STUDIA INFORMATICA 2009
Volume 30 Number 1 (82)

Jan WICIJOWSKI, Bartosz ZIÓŁKO
Akademia Górniczo-Hutnicza, Katedra Elektroniki

ANALIZA SKUPIEŃ I REDUKCJA WYMIAROWOŚCI W HIERARCHICZNYM MODELU KORPUSOWYM JĘZYKA

Streszczenie. Przedstawiono model semantyczny języka polskiego pochodzący z obróbki materiału językowego z polskiej Wikipedii. Model służy weryfikacji hipotez zdaniowych w systemie automatycznego rozpoznawania mowy. Przedstawiono metody filtracji i klasteryzacji dokumentów w celu przyśpieszenia obliczeń. Autorzy kładą nacisk na oddelegowaniu zadań do silnika bazy danych tam, gdzie jest to pożądane ze względu na szybkość.

Słowa kluczowe: analiza skupień, model przestrzeni wektorowej, macierz dokument-temat, macierz rzadka, sqlite3, Wikipedia, mediawiki

CLUSTER ANALYSIS AND DIMENSIONALITY REDUCTION IN A HIERARCHICAL CORPUS MODEL

Summary. The article presents a semantic model of the polish language based on the polish Wikipedia texts. The model is a part of an automatic speech recognition system and verifies sentences hypotheses. Methods of filtering and clustering of the documents, which aim to accelerate the computations, are presented. The authors emphasize the delegation of the processing tasks to the database engine, where it is possible to gain the performance.

Keywords: cluster analysis, vector space model, document-term matrix, sparse matrix, sqlite3, Wikipedia, mediawiki

1. Wstęp

Automatyczna klasyfikacja tekstu pisanego znajduje szerokie zastosowanie w systemach informacyjnych, między innymi w filtrach e-mail, eksploracji danych (ang. *data mining*) oraz korekcie tekstu. W tej pracy opisany jest system bazodanowy, służący pomiarowi stopnia

prawdopodobieństwa hipotez wypowiedzi w systemie automatycznego rozpoznawania mowy. Wypowiedzi są porównywane do tekstów zgromadzonych w hierarchicznie uporządkowanym korpusie.

Model języka polskiego, jaki został przyjęty w analizach, jest modelem przestrzeni wektorowej (ang. *vector space model*). Każdy dokument pochodzący z korpusu jest traktowany jak wektor należący do przestrzeni wektorowej, co pozwala na zastosowanie metod algebry liniowej na korpusie tekstu jako całości.

Autorzy posługują się bazami tekstu o rozmiarach rzędu kilku gigabajtów, dlatego przedstawione zostały metody o liniowej złożoności czasowej względem rozmiaru korpusu. Do przechowywania i przeszukiwania danych użyto relacyjnej plikowej bazy danych typu `sqlite3`.

2. Model przestrzeni wektorowej

Badanie właściwości języka polskiego wymaga użycia metod, które warunkowane są przez szczególne jego cechy. Nie jest on językiem pozycyjnym, więc można przyjąć, że szyk wyrazów w zdaniu nie ma wpływu na niesioną informację[1]. Przedmiotem badania jest pojedyncze zdanie wraz z nieuporządkowanymi wyrazami, które się na to zdanie składają. Każde zdanie jest przedstawione w postaci wektora liczb. Ilości wystąpień poszczególnych wyrazów w zdaniu stanowią wartości liczbowe elementów wektora, których indeksy odpowiadają numerom słów z korpusu. W taki sposób poszczególne słowa stają się wymiarami przestrzeni wektorowej. Wektory zestawione pionowo w macierz tworzą tzw. macierz dokument-temat, w której poszczególnym wierszom odpowiadają kolejne dokumenty, zaś kolumnom – tematy. W pierwszym etapie przetwarzania korpusu pojęcie dokumentu odpowiada pojedynczemu zdaniu. Ze względu na olbrzymi rozmiar korpusu, dokumenty są łączone w większe jednostki pod względem kontekstu w jakich się znajdują – kolejno w akapity, rozdziały, wreszcie artykuły.

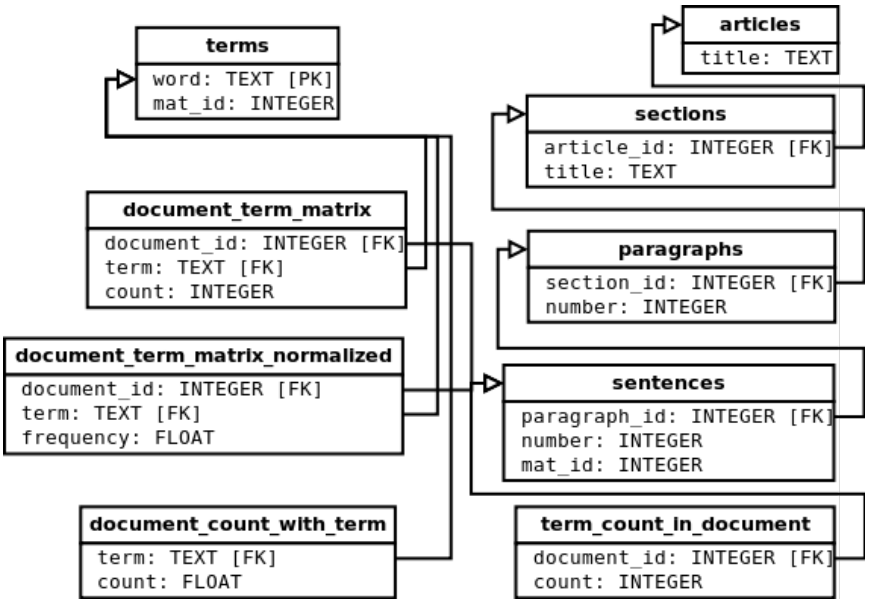
Ze względu na dużą zawartość informacji i bogactwo słownictwa, w analizie za korpus przyjęto polską Wikipedię[1]. Oryginalnym zbiorem danych jest obraz Wikipedii w postaci zbioru XML zawierającego treść wszystkich artykułów zapisanych w formacie mediawiki[1]. Artykuły zostały poddane rozbiorowi składniowemu pod względem znaczników mediawiki przy użyciu parsera mwlib[2], aby otrzymać hierarchiczną strukturę: artykuł-rozdział-akapit-zdanie. Skonstruowano macierz dokument-temat, gdzie tematami są odrębne słowa, zaś

[1] http://www.mediawiki.org/wiki/Markup_spec
[2] http://code.pediapress.com/wiki/wiki/mwlib

dokumentami – zdania. Na przecięciu wiersza i kolumny umieszczona została ilość wystąpień danego słowa w dokumencie.

Macierz tą oznaczono jako *tc* (ang. *term count*), zaś dla odróżnienia zbiór dokumentów w postaci wektorów będzie oznaczany jako *D*.

W projekcie tym skorzystano z projektu bazy danych opisanego przez diagram zależności z Rys. 1. Tabele zostały opisane w tekście. Użyto notacji `$(document)` w miejscach w których występuje odniesienie do jednej z tabel: articles, sections, paragraphs, sentences, jako że w trakcie łączenia dokumentów zmienia się schemat bazy (patrz sekcja 3).



Rys. 1. Struktura bazy danych.
Fig. 1. Database structure.

Trzeba zaznaczyć, że już w pierwszym etapie dokonano redukcji wymiarów macierzy poprzez zamianę wszystkich napotkanych słów na ich odpowiedniki składające się tylko z małych liter oraz poprzez likwidację rozróżnienia między formami gramatycznymi wyrazów (częściowa *lematyzacja*). Wyrazy sprowadzone zostały do następujących form gramatycznych: rzeczownik do mianownika liczby pojedynczej, przymiotnik do mianownika liczby pojedynczej rodzaju męskiego i czasownik do bezokolicznika. W tym celu użyto analizatora składniowego Morfeusz[3]. W przypadku wyrazów mogących mieć różne rdzenie[4] lub spoza słownika analizatora zachowano ich oryginalne formy. Język polski jest językiem fleksyjnym, tj. odmiana wyrazu wpływa na dobór formy gramatycznej powiązanych wyrazów w zdaniu – przyjęto zatem, że utrata części informacji w procesie lematyzacji jest dobrym kompromisem prowadzącym do zwiększenia wydajności systemu.

[3] http://nlp.ipipan.waw.pl/~wolinski/morfeusz
[4] Przykładowo wyraz „mam” może pochodzić od „mieć”, „mamić” lub „mama”.

Powyższe metody łączą grupy słów, czyli redukują ilość odrębnych wymiarów przestrzeni wektorowej. Dzięki ich zastosowaniu liczba wymiarów zmniejszyła się o ok. 33%, w tym 7% jest zasługą zmniejszenia liter.

Niezależnie od łączenia słów, na etapie rozbioru składniowego odrzucono wszystkie artykuły z Wikipedii, które zawierają w tytule przynajmniej jeden znak spoza liter języka polskiego. Dzięki temu unika się dodawania zdań zawierających duże ilości wyrazów pochodzących z języków obcych.

Przykładowe artykuły odrzucone wg tego kryterium to „Martín Fernández de Quintana", „L'Oréal", „Český Krumlov", „Mehmed Köprülü", „Süßen", „Twierdzenie Gödla".

Odrzucone zostały także strony nieprzydatne pod względem badań języka:

- przekierowania i strony ujednoznaczniające,
- duplikaty,
- strony specjalne mediawiki, Wikipedii i Wikiprojektu,
- strony-kategorie artykułów,
- szablony,
- strony pomocy,
- grafiki,
- puste strony.

3. Łączenie dokumentów

Baza danych została tak zaprojektowana, by wektory dokumentów mogły odpowiadać różnym zakresom tekstu. Wyróżniono cztery logiczne jednostki organizacyjne korpusu, wymienione powyżej: artykuł, rozdział, akapit i zdanie. Podczas łączenia zmienia się schemat bazy danych – tworzona jest tabela `document_term_matrix_tmp`, budowana na podstawie tabeli `document_term_matrix`. Kluczem obcym nowej tabeli jest kolumna z nadrzędnej jednostki organizacyjnej. Pole `count` budowane jest z sumy pól przypadających na podrzędne dokumenty. Po dokonanym łączeniu oryginalna tabela jest porzucana, zaś nowa jest przemianowana na jej miejsce. Oto SQL do łączenia zdań w akapity:

```
CREATE TABLE document_term_matrix_tmp
    (document_id INTEGER, term TEXT, count INTEGER,
    PRIMARY KEY(document_id, term),
    FOREIGN KEY(document_id) REFERENCES sentences.id
        ON UPDATE CASCADE ON DELETE CASCADE,
    FOREIGN KEY(term) REFERENCES terms(word)
        ON UPDATE CASCADE ON DELETE CASCADE);

INSERT INTO document_term_matrix_tmp
    SELECT sentences.paragraph_id, document_term_matrix.term,
        sum(document_term_matrix.count)
    FROM document_term_matrix, sentences
```

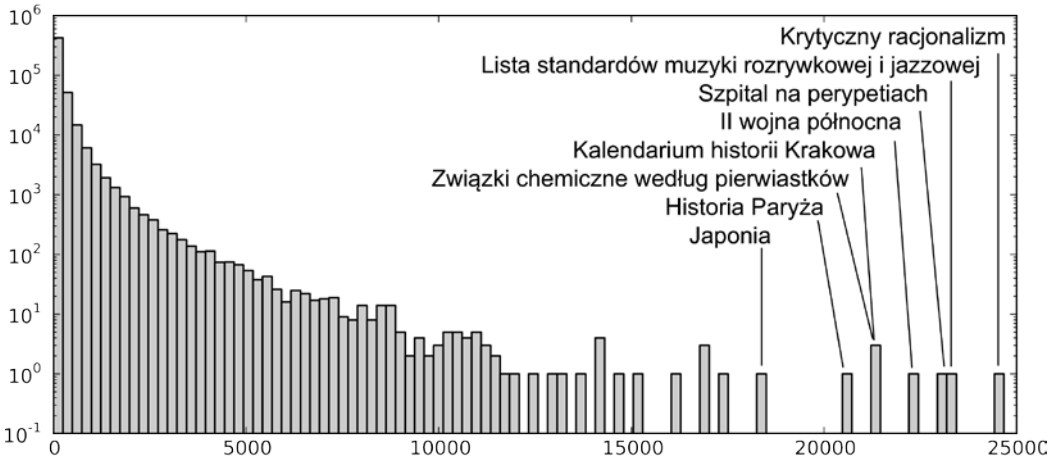
```
    WHERE document_term_matrix.document_id = sentences.id
    GROUP BY sentences.paragraph_id, document_term_matrix.term;

DROP TABLE document_term_matrix;
ALTER TABLE document_term_matrix_tmp RENAME TO document_term_matrix;
```

Powyższy listing został przedstawiony tylko w celach ilustracyjnych. W rzeczywistości wykorzystano uprzednie pogrupowanie rekordów w tabeli `document_term_matrix` względem kolumny `document_id`. Program przechodzi w pętli po wierszach tabeli zapytaniem `SELECT` podobnym do wypisanego, różniącym się brakiem klauzul `ORDER BY`. Krótki program łączy rekordy, a następnie wpisywane są one do nowej tabeli za pomocą polecenia `REPLACE INTO`. Wszystkie te wpisy są zawarte w jednej transakcji, dlatego uniknięto wpisywania do bazy pojedynczych składników dodawania. Użycie opisanych czynności jest o jeden rząd wielkości szybsze od bezpośredniego wywołania zapytania SQL.

4. Odszumianie macierzy

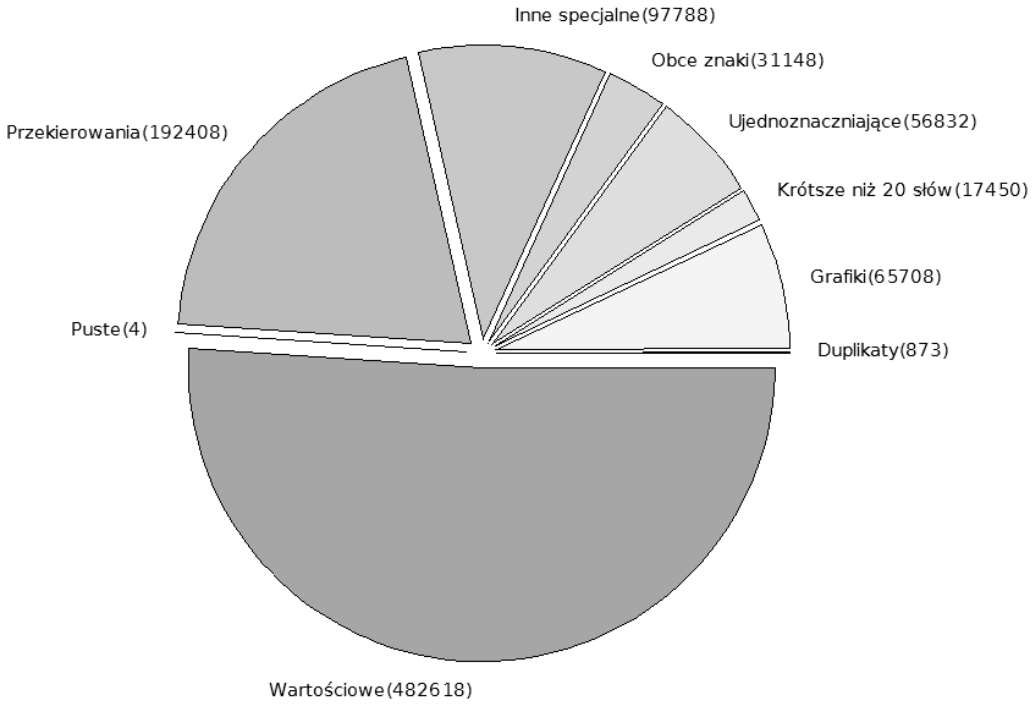
Po tych operacjach poddano korpus dalszemu zmniejszaniu wymiarów, usuwając z bazy wyrazy pojawiające się tylko w jednym artykule. Ilość artykułów o określonej długości została zilustrowana na histogramie na Rys. 1. Zgodnie z empirycznym prawem Zipfa, liczba takich słów wynosząca 54% wszystkich napotkanych, nie może dziwić[5]. Po usunięciu słów występujących w pojedynczych artykułach pozostało ok. 600.000 słów.



Rys. 2. Rozkład długości artykułów w polskiej Wikipedii liczony w ilości niezerowych elementów wektora dokumentu.

Fig. 2. Histogram of polish Wikipedia article length as measured in nonzero document vector dimensions.

[5] Do słów pojawiających się tylko w jednym artykule należą oryginalne nazwy własne, wyrazy obcego pochodzenia, wyrazy błędnie zapisane, ale też szum pochodzący od kolektywnej natury edycji tego zbioru wiedzy. Poniższe przykłady pochodzą z pierwszego tysiąca takich słów: „aaaaau”, „aaameryka”, „aacatgcaggacctggtggaagacttcaagaacaa”, „aajiya”, „aanleiding”, „aaronitami”, „abarskim”, „abbeyshrule”, „abbracciamoci”, „abcdeffedcba”, „abdominoplastykę”, „abdurachmanow”



Rys. 3. Kategorie artykułów w polskiej Wikipedii.
Fig. 3. Article categories in the polish Wikipedia.

Wśród artykułów pojawiają się takie, które nie wnoszą informacji o konstrukcji zdań w języku, np. artykuły zbiorcze, katalogi, tabele. Przy programowaniu parsera wykorzystano spostrzeżenie, że w większości przypadków tabele zawierają dane liczbowe, równoważniki zdań, pojedyncze wyrazy, dane fizyczne, geograficzne itp., więc podczas rozbioru składniowego tabele, które są ograniczone znacznikami `{|` i `|}`, są w całości pomijane. W wyniku tego wycinania powstaje pewna liczba artykułów o niewielkiej liczbie wyrazów. Przykładowo, artykuł „Księżne Walii” jest prawie w całości taką tabelą – po jej usunięciu pozostają dwa tytułowe wyrazy. Przyjęto próg tolerancji 20 wyrazów, poniżej którego artykuły są odrzucane – w sumie 5,5% wszystkich artykułów. Po odszumieniu pozostało około połowy wszystkich artykułów.

Rys. 2. przedstawia na wykresie kołowym liczbę artykułów plasujących się w wymienionych w artykule kategoriach:

a) Puste – artykuły pozbawione treści,
b) Przekierowania typu `redirect`, `patrz`, `tam`, `przekieruj` oraz `słownikowe`,
c) Inne specjalne – strony z kategorii `kategoria:`, `szablon:`, `mediawiki:`, `portal:`, `wikiprojekt:`, `wikipedia:`, `pomoc:`,
d) Obce znaki – artykuły zawierające w nazwie znaki spoza liter języka polskiego, cyfr i znaków przestankowych,
e) Ujednoznaczniające – strony typu `disambig` i `Szablon:Disambig`,
f) Krótsze niż 20 słów,

g) Grafiki rozpoznane po prefiksach `plik:`, `grafika:` lub sufiksach `.jpg`, `.jpeg`, `.gif` lub `.png`,
h) Duplikaty – artykuły o tej samej nazwie, w większości przypadkowo lub błędnie nazwane,
i) Wartościowe dla analizy językowej.

Odszumianie jest realizowane w całości poprzez odpowiednio skonstruowane zapytania SQL. Utworzono w bazie tabele `document_count_with_term` oraz `term_count_in_document`. Pierwsza z nich przypisuje każdemu wyrazowi ilość dokumentów, w których on występuje, np. wyraz „król" występuje w 18141 artykułach Wikipedii. Druga tabela zawiera ilość wyrazów w poszczególnych dokumentach, np. artykuł „Monarcha" składa się ze 108 wyrazów. Konstruowana jest ona zatem na podstawie sumy pól `count` z tabeli `document_term_matrix` dla pasujących rekordów. Obie te tabele posłużą przy normalizacji macierzy (patrz sekcja 5).

5. Normalizacja

Przechowywanie w wektorze dokumentu ilości wystąpień słów jest przydatne tylko w fazie gromadzenia informacji. Znormalizowano te wartości do wielkości, które można ze sobą porównywać, tak by uniknąć następujących problemów:

- długie artykuły zawierają dużą ilość słów, tak kluczowych, jak i popularnych – metody algebraiczne, takie jak iloczyn skalarny, dadzą zawyżone wyniki;
- często pojawiające się słowa, takie jak spójniki, nie niosą tak dużej ilości informacji, jak słowa kluczowe danego artykułu[2].

Popularną metodą normalizacji jest TF-IDF (ang. *term frequency - inverse document frequency*), której przydatność została dowiedziona w wielu opracowaniach[3]. Powstaje nowa macierz *X*, w której wartości komórek równe są iloczynom dwóch niezależnych czynników – częstości tematu *tf* i odwrotnej częstości dokumentu *idf*.

Częstość tematu jest ilorazem wystąpień *i*-tego tematu do sumy wystąpień wszystkich tematów w *j*-tym dokumencie – innymi słowy normalizacja TF skaluje *j*-ty wektor tak, by suma jego składowych dawała 1

$$tf_{i,j} = \frac{tc_{i,j}}{\sum_k tc_{k,j}}. \tag{1}$$

Odwrotna częstość dokumentu jest miarą popularności danego tematu w korpusie. Im częściej występuje dany wyraz, tym mniejsza jego waga *idf*. Jest ona wyrażona za pomocą następującego wzoru

$$idf_i = \log \frac{|D|}{1+\left|\{d \in D : d_i \neq 0\}\right|}, \tag{2}$$

gdzie licznik w ułamku jest ilością dokumentów w korpusie, w mianowniku zaś znajduje się liczba dokumentów zawierających *i*-te słowo.

Znormalizowana macierz dokument-temat powstaje z iloczynu

$$X_{i,j} = tf_{i,j} idf_i . \tag{3}$$

Normalizacja TF-IDF dla korpusu o wspomnianych rozmiarach jest podobnie czasochłonna, jak inne operacje na całości macierzy, ale wymagana przy obliczeniach wektorowych. Zatem, by przy każdym ładowaniu macierzy do pamięci operacyjnej nie powtarzać normalizacji, macierz *X* jest gromadzona w bazie danych. Współczynniki $\sum_k tc_{k,j}$ oraz $1+\left|\{d \in D : d_i \neq 0\}\right|$ zostały policzone w poprzednim kroku odszumiania. Są to kolejno tabele `term_count_in_document` i `document_count_with_term`. Są one wykorzystane do obliczania znormalizowanej macierzy.

```
CREATE TABLE term_document_matrix_normalized
    (document_id INTEGER, term TEXT, frequency REAL,
    FOREIGN KEY(document_id) REFERENCES $(documents)(id),
 FOREIGN KEY(term) REFERENCES terms(word));
```

6. Eksport do formatu numerycznego

Macierz znormalizowana według opisanego wyżej sposobu została wyeksportowana do formatu macierzy rzadkiej, którą można wczytać w dowolnym środowisku inżynierskim. Uprzednio jednak utworzono w tabelach `terms` i `$(documents)` dodatkową kolumnę liczb całkowitych, która będzie służyła jako pomost pomiędzy bazą danych, a macierzą w środowisku inżynierskim. Kolumny te będą zawierały kolejne liczby naturalne odpowiadające *j*-tym wierszom i *i*-tym kolumnom. Unika się dzięki temu posługiwania się domyślną kolumną `rowid` silnika `sqlite3` (tak jak przy niektórych kluczach obcych), gdyż podczas redukcji wymiarów usunięte zostały rekordy, ale odpowiadające im wartości `rowid` nie zostały przesunięte[6], tak by stworzyć sekwencyjny indeks. Obie kolumny zawierają już klucz główny, zatem dla szybkości identyfikacji danego słowa lub dokumentu na podstawie informacji o numerze wiersza lub kolumny, zostają utworzone indeksy na nowo utworzonych kolumnach.

```
ALTER TABLE terms ADD COLUMN mat_id INTEGER;
CREATE INDEX terms_mat_id ON terms (mat_id);

ALTER TABLE $(documents) ADD COLUMN mat_id INTEGER;
CREATE INDEX $(documents)_mat_id ON $(documents)(mat_id);
```

[6] Byłoby to niepożądane ze względu na szybkość działania.

Macierz dokument-temat jest rzadka, co jest typowe dla modelu wektorowego. Po łączeniu dokumentów i odszumianiu macierzy na jeden niezerowy element macierzy przypada 6000 zerowych. Gdyby próbować zapisać taką macierz w postaci pełnej, używając typu `double` standardu IEEE 754-1985, wymagałaby ona 2,1TiB pamięci operacyjnej lub dyskowej. Popularne formaty zapisu i przechowywania macierzy rzadkich wymagają $O(nnz)$ pamięci, gdzie nnz jest liczbą niezerowych elementów macierzy. Przy obliczeniach posłużono się formatem CSC (ang. *Compressed Sparse Column*), który wymaga użycia $(s_{index} + s_{elem})nnz + s_{index}r$ bajtów pamięci, gdzie s_{index} jest rozmiarem zmiennej indeksu (np. `uint32`), r zaś jest liczbą wierszy. Przy konstruowaniu macierzy na podstawie bazy danych posłużono się formatem COO[4] (ang. *COOrdinate format*), wymagającym przechowania $(2s_{index} + s_{elem})nnz$ bajtów pamięci. Wielkości zajętej pamięci w obu przypadkach to 555MiB i 738MiB.

Macierz COO wymaga podania trzech wektorów: niezerowych elementów, indeksów ich wierszy i kolumn. Wektory te są zwracane poprzez jedno zapytanie SQL:

```
SELECT $(documents).mat_id , terms.mat_id,
    document_term_matrix_normalized.frequency
    FROM document_term_matrix_normalized, terms, $(documents)
    WHERE document_term_matrix_normalized.term=terms.word
    AND document_term_matrix_normalized.document_id=$(documents).id;
```

Konwersja do formatu CSC z COO jest błyskawiczna - wymaga jedynie sortowania indeksów. Jej asymptotyczna złożoność czasowa jest liniowa względem ilości elementów. Do eksportu tak utworzonej macierzy do pliku w formacie Matlaba wykorzystano funkcjonalność pakietu SciPy.

7. Analiza skupień

Opisywanymi w literaturze[5] i popularnymi narzędziami służącymi zmniejszaniu wymiarowości macierzy są metody algebry liniowej:

- analiza głównych składowych – PCA (ang. *principal component analysis*) oparta na rozkładzie macierzy kowariancji dokumentów na wartości własne.
- analiza ukrytych grup semantycznych – LSA (ang. *latent semantic analysis*) oparta na rozkładzie macierzy według wartości osobliwych – SVD (ang. *singular value decomposition*)

Obie metody polegają na znalezieniu charakterystycznych grup wyrazów jako kierunków w przestrzeni i-wymiarowej. Są one posortowane według wartości własnych lub osobliwych, co odpowiada ich istotności w korpusie. Dla obu metod dowiedziono, że utworzenie zredukowanych macierzy składających się jedynie ze składników odpowiadających

określonej liczbie największych wartości własnych lub osobliwych, aproksymuje oryginalną macierz najlepiej, jak to możliwe[5].

Metody te mają wielorakie zastosowania w statystyce lingwistycznej, np. przy wyszukiwaniu pokrewnych znaczeniowo stron www[6], kategoryzowaniu dokumentów, tworzeniu i zastosowaniu ontologii, automatycznych przekładach. W opisywanym systemie służą one zmniejszeniu roboczego zestawu danych, tak by ocena wiarygodności hipotezy zdania mogła być obliczana w czasie rzeczywistym, tj. aby mówca nie musiał przerywać toku wypowiedzi. Wykorzystano te metody, by utworzyć mniejsze korpusy, zawierające podzbiór tematów występujących w języku, np. terminologia sądownicza, medyczna.

Komplementarną metodą, której użyto do łączenia dokumentów i tematów w pokrewne grupy, by dodatkowo zmniejszyć złożoność poszukiwań, jest kwantyzacja wektorowa – VQ (ang. *vector quantization*). Typowo używana w algorytmach kompresji stratnej, w opisywanym systemie pełni funkcję algorytmu grupowania działającego na bieżąco (ang. *online clustering*). Zaimplementowano wariant tej metody, zaproponowany przez Kohonena[7]. Dokumenty łączone są w grupy, zaś do reprezentacji w nowej macierzy dokument-temat wyznaczane są centroidy grup. Podobnie postąpiono z wyrazami, łącząc je w grupy o podobnym znaczeniu.

8. Weryfikacja hipotezy

Niższe warstwy systemu rozpoznania tekstu mówionego – fonetyczna i leksykalna – dostarczają hipotetycznych sekwencji słów. Celem działania opisywanej warstwy semantycznej jest nadanie poszczególnym hipotezom oceny prawdopodobieństwa na podstawie opisanego modelu semantycznego. Dzięki temu możliwe jest zwrócenie przez cały system najtrafniejszych rozpoznań, a także modyfikacja algorytmów niższych warstw w trakcie działania systemu poprzez sprzężenie zwrotne (np. system kar i nagród).

Przykładowo, niech rozpoznawaną wypowiedzią będzie „Prezydent aprobuje wejście do strefy Schengen”. Niższe warstwy mogą zwrócić przykładowe hipotezy dla ciągu słów, wraz z własną oceną rozpoznania:

prezydent: 0,4	aprobuje: 0,33	wejście: 0,2	do: 0,4	strefy: 0,2	schengen: 0,5
rezydent: 0,4	aportuje: 0,33	dwieście: 0,4	to: 0,5	trafi: 0,3	szogun: 0,05
prezydium: 0,2	operuje: 0,33	nieście: 0,4	dom: 0,1	stepy: 0,5	szelkę: 0,45

Rys. 4. Hipotezy słów otrzymane z warstw fonetycznej i gramatycznej wraz z miarą rozpoznania.
Fig. 4. Word hypotheses and their likelihood acquired from acoustic and grammatical layers.

Dla tak prostego przypadku należy rozpatrzeć $R = 3^6 = 729$ hipotez zdań, bo tyle ciągów można stworzyć biorąc kolejne hipotezy słów. Zbiór hipotez będzie oznaczony przez *H*.

Każdy ciąg słów utworzony z pojedynczych hipotez słów zostanie porównany z macierzą dokument-temat. Aby móc tego dokonać, ciąg słów jest traktowany w taki sam sposób jak oryginalny tekst korpusu. Hipoteza zostaje poddana

- lematyzacji słów tą samą metodą, co przy budowie korpusu,
- przekształceniu ciągu słów na wektor zliczeń,
- normalizacji wektora metodą TF, tak by suma składowych wynosiła 1,
- przemnożeniu składowych przez współczynniki IDF, określone w korpusie,
- przypisaniu wyrazów do grup.

Tak otrzymany wektor h_r porównano z dokumentami macierzy. Metodą porównania jest obliczenie miary kosinusowej[8] między wektorem-hipotezą a dokumentami korpusu

$$s(d_i, h_r) = \frac{d_i \cdot h_r}{\|d_i\| \cdot \|h_r\|}, \tag{4}$$

gdzie wykorzystany jest iloczyn skalarny oraz długość wektorów. Miara $s(d,h)$ zawiera się w przedziale [0,1], przy czym wartość 0 oznacza brak wspólnych tematów, zaś 1 oznacza identyczność.

Wynikiem mnożenia $d_i \cdot h_r$ jest skalar. Wszystkie te wyniki otrzymane są w postaci wektorowej z mnożenia $s^{(r)} = X' h_r$. Liczenie iloczynu macierz-wektor implementowane jest w postaci szybkich, zoptymalizowanych algorytmów, zatem korzystny jest zapis kodu w postaci macierzowej[9,10].

Miarą dopasowania hipotezy h_r do korpusu jest maksymalna składowa $s^{(r)}$

$$p_r = \max_m s_m^{(r)}. \tag{5}$$

Miara dopasowania do korpusu jest liczona dla każdej z hipotez zdania h_r

$$P = (p_1, p_2, \ldots p_r, \ldots p_R). \tag{6}$$

Ostatecznym decydentem jest użytkownik systemu, który otrzymuje pewną liczbę prawdopodobnych hipotez zdaniowych ze zbioru *H*, dla których odpowiadające miary p_r są największe.

System znajduje się obecnie w fazie rozwojowej, więc przedstawianie działania modelu semantycznego na sztucznych problemach, tak jak wyżej wymieniony, ma jedynie wartość ilustracyjną. Wyniki weryfikacji sztucznych hipotez są poprawne i obiecujące, jednak nie mówią nic o właściwościach systemu w docelowym użyciu.

Praca naukowa finansowana ze środków na naukę w latach 2008-2011 jako projekt rozwojowy.

BIBLIOGRAFIA

1. Ziółko B., Manandhar, S., Wilson, R.C.: Bag-of-words modelling for speech recognition. 2009 International Conference on Future Computer and Communication. ICFCC 2009, strony 646–650, Kwiecień 2009.
2. Salton, G.: Automatic text processing: the transformation, analysis, and retrieval of information by computer. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989.
3. Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. Information Processing and Management, strony 513–523, 1988.
4. Jones, E., Oliphant, T., Peterson, P. et.al.: SciPy: Open source scientific tools for Python. SciPy Documentation: Sparse matrices. http://www.scipy.org/
5. Martinez, W.L., Martinez, A.R.: Exploratory Data Analysis with MATLAB (Computer Science and Data Analysis). Chapman & Hall/CRC, 2004.
6. Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., Harshman, R.: Indexing by latent semantic analysis. Journal of the American Society For Information Science, 41, 1990.
7. Kohonen, T.: Self-Organizing Maps. Springer-Verlag, Berlin 1995/1997.
8. Ntoulas, A., Cho, J., Olston, C.: What's new on the web? : the evolution of the web from a search engine perspective. WWW '04: Proceedings of the 13th international conference on World Wide Web, strony 1–12, New York, NY, USA, 2004. ACM.
9. The Mathworks. Matlab Code Vectorization Guide. http://www.mathworks.com/support/tech-notes/1100/1109.html
10. Jones, E., Oliphant, T., Peterson, P. et.al.: SciPy: Open source scientific tools for Python. SciPy Documentation: A beginners guide to using Python for performance computing. http://www.scipy.org/PerformancePython

Recenzent: tytuły Imię Nazwisko

Wpłynęło do Redakcji ...

Abstract

The work presents a semantic model of the polish language acquired by processing the polish Wikipedia by the vector space model. It is employed in an automatic speech recognition system as the final verification stage, which validates the sequences hypotheses to match their

semantic content to the language model. Tools such as sqlite3, Python, SciPy and mwlib are employed to collect, filter and cluster the data, eventually leading to the versatile sentence validator. Stages which lead to reduction of the problem dimensionality are emphasized, so are the database manipulation queries.

The Wikipedia corpus is first acquired in a form of an xml dump, then parsed with mwlib tool. A number of predicates are employed which filter out the articles irrelevant to the language model, such as: articles with non-polish characters, special Wikipedia pages (help, categories), articles containing enumerations, tabular data etc., redirections and pictures. The proportion of articles containing correct language is presented in Fig. 2.

The candidate articles are split hierarchically into sections, paragraphs and sentences. Then the document-term matrix is constructed, with documents being one of articles, sections, paragraphs or sentences, and terms being the lemmatized words, as presented in Sec. 2.

The resultant document-term matrix can be then joined by documents thanks to the stored hierarchy (Sec. 3.), then denoised from obscure words and short documents (Sec. 4.). The TF-IDF normalization (equations (1,2,3)) is applied to the matrix (Sec. 5.), then exported to the numerical format (Sec. 6.) and presented to clustering algorithms (Sec. 7.). Finally the process of hypotheses verification is presented in the final section, according to the equations (4,5,6).

Adresy

Jan WICIJOWSKI: Akademia Górniczo-Hutnicza im. Stanisława Staszica,
Katedra Elektroniki, Al. Mickiewicza 30, 30-059 Kraków, Polska,
jan.wicijowski@agh.edu.pl

Bartosz ZIÓŁKO: Akademia Górniczo-Hutnicza im. Stanisława Staszica,
Katedra Elektroniki, Al. Mickiewicza 30, 30-059 Kraków, Polska,
bartosz.ziolko@agh.edu.pl