Wyszukiwanie w repozytoriach tekstowych w języku polskim

Maciej Klubiński

Politechnika Warszawska, Wydział Elektroniki i Technik Informacyjnych,
ul. Nowowiejska 15/19, 00-665 Warszawa
M.Klubinski@stud.elka.pw.edu.pl

Abstrakt. Artykuł jest wprowadzeniem w zagadnienia związane z funkcjonowaniem i tworzeniem wyszukiwarki umożliwiającej wyszukiwanie w repozytoriach w języku polskim. Przedstawione zostały wszystkie zagadnienia związane z tworzeniem wyszukiwarki, począwszy od roli i budowy analizatora tekstów, poprzez stemmer, charakterystyczną budowę indeksu, aż na budowaniu zapytań skończywszy. Wszystkie ww. zagadnienia przedstawione zostały w kontekście wyszukiwarki dla języka polskiego, a co za tym idzie, omówione zostały wszystkie problemy, jakie stwarza język polski.

1 Wprowadzenie

Przełom XX i XXI wieku określany jest mianem „eksplozji informacyjnej". Określenie to wzięło swoją nazwę od zjawisk, jakie mają miejsce w otaczającej nas rzeczywistości. Jeszcze trzydzieści lat temu szacowano, że tygodniowo publikowanych było około stu nowych pozycji książkowych oraz około trzech tysięcy artykułów naukowych. W roku 2000 liczby te szacowane były na poziomie odpowiednio trzystu pięćdziesięciu nowych pozycji książkowych oraz prawie dwudziestu tysięcy artykułów naukowych. W ciągu dwudziestu lat każda z tych liczb wzrosła parokrotnie, ciągle rośnie i nie ma najmniejszych oznak wskazujących na to, by proces ten miał ulec spowolnieniu bądź zatrzymaniu.

Zjawisko tak szybkiego przyrostu informacji nie mogłoby mieć miejsca, gdyby nie rozwój i duża popularność Internetu. To właśnie Internet jest głównym źródłem poszukiwania informacji, jak i dzielenia się nimi. Ostatnio liczba nowopowstających stron internetowych rośnie wykładniczo, a w lutym 2007 roku liczba ta szacowana była na 30 bilionów. Obecnie około 80% ogółu informacji przechowywanych jest w postaci dokumentów tekstowych, których znakomita większość umieszczona jest na stronach internetowych w sieci Internet.

Mając na uwadze oba wyżej przedstawione zjawiska można dojść do wniosku, że w całej tej masie dokumentów tekstowych znalezienie informacji, które w danej chwili nas interesują, może okazać się czynnością bardzo trudną i czasochłonną, o ile w ogóle wykonalną dla pojedynczej osoby. Współczesny człowiek nie jest w stanie ogarnąć umysłem tak szybko przyrastającej wiedzy. Nie wynika to bezpośrednio z faktu, że umysł człowieka na to nie pozwala. Wynika to raczej z tego, iż człowiek nie jest w stanie przeczytać wszystkich nowopowstałych dokumentów na bieżąco. A

nawet gdyby mu się to udało, to i tak nie byłby w stanie zapamiętać wszystkiego, co przeczytał. Innymi słowy, pomimo ogromu dostępnej informacji, informacja ta jest nieosiągalna, a co za tym idzie, przestaje być użyteczna dla człowieka. Do tej sytuacji doskonale pasuje współczesna parafraza słów Sokratesa[1]: „Nie wiem co wiem". Oddaje ona sedno problemu - informacja jest dostępna w Internecie lecz nikt nie może z niej skorzystać.

I w tym oto momencie pojawia się ogromne pole do popisu dla twórców wszelkich wyszukiwarek tekstowych - począwszy od najprostszych wyszukiwarek, przeszukujących dokumenty po zawartości słów, poprzez wyszukiwarki tematyczne, aż na wyszukiwarkach wykorzystujących mechanizmy analizy tekstów oraz Semantic Web skończywszy.

2 Budowa wyszukiwarki

Pomimo różnorodności istniejących wyszukiwarek, wszystkie istniejące rozwiązania charakteryzują się jednak bardzo podobną budową. Każda wyszukiwarka składa się z pewnych elementów, które są nieodłącznym składnikiem jej budowy.

Budowę tą najprościej opisać na podstawie ludzkiego ciała. Tułowiem wyszukiwarki, czyli tą częścią, na której opierają się pozostałe fragmenty, jest repozytorium dokumentów. Repozytorium to może mieć różne postaci – zbiór dokumentów w postaci elektronicznej, korpus tekstów, zbiór danych w bazie danych, czy sieć WWW. Bez tego elementu tworzenie wyszukiwarki byłoby bezsensowne, gdyż nie byłoby czego wyszukiwać.

Kolejnym elementem budowy wyszukiwarki jest indekser, czyli odpowiednik ludzkiego serca. Zadaniem indeksera jest analiza repozytorium dokumentów pod względem ich zawartości, której wynikiem jest utworzenie indeksu. Indeks ten przechowywać powinien niezbędne informacje umożliwiające wyszukiwarce wyszukiwanie dokumentów oraz informacje o samym dokumencie, z którego to powyższe dane pochodzą.

Trzecim elementem budowy wyszukiwarki jest moduł budowy zapytań. Odpowiada on głowie w ciele człowieka. Zadaniem tego modułu jest analiza zapytań pochodzących od użytkownika oraz taka ich przebudowa, by jako wynik można było otrzymać logiczne wyrażenie zawierające w sobie wszystkie warunki, jakie nałożył na wyszukiwane dokumenty użytkownik.

Ostatnim elementem budowy wyszukiwarki jest moduł wyszukujący dokumenty relewantne. Moduł ten pełni taką samą rolę jak mózg w ciele człowieka. Jego zadaniem jest właściwe zarządzanie poszczególnymi modułami wyszukiwarki w sposób umożliwiający wydajne i skuteczne wyszukiwanie. Wykorzystuje on wszystkie ww. elementy. Z modułu zapytań pobiera wyrażenie logiczne składające się na zapytanie użytkownika, następnie wykorzystując indeks zbudowany przez indekser, wyszukuje dokumenty relewantne do zapytania, po czym wyświetla znalezione dokumentu w

[1] Oryginalna wersja wyrażenia brzmi: „Wiem, że nic nie wiem" (gr. *Oida ouden eidos*). Wyrażenie to zostało utworzone wtórnie na podstawie słów Sokratesa przytoczonych przez Platona w *Obronie Sokratesa*: „Jemu się zdaje, że coś wie, choć nic nie wie, a ja, ja nic nie wiem, tak mi się nawet i nie zdaje."

wynikach wyszukiwania, pozwalający tym samym na zapoznanie się użytkownikowi ze znalezionymi dokumentami.

Poszczególne elementy różnią się między sobą pod względem skomplikowania budowy oraz funkcjonalności. Najprostszym elementem jest repozytorium tekstów. Jest to nic więcej jak odpowiednio zorganizowany zbiór tekstów w postaci elektronicznej. Pozostałe elementy są już bardziej skomplikowane, przez co każdy z nich zostanie teraz omówiony w kolejnych punktach.

3 Działanie indeksera, czyli jak zbudować indeks

Dobrze zbudowany indeks to połowa sukcesu w procesie tworzenia dobrej wyszukiwarki. Przy wykorzystaniu dobrze zbudowanego indeksu wyszukiwanie będzie odbywać się szybko oraz skutecznie. A do tego właśnie dąży każdy twórca wyszukiwarek. Jednak zbudowanie indeksu to proces skomplikowany i wykonywany w kilku kolejnych krokach. Każdy z kroków zostanie teraz przedstawiony w kolejności, w jakiej jest wykonywany.

3.1 Rozpoznawanie języka dokumentu

Pierwszym krokiem w procesie budowania indeksu jest rozpoznanie języka dokumentu, którego analizę rozpoczynamy. Rozpoznawanie języka odbywać się może na wiele sposobów. Sam sposób jest mało istotny dopóty, dopóki wyniki jego działania zwracają prawidłowe wyniki. Jednak możliwymi sposobami na rozpoznawanie języka może być rozpoznawanie na podstawie:

- unikalnych dla danego języka ciągów znaków (np. eux – francuski, cchi – włoski, der – niemiecki),
- występowania określonych znaków (np. ü – niemiecki, ć – polski),
- wykorzystując cechy składniowe sylab danego języka,
- wykorzystując rozkład prawdopodobieństwa występowania liter oraz dłuższych ciągów znaków (n-gramów),
- porównując dokumenty ze słowami ze stop-listy.

3.2 Tokenizacja

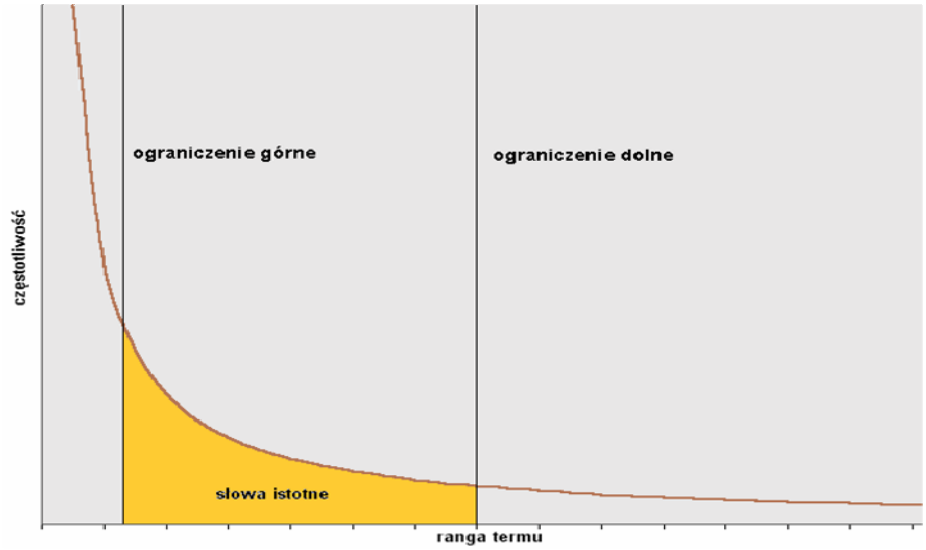
Tokenizacja to proces, w wyniku którego analizowany dokument, widziany jako jedna spójna całość, podzielony zostaje na mniejsze logiczne części. Podział ten składa się z kilku poziomów podziału. Pierwszy podział to podział tekstu na akapity. W analizowanym dokumencie wyszukiwane są fragmentu tekstu, które reprezentują poszczególne akapity tekstu. Akapity te są następnie poddawane dalszym podziałom.

Kolejnym poziomem podziały jest podział na zdania. Każdy akapit dzielony jest na poszczególne zdania, które poddawane są dalszej analizie. Na tym poziomie pojawia się kilka problemów związanych ze specyfiką języków. W językach występują różne znaki interpunkcyjne, przez co nie zawsze znak „.” oznacza jednoznacznie koniec

zdania. Czasem znak ten może być użyty wewnątrz zdania. Może się też zdarzyć, że zdanie kończyć się będzie innym znakiem interpunkcyjnym, jak np. „?”, „!”, itp. Oczywiście są też języki, które nie posiadają żadnych znaków interpunkcyjnych. Najlepszym tego przykładem jest język chiński. W języku tym występują tylko i wyłącznie znaki, które nabierają znaczenia dopiero wtedy, gdy występują w sąsiedztwie innych znaków. W takim przypadku podział na zdania może okazać się niemożliwy.

Kolejnym podziałem jest podział na wyrażenia. Przez wyrażenia rozumiemy tutaj podział np. na związki frazeologiczne. Na tym poziomie następuje jednocześnie podział na poszczególne słowa. W trakcie podziału system analizuje, czy dany ciąg słów tworzy związek frazeologiczny, czy też nie. Jeśli dany ciąg wyrazów tworzy związek frazeologiczny, to informacja o wystąpieniu tegoż związku zostaje zapamiętana. Wynikiem analizy na tym poziomie jest zbiór słów występujących w dokumencie.

Zbiór takich słów poddawany jest ostatniemu etapowi analizy, a mianowicie wyborowi termów do dalszej analizy. Przy wyborze termów wykorzystywane jest prawo Zipfa[2], dzięki czemu do dalszej analizy przechodzą tylko te słowa, które można uznać za istotne z punktu widzenia informacyjnego.



Rys. 1. Wybór słów istotnych na podstawie Prawa Zipfa

3.3 Usunięcie stop-words

Stop-words to nazwa określająca zbiór słów najczęściej występujących w danym języku, a które to słowa nie niosą ze sobą żadnej treści informacyjnej, lecz których występowanie w tekście jest nieodzowną częścią samego tekstu. W tekstach zawarta

[2] Prawo Zipfa: Jeżeli weźmiemy wystarczający zbiór dokumentów z typowymi angielskimi słowami i posortujemy te słowa według częstości występowania, to iloczyn częstości występowania słowa i numeru w tym uporządkowaniu będzie stały.

jest wiedza, którą autor próbuje przekazać czytelnikowi. Jednak by to się udało, tekst musi być napisany w sposób, w którym będzie zrozumiały. Niestety nie ma sformalizowanego sposobu na to, by przekazać wiedzę używając tylko i wyłącznie samych informacji. Pisząc tekst należy zadbać o jego zrozumienie przez czytającego. A żeby to osiągnąć, konieczne jest używanie słów pomocniczych, których zadaniem jest spełnianie roli czegoś w rodzaju „kleju" pomiędzy kolejnymi informacjami zawartymi w tekście. A takimi właśnie słowami są słowa z listy stop-words. Słowa te to głównie spójniki, zaimki i przyimki. Bez nich zdania byłyby niezrozumiałe, a czytanie tekstu byłoby niemożliwe.

Lista słów stop-words jest inna dla każdego języka i jest dla niego charakterystyczna[3]. Słowa z tej listy stanowią ok. 30% wszystkich słów zawartych w tekstach. W niektórych rozwiązaniach porównywanie listy występujących słów w dokumencie z listą słów ze stop-words jest wykorzystywana do rozpoznawania języka (patrz: punkt 3.1). Niemniej, słowa występujące na liście stop-words nie niosą ze sobą żadnej informacji, przez co są nieprzydatne z punktu widzenia procesu indeksowania. Słowa takie są usuwane ze zbioru słów występujących w analizowanym tekście.

3.4 Stemming

Kolejnym etapem tworzenia indeksu jest stemming. Pod pojęciem tym kryje się proces, na wejściu którego podawane są poszczególne słowa, zaś na wyjściu którego otrzymujemy ciąg znaków jednoznacznie reprezentujący rodzinę słów, do której to rodziny dane słowo należy. W zależności od przyjętego rozwiązania, na wyjściu mogą pojawić się początkowe części słowa wejściowego, charakterystyczne dla danej rodziny słów, swoją budową przypominające temat słowa, lecz nie koniecznie będące poprawnymi tematami słów w rozumieniu leksykalnych danego języka. Inne rozwiązania na wyjściu zwracają słowo podstawowe dla danej rodziny, go rodziny którego to słowa słowo podane na wejściu należało.

Niezależnie jednak od przyjętego rozwiązania, wynik działania stemmingu ma dać ciąg znaków jednoznacznie klasyfikujący i jednocześnie jednoznacznie opisujący dane słowo. Dzięki temu, w trakcie dalszych etapów budowania indeksu, dokumenty zawierające słowa z tej samej rodziny, lecz występujące w różnych odmianach, będą mogły być zaklasyfikowane do tej samej grupy. Im lepszy proces stemmingu, tym lepszą jakością zwracanych wyników charakteryzuje się wyszukiwarka.

W istniejących rozwiązaniach w ramach tworzenia stemmerów spotkać się można z trzema typami rozwiązań. Podział ten wykonany jest pod względem sposobu działania stemmera. Typami stemmerów są zatem: stemmery algorytmiczne, stemmery

[3] Lista stop-words dla języka polskiego: a, aby, ale, bardziej, bardzo, bez, bo, bowiem, był, była, było, były, będzie, co, czy, czyli, dla, dlatego, do, gdy, gdzie, go, i, ich, im, innych, iż, jak, jako, jednak, jego, jej, jest, jeszcze, jeśli, już, kiedy, kilka, która, które, którego, której, który, których, którym, którzy, lub, ma, mi, między, mnie, mogą, może, można, na, nad, nam, nas, naszego, naszych, nawet, nich, nie, nim, niż, o, od, oraz, po, pod, poza, przed, przede, przez, przy, również, się, sobie, swoje, są, ta, tak, takie, także, tam, te, tego, tej, ten, też, to, tu, tych, tylko, tym, u, w, we, wiele, wielu, więc, wszystkich, wszystkim, wszystko, właśnie, z, za, zawsze, ze, że
Lista stop-words dla języka angielskiego: a, an, and, are, as, at, be, but, by, for, if, in, into, is, it, no, not, of, on, or, such, that, the, their, then, there, these, they, this, to, was, will, with

słownikowe oraz stemmery hybrydowe. Każdy z typów zostanie teraz pokrótce scharakteryzowany.

Pierwszym typem stemmerów są stemmery algorytmiczne. Stemmery tych typów powstawały jako pierwsze, już w latach ’60 XX wieku. Były one dedykowane pod język angielski i tylko one były głównie rozwijane. Pierwsze rozwiązania bazowały na analizie końcówek słów. W swojej bazie wiedzy zawierały listę znanych i najczęściej występujących końcówek słów, i przy ich wykorzystaniu starały się pozbawić analizowane słowo końcówki, przez co tym samym zwrócić rdzeń słowa. Kolejne unowocześnienia tego typu stemmerów opierały się na rozszerzaniu listy znanych końcówek oraz na zwiększaniu liczby przebiegów algorytmu. Najbardziej znany stemmer tego typu to stemmer Portera. Jest on wieloprzebiegowy. W każdym kolejnym kroku zajmuje się innym zagadnieniem, m.in. sprowadzeniem słowa do liczby pojedynczej, sprowadzenie słowa do czasu teraźniejszego, usunięcie końcówki itd. Algorytm ten jest obecnie najpopularniejszym stemmerem algorytmicznym dla języka angielskiego, przez co jest bardzo często wykorzystywany w systemach do analizy tekstów w języku angielskim. Niestety, stemmery tego typu charakteryzują się dosyć niską jakością, a wynikającą głównie z tego, iż nie zawsze dla tych samych słów potrafią zwrócić dokładnie ten sam ciąg znaków na wyjściu.

Drugim typem stemmerów są stemmery słownikowe. Stemmery te działają na podstawie tylko i wyłącznie słownika, który jest im dostarczony. Słowniki takie zawierają znaczną liczbę różnych form gramatycznych poszczególnych słów oraz odpowiadającej każdej z nich formę podstawową (lemat i rdzeń). Działanie tego typu stemmera jest bardzo proste, a polega wyłącznie na odszukaniu słowa pojawiającego się na wejściu w słowniku, a następnie na zwróceniu podstawowej formy odpowiadającej temu słowu w słowniku. Dzięki takiemu działaniu stemmer taki zwraca zawsze jednakową formę dla tych samych słów i charakteryzuje się wysoką jakością. Jednak posiada dwie znaczące wady. Po pierwsze wymaga przechowywania słownika, który może osiągać duże rozmiary i jest każdorazowo przeglądany w poszukiwaniu kolejnych słów pojawiających się na wejściu. Drugą wadą jest to, iż w momencie, gdy na wejściu takiego stemmera pojawi się słowo niewystępujące w słowniku, algorytm nie wykona na podanym słowie żadnych operacji i zwróci oryginalne słowo. Problem ten potrafi być bardzo uciążliwy w sytuacjach, gdy słownik jest niewielkich rozmiarów, bądź gdy tworzony był na słowach charakterystycznych dla pewnej dziedziny wiedzy, a wykorzystywany jest przy stemmingu tekstów z zupełnie innej dziedziny wiedzy, która również charakteryzuje się specyfiką słów.

Mając na uwadze zarówno zalety, jak i wady stemmerów obu ww. typów, stworzono rozwiązanie pośrednie, stworzono stemmer typu mieszanego, nazwanego stemmerem hybrydowym. Stemmer tego typu łączy w sobie oba typy stemmerów, wykorzystując przy tym zalety obu typów, usuwając jednocześnie ich wady. Stemmer hybrydowy również posiada słownik, zbudowany na identycznych zasadach jak słowniki w stemmerach słownikowych. Również działanie jest identyczne. Gdy pojawia się słowo na wejściu, przeszukiwany jest słownik w poszukiwaniu danego słowa. Gdy słowo zostaje znalezione w słowniku, jego podstawowa forma jest zwracana na wyjście. Pierwszą różnicą w zachowaniu tego stemmera jest sytuacja, gdy dane słowno nie zostanie odnalezione. W takim wypadku oryginalne słowo nie jest zwracane, lecz poddawane jest działaniu stemmera algorytmicznego. Na wyjście przekazywany jest zatem wynik działania stemmera algorytmicznego. Drugą różnicą w

działaniu stemmera hybrydowego względem stemmera słownikowego jest to, iż w przypadku, gdy dane słowo nie zostaje znalezione w słowniku, po zakończeniu działania stemmera algorytmicznego, słowo takie wraz z wynikiem działania stemmera algorytmicznego dodawane jest do słownika. A zatem, gdy dane słowo pojawi się po raz kolejny, jego odpowiadająca mu forma zostanie zwrócona już bezpośrednio ze słownika. Dzięki takiemu rozwiązaniu udało się zniwelować jedną z wad stemmerów słownikowych, a mianowicie ich bezradność w sytuacjach, gdy słowa nie ma w słowniku. Dodatkowo słownik stał się dynamiczny i jest rozbudowywany w trakcie działania stemmera. Jednak dalej pozostaje słownik, który może mieć spore rozmiary oraz który musi być przeglądany. Tej wady nie da się jednak usunąć. Drugą wadą jest też ciągle skuteczność działania stemmera algorytmicznego. Gdy słowa nie ma w słowniku, to właśnie ten stemmer ma znaczenie. I to, jaką formę dla danego słowa zwróci, ma zasadnicze znaczenie na jakość działania całego stemmera hybrydowego. Bo gdy stemmer algorytmiczny zwróci błędną formę, forma taka trafia do słownika stemmera słownikowego i przy każdym kolejnym pytaniu o takie słowo, zwracana będzie niepoprawna forma dla danego słowa. Jednak suma summarum, stemmer hybrydowy i tak działa lepiej niż poszczególne typu stemmerów słownikowych i algorytmicznych działające rozłącznie.

3.5 Usuwanie synonimów

Na tym etapie budowy indeksu pojawia się słownik zawierający synonimy dla poszczególnych słów, tzn. tezaurus. Gdy pojawia się słowo na wejściu, słowo takie jest odszukiwane w słowniku, i jeśli w nim występuje, zwracane jest słowo główne dla danej rodziny synonimów. Zatem wszystkie synonimy danego słowa zostaną zamienione na słowo podstawowe.

Dzięki temu procesowi w znacznym stopniu można ograniczyć ilość słów, jakie docelowo będą zapisane w indeksie. Gdy dane słowo posiada wiele synonimów, ograniczenie ilości różnych słów może być znaczące. Ma to bezpośrednio pozytywne odbicie w wielkości tworzonego indeksu. Indeks taki jest po prostu mniejszych rozmiarów.

Jednak rozwiązanie takie niesie ze sobą również negatywne skutki. Gdy użytkownik szuka dokładnie jednej i tylko jednej formy danego słowa, w wyniku otrzyma dokumenty zawierające to słowo oraz wszystkie dokumenty zawierające synonimy tego słowa. Innymi słowy, w wyniku wyszukiwania otrzyma więcej dokumentów, które uznane zostaną przez system za relewantne. Pogarsza to tym samym dokładność działania wyszukiwarki, poprawiając jednocześnie odzysk dokumentów (dokładność i odzysk zostaną wyjaśnione w dalszej części artykułu).

3.6 Zastąpienie terminów bardziej ogólnymi

Etap ten jest bardzo podobny do poprzedniego etapu, a mianowicie również opiera się o słownik. Wykorzystuje słownik, w którym występuje słowo ogólne oraz wiele jego odpowiedników, bardziej szczegółowych. Gdy któreś ze słów szczegółowych pojawia się na wejściu, zwracane jest słowo ogólne.

Etap ten jest opcjonalny i w standardowych rozwiązaniach jest najczęściej niewykorzystywany. Ma on bardzo podobne zalety i wady, jak etap poprzedni. Zaletami są: dalsze ograniczenie rozmiarów tworzonego indeksu oraz poprawienie odzysku dokumentów. Wadą jest zmniejszenie dokładności działania wyszukiwarki, gdyż w wynikach zwracać ona będzie wszystkie dokumentu dla słowa ogólnego, gdy wyszukiwane będzie którekolwiek ze słów bardziej szczegółowych. Etap ten wykorzystywany jest w systemach dedykowanych, gdzie opisany efekt uznany może być za cechę pozytywną, a nie negatywną działania systemu.

3.7 Rozbijanie zlepków wyrazowych

Etap ten polega na przeprowadzeniu procesu tokenizacji dla wszystkich związków frazeologicznych bądź innych zlepków wyrazowych (np. nazw chemicznych), które nie zostały podzielone na poszczególne słowa w procesie tokenizacji. Etap ten również jest opcjonalny, lecz wykorzystywany jest już stosunkowo częściej, niż etap poprzedni.

Etap ten przeprowadza się w celu poprawienia odzysku wyszukiwarki. Ma to na celu umożliwienie użytkownikowi odnalezienie dokumentów zawierających rozbijane na tym etapie zlepki wyrazowe, bez konieczności wpisywania do szukanej frazy całego zlepku wyrazowego. Dzięki temu możliwe będzie odnalezienie danego zlepku już po jego fragmencie.

Jednocześnie zabieg ten pogarsza dokładność działania wyszukiwarki. W wynikach wyszukiwania pojawią się nie tylko wszystkie dokumenty zawierające szukany zlepek wyrazowy, ale również wszystkie dokumentu, które mają część wspólna z szukanym zlepkiem wyrazowym, i która to część wspólna została w danym momencie wpisana jako szukana fraza.

3.8 Obliczanie wag dla słów kluczowych

Etap ten pojawił się całkiem niedawno w rozwiązaniach wykorzystywanych przy budowie indeksu, wydaje się jednak etapem wpływającym pozytywnie na ten proces. Na etapie tym analizowane jest miejsce wystąpienia poszczególnych słów w dokumencie. Gdy dane słowo występuje np. w tytule, streszczeniu bądź bibliografii, ważność tego słowa w dokumencie wzrasta. Innymi słowy, zakłada się, że skoro dane słowo wystąpiło w tych miejscach, to cały dokument z dużym prawdopodobieństwem dotyczył będzie właśnie tego słowa i jego znaczenia. Taki dokument jest wtedy uznawany za relewantny dla danego słowa i na pewno zostanie zwrócony w sytuacji, gdy użytkownik wpisze do szukanej frazy to słowo.

Etap ten poprawia zatem dokładność działania wyszukiwarki, jak i przyspiesza czas odnalezienia tego dokumentu przez użytkownika (bo dokument taki zostanie zwrócony jako jeden z pierwszych na liście z wynikami wyszukiwania).

3.9 Tworzenie indeksu

Etap ten jest ostatnim etapem tworzenia indeksu i wynikiem jego działania jest właściwe stworzenie indeksu i fizyczne jego utworzenie. Po przeprowadzeniu wszystkich wcześniejszych etapów oraz po zgromadzeniu wszystkich niezbędnych do wyszukiwania dokumentów informacji, informacje te są zbierane w jedną logiczną całość i zapisywane na dysk fizyczny.

Przy zapisie tychże danych ważnym elementem są użyte struktury danych. Struktury te muszą pozwalać szybkie i efektywne przeszukiwanie indeksu w poszukiwaniu danych, a jednocześnie zajmować jak najmniej miejsca fizycznie na dysku. Dzięki temu wyszukiwarka będzie szybko wyszukiwać dokumenty, zajmując jednocześnie stosunkowo mało miejsca.

Co się zaś tyczy samych informacji, jakie zapisywane są do indeksu, to jest to zależne już przede wszystkim od pozostałych elementów wyszukiwarki, a mianowicie od modułu budującego zapytania oraz wyszukującego dokumenty. Muszą zostać zapisane wszystkie te informacje, które będą wykorzystywane przy wyszukiwaniu oraz umożliwiające odnalezienie dokumentu w repozytorium, i żadne inne.

Organizacja danych w indeksie również zależy od ww. elementów. Organizacja danych w indeksie musi być dostosowana do modułu wyszukującego dokumenty, bo możliwe było jakiekolwiek wyszukiwanie.

4 Wyszukiwanie dokumentów

Mając zbudowany indeks można rozpocząć proces wyszukiwania. W zależności od stworzonego systemu, wyszukiwanie to może mieć wiele form. To samo wiąże się z możliwościami budowania zapytań do wyszukiwarki. Im lepszy system, tym więcej kryteriów wyszukiwania dostarcza wyszukiwarka ze swoją funkcjonalnością.

Niemniej, we współczesnych wyszukiwarkach można mówić o pewnym standardzie funkcjonalności, jakie dobre wyszukiwarki powinny dostarczać użytkownikowi. W skład tejże funkcjonalności wchodzą takie możliwości wyszukiwania, jak:

- wyszukiwanie po słowach kluczowych – jest to najprostszy typ wyszukiwania. Użytkownik wpisuje frazę składającą się z ciągu słów, po czym wyszukiwarka wyszukuje te dokumenty, w których występują wszystkie słowa wymienione we frazie.
- wyszukiwanie boolowskie – wyszukiwanie to realizowane jest przy wykorzystaniu zbiorów odwróconych. Zbiory takie przypominają swoją budową hashMapę, gdzie kluczami są poszczególne słowa, zaś wartościami są listy dokumentów, w których słowo z klucza występuje. Wyszukiwanie na podstawie takich zbiorów przeprowadzane jest przy wykorzystaniu algebry zbiorów. Gdy użytkownik wpisze kilka słów do szukanej frazy, wyszukiwane są najpierw zbiory dokumentów zawierających te słowa, po czym przeprowadzana jest operacja iloczynu logicznego wszystkich znalezionych zbiorów. Jako wynik wyszukiwania zwracany jest zbiór będący wynikiem działania tejże operacji. W wyszukiwaniu tym można dopuścić także wykluczanie pewnych słów, które dokumenty nie powinny zawierać. W takich sytuacjach wykonywana jest operacja różnicy zbiorów.

- wyszukiwanie koncepcyjne – wyszukiwanie to wykorzystuje słownik synonimów (tzn. tezaurus). Gdy użytkownik wpisze słowo do frazy do wyszukiwania, wyszukiwane są dokumenty zawierające zarówno to słowo, jak i jego synonimy. Wyszukiwanie takie można zrealizować na dwa sposoby: po pierwsze, w trakcie tworzenia indeksu można wykorzystać słownik synonimów (patrz: punkt 3.5), bądź po drugie, słownik taki można wykorzystać na etapie wyszukiwania, wyszukując dane słowo oraz jego synonimy, na koniec zwracając sumę zbiorów dokumentów znalezionych dla każdego ze słów.
- szukanie frazy – wyszukiwanie takie polega na wyszukiwaniu dokumentów, w których podana fraza występuje w dokładnie takiej formie, w jakiej została podana. Innymi słowy wyszukiwane są dokumenty zawierające ciągi znaków odpowiadające dokładnie ciągowi znaków utworzonych przez frazę. Funkcjonalność taką realizuje się poprzez wyszukiwanie po kolei zbiorów dokumentów dla poszczególnych słów z frazy, a następnie na wykonywaniu kolejno iloczynu zbiorów i analizowaniu, czy podane słowa występują we właściwej kolejności.
- wyszukiwanie z określeniem odległości między słowami – wyszukiwanie to może mieć dwie postacie. Można wyszukiwać dokumenty, w których podane słowa znajdują się w dokładnie takiej odległości, jaką podał użytkownik, albo w maksymalnie takiej odległości, jaką podał użytkownik. Wyszukiwanie takie realizuje się szukając zbiory dokumentów dla poszczególnych słów oraz na wykonaniu iloczynu zbiorów dla tych dokumentów, a następnie na sprawdzeni każdego dokumentu z osobna pod względem spełniania danego wymagania.
- wyszukiwanie z zastosowaniem masek – wyszukiwanie to pozwala podawać do szukanej frazy niepełne słowa, przy czym brakujące części słów zastępowane są specjalnymi znakami reprezentującymi od jednego do kilku znaków. Najczęściej spotykanymi znakami są: „?” – reprezentujący pojedynczy znak oraz „*” reprezentujący dowolny ciąg znaków. Znaki te mogą występować zarówno na końcu, jak i na początku wyrazu, przez co dopasowywanie słów do podanych wzorców odbywać się może w obie strony. Dlatego też bardzo często spotykanym rozwiązaniem w trakcie budowania indeksu jest przechowywanie wyrazów w postaci „normalnej”, jak i w odwróconej kolejności znaków. Dzięki temu zabiegowi można bardzo szybko i skutecznie dopasowywać wzorce na początku wyrazów. Proces ten przebiega dokładnie w ten sam sposób, co standardowe dopasowywanie wzorców, z tym że odbywa się na wyrazach z odwróconą kolejnością znaków.
- wyszukiwanie dokumentów podobnych do już znalezionych – wyszukiwanie to przebiega nieco inaczej niż standardowe dopasowywanie frazy. Wyszukiwanie to odbywa się przy wykorzystaniu informacji o dokumentach, nie zaś po zawartości słów. Informacjami o dokumentach mogą być np. kategorie tematyczne, podobne zagadnienia, zbliżone tytuły, podobna bibliografia i wiele innych.
- wyszukiwanie dokumentów po statystykach odwiedzin – wyszukiwanie to odbywa się przy wykorzystaniu informacji na temat tego, jakie dokumenty były odwiedzane przez użytkownika w powiązaniu z innymi dokumentami. System zapamiętuje informacje na temat serii dokumentów, jakie odwiedzał użytkownik w powiązaniu z danym zagadnieniem. Dzięki temu, gdy inny użytkownik będzie wyszukiwał dokumenty oraz gdy po znalezieniu odwidzi on jeden z dokumentów odwiedzanych wcześniej przez innego użytkownika, system automatycznie wy-

świetli dokumenty, które ów drugi użytkownik odwiedzał przy okazji odwiedzenia danego dokumentu. Dzięki gromadzeniu takich informacji przez system, wyszukiwanie informacji rozmieszczonych w kilku dokumentach może okazać się o wiele prostszym zadaniem, niż przy „standardowym" wyszukiwaniu.

Wszystkie wyżej przedstawione typy wyszukiwań to jedynie niektóre z możliwych, najczęściej wykorzystywane we współczesnych rozwiązaniach. Istnieje jeszcze wiele innych sposobów wyszukiwania dokumentów, jednak są już one najczęściej specyficzne i dedykowane pod konkretne rozwiązania.

5 Miary jakości wyszukiwania

We wcześniejszych punktach artykułu pojawiały się miary dotyczące jakości wyszukiwania, które nie są zbyt intuicyjne, gdy słyszy się je po raz pierwszy. Dlatego też w niniejszym punkcie miary te zostaną przedstawione i wyjaśnione.

Testowanie jakości wyszukiwania wyszukiwarek odbywa się przy wykorzystaniu pewnego zbioru dokumentów, które arbitralnie uznane zostały za relewantne dla zagadnień, które będą służyć jako testowe zapytania oraz które to dokumentu zostały umieszczone wraz z innymi dokumentami w repozytorium dokumentów. Wszystkie miary jakości wyszukiwania bazują na tychże dokumentach. Nim jednak przejdziemy do właściwego definiowania miar jakości wyszukiwania, wprowadzone zostaną podstawowe oznaczenia wykorzystywane do definiowania poszczególnych miar. I tak oto mamy:

- d_s - znalezione dokumenty przez system
- d_r - dokumenty w bazie uznane za relewantne (arbitralnie)
- DB - liczebność bazy danych

Pierwszą z miar wykorzystywanych do oceny jakości wyszukiwania jest precyzja. Precyzja to stosunek liczby wyszukanych relewantnych dokumentów przez wyszukiwarkę do liczby wszystkich wyszukanych dokumentów. Precyzja wyraża się więc następującym wzorem:

$$(|d_s \cap d_r|) / |d_s| \tag{5.1}$$

Drugą miarą jakości wyszukiwania jest odzysk. Odzysk jest to stosunek liczby wyszukanych relewantnych dokumentów przez wyszukiwarkę do liczby wszystkich dokumentów relewantnych zindeksowanych przez wyszukiwarkę. Miara ta wyraża się następującym wzorem:

$$(|d_s \cap d_r|) / |d_r| \tag{5.2}$$

Obie powyższe miary, tj. precyzja i odzysk są względem siebie przeciwstawne. Im większa precyzja działania wyszukiwarki, tym mniejszy odzysk. Jest to sensowne z logicznego punktu widzenia, gdyż zawężając liczbę zwracanych dokumentów do jedynie tych najbardziej relewantnych, zmniejszamy jednocześnie prawdopodobieństwo, że wszystkie relewantne dokumenty zostaną przez wyszukiwarkę zwrócone. W praktycznych rozwiązaniach oba te parametry dobiera się tak, by wyszukiwarka cha-

rakteryzowała się dużą precyzją przy jednoczesnym zachowaniu dużego odzysku. Jednak proces dostrajania wyszukiwarki przeprowadza się już doświadczalnie.

Kolejną miarą jakości wyszukiwania jest dokładność wyszukiwania. Miara ta wyraża stosunek liczby dokumentów uznanych za relewantne przez wyszukiwarkę w procesie wyszukiwania do liczby wszystkich dokumentów, jakie znajdują się w repozytorium dokumentów. Dokładność wyrażana jest wzorem:

$$(|d_s \cap d_r| + |DB - (d_s \cup d_r)|) / |DB| \tag{5.3}$$

Ostatnim z najczęściej wykorzystywanych miar jakości wyszukiwania jest szum. Miara ta wyraża stosunek liczby wyszukanych nierelewantnych dokumentów przez wyszukiwarkę do liczby wszystkich nierelewantnych dokumentów w repozytorium dokumentów. Szum wyraża się wzorem:

$$|d_s - d_r| / |DB - d_r| \tag{5.4}$$

Wszystkie powyżej przedstawione miary wyszukiwania nie stanowią pełnego zestawu miar jakości wyszukiwania, stosowanych do analizy jakości działania wyszukiwarek. Stanowią one jednak zestaw najważniejszych miar, których znajomość pozwala na całkiem dobrą ocenę jakości działania wyszukiwarki.

6 Język polski a tworzenie wyszukiwarek

Język polski uważa się za wybitnie trudny nie tyle z uwagi na fonetykę, gdyż istnieje wiele języków o bardziej zawikłanej i trudniejszej wymowie, ile raczej z uwagi na jego morfologię. Wydaje się, że w polskiej morfologii nie ma zasad bezwyjątkowych, a więcej, można kwestionować zasadność ogólnie przyjętych praw językowych w odniesieniu do polszczyzny, a już na pewno można dyskutować zasadność wszelkich podziałów i zakres znaczeniowy pojęć gramatycznych (np. pojęcie rodzaju gramatycznego).

Język polski należy do języków fleksyjnych. Większość wyrazów złożona jest z pewnej liczby cząstek zwanych morfenami, z których każde niesie odrębne znaczenie. Sytuacja taka nastręcza wiele problemów twórcom wyszukiwarek dla repozytoriów dokumentów w języku polskim. Przedstawione teraz zostaną niektóre z nich, które stanowią tak naprawdę wierzchołek góry lodowej ogółu problemów, jakie pokonać muszą twórcy wyszukiwarek. Problemami tymi są:

- fleksja – czyli wszelka odmiana wyrazów, począwszy od czasowników, przez rzeczowniki, przymiotniki, liczebniki, aż na zaimkach kończąc. Odmiana ta niesie ze sobą wiele problemów przede wszystkim na etapie indeksowania. Jedno słowo może występować w wielu odmianach, przez co należy wbudować takie mechanizmy w wyszukiwarkę, by wszystkie odmiany danego wyrazu traktowane były jako jeden wyraz, nie zaś jako wiele podobnych, aczkolwiek różnych wyrazów.
- pojęcia wielowyrazowe – w języku polskim występuje wiele pojęć, które składają się z kilku wyrazów i które tylko wtedy, gdy występują razem, znaczą to, co

oznacza całe pojęcie. Jest to kolejny problem dla procesu indeksowania, gdyż wyrazy składające się na pojęcie mogą mieć znaczenie również wtedy, gdy występują samodzielnie. Pojawia się zatem problem na etapie tokenizacji.

- homonimia – czyli różne znaczenia tego samego wyrazu w zależności od kontekstu, w jakim dany wyraz został użyty. Pojawia się tutaj problem przy wyszukiwaniu. Użytkownik, wyszukując dokumenty przy wykorzystaniu tego słowa może skupiać się tylko na jednym z jego znaczeń, w odpowiedzi zaś otrzyma dokumenty zawierające to słowo we wszystkich jego znaczeniach.
- synonimia – czyli opisywanie tego samego pojęcia przy wykorzystaniu wielu różnych wyrazów. Jest to kolejny problem przy wyszukiwaniu. Użytkownik może podawać tylko jedno ze słów, a oczekiwać, że otrzyma dokumenty zawierające również synonimy danego słowa.
- niezgodność semantyki słów z semantyką tekstu – czyli brak analizy składniowej i semantycznej. Innymi słowy pewne ciągi wyrazów mogą łącznie znaczyć zupełnie co innego, niż każde z osobna. Z perspektywy wyszukiwarki układ wyrazów nie ma znaczenia, z perspektywy użytkownika układ wyrazów ma już zasadnicze znaczenie.
- błędy ortograficzne – czyli możliwość pojawienia się teoretycznie tych samych wyrazów, jednak z analizy ciągu znaków w wyrazie zupełnie różnych. Jest to problem dla procesu indeksowania.
- swobodna składania – czyli to samo znaczenie zdania przy wykorzystaniu różnych szyków wyrazów w zdaniu. Dla użytkownika szyk wyrazów w zdaniu jest stosunkowo mało istotny. Dla systemu analizującego układ wyrazów w zdaniu oraz wyciągającego na tej podstawie wniosków może stanowić już bardzo duże wyzwanie.

Przedstawione powyżej problemy to jedynie najważniejsze z ogółu wszystkich występujących. System, od którego wymaga się dużej jakości wyszukiwania musi sprostać jeszcze wielu innym. Jednak nie zostaną one przedstawione w niniejszym artykule.

7 Podsumowanie

Artykuł ten stanowił ogólne wprowadzenie w zagadnienia związane z procesem tworzenia wyszukiwarek dla repozytoriów dokumentów tekstowych. Przedstawione zostały kolejne kroki tworzenia wyszukiwarki oraz ukazane zostały problemy na poszczególnych etapach tego procesu. Oczywiście artykuł ten nie jest dokładnym opisem pełnego procesu budowania wyszukiwarki, a jedynie jego ogólnym ujęciem. Czytelnicy, chcący samodzielnie zbudować wyszukiwarkę dla repozytoriów tekstowych będą zmuszeni do zapoznania się z dokładniejszymi artykułami, a popełnionymi w kontekście poszczególnych etapów procesu budowania wyszukiwarki jako pełnego tematu artykułu. Niemniej, artykuł ten może być uznany za bazę do dalszych prac w kontekście tego tematu.

Bibliografia

1. Christopher D. Manning, Prabhakar Raghavan, Hinrich Schutze – *An Introduction to Information Retrieval*
2. Gerard Salton, Michael J. McGill - *Introduction to Modern Information Retrieval*