Automatyczna klasyfikacja i ekstrakcja tematu krótkich notatek w języku polskim

Paweł Obrok
pod kierunkiem dr. Michała Korzyckiego

4 września 2012

Spis treści

1 Wstęp

2 Terminy i oznaczenia

- Dokument — tekst lub fragment tekstu
- Korpus — zbiór dokumentów
- Skojarzony — (o dokumencie) zgodny z kryteriami wyszukiwania

3 Podstawy teoretyczne

3.1 Vector Space Model

??

3.1.1 Log-Entropy

3.2 Latent Semantic Indexing

3.3 Latent Dirichlet Allocation

3.4 Perplexity

3.5 Dokładność i kompletność

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 **??**), druga natomiast omawia po krótce wykorzystane w badaniach biblioteki programistyczne.

4.1 Procedura przetwarzania

This is most likely not complete

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 **??**), jak i sposób obliczania metryk jakości rozwiązania takich jak współczynnik perplexity.

4.1.1 Sprowadzenie do form podstawowych

Metody macierzowe zachowują się znacznie lepiej dla mniejszych rozmiarów leksykonu, a co za tym idzie mniejszych rozmiarów wektorowej reprezentacji dokumentów. Poza oczywistym zyskiem polegającym na krótszym czasie przetwarzania pomniejszenie leksykonu 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 [**?**] 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 większych 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 przeszłałcenia opisanego równaniami 1–3 uzyskując macierz a_{ij} zawierającą nowe wagi wyrazów.

$$p_{ij} = \frac{tf_{ij}}{gf_i} \tag{1}$$

$$g_i = 1 + \sum_{j=1}^{n} \frac{p_{ij} log(p_{ij})}{log(n)} \quad log(p_{ij}) = -\infty? \tag{2}$$

$$a_{ij} = g_i log(tf_{ij} + 1) \tag{3}$$

Drobiazgowe omówienie i porównanie różnych schematów wagowych znaleźć można w [5].

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 4-6.

$$\langle d_i, d_j \rangle = \sum_k^n d_{ik} d_{jk} \tag{4}$$

$$||d|| = \langle d, d \rangle \tag{5}$$

$$r_i = cos(d_i, d_j) = \langle \frac{d_i}{||d_i||}, \frac{d_j}{||d_j||} \rangle \tag{6}$$

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

Współczynnik perplexity który zaproponowany był do oceny modeli języka [8] ma oddawać niejako poziom "zaskoczenia" modelu niewidzianymi dotąd danymi. Jest on tak dobrany, że rzut symetryczną, k-ścienną kością będzie miał perplexity dokładnie k - możemy co najwyżej przewidywać, że nastąpi jeden z k rezultatów. Można też zdefiniować go jako eksponentę z entropii. Oczywiście jeżeli współczynnik ten się zmniejsza, to można powiedzieć, że model lepiej opisuje dane zjawisko.

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 7.

$$\frac{2^{-\sum_{i/1}^n p_i log_2(p_i)}}{n} \tag{7}$$

Aby obliczyć prawdopodobieństwo p_i obliczamy odległość kosinusową d_i między danym wyrazem, a dokumentem, zgodnie z 4.1.4. Następnie dokonujemy transformacji 8, gdzie d_j to odległość kosinusowa j-tego wyrazu rozponawanego przez system od danego dokumentu.

$$p_i = \frac{\pi - arccos(d_i)}{\sum_j^N \pi - arccos(d_j)} \tag{8}$$

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 relewantne 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 relewantnych dokumentów znajduje się na pierwszych n pozycjach rankingu. Wzór 9 podsumowuje te obliczenia — r_i oznacza pozycję i-tego dokumentu w odpowiedzi systemu.

$$\mathrm{M} = \frac{\sqrt{\sum_i^n r_i^2}}{\sqrt{\sum_i^n i^2}} \tag{9}$$

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 [7].

4.2.2 Słownik fleksyjny CLP

Do sprowadzenia wyrazów występujących w tekście do form podstawowych zastosowany został słownik fleksyjny języka polskiego CLP [6] 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 [9], 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 Opis danych

Działanie algorytmów LDA i LSI analizowane było na zbiorze około 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.

5.1 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 - poinformo-

wał 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.

6 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.

6.1 Tematy

Tabele 1 i 2 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 1 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 2, 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 1: 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 2: 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

6.2 Czas działania

Czy to ma sens?

6.3 Metryki z nadzorem

W tym rozdziale omówiono wyniki otrzymane za pomocą algorytmów LDA i LSI dla przykładowego problemu opisanego w 5.1. 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.

6.3.1 Ranking dokumentów

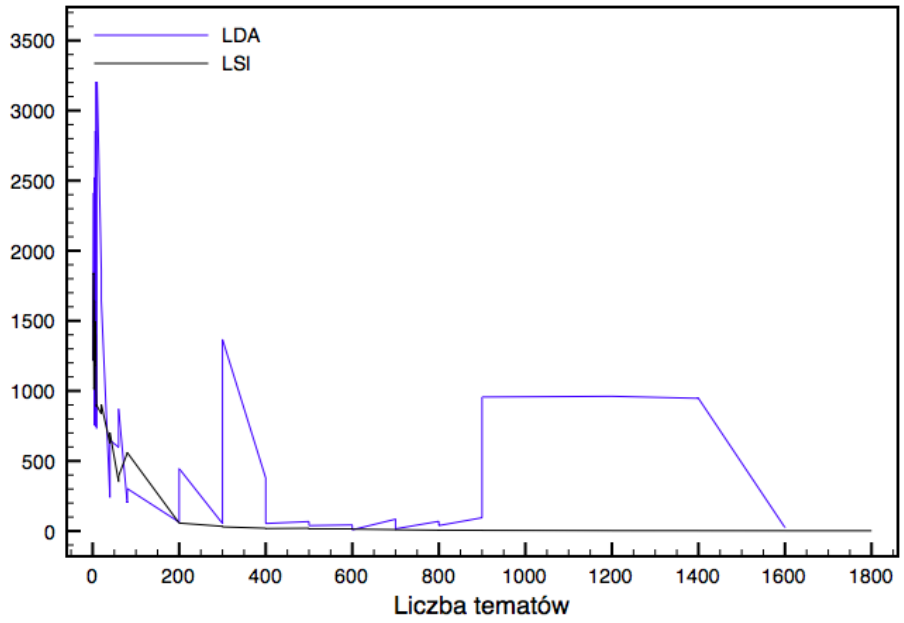
Wykresy 1 i 2 przedstawiają sumę kwadratów ranków dokumentów (patrz 4.1.6) z wzorca przygotowanego ręcznie dla danego zapytania w wynikach działania odpowiednio algorytmów LDA i LSI dla różnej liczby tematów.

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.

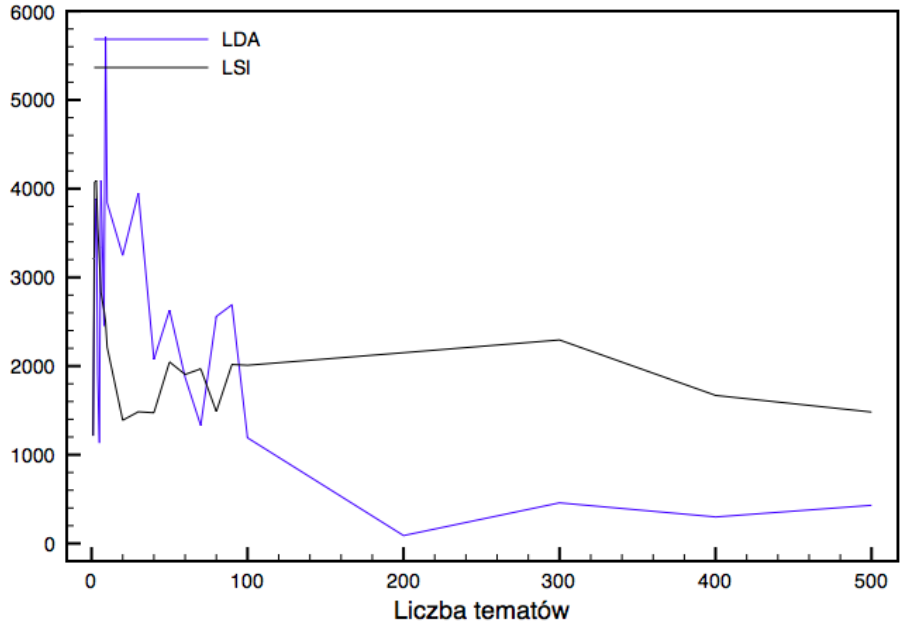
Polepszenie wyników dzięki zastosowaniu stemmingu jest widoczne na pierwszy rzut oka — polski jako język silnie fleksyjny jest znakomitym kandydatem do zastosowania tego typu techniki. W [11] zasugerowano, że ze stemmingu można zrezygnować dyponując odpowiednio dużym zbiorem danych jednak wyniki te uzyskano 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 [10].

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



Rysunek 2: Suma kwadratów ranków dokumentów ze wzorca dla testowego zapytania (bez wykorzystania stemmingu)



6.3.2 Krzywe ROC

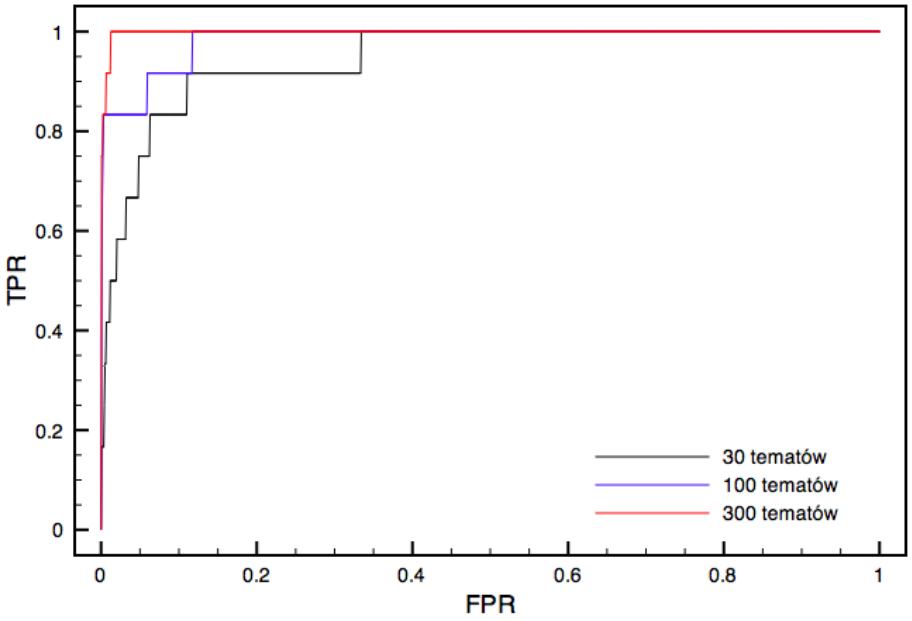
Krzywa ROC [12] (Receiver Operation Characteristic) to wykres przedstawiający dla danego klasyfikatora zależność między stosunkiem liczby znalezionych dokumentów relewantnych do liczby wszystkich zwróconych dokumentów (TPR — True Positive Rate), a stosunkiem liczby odrzuconych dokumentów relewantnych do liczby wszystkich odrzuconych dokumentów (FPR - False Positive Rate) w miarę zmiany progu detekcji. W tym wypadku ten zmienny próg to po prostu liczba n - pierwszych n dokumentów jest traktowane jako odnalezione, a pozostałe jako odrzucone.

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.

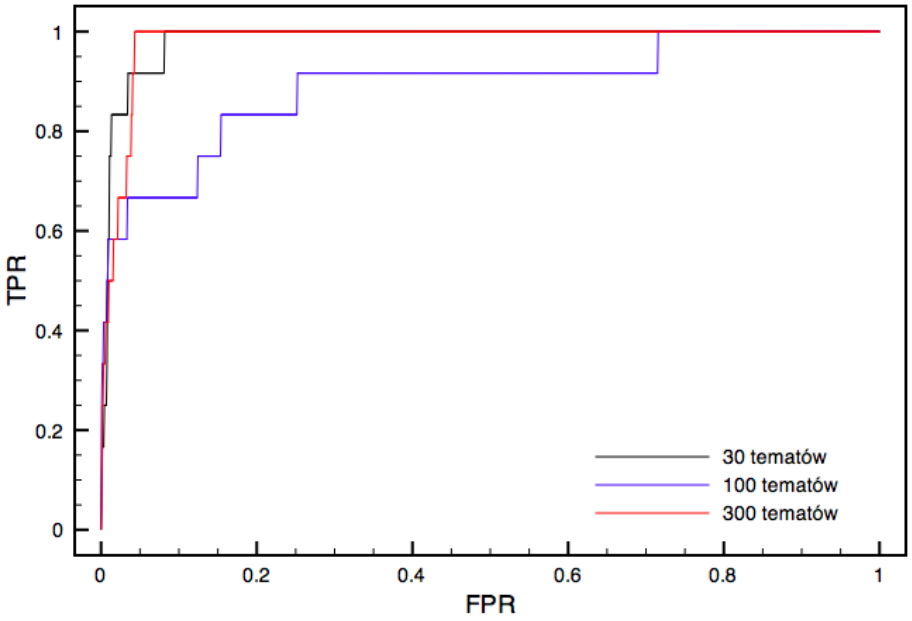
Wykresy 3 i 4 przedstawiają krzywe ROC dla algorytmów LDA i LSI dla różnych liczb tematów. Dla dużych liczb tematów algorytm LDA spisuje się gorzej, jednak można zauważyć, że klasyfikator uzyskany dla 30 tematów jest podobnej jakości lub lepszy jak ten uzyskany przy użyciu LSI dla 100 tematów.

Na wykresach 5 i 6 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.

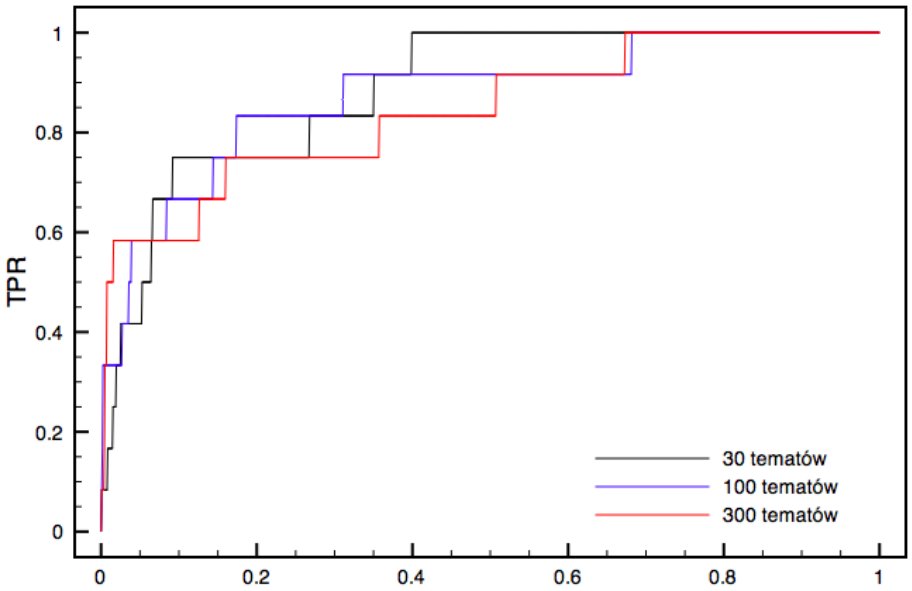
Rysunek 3: Krzywe ROC dla algorytmu LSI dla wybranych liczb tematów



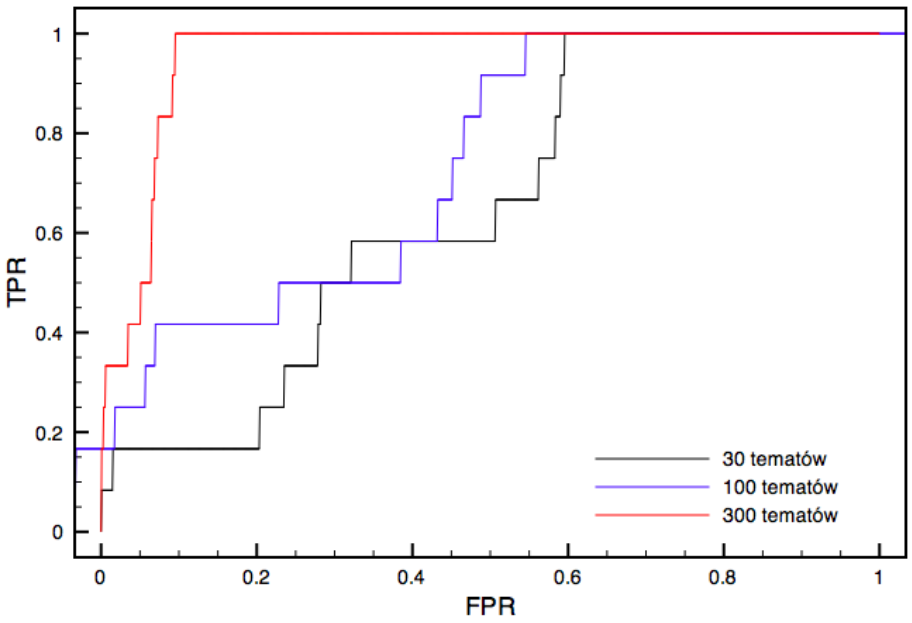
Rysunek 4: Krzywe ROC dla algorytmu LDA dla wybranych liczb tematów



Rysunek 5: Krzywe ROC dla algorytmu LSI dla wybranych liczb tematów bez wykorzystania stemmingu



Rysunek 6: Krzywe ROC dla algorytmu LDA dla wybranych liczb tematów bez wykorzystania stemmingu



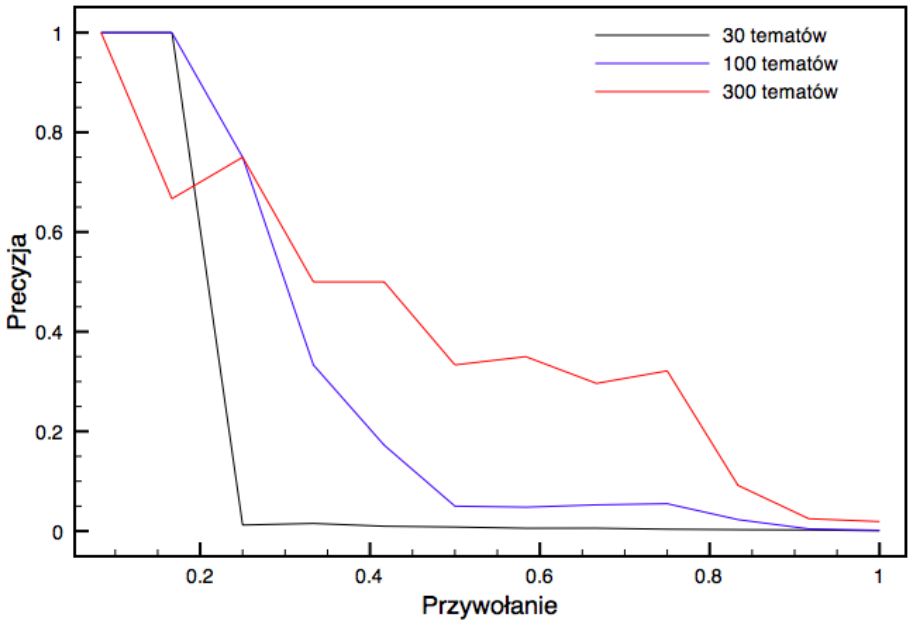
6.3.3 Przywołanie i precyzja

Przywołanie (stosunek liczby zwróconych relewantnych dokumentów do liczby wszystkich relewantncyh dokumentów) i prezycja (stosunek liczby zwróconych relewantncyh dokumentów do liczby wszystkich zwróconych dokumentów) to częste metryki w zadaniach typu information retrieval. Wybranie jakiegoś poziomu przywołania reprezentuje pewien kompromis między kompletnością zwróconych danych, a częstością występowania w nich danych relewantnych, a więc ilością czasu, które musi poświęcić operator systemu na ich dalsze przetworzenie.

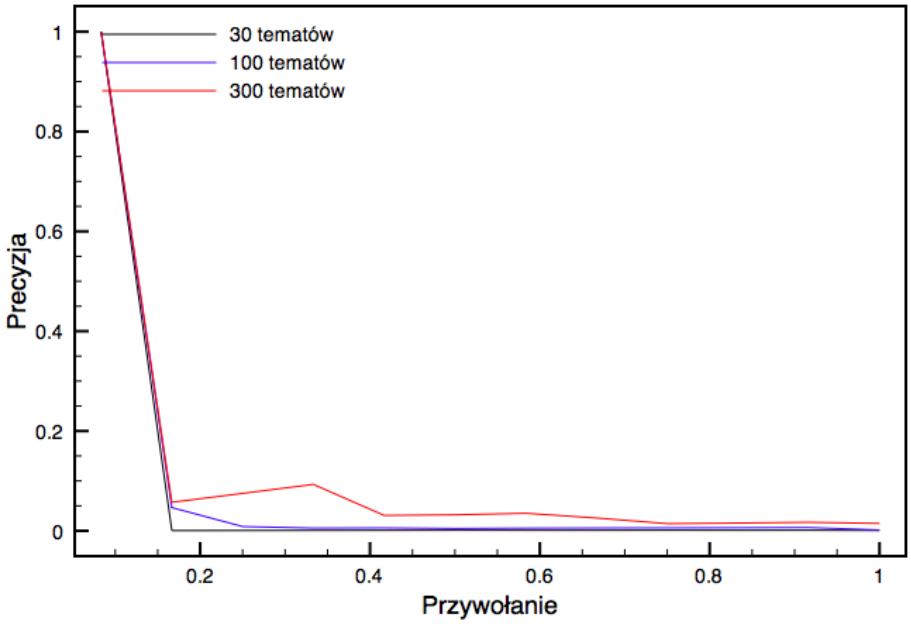
Wykresy 7 i 8 prezentują precyzę osiąganą przez algorytmy LDA i LSI na różnych poziomach przywołania dla przykładowego problemu.

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

Rysunek 7: Precyzja na różnych poziomach przywołania dla algorytmu LSI



Rysunek 8: Precyzja na różnych poziomach przywołania dla algorytmu LDA



6.4 Metryki bez nadzoru (perplexity)

Współczynnik perplexity, który dla pewnych prawdopodobieństw p_i przypisywanych przez model zdarzeniom obliczyć można zgodnie ze wzorem 10, 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.

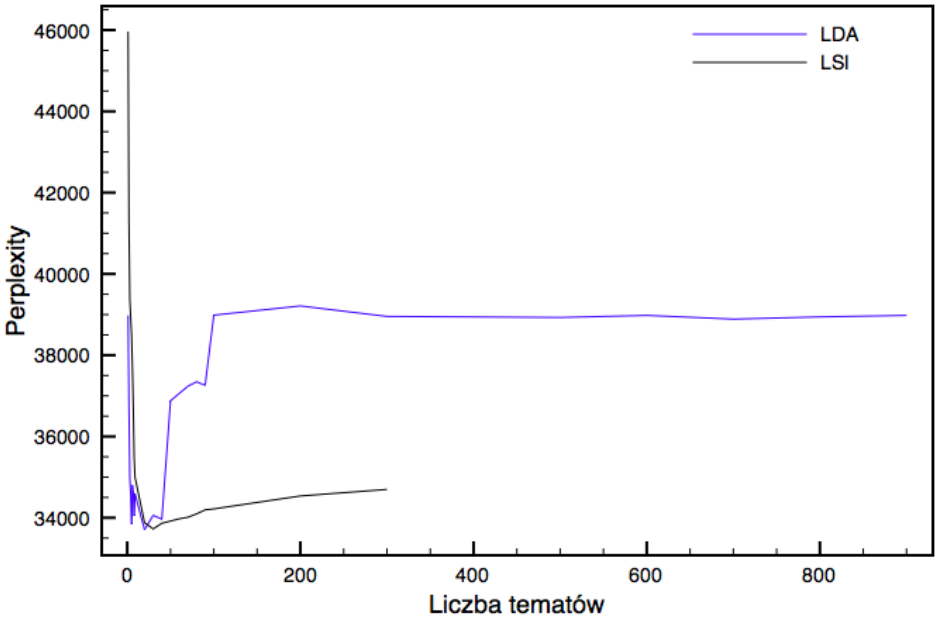
$$\frac{2^{-\sum_{i/1}^{n} p_i log_2(p_i)}}{n} \tag{10}$$

Wartości współczynnika perplexity dla omawianych metod w zależności od liczby tematów są przedstawione na wykresie 9, 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 9: Współczynnik perplexity dla LDA i LSI w zależności od liczby tematów



6.5 Wnioski

Algorytm LDA daje gorszej jakości (a przynajmniej mniej stabilne) wyniki dla typowych problemów klasyfikacji i wyszukiwania informacji spotykanych w codziennej praktyce. 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.

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

7 Podsumowanie

A Sposób użycia kodu

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] 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.

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

[7] 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.

[8] 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.

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

[10] 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.

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

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