę. Była to siódma operacja rozdzielenia bliźniaków syjamskich dokonana w Instytucie w Prokocimiu.

***** #008779 *****
Bracia syjamscy Kamil i Patryk przeszli w poniedziałek pierwszy zabieg przygotowujący ich do operacji rozdzielenia, planowanej za trzy miesiące w Polsko-Amerykańskim Instytucie Pediatrii UJ w Krakowie-Prokocimiu.

***** #005662 *****
W Polsko-Amerykańskim Instytucie Pediatrii w Krakowie-Prokocimiu zmarły siostry syjamskie, które urodziły się przed tygodniem w Wejherowie.

***** #000469 *****
Bliźniaczki syjamskie z Poznania - Małgosia i Dorota - przeszły już pierwsze badania w Polsko-Amerykańskim Instytucie Pediatrii UJ w Krakowie-Prokocimiu.Ź przeprowadzonych badań wynika, że siostry mają wspólną watrobę i przeponę oraz prawdopodobnie wspólne drogi żółciowe i serce- powiedział PAP opiekujący się bliźniaczkami dr Adam Bysiek z Instytutu. Siostry urodziły się 11 lutego w poznańskiej klinice św. Rodziny. Dziewczynki zrośnięte są brzuszkami i klatkami piersiowymi.

***** #005855 *****
Liczba urodzeń bliźniąt syjamskich nie odbiega w Polsce od statystycznej normy - powiedział PAP prof. Jan Grochowski, dyrektor Polsko-Amerykańskiego Instytutu Pediatrii UJ, w którym przebywają dwie pary bliźniąt syjamskich.

***** #010677 *****
Lekarze z Polsko-Amerykańskiego Instytutu Pediatrii UJ w Krakowie Prokocimiu rozdzielili 9-miesięcznych braci syjamskich Kamila i Patryka - poinformował PAP doc. Adam Bysiek, szef zespołu opiekującego się bliźniętami.

***** #007320 *****
Prof. Louis Gerald Keith, który w ubiegłym roku przeprowadził operację rozdzielenia polskich bliźniaczek syjamskich Weroniki i Wiktorii, został odznaczony przez prezydenta Aleksandra Kwaśniewskiego Krzyżem Oficerskim Zasługi RP.

***** #007872 *****
Komitet etyki szpitala w Palermo na Sycylii wydał zgodę na

operację peruwiańskich bliźniaczek syjamskich, w wyniku której jedna z nich ma szansę na ocalenie kosztem życia drugiej - doniosła prasa włoska.

***** #012193 *****
Jeden z 9-miesięcznych braci syjamskich, rozdzielonych przez lekarzy w Krakowie pod koniec czerwca, zmarł w środę wieczorem z powodu niewydolności krążenia - poinformował PAP docent Adam Bysiek.

4 Architektura rozwiązania

W tym rozdziale zawarty został szczegółowy opis rozwiązań użytych w tej pracy. Pierwsza część omawia sposób przetwarzania danych wejściowych (zbioru dokumentów, patrz 3), druga natomiast omawia po krótce wykorzystane w badaniach biblioteki programistyczne.

4.1 Procedura przetwarzania

Poniżej znajduje się omówienie krok po kroku sposobu w jaki osiągnięto prezentowane wyniki. Omawiane są zarówno operacje wykonywane na danych, wybór konkretnych możliwości w takich kwestiach jak schemat wagowy (patrz 2.1.1), jak i sposób obliczania metryk jakości rozwiązania: współczynnika perplexity oraz współczynnika M opisującego jakość zwróconego rankingu dokumentów.

4.1.1 Sprowadzenie do form podstawowych

Metody macierzowe zachowują się znacznie lepiej dla mniejszych rozmiarów słownika, a co za tym idzie mniejszych rozmiarów wektorowej reprezentacji dokumentów. Poza oczywistym zyskiem polegającym na krótszym czasie przetwarzania pomniejszenie słownika w stosunku do liczby i długości dokumentów sprawia także, że metody te mogą łatwiej wykrywać, które wyrazy są powiązane — każdy wyraz występuje w większej liczbie kontekstów, umożliwiając zebrania na jego temat więcej informacji.

W tym wypadku zastosowano słownik fleksyjny (patrz 4.2.2) do sprowadzenia wyrazów występujących w tekście do form podstawowych — ta operacja nie tylko zmniejsza liczbę różnych tokenów, ale także umożliwia na dalszych etapach rozpoznanie faktu, że kilka różnych form jest w istocie tym samym wyrazem. Bez niej przykładowo wyrazy „bliźniak“ i „bliźniaka“ byłyby traktowane jako całkowicie różne i nie wnosiły by nic do oceny przez algorytm tekstów, w których występują.

W [14] sugerowano, że tego typu krok można pominąc posiadając odpowiednio duży korpus danych, jednak uwaga ta dotyczy języka angielskiego, w którym liczba form wyrazów jest stosunkowo mała. Polski jako język silnie fleksyjny wymagałby w tym wypadku zapewne zbioru danych o dużo wiąksych rozmiarach, którego przetwarzanie mogłoby być niepraktyczne.

4.1.2 Preprocessing

Aby zmniejszyć rozmiar przetwarzanych danych i poprawić ich uwarukowanie po sprowadzeniu wszystkich wyrazów danego dokumentu do

form podstawowych zastosowano dodatkowy preprocessing polegający na odrzuceniu wyrazów występujących w więcej niż 70% dokumentów i takich, które wystąpiły w co najwyżej jednym dokumencie. Te pierwsze nie niosą żadnej informacji o rodzaju tekstu, w którym występują ze względu na swoją pospolitość. Powiązania tych drugich nie mogłyby być wychwycone przez algorytmy redukcji wymiaru macierzy.

Zastosowano również kilka innych prostych operacji, jak na przykład zamienienie wszystkich liczb na token "NUMBER" a wyrażeń typ "2:3" na token "RATIO".

Przykładowo następujący dokument:

> #000064
> Drużyna Krzysztofa Oliwy - New Jersey Devils, mająca najlepszy bilans w całej NHL, odniosła kolejne zwycięstwo, pokonując w środę na własnym lodowisku New York Rangers 4:1.

po poddaniu preprocessingowi przyjmie postać:

> NUMBER
> drużyna Krzysztof oliwa - new jersey devils , mający najlepszy bilans cały NHL , odnieść kolejny zwycięstwo , pokonując środa własny lodowisko new York rangers RATIO

4.1.3 Dobór schematu wagowego

Bezpośrednie zastosowanie podejścia bag-of-words może dawać mylny obraz dokumentu — pewne wyrazy mogą pojawiać się bardzo często a mimo to nie dostarczać żadnej pożytecznej informacji o danym dokumencie ze względu na fakt występowania bardzo często w danym korpusie. Innym problemem może być przecenianie wielokrotnego występowania danego wyrazu — trzykrotne pojawienie się pewnego wyrazu raczej nie sugeruje trzykrotnie większego prawdopodobieństa, że jest on kluczowy dla tekstu. Aby rozwiązać pierwszy z tych problemów stosuje się normalizowanie wag wyrazów w danym dokumencie przez jakiś współczynnik charakteryzujący częstość występowania tego wyrazu w całym korpusie. Drugi z nich można rozwiązać przez zastosowanie funkcji typu logarytm czy pierwiastek. Dokładny dobór zastosowanych na tym etapie przekształceń nazywamy schematem wagowym.

W tej pracy zastosowano schemat wagowy nazywany "Log Entropy". Intuicyjnie polega on na wykorzystaniu ilości informacji w sensie Shannona niesionej przez dany wyraz jako czynnika normalizacyjnego (dla często spotykanych wyrazów będzie on niski) i zastosowaniu logarytmu jako funkcji wygładzającej.

Mając daną macierz w_{ij}, której j-ty wiersz odpowiada j-temu dokumentowi, a jego i-ta pozycja zawiera liczbę wystąpień i-tego wyrazu w tym dokumecie dokonujemy przeształcenia opisanego równaniami 9–11 uzyskując macierz a_{ij} zawierającą nowe wagi wyrazów.

$$p_{ij} = \frac{tf_{ij}}{gf_i} \tag{9}$$

$$g_i = 1 + \sum_{j=1}^{n} \frac{p_{ij} log(p_{ij})}{log(n)} \tag{10}$$

$$a_{ij} = g_i log(tf_{ij} + 1) \tag{11}$$

Drobiazgowe omówienie i porównanie różnych schematów wagowych znaleźć można w [8].

4.1.4 Przetwarzanie zapytania

Mając dany model macierzowy dla pewnego korpusu pojawia się standardowy problem wykonania zapytania do modelu i uzyskania z niego informacji. Dla pewnego dokumentu stanowiącego zapytanie d_q (może to być dokument, podobny do tych, które znajdują się w korpusie, albo po prostu zbitek wyrazów w rodzaju zapytań wprowadzanych zwykle w wyszukiwarce internetowej) chcielibyśmy uzyskać teksty z korpusu d_i posortowane według malejącej oceny podobieństwa tych tekstów do d_q. W tym celu obliczamy dla każdego d_i ranking r_i według wzorów 12-14.

$$\langle d_i, d_j \rangle = \sum_{k}^{n} d_{ik} d_{jk} \tag{12}$$

$$||d|| = \langle d, d \rangle \tag{13}$$

$$r_i = cos(d_i, d_j) = \langle \frac{d_i}{||d_i||}, \frac{d_j}{||d_j||} \rangle \tag{14}$$

Nietrudno zauważyć, że obliczone rankingi to po prostu odległość kosinusowa między wektorami reprezentującymi dokumenty. Warto zauważyć, że same rankingi r_i także mogą okazać się przydatne, gdyż zawierając informację o ocenie stopnia podobieństwa między dokumentami przez system. Można sobie wyobrazić sytuację, w której na ich podstawie obliczan i wyświetlane operatorowi będzie prawdopodobieństow, że dany dokument jest tym, czego szuka.

4.1.5 Obliczanie perplexity

Jeżeli p_i to prawdopodobieństwo przypisywane przez nasz model zdarzeniu polegającemu na wygenerowaniu i-tego wyrazu w danym dokumencie, znając temat lub tematy i ich wagi przyporządkowane danemu dokumentowi przez system, to znormalizowany współczynnik perplexity dla tego wyrazu opisany jest wzorem 15.

$$\frac{2^{-\sum_{i/1}^{n} p_i log_2(p_i)}}{n} \tag{15}$$

Aby obliczyć prawdopodobieństwo p_i obliczamy odległość kosinusową d_i między danym wyrazem, a dokumentem, zgodnie z 4.1.4. Następnie dokonujemy transformacji 16, gdzie d_j to odległość kosinusowa j-tego wyrazu rozponawanego przez system od danego dokumentu.

$$p_i = \frac{\pi - arccos(d_i)}{\sum_j^N \pi - arccos(d_j)} \tag{16}$$

Warto tutaj zaznaczyć, że w badaniach współczynnik perplexity obliczany był jedynie dla wyrazów znanych — takich, które wystąpiły w danych treningowych. Aby być całkowicie dokładnym należałoby zdarzeniu polegającemu na wystąpieniu wyrazu nieznanego przypisać pewne prawdopodobieństwo i odpowiednio obniżyć prawdopodobieństwo wystąpienia wszystkich innych wyrazów, jednak nie ma to znaczenia dla przeprowadzanego porównania, gdyż zbiór wyrazów znanych jest taki sam dla obu modeli, ze względu na fakt stosowania tego samego preprocessingu w obu przypadkach.

4.1.6 Ocena uszeregowania zwracanych dokumentów

W celu obiektywnego porównania odpowiedzi systemu na zapytanie, która ma postać listy dokumentów uporządkowanych według malejącego podobieństwa do tego zapytania wprowadzamy metrykę mającą opisywać jakość takiej odpowiedzi. Jako, że głównym celem jest ocena jak wysoko w wynikach wyszukiwania znalazły się dokumenty skojarzone możemy do takiej oceny zastosować sumę ich ranków lub — jak ostatecznie zrobiono — sumę kwadratów ranków, aby lepiej oddać sposób korzystania z tego typu systemów przez jego potencjalnych użykowników, któży zwykle zwracają znacznie mniejszą uwagę na dokumenty nie znajdujące się w ścisłej czołówce zwróconego rankingu.

W przedstawionych wynikach metryka została znormalizowana przez wynik dla idealnego uszeregowania dokumentów, to znaczy takiego, gdzie wszystkie n skojarzonych dokumentów znajduje się na pierwszych n pozycjach rankingu. Wzór 17 podsumowuje te obliczenia — r_i oznacza pozycję i-tego dokumentu w odpowiedzi systemu.

$$M = \frac{\sqrt{\sum_i^n r_i^2}}{\sqrt{\sum_i^n i^2}} \tag{17}$$

4.2 Wykorzystne rozwiązania/biblioteki

Poniżej zawarto opisy najważniejszych bibliotek programistycznych wykorzystywanych w przeprowadzonych badaniach. Należy zauważyć, że dla niektórych z nich istnieją alternatywy — przykładowo dostępnych jest wiele implementacji algorytmu LDA innych niż wykorzystana [1, 2, 3], jednak w większości różnią się one szczegółami implementacyjnymi, a ich porównanie wykracza poza zakres tej pracy.

4.2.1 Biblioteka gensim

Do badań wykorzystano pakiet gensim [4]. Jest to biblioteka napisana w języku Python tworzoan przez Radima Řehůřeka i stawiająca sobie za cel udostępnienie narzędzi do łatwego i wydajnego przetwarzania tesktów w językach naturalych. Zawiera ona szereg funkcjonalności użytecznych w zadaniach dotyczących przetwarzania języka naturalnego takich jak:

- Implementacje algorytmów LDA i LSI wraz z wersjami roproszonymi, pozwalające na uaktualnianie modeli w locie
- Implementacje kilku popularnych schematów wagowych (patrz 4.1.3)
- Zarządzanie korpusami tekstów i obsługa formatów stosowanych przez SVMlight [1], LDA-C [2] i GibbsLDA++ [3]
- Wydajne obliczanie podobieństwa między dokumentami w sensie danego modelu
- Wyszukiwanie kolokacji

Algorytm stosowany w pakiecie gensim do estymacji parametrów LDA opisany jest w pełni w [10].

4.2.2 Słownik fleksyjny CLP

Do sprowadzenia wyrazów występujących w tekście do form podstawowych zastosowany został słownik fleksyjny języka polskiego CLP [9] rozwijany w Zespole Przetwarzania Języka Naturalnego Instytutu Informatyki AGH. Jest to biblioteka napisana w języku C, udostępniająca interfejs pozwalający przeszukiwać zawarte w słowniku dane. Słownik obejmuje obecnie ponad 150 tysięcy wpisów co pokrywa niemal całkowicie zbiór

polskich wyrazów pospolitych. Zawiera on także najczęściej występujące nazwy własne i skróty.

CLP oferuje następujące funkcjonalności:

- Zwrócenie listy możliwych form podstawowych dla danego wyrazu
- Zwrócenie etykiety fleksyjnej, w której zawarty jest sposób odmiany danej formy podstawowej
- Znalezienie dla danego wyrazu wektora odmiany opisującego formę w jakiej występuje

Ponadto zastosowane zostało rozszerzenie słownika zaproponowane w [12], które umożliwia automatyczną ekstrakcję ogólnych reguł fleksyjnych z podstawowedo słownika. Pozwala to na sprowadzanie do form podstawowych również wyrazów, które nie zostały uwzględnione w słowniku.

Przykładowo, dla wyrazu "zamek" uzyskać można następujące informacje:

ID: 286975056
Forma podstawowa: zamek
Formy: zamek, zamka, zamkowi, zamkiem, zamku, zamki, zamków, zamkom, zamkami, zamkach
Etykieta: ACABBA
Opis etykiety: rzeczownik / męski nieżyw. / M.Lp.-0 / M.Lm.-i / D.Lp.-a / D.Lm.-ów
Wektor odmiany: [1, 4]

ID: 286975040
Forma podstawowa: zamek
Formy: zamek, zamku, zamkowi, zamkiem, zamki, zamków, zamkom, zamkami, zamkach
Etykieta: ACABA
Opis etykiety: rzeczownik / męski nieżyw. / M.Lp.-0 / M.Lm.-i / D.Lp.-u
Wektor odmiany: [1, 4]

5 Wyniki i analiza

Niniejszy rozdział zawiera porównanie algorytmów LDA i LSI pod kątem jakości otrzymywanych wyników. Zawarte w nim zostały zarówno porównanie ich przez pryzmat metryk z nadzorem (metryka M, dokładność, kompletność), które dobrze odpowiadają prawdziwym zastosowaniom tego typu rozwiązań jak i współczynnika perplexity, który może dawać sugestie na temat zdolności danej metody do uogólniania na nieznane dane (na przykład przy zastosowaniu do generowania automatycznych podsumowań) i odporności na przeuczenie.

Przedstawione wyniki uzyskane są każdorazowo dla różnych liczb tematów w celu przedstawienia w pełni zachowania omawianych metod w różnych warunkach wraz z wpływem tego czynnika na ich zachownie. Jako, że parametr ten można dowolnie regulować, może pozwalać on sterować jakością osiąganych wyników kosztem zwiększonego nakładu obliczeniowego. Rozdział zawiera także porównanie czasu działania obydwu metod. Na jego końcu znajdują się wnioski jakie można wyciągnąć z zebranych danych.

5.1 Tematy

Tabele 3 i 4 zawierają niektóre tematy wygenerowane przez algorytmy LSI i LDA skonfigurowane na 100 tematów (po dziesięć najbardziej znaczących słów w każdym temacie). Pojedynczy wiersz tabeli zawiera jeden temat - liczby przy tokenach oznaczają wagi poszczególnych słów w danym temacie.

Tematy uzyskane przy pomocy LDA wydają się bardziej odpowiadać postrzeganiu tekstu przez człowieka niż te wygenerowane przez LSI. Przykładowo temat numer 4 w tabeli 3 można interpretować jako „pogoda", jednak ma on znak ujemy — dokument, dla którego podobieństwo do tego tematu będzie duże z dużą pewnością nie traktuje o pogodzie. Temat numer 8 w tej samej tabeli prezentuje inne zachodzące zjawisko — możliwość połączenie dwóch konceptów jednak z przeciwnymi znakami w jeden temat, w tym wypadku „policja" ze znakiem dodatnim i „giełda" ze znakiem uejmnym. Podobieństwo do tego tematu może wskazywać, że danych tekst traktuje o pracy policji, lecz łatwo je przecenić — wiele tekstów, które zawierając co innego niż informacje giełdowe będzie wykazywało takie podobieństow.

Tematy wygenerowane przez LDA bywają złożeniami dwóch różnych konceptów, jednak zawsze mają ten sam znak, jak na przykład temat numer 4 w tabeli 4, który wydaje się łączyć koncepty „sport" i „giełda". Tego typu tematy powodują podobny problem jak przy LSI, ale jeżeli na przykład stosować te metody do klastowania tekstów, to zbiór zawierający przemieszane teksty sportowe i giełdowe wydaj się bardziej przydatny

niż taki, który zawiera różnorodne teksty, które akurat nie traktują o pogodzie.

Tablica 3: Tematy wyekstrahowane przez algorytm LSI

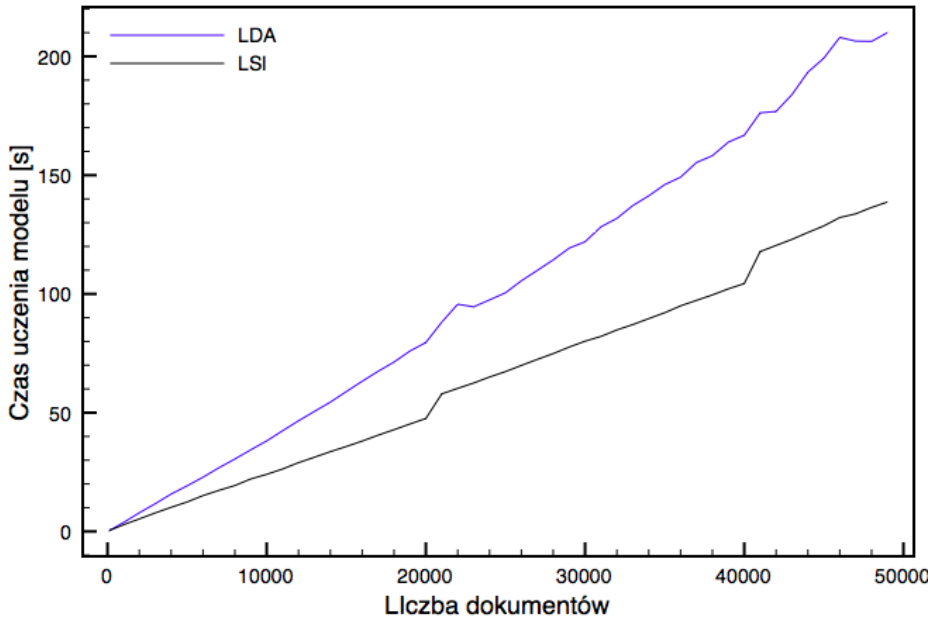
Lp.	Temat
1	0.208*procent + 0.160*rok + 0.153*złoty + 0.152*polski + 0.122*spółka + 0.121*milion + 0.111*wzróść + 0.110*punkt + 0.104*akcja + 0.099*powiedzieć
2	-0.275*procent + -0.257*wzróść + -0.252*punkt + -0.213*WIG + -0.189*spaść + -0.182*wynieść + -0.153*sesja + -0.147*akcja + -0.146*złoty + -0.144*spółka
3	0.477*RATIO + 0.264*mecz + 0.186*pokonać + 0.181*mistrzostwo + 0.146*turniej + 0.117*piłkarski + 0.114*wygrać + 0.111*przegrać + 0.107*świat + 0.107*reprezentacja
4	-0.281*stopień + -0.234*temperatura + -0.224*maksymalny + -0.213*wiatr + -0.208*umiarkowany + -0.202*deszcz + -0.198*słaby + -0.195*opad + 0.194*złoty + -0.170*południe
5	-0.380*złoty + -0.307*grosz + -0.259*dolar + -0.248*euro + 0.200*punkt + -0.197*osiągać + -0.158*milion + 0.148*WIG + -0.148*umocnić + 0.139*procent
6	-0.376*spółka + -0.315*Akcyjna + 0.241*grosz + -0.226*milion + 0.204*zamknięcie + 0.176*euro + 0.168*osiągać + 0.168*punkt + -0.152*bank + 0.143*dolar
7	-0.444*procent + 0.271*spółka + -0.244*rok + 0.205*akcja + -0.186*proca + 0.165*Akcyjna + -0.156*milion + 0.152*giełda + 0.131*zmienić + -0.129*finanse
8	0.237*sąd + -0.169*RATIO + -0.163*spółka + 0.146*policja + -0.144*Akcyjna + -0.138*unia + -0.137*AWS + 0.135*lato + 0.129*więzienie + 0.126*tysiąc
9	0.222*europejski + -0.206*AWS + -0.195*sąd + -0.153*procent + 0.151*unia + 0.144*UE + -0.130*wyborczy + -0.120*wybór + 0.119*milion + -0.117*SLD
10	-0.321*RATIO + 0.227*świat + 0.223*mistrzostwo + -0.160*sąd + -0.155*mecz + 0.149*wyścig + 0.131*medal + -0.127*milion + 0.120*miejsce + 0.119*procent

Tablica 4: Tematy wyekstrahowane przez algorytm LDA

Lp.	Temat
1	0.040*PKB + 0.038*pieniężny + 0.025*polityka + 0.022*deficyt + 0.021*rada + 0.020*procent + 0.019*RPP + 0.018*analityk + 0.014*obrót + 0.014*sport
2	0.026*prokuratura + 0.019*sąd + 0.018*letni + 0.016*gang + 0.016*oskarżona + 0.016*okręgowy + 0.013*akt + 0.013*oskarżenie + 0.013*efekt + 0.013*przestępstwo
3	0.025*Bush + 0.020*ii + 0.019*papież + 0.015*Paweł + 0.014*Jan + 0.013*kościół + 0.012*wiek + 0.011*George + 0.011*święty + 0.009*kardynał
4	0.034*TechWI + 0.027*tour + 0.024*France + 0.022*de + 0.018*sportowy + 0.014*bankowy + 0.014*oszczędność + 0.013*Vivendi + 0.012*bank + 0.010*instancja
5	0.012*choroba + 0.009*of + 0.008*najnowszy + 0.008*zdrowie + 0.007*lek + 0.007*informować + 0.006*Ameryka + 0.006*naukowy + 0.006*obywatelski + 0.006*numer
6	0.048*kwartał + 0.027*Laden + 0.026*bin + 0.020*Osama + 0.017*zwierzę + 0.014*zadłużenie + 0.013*transakcja + 0.013*IV + 0.012*tom + 0.012*pies
7	0.052*spółka + 0.044*skarb + 0.044*Akcyjna + 0.030*akcja + 0.028*Spółka Akcyjna + 0.023*prywatyzacja + 0.023*oferta + 0.020*sprzedaż + 0.020*TP + 0.019*procent
8	0.032*NBP + 0.026*wczorajszy + 0.020*milion + 0.019*otwierać + 0.016*dolar + 0.016*dekada + 0.015*off + 0.015*podaż + 0.014*play + 0.014*uzgodnić
9	0.027*huta + 0.020*przejęcie + 0.016*energia + 0.016*belgijski + 0.013*Toruń + 0.012*domniemany + 0.012*gazociąg + 0.012*białostocki + 0.011*rolnik + 0.010*D
10	0.015*koncert + 0.015*muzyka + 0.010*www + 0.010*podlaski + 0.007*twórca + 0.007*zespół + 0.007*muzyczny + 0.007*festyn + 0.007*proponowany + 0.006*Stefan

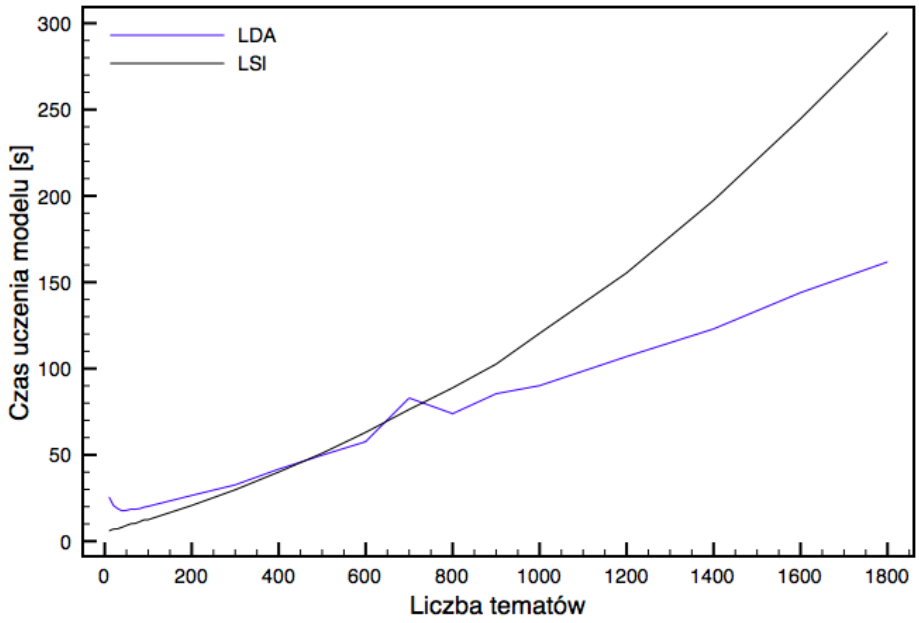
5.2 Czas działania

Poniższe wykresu (1, 2) przedstawiając zależność czasu uczenia modelu dla metod LSI i LDA od odpowiednio wielkości korpusu dla stałej liczby tematów i liczby tematów dla stałej wielkości korpusu.



Rysunek 1: Czas uczenia modelu dla 100 tematów

Complexity discussion Conclusions



Rysunek 2: Czas uczenia modelu dla 5000 dokumentów

5.3 Metryki z nadzorem

W tym rozdziale omówiono wyniki otrzymane za pomocą algorytmów LDA i LSI dla przykładowego problemu opisanego w 3.2. Należy zauważyć, że tego rodzaju ewaluacja wymaga ręcznego przygotowania danych testowych przez człowieka, co może być nieprakyczne dla dużych zbiorów danych. Jej zaletą jest fakt, że mierzy ona faktyczne osiągi danego rozwiązania w rzeczywistych problemach.

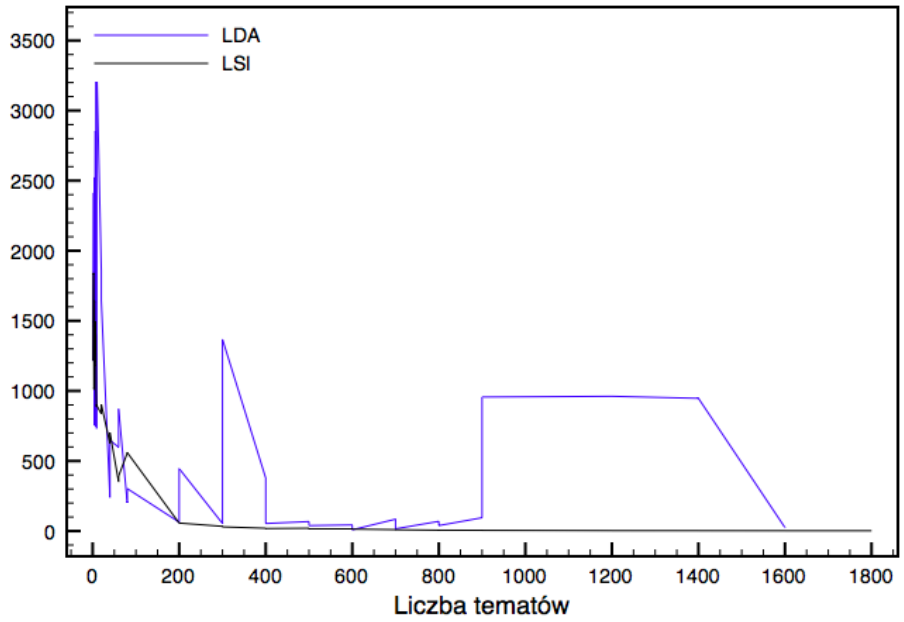
5.3.1 Ranking dokumentów

Wykresy 3 i 4 przedstawiają sumę kwadratów ranków dokumentów (patrz 4.1.6) z wzorca przygotowanego ręcznie dla danego zapytania w wynikach działania odpowiednio algorytmów LDA i LSI dla różnej liczby tematów.

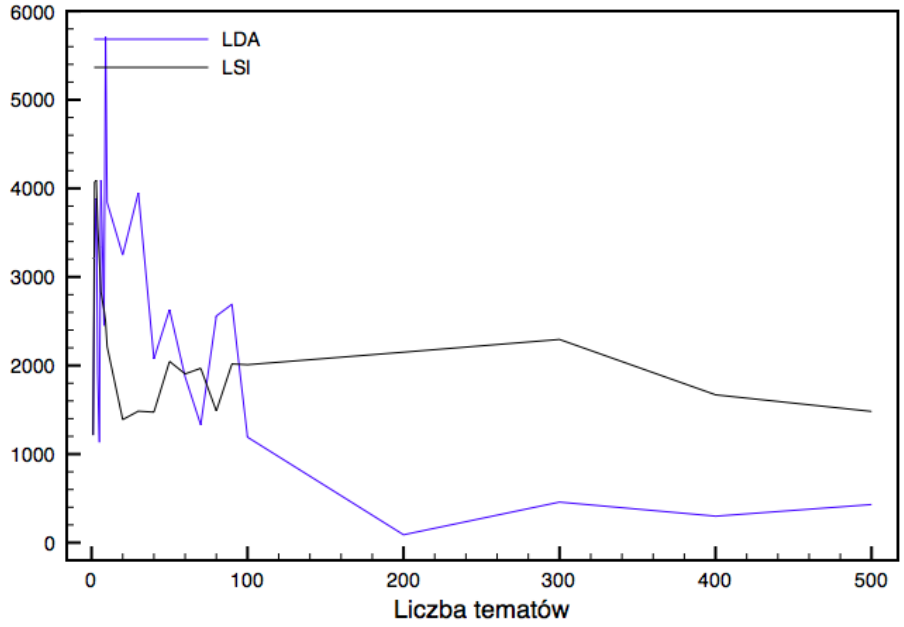
Algorytm LDA osiąga ogólnie gorsze wyniki niż LSI - poza przedziałem 50 – 100 tematów. Gorszy jest też (aczkolwiek niewiele) najlepszy wynik jaki udałoby się osiągnąć odpowiednio dobierając liczbę tematów. Na wykresie daje się także zauważyć stochastyczna natura LDA - podczas gdy dla LSI wyniki niemal monotonicznie poprawiają się wraz ze wzrostem liczby tematów dla LDA zdarza się znaczne pogorszenie wyników przy zwiększeniu tej liczby.

Polepszenie wyników dzięki zastosowaniu stemmingu jest widoczne na pierwszy rzut oka — polski jako język silnie fleksyjny jest znakomitym kandydatem do zastosowania tego typu techniki. W [14] zasugerowano, że ze stemmingu można zrezygnować dyponując odpowiednio dużym zbiorem danych jednak wyniki te uzyskano dla języka angielskiego, którego fleksja jest znacznie mniej rozbudowana. W tym wypadku zebranie tak dużej ilości danych może być mniej praktyczne niż skonstruowanie słownika fleksyjnego takiego jak na przykład ten opisany w [13].

Co ciekawe algorytm LDA radzi sobie znacznie lepiej od LSI bez wykorzystania stemmingu. Może to być spowodowane trudnością w przypadku LSI połączenia ze sobą słów, które różnią się formą fleksyjną i są w tym wypadku traktownae całkowicie osobno.



Rysunek 3: Suma kwadratów ranków dokumentów ze wzorca dla testowego zapytania (z wykorzystaniem stemmingu)



Rysunek 4: Suma kwadratów ranków dokumentów ze wzorca dla testowego zapytania (bez wykorzystania stemmingu)

5.3.2 Krzywe ROC

Krzywa ROC [15] (Receiver Operation Characteristic) to wykres przedstawiający dla danego klasyfikatora zależność między stosunkiem liczby znalezionych dokumentów skojarzonych do liczby wszystkich zwróconych dokumentów (TPR — True Positive Rate), a stosunkiem liczby odrzuconych dokumentów skojarzonych do liczby wszystkich odrzuconych dokumentów (FPR - False Positive Rate) w miarę zmiany progu detekcji. Obliczanie tych wartości podsumowując wzory 18 i 19.

W tym wypadku ten zmienny próg to po prostu liczba n - pierwszych n dokumentów jest traktowane jako odnalezione, a pozostałe jako odrzucone.

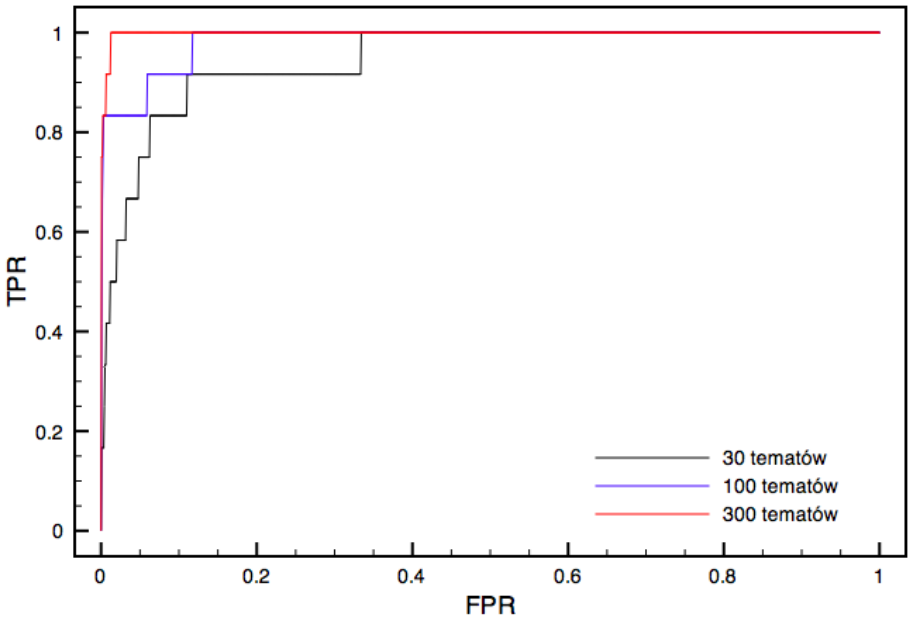
$$TPR = \frac{TP}{TP + FN} \tag{18}$$

$$FPR = \frac{FP}{FP + TN} \tag{19}$$

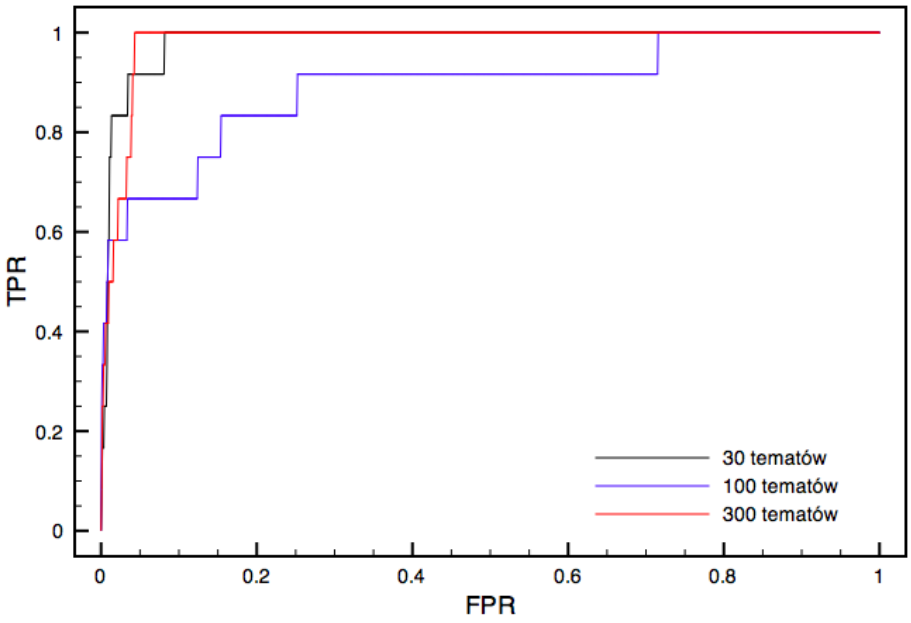
Lepsze klasyfikatory charakteryzują się krzywymi ROC położonymi dalej od linii $x = y$. Klasyfikatory blisko, lub na tej linii nie wykonują żadnej użytecznej pracy. Analiza odległości krzywej ROC od linii $x = y$ w różnych miejscach wykresu może dać wskazówkę co do najlepszego dobrania progu detekcji dla danego problemu.

Wykresy 5 i 6 przedstawiają krzywe ROC dla algorytmów LDA i LSI dla różnych liczb tematów. Dla dużych liczb tematów algorytm LDA spisuje się gorzej, jednak można zauważyć, że klasyfikator uzyskany dla 30 tematów jest podobnej jakości lub lepszy jak ten uzyskany przy użyciu LSI dla 100 tematów.

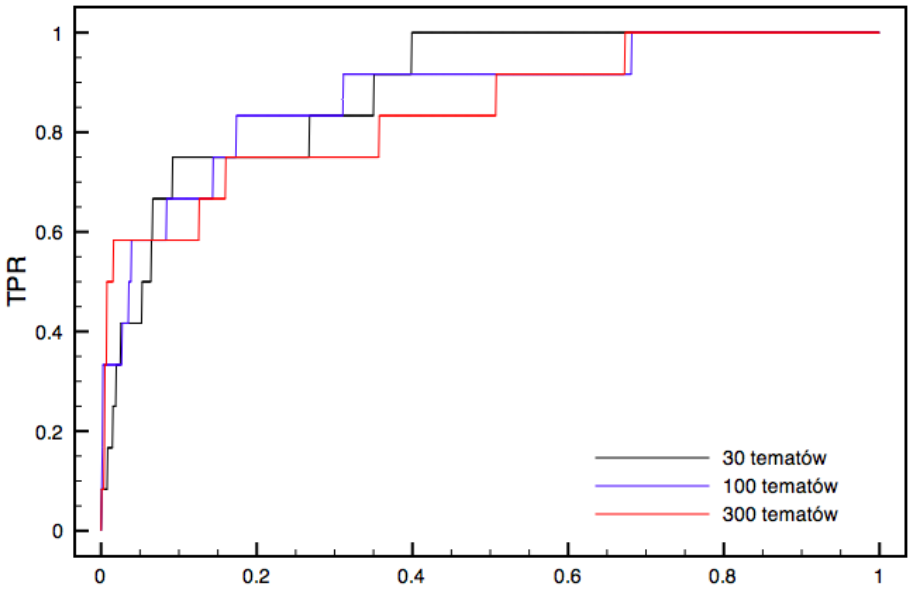
Na wykresach 7 i 8 przedstawione zostały krzywe ROC dla algorytmów LSI i LDA bez wykorzystania stemmingu. Ponownie daje się zauważyć lepsze działanie algorytmu LDA w tym wypadku - klasyfikator uzyskany dla 300 tematów jest znacznie lepszy od tego uzyskanego przy pomocy LSI.



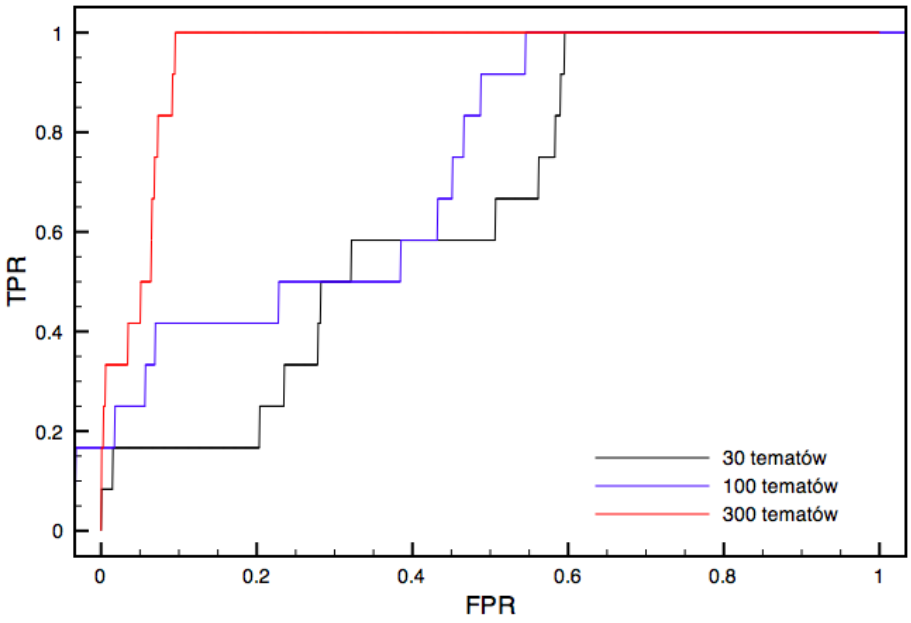
Rysunek 5: Krzywe ROC dla algorytmu LSI dla wybranych liczb tematów



Rysunek 6: Krzywe ROC dla algorytmu LDA dla wybranych liczb tematów



Rysunek 7: Krzywe ROC dla algorytmu LSI dla wybranych liczb tematów bez wykorzystania stemmingu



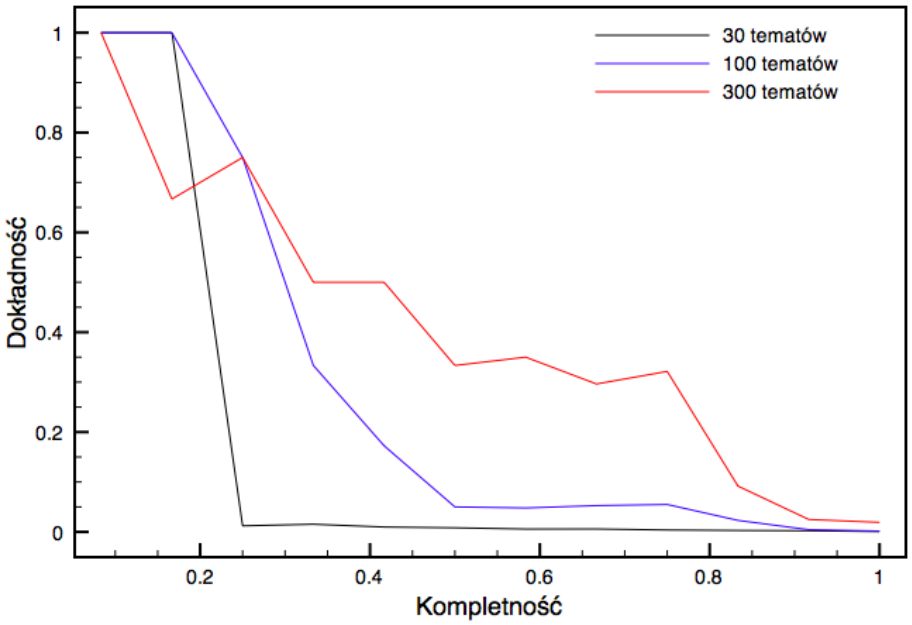
Rysunek 8: Krzywe ROC dla algorytmu LDA dla wybranych liczb tematów bez wykorzystania stemmingu

5.3.3 Dokładność i kompletność

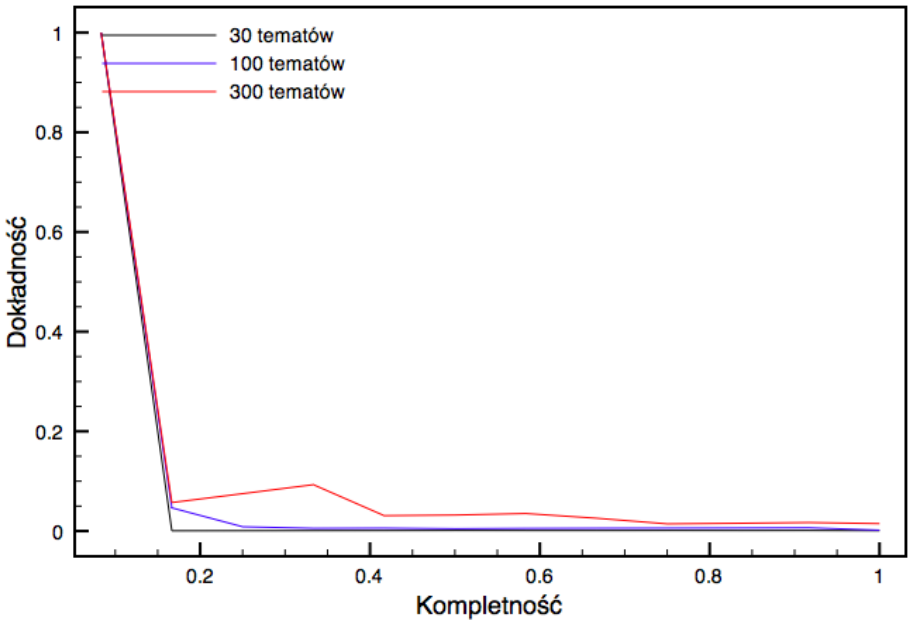
Kompletność (stosunek liczby zwróconych skojarzonych dokumentów do liczby wszystkich skojarzonych dokumentów) i dokładność (stosunek liczby zwróconych skojarzonych dokumentów do liczby wszystkich zwróconych dokumentów) to częste metryki w zadaniach typu information retrieval. Wybranie jakiegoś poziomu kompletności reprezentuje pewien kompromis między prawdopodobieństwem zawarcia wszystkich interesujących wyników w zwróconych danych, a częstością występowania w nich danych skojarzonych, a więc ilością czasu, który musi poświęcić operator systemu na ich dalsze przetworzenie.

Wykresy 9 i 10 prezentują dkoładność osiąganą przez algorytmy LDA i LSI na różnych poziomach kompletności dla przykładowego problemu.

Można zauważyć, że LDA daje znacznie gorszą dokładność niż LSI. Nawet najlepiej dobrana liczba tematów (w tym wypadku 300) pozwala osiągać dokładność porównywalną jedynie z LSI dla 30 tematów.



Rysunek 9: Dokładność na różnych poziomach kompletności dla algorytmu LSI



Rysunek 10: Dokładność na różnych poziomach kompletności dla algorytmu LDA

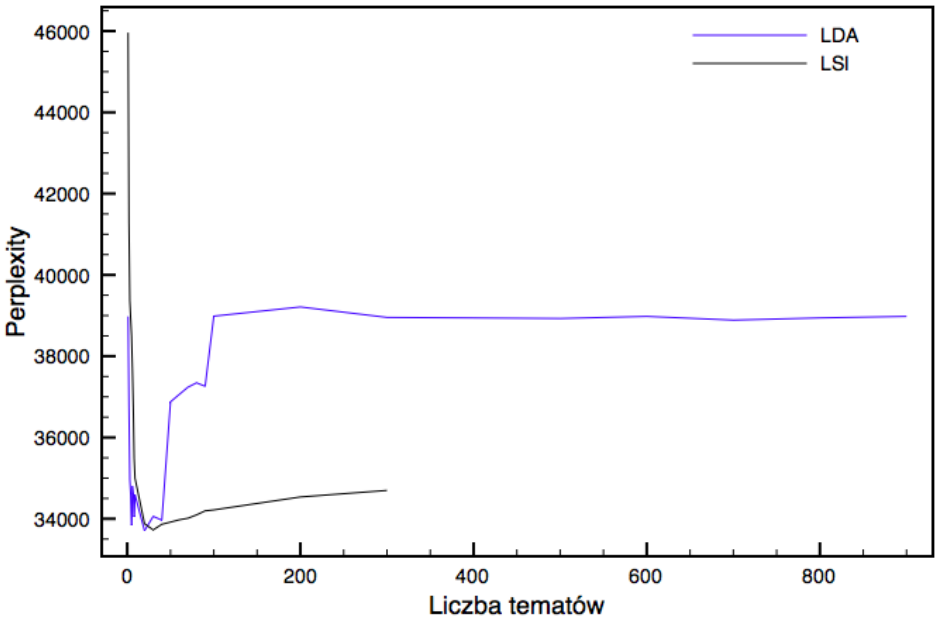
5.4 Metryki bez nadzoru (perplexity)

Współczynnik perplexity, który dla pewnych prawdopodobieństw p_i przypisywanych przez model zdarzeniom obliczyć można zgodnie ze wzorem 20, daje pewne pojęcie o tym jak dobrze model jest w stanie przewidzieć nowe dane. Wysokie wartości współczynnika mogą wskazywać, że model jest przeuczony i będze słabo uogólniał swoje działania na nieznane dane. Jest on dobrym wskaźnikiem jak dobrze model będzie sobie radził z klastrowaniem tego typu danych.

$$\frac{2^{-\sum_{i/1}^{n} p_i log_2(p_i)}}{n} \tag{20}$$

Wartości współczynnika perplexity dla omawianych metod w zależności od liczby tematów są przedstawione na wykresie 11, Wykres demonstruje, że optymalne wartości współczynnika perplexity zostają osiągnięte w okolicach 50 tematów. W tym wypadku może to sugerować, że mniej więcej na tyle właśnie grup tematycznych należałoby podzielić ten zbiór danych.

Algorytm LDA zachowuje niski współczynnik perplexity tylko stosunkowo blisko optymalnej liczby tematów. Takie zachowanie może wymagać dokładnego strojenia algorytmu do każdego zastosowania, co bywa uciążliwe i czasochłonne.



Rysunek 11: Współczynnik perplexity dla LDA i LSI w zależności od liczby tematów

5.5 Wnioski

Algorytm LDA daje gorszej jakości (a przynajmniej mniej stabilne) wyniki dla typowych problemów klasyfikacji i wyszukiwania informacji spotykanych w codziennej praktyce co nie jest zaskakujące, zważywszy na fakt, że LDA można traktować jako metodą analogiczną do LSI z nałożonym dodatkowym ograniczeniem o dodatniości wag. Wydaje się za to być w stanie działać w sytuacji, gdy wiele różnych tokenów oznacza to samo (przypadek bez wykorzystania stemmingu), w odróżnieniu od LSI, którego wyniki są wtedy całkowicie nieprzydatne. To bardzo porządana cecha w przypadku braku odpowiedniego słownika fleksyjnego dla danego języka.

Metryka perplexity zdaje się sugerować, że LDA jest w stanie uogólniać się na nieznane dane podobnie dobrze jak LSI, jednak trudniej dobrać parametry, dla których tak się stanie.

Zarówno LSI jak i LDA skalują się liniowo z liczbą analizowanych dokumentów, jednak LSI potrzebuje ponad-liniowego czasu w zależności od liczby tematów. Bardzo duże liczby tematów będą praktyczne dopiero dla wielkich korpusów, jednak jest to kolejne potencjalne zastosowanie dla LDA.

Przewagą LDA wydaje się być jakość generowanych tematów — przez wymuszenie dodatnich wag otrzymujemy na najbardziej znacz-

cych pozycjach (z najwyższymi wagami) słowa opisujące dany dokument/temat, podczas gdy w przypadku LSI mogą to być słowa najodleglejsze. Takie zachowanie może okazać się korzystne w zastosowaniach typu tagowanie dokumentów lub automatyczne generowanie podsumowań czy słów kluczowych, odpowiada ono lepiej postrzeganiu tekstu przez człowiek, a przecież to on będzie odbiorcą takich podsumowań.

6 Podsumowanie

W tej pracy opisano i poddano porównaniu metody Latent Semantic Indexing i Latent Dirichlet Analysis pod kątem ich przydatności do automatycznego porównywania i ekstrahowania tematu dokumentów w języku polskim. W typowych zadaniach metoda LSI daje lepsze wyniki i łatwiej jest dobrać liczbę tematów dla danego odpowiednią dla danego korpusu. Metoda LDA pozwala ekstrahować tematy lepiej odpowiadające ludzkiemu postrzeganiu tekstu przez co lepiej nadaje się do tagowania czy generowania skrótów dokumentów.

Z braku czasu nie rozważono pewncyh aspektów tego zagadnienia, które mogą być tematem dalszych badań, w tym:

- Wpływ parametrów treningu, przykładowo kryterium stopu, na jakość modelu LDA
- Wpływ parametrów korpusu, głównie rozmiaru słownika, ale także udziału słów zapożyczonych czy terminów technicznych na działanie obu metod
- Porównanie działania modeli dla różnych języków z uwzględnieniem złożoności ich fleksji

A Terminy i oznaczenia

- df_i (Document Frequency) — liczba dokumentów zawierających i-ty token
- gf_i (Global Frequency) — stosunek liczby wystąpień i-tego tokenu do liczby wszystkich wyrazów w korpusie
- tf_i^j (Term Frequency) — stosunek liczby wystąpień i-tego tokenu do liczby wszystkich wyrazów w j-tym dokumencie
- Bag-of-words (worek ze słowami) — założenie o możliwości pominięcia kolejności słów w dokumencie z małą utratą znaczenia
- Dokument — tekst lub fragment tekstu
- FN (False Negatives) — liczba odrzuconych dokumentów skojarzonych
- FP (False Positives) — liczba zwróconych dokumentów skojarzonych
- Forma podstawowa — wyróżniona, najprostsza forma wyrazu, przykładowo mianownik rzeczownika
- Korpus — zbiór dokumentów
- LDA — Latent Dirichlet Allocation
- LSI — Latent Semantic Indexing
- Leksykon — zbiór wszystkich tokenów występujących w korpusie
- Polisemia — możliwość wyrażania różnych koncepcji przez ten sam wyraz, przykładowo zamek
- Przeuczenie — zjawisko polegające na zbyt dokładnym odwzorowaniu danych treningowych przez model i utracie zdolności uogólniania
- SVD — Singular Value Decomposition
- Schemat Wagowy — sposób w jaki dokument jest transformowany do reprezentacji wektorowej
- Skojarzony — (o dokumencie) zgodny z kryteriami wyszukiwania, uznany za podobny do dokumentu wzorcowego
- Słownik — zbiór wyrazów występujących w korpusie

- TN (True Negatives) — liczba odrzuconych dokumentów nieskojarzonych
- TP (True Positives) — liczba zwróconych dokumentów skojarzonych
- Token — wyraz, znak interpunkcyjny lub grupa znaków traktowana jako całość (na przykład liczba)
- VSM (Vector Space Model) — modelowanie języka przez przedstawianie dokumetów jako wektorów

B Sposób użycia kodu

Kod wykorzystany do przeprowadzenia badań w tej pracy znaleźć można pod [5]. Aby z niego skorzystać konieczne jest zainstalowanie pakietu gensim. Katalog *bin* zawiera skrypty uruchomieniowe dla przeprowadzonych eksperymentów. W poniżych poleceniach *model* oznaczna ciąg *LsiModel* lub ciąg *LdaModel* w zależności od rodzaju modelu, który ma być testowany. Plik z danymi powinien zawierać dokumenty, każdy w jednej linii, wyrazy powinny być sprowadzone do form podstawowych, jeżeli takie rozwiązanie ma być zastosowane.

- *bin/test_lsi [plik z korpusem] [model]* — wypisuje na ekran listę wygenerowanych tematów
- *bin/map_topics [plik z korpusem] [model]* — oblicza i wypisuje metrykę M dla różnych liczb tematów
- *bin/perplexity [plik z korpusem] [model]* — oblicza i wypisuje współczynnik perplexity dla różnych liczb tematów
- *bin/precision [plik z korpusem] [model] [liczba tematów]* — oblicza i wypisuje zależność między dokładnością a kompletnością w formie par [dokładność kompletność] dla zadanej liczby tematów
- *bin/roc [plik z korpusem] [model] [liczba tematów]* — oblicza i wypisuje zależność między TPR a FPR w formie par [TPR FPR] dla zadanej liczby tematów

C Tabele tematów

Spis tablic

Spis rysunków

Literatura

[1] http://svmlight.joachims.org/.

[2] http://www.cs.princeton.edu/~blei/lda-c/.

[3] http://gibbslda.sourceforge.net/.

[4] http://radimrehurek.com/gensim/.

[5] https://github.com/obrok/thesis.

[6] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[7] P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, J. C. Lai, and R. L. Mercer. An estimate of an upper bound for the entropy of English. *Computational Linguistics*, 18(1):31–40, 1992.

[8] A. Figiel. Tekst jak wzorzec informacyjny — automatyczna ocena podobieństwa tematycznego tekstów za pomocą Latent Semantic Analysis. *Słowniki Komputerowe i Automatyczna Ekstrakcja Informacji z Tekstu*, 2009.

[9] M. Gajęcki. Słownik fleksyjny jako biblioteka języka C. *Słowniki Komputerowe i Automatyczna Ekstrakcja Informacji z Tekstu*, 2009.

[10] M. D. Hoffman, D. M. Blei, and F. R. Bach. Online learning for Latent Dirichlet Allocation. In J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, editors, *NIPS*, pages 856–864. Curran Associates, Inc., 2010.

[11] F. Jelinek, R. L. Mercer, L. R. Bahl, and J. K. Baker. Perplexity – a measure of the difficulty of speech recognition tasks. *Journal of the Acoustical Society of America*, 62:S63, November 1977. Supplement 1.

[12] M. Korzycki. A dictionary based stemming mechanism for polish. In *9th International Workshop on Natural Language Processing and Cognitive Science*, 2012.

[13] W. Lubaszewski, H. Wróbel, M. Gajecki, B. Moskal, A. Orzechowska, P. Pietras, P. Pisarek, and T. Rokicka. *Słownik Fleksyjny języka polskiego*. Lexis Nexis, Kraków, 2001.

[14] C. D. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA, 1999.

[15] D. K. McClish. Analyzing a portion of the ROC curve.