WYSZUKIWANIE I PRZETWARZANIE INFORMACJI
LISTA KONTROLNA PYTAŃ, WSKAZÓWEK I PODPOWIEDZI PRZED KOLOKWIUM ZALICZENIOWYM

Termin i miejsce: **9 czerwca 2020, on-line**

Liczba zadań: ok. 12-14 (wiele z podzadaniami, więc trzeba być biegłym w ich rozwiązywaniu)

Zakres materiału (Miłosz Kadziński):
- Analiza połączeń (algorytmy grafowe)
- Analiza użytkowania (logi, zbiory częste, wzorce nawigacyjne)
- Klasyfikacja i rekomendacja
- Grupowanie (analiza skupień + odległość edycyjna)
- Map-Reduce
- Adwords

Zakres materiału (Irmina Masłowska):
- Reprezentacje dokumentów tekstowych (wstępne przetwarzanie tekstów, reprezentacje binarne, *bag-of-words*, inne)
- Modele Information Retrieval (boole'owski, wektorowy VSM, probabilistyczny BIR)
- Miary oceny wyników wyszukiwania
- Relevance feedback i modyfikacja zapytań
- Indeksowanie (indeksy odwrotne, drzewa i tablice sufiksów), indeksowanie rozproszone, kompresja indeksów
- N-gramowe modele lingwistyczne i własność Markova
- Oznaczanie części mowy (POS-tagging)
- Spamowanie wyszukiwarek internetowych

Potencjalne zadania „praktyczne" (jest ich więcej niż będzie, więc jest z czego wybierać). W ramach zadań praktycznych mogą pojawić się pytania uzupełniające o wytłumaczenie/uzasadnienie. Oprócz tego mogą pojawić się pytania dotyczące „praktycznej teorii" w wersji prawda/fałsz lub tak/nie lub „uzupełnianek".

Analiza połączeń (algorytmy grafowe)

I. **PageRank** - dla określonego grafu sieci (rozmiar 3-5 stron/wierzchołków):

- zapisz macierz stochastyczną (uwaga na kolejność wierszy i kolumn (najłatwiej wypełniać kolumnami) oraz wpisywane liczby (0 lub 1/n, gdzie n jest liczbą linków wychodzących od strony z kolumny; oznacz wiersze i kolumny, jeśli kolejność nie jest narzucona z góry; strona może linkować do samej siebie));
- zapisać równania pozwalające na obliczenie wartości PageRank w wersji i) z narzuconym damping factor q (np. 0.2) lub ii) bez jego uwzględnienia (w tym drugim przypadku pamiętaj wtedy o równaniu normalizującym sumę wartości PageRank wszystkich stron);
- rozwiąż prosty układ równań metodą przez podstawianie (uzasadnij, dlaczego PageRank określonej strony jest dobry/słaby);
- bez obliczania dokładnych wartości PageRank uzasadnij na podstawie struktury połączeń, które strony mają najwyższy lub najniższy PageRank;
- inne zagadnienia: interpretacja PageRank (prawdopodobieństwa, zasady kierujące rozdziałem i agregacją PageRank); sposoby obliczania; wskazanie/rozpoznanie dead end lub spider trap; farma linków - sposób manipulacji i sposoby przeciwdziałania.

II. **Inverse PageRank** – zadania jak dla PageRank (pamiętaj o zanegowaniu linków w oryginalnej strukturze grafu sieci).

- inne zagadnienia: interpretacja Inverse PageRank i wykorzystanie w ramach TrustRank.

III. **TrustRank** - dla określonego grafu sieci (rozmiar 3-5 stron/wierzchołków):

- zapisz macierz stochastyczną (jak w PageRank);
- zapisanie wektora początkowego wynikającego z ocen wyroczni (1 - dla stron ocenionych jako zaufane; 0 - dla stron ocenionych jako niezaufane lub nieocenionych; normalizacja);
- zapisać równania pozwalające na obliczenia wartości TrustRank (biased PageRank – uwzględnienie mnożenia q przez początkową ocenę wyroczni); TrustRank oblicza się metodą iteracyjną;
- inne zagadnienia: wybór próbki stron do ocen przez wyrocznię (podstawowe algorytmy: PageRank (dlaczego?) i Inverse PageRank (dlaczego? - idea rozprzestrzenia po sieci; zasady kierujące propagacją zaufania w sieci (podział, dystancs, agregacja) oraz pożądane cechy stron, które powinna ocenić wyrocznia (rozprzestrzenie (braku) zaufania; jaką własność powinna spełniać próbka stron ocenionych przez wyrocznię, aby dojść do każdej strony w sieci?).

IV. **HITS** - dla określonego grafu sieci (rozmiar 3-5 stron/wierzchołków):

- zapisz macierz połączeń (uwaga na kolejność wierszy i kolumn (najłatwiej wypełniać wierszami) oraz wpisywane liczby (0 - brak linka lub 1 - link od strony z wiersza do strony z kolumny));
- obliczenie macierzy LL^T oraz L^TL, której analiza pozwoli na uzyskanie koncentratorów lub autorytetów (transpozycja macierzy i wymnożenie dwóch macierzy);
- zapisać równania pozwalające na obliczenia wartości koncentratorów w funkcji autorytetów na podstawie macierzy połączeń lub autorytetów w funkcji koncentratorów na podstawie transponowanej macierz połączeń;

dla podanych wartości i wektorów własnych dla macierzy LL^T oraz L^TL:

- rozpoznaj moce stron jako koncentratorów i autorytetów (którą macierz odpowiada za koncentratory/autorytety?);
- wskaż najlepsze/najgorsze koncentratory/autorytety i uzasadnij, dlaczego są dobre/złe, wykorzystując sprzężenie zwrotne koncentratory-autorytety (dobry autorytet jest wskazywany przez dobre koncentratory, itd.);
- inne zagadnienia: interpretacja współczynników koncentratorów i autorytetów (wskazuje gdzie znaleźć istotne informacji vs. zawiera istotne informacje; autorytet/koncentrator pobiera swoją moc z ...); sposoby obliczania; rola pierwszego etapu (zbiór korzeń, zbiór bazowy, działanie na podgrafie sieci zależnym od zapytania).

Analiza użytkowania

I. Analiza logów - dla małej próbki (do 10 linii) pliku logu i ew. podanej struktury serwisu (linki między stronami):

- identyfikacja użytkowników, korzystając z pola adresu IP oraz przeglądarki;
- identyfikacja sesji (po uprzedniej identyfikacji użytkowników), korzystając z heurystyki h1, h2 lub href;
- uzupełnianie ścieżek;
- inne zagadnienie: czyszczenie plików log; znaczenie poszczególnych pól w formacie CLF i ECLF.

II. **Zbiory częste i reguły asocjacyjne** – dla podanych sesji użytkowników/koszyków:

- zidentyfikuj zbiory częste przy zadanym minimalnym progu wsparcia, korzystając z algorytmu Apriori (prawidłowa generacja kandydatów na zbiory częste 1, 2, 3, …- elementowe (jakich kandydatów nie można tworzyć?), identyfikacja zbiorów częstych L1, 2, 3, ... i odsianie zbiorów kandydatów C1, 2, 3, ... niespełniających minimalnego progu wsparcia);
- generacja reguł spełniających minimalny próg wsparcia (na postawie zbiorów częstych) i pewności (weryfikacja spełnienia progu współczynnika pewności dla reguły; ile przykładów spełnia regułę/ile przykładów spełnia część warunkową reguły) – postać reguły może być narzucona (np. $item_1 \rightarrow item_2$), ale nie musi (wtedy trzeba wygenerować i zweryfikować wszystkie możliwe reguły);
- inne zagadnienia: własność Apriori pozwalająca na dokonywanie cięć w algorytmie generacji kandydatów; algorytm Apriori (funckja join, weryfikowany warunek, itd.) i jego wykorzystanie w innych dziedzinach (np. dokumenty-termy).

III. **Wykorzystanie łańcuchów Markowa do analizy wzorców nawigacyjnych**:

- dla podanego kilku- lub kilkunastu-sesyjnego pliku log narysuj łańcuch Markova (wierzchołki (Start, Final, strony), połączenia z niezerowymi prawdopodobieństwami, obliczenia prawdopodobieństw na postawie analizy bezpośrednich następstw stron w pliku log);
- dla określonego łańcucha Markova, oblicz prawdopodobieństwa pokonania ścieżki lub prawdopodobieństwo warunkowe pojawienia się jednej strony, jeśli wcześniej wyświetlona stronę inną (sumowanie prawdopodobieństw pokonania wielu ścieżek; uwzględnij, że jedną ze strony może być Start lub Final);
- inne zagadnienia: własność, na której opiera się tworzenie/wykorzystanie łańcucha Markova ("brak pamięci"); wzorce nawigacyjne (ścieżki z wysokim prawdopodobieństwem); zagregowane drzewo sekwencji traktujemy jako ciekawostkę (nie będzie wymagane na kolokwium).

Klasyfikacja i rekomendacja

I. Algorytm k najbliższych sąsiadów (k-NN):

- dla określonej kolekcji dokumentów i ich przydziału do klas, dany jest dokument testowy i jego podobieństwo do dokumentów uczących: zdecyduj i uzasadnij obliczeniami, do jakiej klasy zostanie przypisany wg k-NN z narzuconym k oraz sposobem agregacji głosów (algorytm prosty lub ważony);
- inne zagadnienia: wybór k, różnica w stosunku do klasyfikatora Rocchio, dla jakich grup jest reprezentatywny (distance-based, lazy) i czym się one charakteryzują?;

II. **Naiwny klasyfikator Bayesowski**

- dla określonej kolekcji dokumentów i ich przydziału do klas, dany jest dokument testowy i jego reprezentacja binar/bag-of-words: zdecyduj i uzasadnij obliczeniami (potrafić zapisać na symbolach prawdopodobieństwa warunkowe), do jakiej klasy zostanie przypisany wg naiwnego Bayesa (potencjalnie z koniecznym wygładzeniem danych np. techniką add one) - za przybliżenie końcowego prawdopodobieństwa P(C/Doc) uznajemy wartość licznika (mianownik dla wszystkich klas jest taki sam);
- inne zagadnienia: schemat walidacji krzyżowej (cross) i bootstrap.

III. **Item- oraz user-based collaborative filtering**:

- dla podanych podobieństw przedmiotów/użytkowników (liczby podane, ale trzeba umieć je interpretować i wiedzieć, która wartość oznacza wysokie/niskie podobieństwo), oblicz nieznaną ocenę, korzystając z narzuconej metody (item- lub user-based CF ze średnią, średnią ważoną lub modyfikacją średniej o ważoną odchyłkę od średnich) oraz algorytmu k-NN z danym k;
- inne zagadnienia: różnica między klasyfikacją i predykcją; miary oceny systemów rekomendacyjnych (MAE i RMSE).

IV. **Slope-one predictor**:

- predykcja oceny na podstawie analizy ocen innych przedmiotów; agregacja ocen uzyskanych z porównania z różnymi przedmiotami z wykorzystaniem średniej ważonej (wagi odpowiadają liczbie użytkowników na podstawie których dokonano predykcji oceny).

Grupowanie

I. Algorytm K-means:

- dla podanej reprezentacji dokumentów i macierzy podobieństwa między dokumentami; dokonaj podziału korzystając z K-means, jeśli centroidami w I etapie są np. dokumenty najmniej/najbardziej podobne; oblicz miarę podobieństwa J; oblicz nowe centroidy dla grup, który zostałyby wykorzystane w kolejnym etapie;
- inne zagadnienia: warunki stopu (liczba iteracji, brak zmiany); optimum lokalne (jak dać sobie szanse na globalne?); obliczenia w K-medoids.

II. Aglomeracyjne grupowanie hierarchiczne:

- dla podanej macierzy podobieństwa między dokumentami; dokonaj aglomeracyjnego grupowania hierarchicznego aż do uzyskania zadanej liczby grup (pamiętaj, że zawsze łączymy grupy najbardziej podobne), korzystając z zadanej miary podobieństwa między grupami (np. maksymalne lub minimalne podobieństwo); narysować dendrogram z naniesionymi podobieństwami łączenia grup; warunek stopu (aż do uzyskania jednej grupy lub określone liczby grup >1 lub próg podobieństwa poniżej którego zaprzestajemy łączenia);
- inne zagadnienia: różnica w grupowanie aglomeracyjnym i rozdziałowym.

III. DBSCAN:

- dla podanych obserwacji (punktów), promienia sąsiedztwa oraz minimalnej liczby przykładów niezbędnej do uznania sąsiedztwa za gęste, wskaż dla których punktów sąsiedztwo jest gęste/rzadki; wskaż rdzenie, punkty graniczne i odstające; wskaż grupę, która powstałaby przy starcie z narzuconego punktu (wykorzystanie gęstościowej osiągalności);

IV. Inne zagadnienia:

- obliczenie profilu użytkownia lub zawartości; własności dobrego pogrupowania (podobieństwo wewnętrzne i zewnętrzne); podział algorytmów klasyfikacji;

V. Obliczenie odległości edycyjnej Levenshteina dla dwóch ciągów znaków przy założeniu jednostkowego kosztu operacji edycyjnych z wykorzystaniem programowania dynamicznego:

- wypełnienie macierzy (co w pierwszej kolumnie i wierszu?; wypełnianie wierszami; wypełnianie komórki na podstawie analizy trzech sąsiadów oraz zgodności porównywanych znaków; udzielenie odpowiedzi).

Map-Reduce

- rola Mapper, Reducera, Combinera, Partitionera oraz fazy grupowania i sortowania;
- korzenie Map-Reduce w programowaniu funkcyjnym (Map-Fold w języku Lisp);
- czy typy i wartości w Map-Reduce mogą być złożone? zgodność typów w przypadku wykorzystania Combinera;
- stosowalność Map-Reduce w tworzeniu statystyk na bazie tekstów; indeksowaniu, algorytmach grafowych, wyszukiwaniu (kiedy się dobrze/słabo sprawdza?)

Problem Adwords

- dla zadanego ciągu zapytań oraz reklamodawców z określonymi budżetami, obstawiających różne słowa kluczowe, wykorzystać prosty algorytm BALANCE lub prosty algorytm Adwords; wskazać rozwiązanie optymalne obliczenie w wersji nie online; obliczyć competitive ratio.

Reprezentacje dokumentów tekstowych

I. Wstępne przetwarzanie tekstów:

- standardowe etapy wstępnego przetwarzania tekstów, problemy występujące w rzeczywistych dokumentach/językach naturalnych;
- wpływ wstępnego przetwarzania na ostateczną reprezentację dokumentów/zapytań;
- prawo Zipfa a eliminacja stopwords; cele filtrowania stopwords;
- stemming a lematyzacja – podejścia regułowe, słownikowe i mieszane; cele zastosowania do dokumentów/zapytań;
- dobór termów indeksujących (pojedyncze słowa kluczowe, n-gramy, frazy, etc.).

II. Reprezentacje:

- dla zadanej kolekcji dokumentów tekstowych (np. pojedyncze krótkie zdania) przedstaw ich reprezentację *binarną*, *tf* (z normalizacją lub bez), *tf-idf*;
- oblicz podobieństwo między danymi dwoma dokumentami lub dokumentem a zapytaniem z wykorzystaniem miary Jaccarda, kosinusowej;
- jakie reprezentacje są wykorzystywane w klasycznych modelach IR (boole'owskim, wektorowym VSM, probabilistycznym BIR);
- czy uwzględnienie w reprezentacji dokumentów/zapytań miary *idf* może mieć znaczący wpływ na ostateczny ranking – dlaczego?

Modele Information Retrieval

- schematy działania poszczególnych klasycznych modeli IR;
- założenie o niezależności termów indeksujących w poszczególnych modelach IR;
- wady i zalety poszczególnych modeli w kontekście wyszukiwania informacji (*ranked retrieval*).

Miary oceny wyników wyszukiwania

- miary dokładności (precision) i kompletności (recall) do oceny wyników wyszukiwania, czy polepszanie jednej będzie zwykle pogarszać drugą – dlaczego?, jak może na nie wpłynąć zastosowanie *stemmingu* lub zawężanie/rozszerzanie zapytań?
- miary dla *ranked retrieval* – dla danego rankingu kilku/kilkunastu wyników oblicz wartość miar *P@k*, *R@k*, *MAP*, *R*-precision).

Relevance feedback i modyfikacja zapytań

- cel stosowania i schematy działania technik *explicit, implicit, pseudo relevance feedback*;
- metoda Rocchio dla modelu wektorowego – wyznacz postać zmodyfikowanego zapytania przy danych wagach α, β, γ;
- cele i metody modyfikacji zapytań.

Indeksowanie

I. Indeksy w zadaniach IR:

- indeks odwrotny (pełny (pozycyjny)/blokowy) – budowa (składowe), konstrukcja, złożoność pamięciowa, eksploatacja w obsłudze zapytań typu *AND*, *OR*, wyszukiwaniu fraz;
- wykorzystanie indeksu odwrotnego do obsługi zapytań w modelu wektorowym (*ranked retrieval*);
- wykorzystanie struktury *trie* w konstrukcji indeksu odwrotnego;
- dla zadanego tekstu narysuj strukturę *trie*, *suffix trie*, *suffix tree lub suffix array* indeksującą tekst z dokładnością do pojedynczych znaków lub początków termów indeksujących;
- podobieństwa i różnice (w budowie, eksploatacji) między indeksem odwrotnym a indeksem w postaci tablicy sufiksów z supraindeksem po termach indeksujących.

II. Indeksowanie rozproszone:

- motywacja, realizacja, zalety/wady rozproszenia indeksu względem termów/dokumentów;
- która technika jest typowo wykorzystywana w rzeczywistych wyszukiwarkach?; rozproszenie a replikacja.

III. Kompresja indeksów:

- podatność na kompresję (poszczególnych składowych) indeksu odwrotnego/tablic sufiksów – z czego wynika?;
- kodowanie liczb całkowitych; zapisz daną liczbę z wykorzystaniem kodowania γ, δ; liczbę zapisaną za pomocą kodowania γ, δ zapisz w systemie dziesiętnym.

N-gramowe modele lingwistyczne i własność Markova

- wyznacz prawdopodobieństwa warunkowe wystąpień słów dla modelu n-gram (n=1,2,3,…) na podstawie danej kolekcji zdań;
- dla podanej sekwencji słów ustal najbardziej prawdopodobne kolejne słowo dla modelu n-gram (n=1,2,3,…) na podstawie danej kolekcji zdań;
- oblicz prawdopodobieństwo zdania w modelu *bigram* lub *trigram* na podstawie danej kolekcji uczącej;
- wymień zastosowania modeli n-gramowych.

Oznaczanie części mowy (POS-tagging)

- założenia w HMM POS-tagging;
- mając dane macierze *T* i *E* dla max. 3 stanów ukrytych i tyluż widzialnych wyznaczyć najbardziej prawdopodobny ciąg stanów ukrytych dla podanego ciągu (długości max. 3) stanów widzialnych.

Spamowanie wyszukiwarek internetowych

- techniki *content*, *link*, *click spamming*;
- techniki ukrywania spamu („optyczne”, skrypty*, cloaking*, przekierowania);
- zwalczanie spamu jako problem klasyfikacji – istotne cechy (*features*) stron.