Metody i narzędzia automatycznego przetwarzania informacji tekstowej i ich wykorzystanie w procesie zarządzania wiedzą

Article · January 2011

CITATION
1

READS
79

1 author:

Piotr Potiopa
AGH University of Science and Technology in Kraków
9 PUBLICATIONS **3** CITATIONS

SEE PROFILE

AUTOMATYKA • 2011 • Tom 15 • Zeszyt 2

Piotr Potiopa*

Metody i narzędzia automatycznego przetwarzania informacji tekstowej i ich wykorzystanie w procesie zarządzania wiedzą

1. Wprowadzenie

Przeważająca większość informacji wykorzystywanych we współczesnych firmach i instytucjach przechowywana jest nadal w postaci informacji w języku naturalnym. Stanowi on podstawowy nośnik i środek komunikacji w procesach dzielenia się wiedzą. Jednocześnie coraz większe znaczenie mają systemy i narzędzia informatyczne, które umożliwiają łatwe i zautomatyzowane przetwarzanie danych i informacji przechowywanych właśnie za pomocą języka naturalnego. Problematyka analizy tekstów, dokumentów od dłuższego czasu w różnych organizacjach nabierają coraz większego znaczenia. Zasadnie jest zatem, aby zadania realizowane w zakresie zarządzania wiedzą, najbardziej w obszarze jej reprezentacji i ekstrakcji, spełniały systemy przetwarzania języka naturalnego.

Jednym z ważnych aspektów realizacji takich zadań jest budowa odpowiednich ontologii dziedzinowych. Służą one poprawnemu modelowaniu i reprezentacji struktur wiedzy w sposób zarówno czytelny dla człowieka, jak i umożliwiający jej przetwarzanie przez komputer. Obecnie ontologie są obiektem badań w rożnych środowiskach naukowych, m.in. w inżynierii języka naturalnego, w inżynierii systemów informatycznych, w inżynierii wiedzy, a także w teorii zarządzania wiedzą [1].

Drugim ważnym obszarem działań w zakresie zarządzania wiedzą jest wyszukiwanie, analiza i obróbka dokumentów zawierających potrzebne nam informacje. Nierzadko informacja zawarta w tych dokumentach stanowi bazę do rozwiązania aktualnych, nowych problemów wystepujacych w danym systemie wiedzy. Zdobywanie wiedzy i dzielenie się przeszłymi doświadczeniami do ponownego wykorzystania jest coraz bardziej istotne ze względu na ilość technicznych informacji, od których jestesmy uzależnieni obecnie. W systemach zarządzania wiedzą techniki te są określane mianem wnioskowania na podstawie przypadków (*Case-Based Reasoning*) [2].

* AGH Akademia Górniczo-Hutnicza, Wydział Zarządzania, Katedra Informatyki Stosowanej

Analizy tekstów w języku naturalnym są możliwe dzięki, już dziś rozwiniętym, metodom jego przetwarzania. Wykorzystanie podejść i metod takich jak: *information retrieval*, *information extraction*, *text mining* czy *natural language processing* pozwala na budowanie bazy wiedzy i na jej usystematyzowanie, a co za tym idzie na efektywne nią zarządzanie.

W tym artykule zostaną przedstawione metody analizy tekstów wykorzystujące znane algorytmy wspomagające ich przetwarzanie. Nacisk położono na aspekty podobieństwa dokumentów i technik związanych z metodami jego określania. Przedstawiono też przykłady istniejących narzędzi obróbki i analizy dokumentów.

2. Metody wyszukiwania i analizy tekstu

W procesach wyszukiwania i analizy dokumentów tekstowych wyróżnia się m.in. następujące metody:

- Systemy wyszukiwania informacji (*Information Retrieval*, IR).
- Rozumienie języków naturalnych (*Natural Language Processing*).
- Metody ekstrakcji informacji (*Information Extraction*, IE).
- Metody eksploracji tekstu (*Text Mining*).

Poniżej zostały one wyjaśnione i opisane z uwzględnieniem technik jakie umożliwiają działanie danej metody.

2.1. Information Retrieval

Information Retrieval, IR (wyszukiwanie informacji) jest określeniem powszechnie używanym, chociaż niezupełnie trafnie. System wyszukiwania informacji nie tyle informuje użytkownika na temat, który go interesuje, co informuje o istnieniu (lub jego braku) miejsca, gdzie dokument odpowiadający wymaganiom użytkownika się znajduje [3]. W typowym IR użytkownik tworzy zapytanie, złożone z jednego lub kilku wyrazów, na podstawie którego system wyszukuje dokumenty. Wyróżnia się dwa główne podejścia IR: model boolowski (*Boolean Logic Model*, BLM) oraz rankingowy (*ranked-output systems*) [3, 8]. Zapytanie BML składa się ze słów lub fraz połączonych logicznymi operatorami AND, OR oraz NOT. Rezultatem zapytania jest zazwyczaj podział zbioru dokumentów na dwie części: jedną zawierającą dopasowane dokumenty oraz drugą zawierającą dokumenty niedopasowane. System rankingowy, stosując algebrę wektorów ocenia podobieństwo treści dokumentów z treścią zapytania i na tej podstawie dokonuje rankingu znalezionych dokumentow. Systemy rankingowe wykorzystują najczęściej następujące modele do oceny podobieństwa dokumentów: model wektorowy (*Vector Space Model*, VSM), model probabilistyczny (*Probabilistic Model*, PM), a także inne, m.in. Inference Network Model (INM). Zaletą systemów IR jest dziedzinowa niezależność oraz elastyczność językowa (zmiana języka nie wymaga zbyt wielu adaptacji). Natomiast do najważniejszych ograniczeń tych

systemów należy założenie niezależności indeksów termów, co może prowadzić do oszacowania zerowego podobieństwa między dokumentami zawierającymi synonimiczne wyrażenia [3, 8].

2.2. Information Extraction

Zadaniem ekstrakcji informacji (*Information Extraction*, IE) jest zidentyfikowanie instancji pewnej predefiniowanej klasy zdarzeń, ich powiązań oraz wystąpień w dokumentach pisanych w języku naturalnym [4]. W odróżnieniu od systemów IR, systemy IE nie wyszukują samych dokumentów, ale zgodnie z nazwą dokonują ekstrakcji informacji z ich treści. Pozyskane informacje mogą zostać umieszczone w bazie danych. Informacja, jaka będzie pozyskiwana z dokumentu, jest specyfikowana przez użytkownika, który tworzy wzorzec. Zawiera on określone sekcje – „dziury" (*slots*), ktore wypełniane są fragmentami tekstu. Jądro systemu ekstrakcji informacji składa się z dwóch komponentów: procesora tekstów (którym może być jedna z metod NLP) oraz generatora wzorców, które osadzone są w wiedzy dziedzinowej. Zadaniem procesora tekstów jest analiza leksykalna tekstu (obecnie najczęściej stosuje się płytką analizę).

2.3. Text Mining

Text mining jest metodologią wywodzącą się z data mining, wyszukiwania informacji, ekstrakcji danych, kategoryzacji tekstu, modelowania probabilistycznego, algebry liniowej, uczenia maszynowego zastosowanych w celu wykrycia wiedzy z dokumentów tekstowych [4]. Definicja wskazuje na podobieństwo z technikami IE. Ekstrakcji informacji dokonuje się jednak zwykle w oparciu o znane wzorce, w przypadku text mining wzorce wychwytywane są dopiero w procesie przetwarzania dokumentu. Do typowych zadań text mining należy: znajdowanie dokumentów najbardziej pasujących do zapytania użytkownika, tworzenie rankingow dokumentów, grupowanie dokumentów (analiza skupień), klasyfikowanie dokumentów (kategoryzacja), analiza powiązań między jednostkami tekstu, dokonywanie automatycznych streszczeń dokumentów [4].

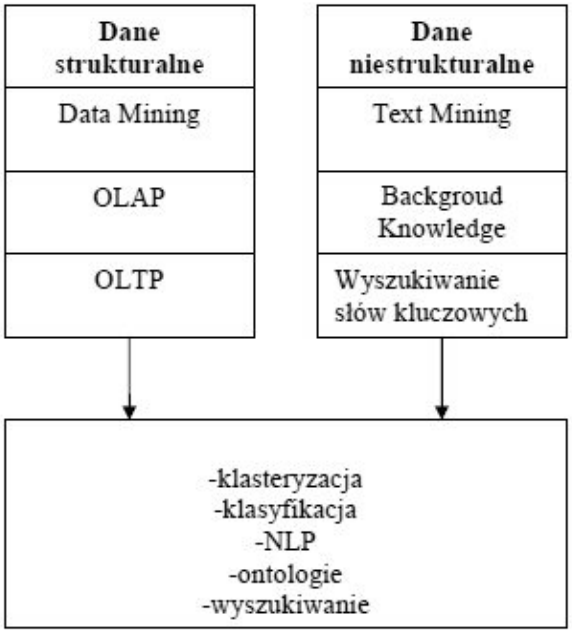
2.4. Natural Language Processing

Metody *Natural Language Processing* – NLP (rozumienie języków naturalnych) zawierają mechanizmy próbujące dokonać „zrozumienia" kontekstu tekstu. W metodach tych nie oblicza się podobieństwa termów, ale oznacza się poszczególne części mowy (analiza płytka) oraz szuka się znaczenia danego wyrażenia w kontekście poprzez pełną analizę gramatyczną (analiza głęboka) [4].

Niektóre serwisy internetowe proponują użytkownikom alternatywnie dla metod IR – wyszukiwanie informacji poprzez systemy wzbogacone o NLP, co może dać w wyniku lepiej dopasowane do danego zapytania dokumenty. Metody NLP mają jednak swoje wady, należą do nich: większa złożoność i czasochłonność (zwłaszcza głęboka analiza), są silnie

związane z danym językiem (adaptacja na inne języki wymaga wiele pracy), tracą znacznie swoją skuteczność w przypadku występowania w tekście terminów spoza słowników oraz w przypadku analizy tekstów sporządzonych jako krótkie notatki (dość często pozbawione poprawnej struktury gramatycznej).

Analiza dokumentów tekstowych jest łączona zwykle z metodami NLP (*Natural Language Processing*). Skupia się ona na pojedynczym dokumencie. Natomiast bazowy charakter ontologii z danego obszaru tematycznego wymaga działań szerszych, analizy dużych wolumenów dokumentów (korpusu), na podstawie których będzie ona tworzona. Do tego celu można zastosować technikę wykorzystywaną przy analizie danych strukturalnych DM (*Data Mining*). Często określa się TM (*Text Mining*) jako DM dla dokumentów niestrukturalnych (rys. 1) [5], w których szuka wzorców i szablonów.

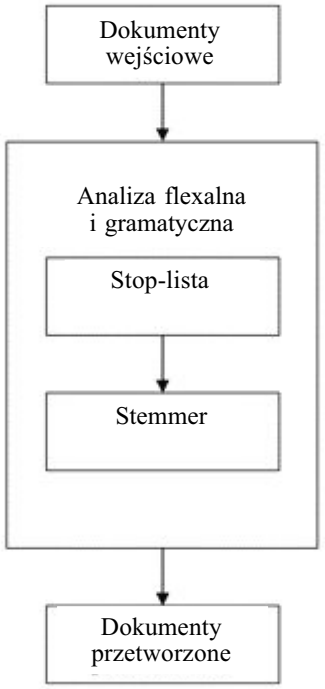


Rys. 1. Techniki analizy danych strukturalnych i niestrukturalnych

Automatyczne przetwarzanie dokumentów języka naturalnego obejmuje następujące zasadnicze fazy:

- podział tekstu wejściowego na zdania, tokeny, słowa;
- odrzucenie słów (tagów) nieistotnych (z tzw. stop-listy);
- tematyzacja tzn. wybór słów istotnych i sprowadzenie ich do postaci podstawowe (stemmer); są stosowane dwie metody: reguły gramatyki w algorytmie lub słowniki;
- automatyczne generowanie słów kluczowych, klasteryzacja dokumentów, ontologie, tezaurusy itp.

Z kolei sam proces analizy tekstu przebiega wg schematu przedstawionego na rysunku 2 [5].



Rys. 2. Etapy analizy dokumentów

Tokeny pozwalają na podział tekstu na proste elementy, czyli np.: liczby, punktacja, słowa. Proces ten jest uzależniony od języka, w jakim dany tekst został zbudowany i do jakiego obszaru tematycznego się odnosi. Dosyć łatwo go przeprowadzić dla języka niemieckiego czy angielskiego, ale o wiele trudniej dla języka polskiego, ponieważ gramatyka jest tu bardziej złożona i wymaga złożonej analizy tekstu wejściowego [5].

Ważnym aspektem jest również brak równoważności między tekstami. Inny jest tekst literacki, z publicznie dostępnych gazet czy też naukowy często zawierający obok terminów naukowych rysunki wykresy, wzory matematyczne czy chemiczne, litery alfabetu greckiego. Jak na razie brakuje idealnego programu, który by potrafił bezbłędnie przetworzyć każdy dokument. Co prawda są dostępne narzędzia zarówno komercyjne, jak i bezpłatne umożliwiające analizę tekstów, ale ich jakość jest różna, od bardzo prostych po bardziej zaawansowane. Wiele z nich dostosowana jest do języków zachodnich, niektóre z nich potrafią analizować nawet teksty chińskie czy też japońskie. Gorzej jest z językiem polskim ze względu na mały rynek dla takiego produktu. Aczkolwiek są już dostępne pewne rozwiązania, które zostaną przedstawione w rozdziale 4.

Osobne zagadnienie to wymagany format wejściowy danych do systemu analizującego dokumenty. Na ogół jest to *.txt*, *.html*, rzadziej *.doc* czy *.pdf*. W przypadku innych formatów niż *.txt* systemy mają wbudowane własne konwertery do wymaganego formatu. A zatem często trzeba dokonać konwersji dokumentu np. z formatu *.pdf* do *.txt*. Jednak w przypadku dokumentów naukowych zawierających wzory matematyczne, chemiczne, specyficzne litery z różnych języków, otrzymuje się postać wynikową mocno zniekształconą i bardzo odbiegającą od oryginału.

Do analizy dokumentów testowych uzywa się dedykowanych narzędzi informatycznych. Ogólnie można je podzielić na:

- proste – umożliwiające uzyskanie podstawowych statystyk w dokumentach takich jak częstość występowania, współwystępowania słów) (np.TextSTAT, AntConc);
- silniki indeksowania i wyszukiwania informacji (np. Lucene, Windows Desktop Search, Google, Yahoo);
- zaawansowane – pozwalające na złożoną analizę tekstów, z wykorzystaniem technik klasteryzacji, wizualizacją wyników i możliwością budowania ontologii (np. SAS Text Miner, Oracle Text, OntoGen Text Garden).

Wynikiem analizy fleksalnej i gramatycznej dokumentu jest zbiór słów. Tylko część z nich jest istotna dla treści. Słowa, które najczęściej występują w większości dokumentów powinny zostać pominięte, ponieważ są to zaimki, przyimki i spójniki. Następny etap – stemming – usuwa przyrostki i przedrostki oraz sprowadza słowa do formy podstawowej w oparciu o algorytmy rozpoznające reguły gramatyczne lub też poprzez odwołanie się do stosownych słowników. Przykładem słownika dla języka angielskiego może być WordNet. Natomiast prace nad stworzeniem polskiego WordNetu są prowadzone przez zespół kierowany przez Politechnikę Wrocławską. Więcej informacji na ten temat jest dostępnych pod adresem [6]. Dokumenty są przedstawiane jako zbiory słów (*bag of words*) z wyliczeniem, jak często każde z nich występuje w każdym dokumencie (*term-by-document frequency*) [8].

3. Podobieństwo dokumentów – podstawy matematyczne

Przeszukiwanie danych niestrukturalnych (wyszukiwanie pełnotekstowe – FTS) jest techniką wydajnego przeszukiwania dokumentów o charakterze tekstowym, wykorzystującą specjalny rodzaj indeksów – tzw. indeksy pełnotekstowe. FTS jest oparty na modelu przestrzeni wielowymiarowej. Tworzą ją wszystkie słowa zawarte w przetwarzanych dokumentach. Dokument można interpretować jako wektor składający się z n słów, gdzie każda współrzędna określa częstość wystąpień danego słowa w danym dokumencie. Analizy zbioru dokumentów można dokonać, budując macierz *term-by-document frequency* [7–8].

Dla przykładowych dwóch dokumentów może ona wyglądać następująco (tab. 1):

D1 – The cat is black

D2 – Black cat is my cat

Tabela 1

Przykładowa macierz *term-by-document frequency*

	the	**cat**	**is**	**black**	**my**
D1	1	1	1	1	0
D2	0	2	1	1	1

Ważność słów w macierzy można zwiększać lub zmniejszać, stosując współczynniki zwane wagami (a_{ij}, gdzie i, j to odpowiednio indeksy wierszy i kolumn w rozpatrywanej macierzy). Otrzymujemy wówczas macierz ważonej częstotliwości.

Rozróżniamy następujące wagi:

– frequency Weight (dotyczy występowania samego wyrażenia),
– term Weight (dotyczy liczby wystąpień danego wyrażenia w całej kolekcji – zbiorze dokumentów).

Frequency Weight – precyzuje metodę określania częstości występowania określonych zwrotów w dokumencie. Można tutaj wymienić następujące metody [8]:

– binarna (waga $w_{ij} = 1$ w przypadku występowania zwrotu, a $w_{ij} = 0$ przypadku jego braku;
– logarytmiczna $w_{ij} = \log_2(a_{ij} + 1)$ (logarytm przy podstawie 2 z liczby określającej częstość występowania słowa – pomniejsza wagę słów, które się często powtarzają);
– none (częstotliwość występowania słów bez modyfikacji: $w_{ij} = a_{ij}$).

Term weight – wagowanie zwrotu można określać m.in. za pomocą metod [8]:

– Entropy – przypisuje najwyższą wagę słowom, które wystąpiły najrzadziej w danym dokumencie;
– IDF (*Inverse Document Frequency*) – waga jest odwrotnością liczby dokumentów, w których pojawił się dany zwrot;
– GF-IDF (*Global Frequency-Inverse Document Frequency*) – obliczamy mnożąc IDF przez całkowitą częstotliwość;
– Normal – waga ta jest proporcjonalna to ilości wystąpienia danego słowa w dokumencie;
– None – każdemu zwrotowi przypisuje się wagę 1;
– Chi-Squared – wykorzystuje wartość testu Chi-kwadrat;
– Mutual Information – pokazuje jak rozkład dokumentów z wyrażeniem i, znajduje się blisko rozkładu dokumentów w całym zbiorze;
– Information Gain – określa oczekiwaną redukcję w Entropy w przypadku podzieleniu zbioru dokumentów według tego wyrażenia i.

Istnieje wiele algorytmów wagowania macierzy, takich jak algorytm modelu przestrzeni wektorowej, algorytm TF-IDF i tak dalej. Algorytmy wagowania w połączeniu z algorytmem mierzenia podobieństwa wektorów, takim jak na przykład miara kosinusowa lub współczynnik Jaccarda tworzą skuteczną metodę miary podobieństwa dokumentów.

3.1. TF-IDF

Waga TF-IDF (*term frequency–inverse document frequency*) jest często używana w metodach information retrieval i text mining. Mimo że TF-IDF jest dość wiekowym algorytmem wagowania, jest prosty i skuteczny. TF-IDF polega na ustalaniu względnej częstotliwości słów w danym, lokalnym dokumencie i porównaniu z odwróconą częstotliwością słowa w całej kolekcji dokumentów. Dla każdego słowa jego TF (*term frequency*) jest względną częstotliwością wystąpień tego słowa w kolekcji dokumentów, które stanowi ważność słowa wewnątrz danego dokumentu, a jego IDF (*inverse document frequency*) jest odwrotnie proporcjonalna do wystapień słowa w odniesieniu do korpusu dokumentu, czyli przedstawia znaczenie tego słowa w całej kolekcji dokumentów [8–9].

Algorytm działa w następujący sposób:
mając:

D – kolekcja dokumentów,
d – dany dokument, dla którego $d \in$ D,
w – słowo występujące w dokumencie d,

obliczamy:

$$w_d = f_{w,d} * \log(|\mathrm{D}| / f_{w,D})a \tag{1}$$

gdzie $f_{w,d}$ jest ilością wystąpień słowa w dokumencie d, $|D|$ jest rozmiarem korpusu dokumentu oraz $f_{w,D}$ jest ilością dokumentów, w których występuje słowo w. Czasami przy dużych kolekcjach dokumentów możemy dokonać normalizacji części TF, stosując technikę redukcji wymiaru SVD (*Singular Value Decomposition*). Redukcja pomogą nam zmniejszyć ilość wymiarów i przybliżyć macierz ważonej częstotliwości [8–9].

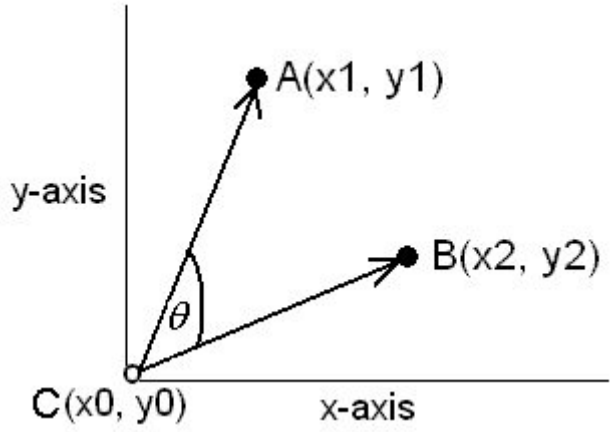
3.2. Miara kosinusowa

Miara kosinusowa jest wydajnym algorytmem obliczania podobieństwa w przypadku tekstów. Podstawowym założeniem tej metody obliczania podobieństwa jest następująca idea:

Dla dwóch punktów A, B na skali *xy* jak pokazuje rysunek 3, podobieństwa między A i B są zdefiniowane następująco:

$$Sim(A,B) = \cos\Theta = A \cdot B / |A||B| \tag{2}$$

gdzie *Sim*(*A*, *B*) jest podobieństwem dokumentu *A* do dokumentu *B*, *A*·*B* jest iloczynem skalarnym wektorów *A* i *B*, który to równa się: $x1*x2 + y1*y2$, $|A||B|$ określa odległość pomiędzy A i B, która jest określona wzorem: $(x_1^2 + y_1^2)^{1/2} (x_2^2 + y_2^2)^{1/2}$ [8, 10].



Rys. 3. Współrzędne punktów A i B na dwuwymiarowej skali liczbowej

4. Przykłady narzędzi do analizy tekstów

Do analizy danych tekstowych dostępne są narzędzia zarówno ogólnodostępne typu open source, jak i komercyjne. Ich możliwości są bardzo zróżnicowane – od prostych podających podstawowe informacje statystyczne na temat dokumentów po bardziej wyrafinowanie systemy budujące ontologie pojęć lub mające wbudowane zaawansowane algorytmy analizy składni. Wszystkie dobrze sobie radzą z językami zachodnimi, chińskim czy nawet japońskim. Problem jest z językiem polskim. Nie dotyczy on tylko sposobu kodowania polskich znaków, ale i programów analizujących składnię. W ramach opracowania przeanalizowano kilka wybranych narzędzi.

4.1. TextSTAT

TextSTAT to prosty program do analizy tekstów. Potrafi on obsługiwać pliki ASCII/ANSI, HTML, formaty MS Word (.doc i .docx) oraz OpenOffice (sxw i .odt), z których tworzy listę częstotliwości występowania poszczególnych słów, ma możliwość tworzenia konkordancji oraz list frekwencyjnych. *TextStat* posiada aż 6 wersji językowych interfejsu (również j. polski) i pracuje we wszystkich systemach operacyjnych. To co wyróżnia go spośród innych darmowych programów tego typu, to możliwość tworzenia korpusu ze stron internetowych wczytywanych przez program bezpośrednio z sieci. Niestety, program nie posiada kilku istotnych funkcji, takich jak tworzenie listy słów kluczowych czy wyszukiwanie kolokacji oraz ciągów wielowyrazowych [11].

4.2. AntConc

AntConc to darmowy program do analizy tekstów oferujący szeroki wachlarz funkcji. Wśród nich znajduje się tworzenie konkordancji, list frekwencyjnych, list słów kluczowych i wykresów dystrybucji, a także wyszukiwanie ciągów wielowyrazowych i kolokacji. Przyjazny interfejs, szybkość wykonywanych analiz i funkcjonalność dorównująca wielu komercyjnym aplikacjom sprawiają, że AntConc jest szczególnie godny polecenia zarówno dla osób stawiających swoje pierwsze kroki w pracy z korpusami dokumentów, jak i dla bardziej zaawansowanych użytkowników [12].

4.3. WordSmith

WordSmith Tools to prawdopodobnie najbardziej popularny w ośrodkach akademickich pakiet narzędzi do analizy danych tekstowych. Oferuje imponujący wachlarz funkcji oraz możliwości dostosowania poszczególnych narzędzi do konkretnych zadań. Obsługuje znaczniki, działa szybko i dobrze radzi sobie nawet z dużymi korpusami. *WordSmith Tools* działa w Windows oraz Mac OS X. Pełna wersja oprogramowania jest płatna, ale istnieje możliwość wypróbowania wersji demo o ograniczonej funkcjonalności. Program opiera się na trzech podstawowych funkcjach: konkordancja (Concord), lista słów kluczowych (KeyWord) oraz lista frekwencyjna (WordList) [12].

4.4. Poliqarp

Poliqarp to darmowe oprogramowanie do przeszukiwania dużych korpusów. Powstał w efekcie prac nad Korpusem IPI PAN i obsługuje ten korpus zarówno w wersji on-line, jak i off-line. Dzięki przejrzystemu interfejsowi korzystanie z podstawowych funkcji programu oraz wykorzystanie jego możliwości konfiguracyjnych nie powinno sprawiać trudności nawet początkującym użytkownikom. Program można uruchamiać zarówno w środowisku Windows, jak i Linux. Dodatkowym atutem jest fakt, że istnieją dwie wersje językowe – polska i angielska. *Poliqarp* daje możliwość wyszukiwania określonych słów czy fraz. Pozwala także na znajdowanie sekwencji określanych za pomocą wyrażeń regularnych, na przykład: wszystkich występujących w korpusie fraz składających się z rzeczownika i przymiotnika lub wszystkich form fleksyjnych wybranego wyrazu (funkcja szczególnie przydatna w przypadku badań nad językiem polskim). Operacje te, zarówno w wersji on-line, jak i off-line, przebiegają dość szybko – przy prostych zapytaniach wyszukiwanie nie zajmuje więcej niż kilka sekund. [12-13]

5. Podsumowanie

Technologie przetwarzania języka naturalnego można wskazywać jako jedne z podstawowych dla technologii zarządzania wiedzą, ponieważ umożliwiają:

– automatyczne przetwarzanie dokumentów (treści) np. WWW,
– maszynowo przetwarzane opisywanie (*annotation*) tekstów w języku naturalnym za pomocą pojęć zawartych w ontologii,
– odkrywanie nowych elementów ontologii (tj. pojęć, klas, instancji, atrybutów, relacji, twierdzeń),
– automatyczne wyszukiwanie elementów wiedzy.

Wymienione aspekty można traktować w świetle automatyzacji tłumaczenia tekstów zapisanych w języku naturalnym na sformalizowany język reprezentacji wiedzy. Tak postawione zagadnienie, tzn. automatyzacja translacji tekstów w języku naturalnym na język formalny, jest jednym z najbardziej pożądanych i obiecujących kierunków współczesnych badań w dziedzinie systemów zarządzania wiedzą. Celem automatyzacji jest tworzenie baz wiedzy zapisanej w języku sformalizowanym, umożliwiającym operowanie tą wiedzą w sposób automatyczny.

Literatura

[1] Gołuchowski J., *Technologie informatyczne w zarządzaniu wiedzą w organizacji*. AE, Katowice 2005.
[2] Aamodt A., Plaza E., *Case-Based Reasoning: Foundational Issues*. Methodological Variations, and System Approaches, „AICom, Artificial Intelligence Communications", IOS Press 1994.
[3] Tomassen S.L., *Semi-automatic generation of ontologies for knowledge-intensive CBR*. Norwegian University of Science and Technology, 2002.
[4] Filipowska A., *Jak zaoszczędzić na czytaniu? Automatyczne tworzenie abstraktów z dokumentów*. http://www.gazeta-it.pl/pl/trendy/6011, Gazeta IT nr 3, marzec 2004.
[5] *Wybrane problemy zarządzania wiedzą*. Instytut Łączności, Państwowy Instytut Badawczy, Praca nr 06300017, 2007.
[6] http://plwordnet.pwr.wroc.pl, 2011.
[7] Ikonomakis M., Kotsiantis S., Tampakas V., *Text Classification Using Machine Learning Techniques*. WSEAS TRANSACTIONS on COMPUTERS, Issue 8, vol. 4, August 2005, 966–974.
[8] Kłopotek M.A., *Inteligentne wyszukiwarki internetowe*. Exit, 2001.
[9] Ramos J., *Using TF-IDF to Determine Word Relevance in Document Queries*. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.121.1424&rep=rep1&type=pdf, 2011.
[10] http://www.miislita.com/information-retrieval-tutorial/cosine-similarity-tutorial.html, 2011.
[11] http://neon.niederlandistik.fu-berlin.de/en/textstat/, 2011.
[12] http://www.korpusy.net/index.php/narzdzia/programy-do-analizy, 2011.
[13] http://korpus.pl/index.php?page=poliqarp 2011.