Wyszukiwanie i Przetwarzanie Informacji WWW

Wyszukiwanie w korpusach tekstowych: ranking i ewaluacja

Marcin Sydow

Web Mining Lab, PJWSTK

Plan dzisiejszego wykładu:

- Wprowadzenie
- Ranking - Model Wektorowy
- Ewaluacja Systemów Wyszukiwawczych
- Podsumowanie Wykładu

Klasyczny tekstowy system IR (tzw. "boolowski")

Zasada działania (przypomnienie):

- Mamy korpus dokumentów tekstowych D.
- Mamy zapytanie boole'owskie q traktowane jako zbiór albo lista słów kluczowych.
- System ma zwrócić dokumenty z D odpowiadające zapytaniu q.

Istotne jest to, że zwraca się **wszystkie i tylko te** dokumenty, które **dokładnie** pasują do zapytania. Stąd nazwa "boole'owskie".

Założenia w klasycznym IR

Zakłada się:

- wysoką jakość tekstów w korpusie (przygotowane przez ludzi)
- brak zaszumienia i jednorodność dokumentów (język, rozmiar, format, etc.)
- brak czynnika "wrogości"

Założenia te są istotne dla modelu - warunkują metody wyszukiwania. Nie są one spełnione np. w WWW (WIR).

Problem nadmiaru wyników w "boolowskim" IR

Do tej pory zajmowaliśmy się głównie tym, jak obliczyć zbiór wszystkich dokumentów, które zawierają słowa kluczowe wg zapytania.

W praktyce, bardzo istotnym problemem jest nadmierna ilość dokumentów spełniająca kryteria wyszukiwania, przy ograniczonych możliwościach ich prezentacji i przetworzenia przez użytkownika. Co zrobić, gdy np. dokładnie 10000 dokumentów spełnia zapytanie? (przy czym, zauważmy, że każdy dokument spełnia "tak samo dobrze" zapytanie - stąd właśnie nazwa: "boolowski") Jak wybrać dokumenty do prezentacji? W czystym modelu "boolowskim" nie ma dobrego naturalnego rozwiązania tego problemu.

Omówimy teraz jak **rozszerzyć model** by rozwiązać ten istotny problem.

Ranking Wyników

Najpopularniejszym sposobem na rozwiązanie problemu ograniczonych możliwości prezentacji i przetwarzania wyników wobec ich nadmiaru w modelu "boolowskim" jest **uporządkowana forma prezentacji**.

Dla każdego dokumentu, spełniającego kryteria zapytania, obliczana jest tzw. **miara odpowiedniości** (ang. relevance measure) i następnie wszystkie dokumenty są prezentowane w kolejności od "najbardziej odpowiadającego" zapytaniu do "najmniej odpowiadającego".

W ten sposób "sztywny" model "boolowski" zastąpiony jest nieco bardziej "rozluźnionym", gdzie dokumenty mogą pasować "bardziej" lub "mniej" do zapytania.

Miara odpowiedniości oparta na podobieństwie tekstowym: model wektorowy

Miarę **odpowiedniości** (ang. relevance) wylicza się m.in. na podstawie podobieństwa tekstowego pomiędzy reprezentacją zapytania a reprezentacją dokumentu.

W szczególności, podobieństwo tekstowe można oprzeć na tzw. **modelu wektorowym**.

Model wektorowy, TF/IDF

Każdy dokument to wektor. Osie (wymiary) odpowiadają tokenom.

Współrzędna t dokumentu d zależy od dwóch rzeczy:

- Częstość termu w dokumencie: TF(d,t)
- Odwrotność ilości dokumentów zawierających t: IDF(t)

Intuicja dla IDF(t) - nie wszystkie tokeny mają równą wartość dyskryminacyjną - jeśli token pojawia się w bardzo wielu dokumentach, jego wartość informacyjna jest niska.

Ogólniej, im więcej dokumentów zawiera token t, tym mniej niesie on informacji.

Częstość dokumentu TF(d,t)

Klasycznie jest to liczba wystąpień termu t w dokumencie d (ozn. n(d,t)) podzielona przez czynnik normalizacyjny.

Czynnik normalizacyjny N(d) (przykłady):

- długość dokumentu: $N(d) = \sum_\tau n(d, \tau)$
- maksymalna częstość wystąpień tokenu w dokumencie: $N(d) = max_\tau n(d, \tau)$

Wzór na TF(d,t) jest wtedy postaci: $TF(d, t) = n(d, t)/N(d)$

Przykład: wariant TF(d,t)

Np. w systemie SMART (Cornell University) użyto nieco innej miary:

$$TF(d,t) = \begin{cases} 0 & n(d,t) == 0 \\ 1 + log(1 + log(n(d,t))) & w.p.p. \end{cases}$$

IDF(t)

Wielkość IDF(t) maleje ze wzrostem ilości dokumentów zawierających t. D - zbiór wszystkich dokumentów w kolekcji, D_t - zbiór dokumentów zawierających t

Definition

$$IDF(t) = log \frac{1+|D|}{1+|D_t|}$$

Używa się też innych wariantów funkcji $\frac{|D|}{|D_t|}$

Model TF-IDF

Ostatecznie, w modelu wektorowym TF-IDF reprezentacji dokumentów, dokument-wektor d ma na każdej współrzędnej t wartość:

Definition

$$d(t) = TF(d,t) \cdot IDF(t)$$

Reprezentacja zapytania

Zapytanie q też może być reprezentowane w takim samym modelu (tzn. jako wektor indeksowany tokenami ze słownika).

Wtedy ranking dokumentów w odpowiedzi na zapytanie q oblicza się stosując miary podobieństwa wektorów.

Miary podobieństwa wektorów

Podstawowymi miarami (nie)podobieństwa wektorów (np. d i q) są:

- odległość wektorów $|d - q|$ (niepodobieństwo)
- kosinus kąta między wektorami $cos(d, q)$ (podobieństwo)

Odległość wektorów

Odległość euklidesowa dana jest wzorem:

Definition

$$|d - q| = \sqrt{\sum_t (d(t) - q(t))^2}$$

(można też brać sumę modułów różnic - tzw. "metryka miejska")

Zauważmy, że przy takiej mierze długie dokumenty są poszkodowane - są "dalej" od (z zasady krótkich) zapytań.

Aby to poprawić, stosuje się normalizację długości dokumentów.

Kosinus kąta między wektorami

Mierzy podobieństwo kierunku wektorów. Im podobniejsze wektory tym mniejszy kąt między nimi (a tym samym większy kosinus). Dla identycznych: 1, dla prostopadłych: 0 (zauważmy: boolowski operator negacji!)

Definition

$$cos(q, d) = \frac{d \cdot q}{|d||q|}$$

W tym wypadku, z kolei, krótsze dokumenty są poszkodowane, gdyż jest mniejsza szansa na zawieranie tokenów z zapytania.
Mimo to, częściej używa się miar bazujących na kosinusie niż na odległości.

Ewaluacja Systemu IR

- Pełność (ang. Recall)
- Pełność na pozycji k (ang. “at k”)
- Precyzja (ang. Precision)
- Precyzja przeciętna
- F-miara (ang. F-measure)

Recall/Precision: Pojęcia pomocnicze

Kolekcja D wszystkich N dokumentów i zapytanie q.

- $Returned_q$ - zbiór dokumentów zwróconych przez system na zapytanie q.
- Rel_q - zbiór wszystkich dokumentów w kolekcji istotnie odpowiednich dla zapytania q (ang. relevant to q)
- R_q - uporządkowana lista wyników zapytania zwrócona przez system
- $R_q[i]$ - i-ty dokument na powyższej liście
- $rel_q(i) = \lfloor R_q[i] \in Rel_q \rceil$ ("czy i-ty zwrócony dokument jest odpowiedni")

Recall

Definition

$$Recall_q = \frac{|Returned_q \cap Rel_q|}{|Rel_q|}$$

Czyli: "jaki procent wszystkich odpowiednich dokumentów zwrócił system".

Rzadziej używane: "Recall @ k" ("Recall at k"):

Definition

$$Recall_q(k) = \frac{1}{|Rel_q|} \sum_{1 \leq i \leq k} rel_q(i)$$

Czyli: "jaki procent z wszystkich odpowiednich dokumentów jest na pierwszych k pozycjach". Przyjmuje bardzo niskie wartości dla niskich k i bogato reprezentowanych zapytań.

Precyzja (ang. precision)

Definition

$$Precision_q = \frac{|Returned_q \cap Rel_q|}{Returned_q}$$

Czyli: "jaki procent zaprezentowanych wyników jest rzeczywiście odpowiedni"

Precision @ k (bardzo ważna dla wyszukiwarek!):

Definition

$$Precision_q(k) = \frac{1}{k} \sum_{1 \leq i \leq k} rel_q(i)$$

Czyli: jaki procent pierwszych k wyników jest rzeczywiście odpowiednich

Inne pochodne miary

Definition

F-miara (ang. F-measure):

$$F = \frac{2 \cdot P \cdot R}{P + R}$$

Definition

Przeciętna precyzja:

$$averagePrecision_q = \frac{1}{|Rel_q|} \sum_{1 \leq k \leq |Returned_q|} rel_q(k) \times Precision_q(k)$$

Podsumowanie Precision/Recall

Recall: (bogactwo wyników) jak dużo odpowiednich wyników system wychwycił spośród dostępnych.

Precision: (czystość wyników) jak dużo spośród wychwyconych wyników jest odpowiednich.

W wyszukiwarkach ważne są te wartości szczególnie dla k pierwszych pozycji (gdzie k to ilość wyników np. na pierwszym ekranie)

Można powiedzieć, że zbieracz (ang. crawler) i indeks dba o wysoką wartość Recall. Natomiast algorytmy rankingowe dbają o wysoką wartość Precyzji.

Naturalnie, Recall nie można obliczyć dla całego WWW (ewentualnie dla jego zindeksowanej pod-kolekcji).

Zależność Recall/Precision

Ustalmy zestaw progów np. 0, 0,1, 0,2, ..., 1. Ustalmy zapytanie q i uporządkujmy wszystkie dokumenty z kolekcji. Dla każdego progu można wtedy zmierzyć jaka jest najwyższa precyzja dla dowolnej wartości Recall większej lub równej od danego progu (dla Recall 0 przyjmuje się wartość precyzji 1). Nazywa się to (ang.) "Interpolated Precision".
Można wtedy zrobić wykres (x: progi, y: precyzja) – zwany Precision/Recall.
Można też uśrednić te wartości po pewnym zbiorze zapytań Q.
Dobry algorytm rankingowy sprawia, że krzywa nie jest nigdzie rosnąca.
Można w ten sposób porównywać systemy: np. krzywą leżąca powyżej oznacza lepszy system (można również porównywać pola pod krzywymi)

Przykłady innych modeli wyszukiwania

Wyszukiwanie na podstawie przykładu

Jeśli dokumenty nie są tekstem, ale np. plikami graficznymi lub muzycznymi, można zastosować metodę wyszukiwania na podstawie "przykładu" (ang. query by example) . Dokładniej:

- korpus składa się z dokumentów multimedialnych określonego typu (np. pliki grafiki 2-D)
- zapytanie q jest również plikiem takiego samego typu

W modelu takim, zapytanie jest interpretowane następująco: "znajdź dokumenty **podobne** do q". System oblicza wtedy (np. na podstawie pewnych atrybutów q i dokumentów z korpusu, takich jak spektrum kolorów, kształty, etc.) pewną **miarę podobieństwa** między q i dokumentami i zwraca te ostatnie posortowane niemalejąco wg wartości tej miary.

Wyszukiwanie XML

W przeciwieństwie do wyszukiwania w bazach danych, wyszukiwanie w kolekcjach dokumentów tekstowych czy WWW dotyczy dokumentów bardzo słabo ustrukturyzowanych.

Pewną formą pośrednią w sensie stopnia ustrukturyzowania jest wyszukiwanie w kolekcjach o wyraźniejszej strukturze niż "wolny" tekst i jednocześnie słabszej niż w bazach daych. Przykładem takich kolekcji są kolekcje dokumentów XML (Extensible Markup Language), gdzie stosuje się pewne specjalne techniki (m.in. związane z eksploracją **struktury drzewa dokumentu** XML).

Wyszukiwanie Semantyczne

Ostatnio, rosnącą rolę mają tzw systemy "wyszukiwania semantycznego":

- baza wiedzy (np. w formie grafu wiedzy typu RDF)
- zapytanie (np. w języku SPARQL)

Na razie systemy te są w fazie prototypów, ale pozwalają na formułowanie całkiem złożonych zapytań typu "semantycznego", będących poza możliwościami klasycznych wyszukiwarek WWW, np.:

"Podaj nazwę miasta, gdzie zmarła polska badaczka, która w XX w. dostała tę samą prestiżową nagrodę co Niels Bohr."

Powyższe zapytanie jest praktycznie nie do wykonania w klasycznej wyszukiwarce.

Lektury

Uzupełnić wiedzę można np. w poniższych publikacjach:

Podstawy IR są opisane w klasycznych pozycjach:

- G.Salton et al. "Introduction to Modern Information Retrieval", McGraw-Hill, 1983
- W.B. Frakes, R. Baeza-Yates "Information Retrieval: Data Structures and Algorithms", Prentice Hall, 1992

Tworzenie i kompresję indeksu opisano w książce:

- I.H. Witten, A. Moffat, T.C. Bell "Managing Gigabytes: Compressing and Indexing Documents and Images", Morgan Kaufmann, 1999

Ponieważ materiał tej prezentacji jest podstawowy, nie wymienia się tutaj specjalistycznych publikacji naukowych.

Na zaliczenie tego wykładu:

1. dlaczego model "boolowski" jest rozszerzany o ranking wyników?
2. czynniki uwzględniane przy obliczaniu "odpowiedniości" tekstowej
3. model wektorowy dla tekstu
4. miary podobieństwa wektorów
5. TF/IDF
6. ewaluacja systemu
7. precyzja
8. pełność
9. pochodne miary ewaluacji (np. F-miara)
10. dokumenty nietekstowe: wyszukiwanie na podstawie "przykładu"

Dziękuję za uwagę

Dziękuję za uwagę.