Eksploracja zasobów internetowych

Wykład 2

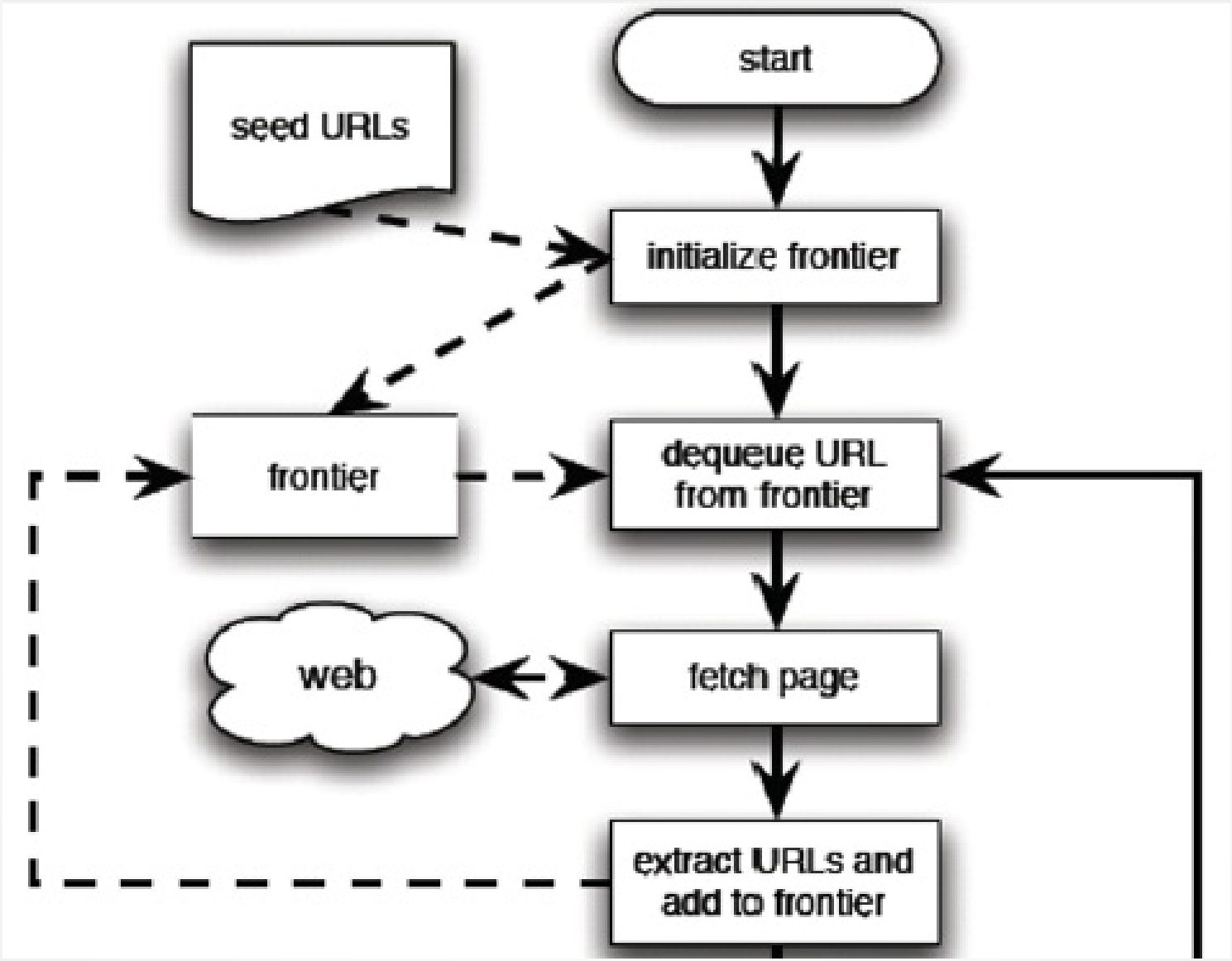
Badanie struktury sieci WWW i pobieranie treści ze stron WWW

Piotr Hońko
Wydział Informatyki
Politechnika Białostocka

Czym jest robot sieciowy?

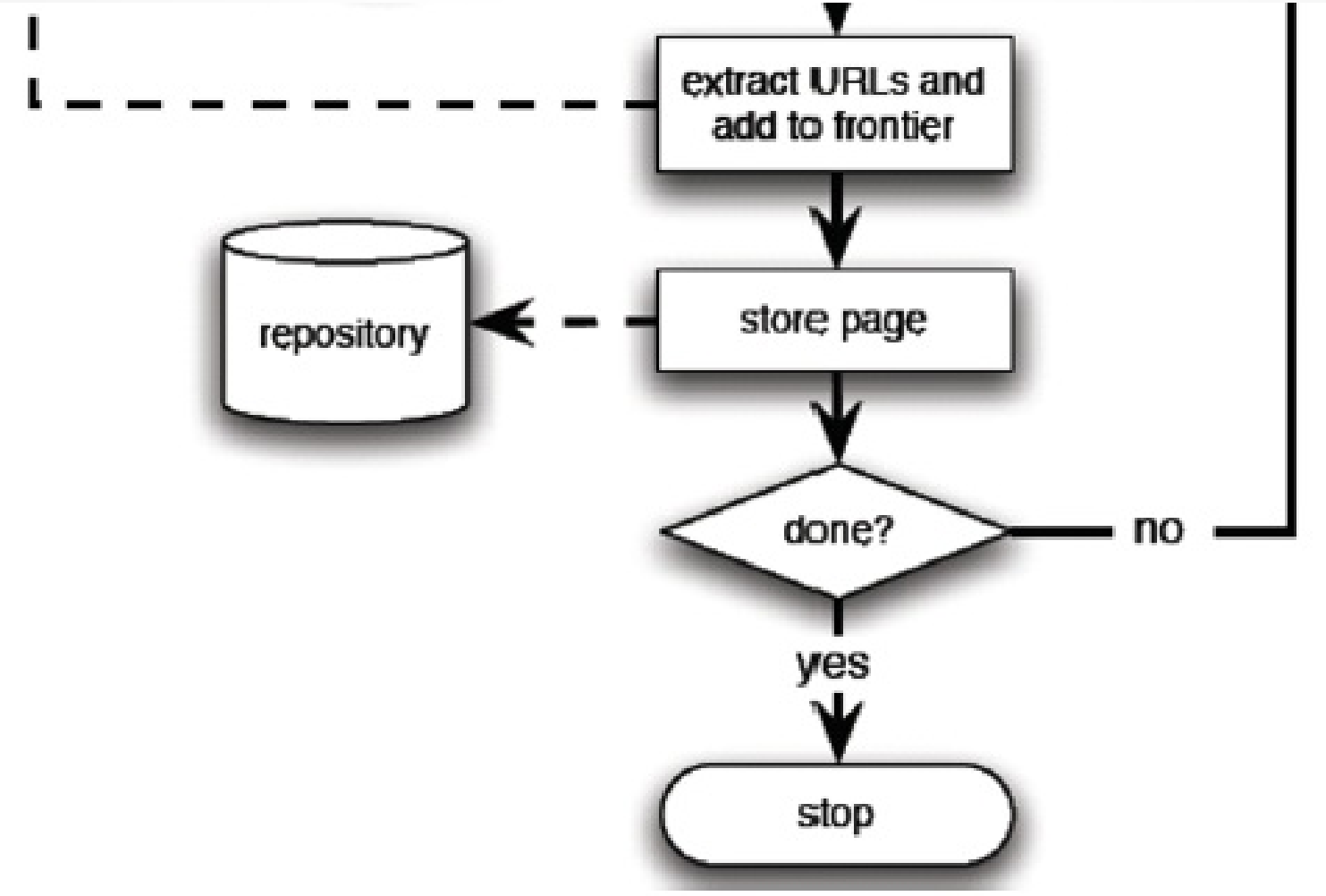
- Robot sieciowy (pająk, pełzacz (ang. *crawler*)) – program, który automatycznie przeszukuje strony internetowe w celu pobrania z nich istotnych danych lub informacji.
- Robot internetowy jest źródłem danych potrzebnych do eksploracji zasobów internetowych.

Ogólny schemat działania robota sieciowego



Źródło: Liu B.: Web Data Mining. Exploring Hyperlinks, Contents, and Usage Data, Second Edition, Springer-Verlag, Berlin Heidelberg, 2011

Ogólny schemat działania robota sieciowego c.d



Źródło: Liu B.: Web Data Mining. Exploring Hyperlinks, Contents, and Usage Data, Second Edition, Springer-Verlag, Berlin Heidelberg, 2011

Metody przeszukiwania sieci

- Metody ślepe
  - Wszerz (tworzenie kolejki FIFO),
  - W głąb (tworzenie kolejki LIFO).
- Metody heurystyczne
  - Pierwszy najlepszy (tworzenie kolejki priorytetowej).
- Zastosowana metoda powinna umieć radzić sobie z grafami cyklicznymi.

Robot sieciowy jako element silnika wyszukiwarek internetowych

- Główne elementy składowe wyszukiwarki internetowej:
  - robot przeszukujący sieć WWW,
  - parser kodu HTML,
  - indekser stron WWW,
  - moduł wyszukiwania dokumentów WWW,
  - interfejs wyszukiwarki.

Robot przeszukujący sieć WWW

- Celem robota jest pobranie adresów stron internetowych oraz danych, które potencjalnie są interesujące/ważne dla użytkowników wyszukiwarek.
- Praca robota może zostać zakończona po przejrzeniu wszystkich dostępnych stron (na najgłębszym poziomie) lub określonej z góry ich liczby.

Parser HTML

- Rolą parsera jest analiza składniowa kodu HTML w celu identyfikacji linków znajdujących się na danej stronie.
- Parser może również dokonać bardziej zaawansowanej analizy składni HTML, przykładowo może określić poprawność budowy danej strony.
- W celu radzenia sobie z niepoprawnie napisanymi stronami parser może wykorzystać dodatkowe narzędzie, które automatycznie oczyszcza kod z błędów.

Indekser stron WWW

- Celem indeksera jest zarejestrowanie nowego linku strony w celu zapisania go w repozytorium.
- Dzięki indeksacji nowa strona staję się dostępna z poziomu danej wyszukiwarki.
- Indeksacja jest również wykonywana w celu przyśpieszenia automatycznej aktualizacji stron, dokonywanej przez robota sieciowego.

Moduł wyszukiwania dokumentów WWW

- Moduł jest odpowiedzialny za
  - przekształcenie żądania użytkownika zgłaszanego do danej przeglądarki do postaci poprawnego zapytania do baza dokumentów WWW;
  - pobranie z bazy wyników zapytania;
  - posortowanie wyników względem ich zgodności z zapytaniem użytkownika.

Interfejs wyszukiwarki

- Interfejs jest odpowiedzialny za komunikowanie się użytkownika z bazą zindeksowanych stron (formułowanie zapytania, prezentacja wyniku) oraz za bezpośredni dostęp do stron wyświetla-nych jako wynik zapytania.

Ukierunkowany robot sieciowy (ang. *focused crawler*)

- W odróżnieniu od ogólnego robota sieciowego robot ukierunkowany przeznaczony jest do przeszukiwania sieci pod szczegółowo określonym kątem.
- Ukierunkowany robot jest bardziej selektywny; wyszukuje stron które spełniają określone kryterium jakościowe lub semantyczne.

Robot sieciowy ukierunkowany na kategorie stron – przykład

- Robot wyszukuje stron, które w największym stopniu pasują do kategorii określonych przez użytkownika.
- Robot jest nawigowany przez klasyfikator (naiwny klasyfikator bayesowski).
- Klasyfikator jest w stanie obliczyć dla każdej kategorii jej prawdopodobieństwo występowania na danej stronie.

Tematyczny robot sieciowy (ang. *topical crawler*)

- Celem robota jest znalezienie stron (dokumentów), które są zgodne pod względem tematycznym z dostarczoną przez użytkownika charakterystyką.
- Tematyczność może być określona poprzez opis lub wskazanie przykładowych stron, które spełniają oczekiwania użytkownika.
- Fragment sieci, który może być odpowiedni do znalezienia oczekiwanych stron, jest zwykle przeszukiwany w czasie rzeczywistym.

Strategie weryfikacji stron

- Topologia leksykalna – dwie strony są bliskie sobie, jeżeli zawierają podobne treści.
  - Wykorzystanie modelu wektorowego do reprezentacji stron oraz zastosowanie miar podobieństwa operujących na wektorach.
- Topologia połączeń – dwie strony są bliskie sobie, jeżeli ścieżka, łącząca je jest krótka.
  - Wykorzystanie grafu do reprezentacji połączeń pomiędzy stronami i narzędzi mierzenia długości ścieżki.

Topologia leksykalną a topologia połączeń

- Topologia leksykalna jest zbieżna z topologią połączeń, jeżeli prawdopodobieństwo występowania podobieństwa leksykalnego dwóch strony, które są połączone ze sobą jest większe niż dwóch losowo wybranych stron.
- Obie topologie mogą być porównane pośrednio poprzez określenie zbieżności każdej z nich z semantyką badanych stron (sieć semantyczna).

Struktury do reprezentowania grafu

- Graf skierowany, który odzwierciedla fragment sieci może być zapisany za pomocą:
  - macierzy sąsiedztwa

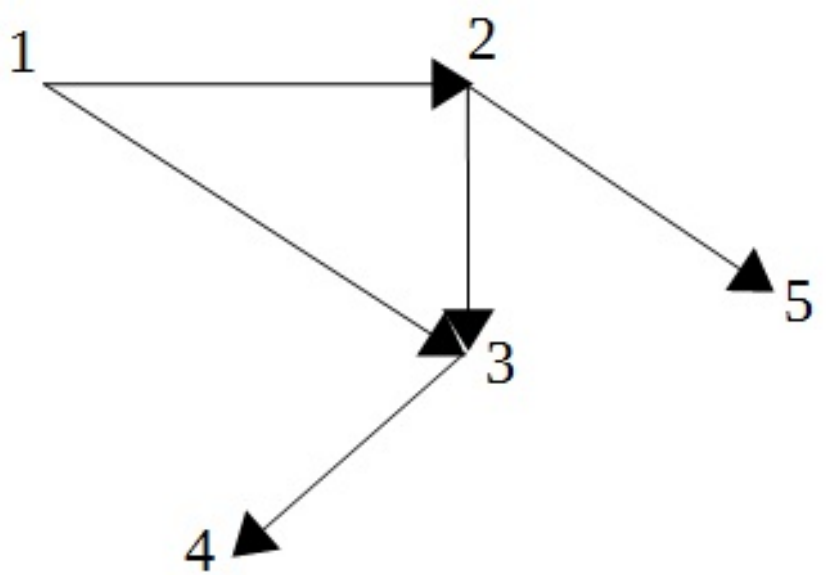
$$M[i,j]=\begin{cases}1 & \text{gdy wierzchołki } i \text{ i } j \text{ są połaczone},\\ 0 & \text{w przeciwnym przypadku.}\end{cases}$$

  - listy incydencji

$$j \in L[i], \text{ gdy wierzchołki } i \text{ i } j \text{ są połaczone}.$$

Struktury do reprezentowania grafu – przykład

- Graf



- Macierz sąsiedztwa

	1	**2**	**3**	**4**	**5**
1	0	1	1	0	0
2	0	0	1	0	1
3	0	0	0	1	0
4	0	0	0	0	0
5	0	0	0	0	0

- Lista incydencji

L[1]: 2, 3
L[2]: 3, 5
L[3]: 4

Wstępne przetwarzanie danych sieciowych

- Wstępne przetwarzanie pobieranych danych może przebiegać dwuetapowo:
  - Przetwarzanie stron HTML w celu identyfikacji istotnych elementów, które mają posłużyć do utworzenia bazy danych.
  - Przetwarzanie tekstu pobranego ze treści strony HTML w celu identyfikacji termów.

Wstępne przetwarzanie stron HTML

- Etapy przetwarzania stron HTML:
  - Identyfikacja różnego rodzaju pól tekstowych;
  - Identyfikacja tekstu kotwic;
  - Usuwanie znaczników HTML;
  - Identyfikacja głównych bloków treści strony.

Identyfikacja różnego rodzaju pól tekstowych

- Kod HTML zawiera pola tekstowe różnego typu, np. tytuł, metadane, ciało dokumentu.
- Sposób pozyskiwania danych z tych pól i ich znaczenie mogą być uzależniony od ich typu.
- Dane pobrane z bardziej istotnych pól, np. tytuł, są traktowane priorytetowo poprzez przypisanie im odpowiednio wysokich wag (stopień istotności).

Identyfikacja tekstu kotwic

- Tekst kotwicy zawiera pewną informację o treści strony, do której prowadzi link opisany tym tekstem.
- Odpowiednio dobrany tekst kotwicy stanowi w miarę precyzyjny opis wskazywanej strony.
- Teksty kotwic do stron zewnętrzny ogólnie są bardziej wiarygodne niż te, które prowadzą do stron danego serwisu.

Usuwanie znaczników HTML

- Znaczniki HTML stanowią jedynie informację o organizacji strony; nie są one zapisywane w tworzonej bazie danych.
- Przed usuwaniem znaczników powinna być określona strategia przetwarzania zawartego w nich tekstu. Nie wskazane jest traktowanie jak całości treści, które pozostanie po usunięciu znaczników.

Identyfikacja głównych bloków treści strony

- Strona HTML może składać się z wielu bloków zawierających treść, z których tylko część jest istotna.
- Główny blok może zostać zidentyfikowany na podstawie:
  - analizy szablonu, który został wykorzystany do budowy strony (w przypadku różnych stron bloki głównego tekstu bardziej różnią się od siebie niż pozostałe bloki);
  - lokalizacji fizycznej na wyświetlanej stronie (bloki głównego tekstu są zwykle umieszczane na środku strony).

Przetwarzanie pobranego tekstu

- Etapy przetwarzania tekstu pobranego ze strony HTML (tworzenie termów):
  - Usuwanie nieistotnych słów;
  - Identyfikacja rdzeni słów;
  - Usuwanie nieistotnych cyfr i znaków interpunkcyjnych oraz formatowanie tekstu.

Usuwanie nieistotnych słów (ang. stopwords)

- Częste słowa, które są niezbędne do stworzenie poprawnej treści pod względem gramatycznym i logicznym, nie stanowią istotnego źródła informacji o treści strony (spójniki, przyimki, zaimki (?)).
- Słowa takie również są usuwane z zapytań użytkowników.

Identyfikacja rdzeni słów (ang. stemming)

- Syntaktyczna postać danego słowa jest zależna od budowy gramatycznej zdania, w którym owo słowo występuje (odmiana przez przypadki, osoby, liczby, czasy; dodanie prefiksu lub sufiksu).
- Syntaktyczna postać słowa jest mniej istotna niż semantyczna.
- Określenie semantycznej postaci może być wykonane poprzez identyfikacje rdzenia danego słowa – fragment słowa, który jest stały niezależnie od odmiany danego słowa, np.

  rdzeniem słów: **opisać**, **przepisać**, **dopisać** jest **pisać***;*

  ale rdzeniem słów: **piszę**, **piszesz**, **pisał** jest **pis** (?).

Usuwanie nieistotnych cyfr i znaków interpunkcyjnych oraz formatowanie tekstu

- Cyfry, które nie tworzą istotnych ciągów znaków są usuwane (np. data – istotny ciąg cyfr).
- Znaki interpunkcyjne są usuwane lub zastępowane spacją,
  np. state-of-art → state of art.
- Wszystkie słowa są zapisywane według tego samego szablonu, np. małymi literami.

Struktury do zapisu danych pobranych z sieci WWW

- Format zapisu danych uzależniony jest od przeznaczenia tworzonej bazy danych (np. baza wyszukiwarki internetowej, baza do eksploracji treści stron).
- Struktury, które mogą być wykorzystane do różnych zastosowań:
  - Model boolowski;
  - Model ilościowy;
  - Model pozycyjny;
  - Model wektorowy.

Model boolowski

- Dokumenty są reprezentowane przez (uporządkowane) zbiory wag termów.
- Model określa wystąpienie termu t_i w dokumencie d_i:

$$w_{ij} = \begin{cases} 1 \text{ gdy } t_i \text{ występuje w } d_j, \\ 0 \text{ w przeciwnym przypadku.} \end{cases}$$

Model boolowski – przykład

DID	lab	laboratory	programming	computer	program
d_1	0	0	0	0	1
d_2	0	0	0	0	1
d_3	0	1	0	1	0
d_4	0	0	0	1	1
d_5	0	0	0	0	0
d_6	0	0	1	1	1
d_7	0	0	0	0	1
d_8	0	0	0	0	1
d_9	0	0	0	0	0
d_{10}	0	0	0	0	0

Źródło: Markov Z., Larose D.T.: Eksploracja zasobów internetowych. Analiza struktury, zawartości i użytkowania sieci WWW, Wydawnictwo Naukowe PWN, Warszawa, 2009.

Model boolowski – cechy

- Przydatny w systemach opartych o logikę dwuwartościową, np. wynik zapytania do bazy albo spełnia kryteria wyszukiwania w pełni albo wcale.
- Rzadko wykorzystywany w praktyce, gdyż w mniejszym stopniu umożliwia operowanie na częściowej zgodności dokumentów ze sobą lub z zapytaniem.

Model ilościowy

- Dokumenty są reprezentowane przez (uporządkowane) zbiory wag termów.
- Model określa liczbę wystąpień termu t_i w dokumencie d_i:

$$w_{ij} = \sum_{k=1}^{n} \left(term\left(d_j, k\right) = t_i \right),$$

gdzie $term\left(d_j, k\right)$ zwraca term występujący na pozycji *k* w dokumencie d_j, a *n* jest liczbą termów występujących w dokumencie d_j.

Model ilościowy – przykład

DID	lab	laboratory	programming	computer	program
d_1	0	0	0	0	1
d_2	0	0	0	0	1
d_3	0	2	0	1	0
d_4	0	0	0	1	2
d_5	0	0	0	0	0
d_6	0	0	2	6	3
d_7	0	0	0	0	2
d_8	0	0	0	0	1
d_9	0	0	0	0	0
d_{10}	0	0	0	0	0

Źródło: Markov Z., Larose D.T.: Eksploracja zasobów internetowych. Analiza struktury, zawartości i użytkowania sieci WWW, Wydawnictwo Naukowe PWN, Warszawa, 2009.

Model ilościowy – cechy

- Przydatny, gdy czynnik ilościowy jest uwzględniany przy porównywaniu dokumentów ze sobą lub z zapytaniem.
- Stanowi raczej reprezentację początkową niż bazę docelową, np. umożliwia obliczenie częstotliwości występowania termu w dokumencie.
- Umożliwia utworzenie modelu boolowskiego.

Model pozycyjny

- Dokumenty są reprezentowane przez (uporządkowane) zbiory termów.
- Model określa pozycje wystąpień termu t_i w dokumencie d_i:

$$w_{ij} = [k : term(d_j, k) = t_i, k = 1, \ldots, n],$$

gdzie $term(d_j, k)$ zwraca term występujący na pozycji *k* w dokumencie d_j, a *n* jest liczbą termów występujących w dokumencie d_j.

Model pozycyjny – przykład

DID	lab	laboratory	programming	computer	program
d_1	0	0	0	0	[71]
d_2	0	0	0	0	[7]
d_3	0	[65,69]	0	[68]	0
d_4	0	0	0	[26]	[30,43]
d_5	0	0	0	0	0
d_6	0	0	[40,42]	[1,3,7,13,26,34]	[11,18,61]
d_7	0	0	0	0	[9,42]
d_8	0	0	0	0	[57]
d_9	0	0	0	0	0
d_{10}	0	0	0	0	0

Źródło: Markov Z., Larose D.T.: Eksploracja zasobów internetowych. Analiza struktury, zawartości i użytkowania sieci WWW, Wydawnictwo Naukowe PWN, Warszawa, 2009.

Model pozycyjny – cechy

- Umożliwia określenie odległość pomiędzy dowolnymi termami danego dokumentu.
- Stanowi raczej reprezentację pomocniczą niż docelową bazę.
- Umożliwia stworzenie modelu ilościowego lub boolowskiego.

Model wektorowy

- Dokumenty są reprezentowane przez wektory wag termów.
- Model przyjmuje rożne postaci w zależności od sposobu wyliczania wag:
  - Model częstotliwości termów (ang. term frequency (TF));
  - Model odwróconej częstotliwości w dokumentach (ang. inverse document frequency (IDF));
  - Model częstotliwości termów-odwróconej częstotliwości w dokumentach (ang. term frequency-inverse document frequency (TF-IDF)).

Model wektorowy c.d

- Pojedynczy wiersz opisujący dokument d_j w modelu wektorowym reprezentowany jest przez wektor

$$\vec{d}_j = (w_{1j}, \ldots, w_{nj}),$$

gdzie *n* jest liczbą termów w dokumencie d_j.

Obliczanie współczynnika TF

- Niech n_{ij} jest liczbą wystąpień termu t_i w dokumencie d_j.
- TF (suma termów): $$w_{ij}=\begin{cases} 0 \text{ gdy } n_{ij}=0, \\ \dfrac{n_{ij}}{\sum_{k=1}^{m} n_{kj}} \text{ gdy } n_{ij}>0. \end{cases}$$
- TF (maksimum): $$w_{ij}=\begin{cases} 0 \text{ gdy } n_{ij}=0, \\ \dfrac{n_{ij}}{max\{ n_{kj} : k=1,\ldots,m \}} \text{ gdy } n_{ij}>0. \end{cases}$$
- TF (logarytm) $$w_{ij}=\begin{cases} 0 \text{ gdy } n_{ij}=0, \\ 1+\log\left(1+\log n_{ij}\right) \text{ gdy } n_{ij}>0. \end{cases}$$

Obliczanie współczynników IDF i TF-IDF

- Niech D będzie zbiorem wszystkich dokumentów, zaś $D_{t_i} = \{ d \in D : n_{ij} > 0 \}$ zbiorem dokumentów zawierających term t_i.

- IDF (ułamek): $w_{ij} = \frac{|D|}{|D_{t_i}|}$

- IDF (logarytm): $w_{ij} = \log \frac{|D|}{|D_{t_i}|}$ lub $w_{ij} = \log \frac{1+|D|}{|D_{t_i}|}$

- TF-IDF: $w_{ij}^{TF} \times w_{ij}^{IDF}$

Model TF – przykład

Document ID	TF Coordinates				
$\vec{d}_1$	0	0	0	0	0.012
$\vec{d}_2$	0	0	0	0	0.010
$\vec{d}_3$	0	0.022	0	0.011	0
$\vec{d}_4$	0	0	0	0.017	0.034
$\vec{d}_5$	0	0	0	0	0.011
$\vec{d}_6$	0	0	0.026	0.078	0.039
$\vec{d}_7$	0	0	0	0	0.033
$\vec{d}_8$	0	0	0	0	0.013
$\vec{d}_9$	0	0	0	0	0
$\vec{d}_{10}$	0	0	0	0	0

Źródło: Markov Z., Larose D.T.: Eksploracja zasobów internetowych. Analiza struktury, zawartości i użytkowania sieci WWW, Wydawnictwo Naukowe PWN, Warszawa, 2009.

Model wektorowy – cechy

- Współczynniki TF preferuje termy, które często występują w danym dokumencie.
- Współczynnik IDF preferuje termy, które rzadko występują w badanym zbiorze dokumentów.
- Współczynnik TF-IDF preferuje termy, które są częste w pojedynczych dokumentach, ale rzadkie w całym zbiorze dokumentów.