Praca-magisterska-przykład-5

PRACA MAGISTERSKA

AUTOMATYCZNA KLASYFIKACJA, GRUPOWANIE I

´ .

ZNAJDOWANIE UOGÓLNIEN DLA J EZYKA¸ POLSKIEGO

´

OSWIADCZENIE AUTORA PRACY
O ´ , ´ ´ ´ -
SWIADCZAM SWIADOMY ODPOWIEDZIALNOSCI KARNEJ ZA POSWIAD
CZENIE NIEPRAWDY,
.

ZE NINIEJSZ A¸ PRAC E¸ DYPLOMOWA¸ WYKONAŁEM

´

´

OSOBISCIE I SAMODZIELNIE,
I NIE KORZYSTAŁEM ZE ZRÓDEŁ INNYCH

. .
NIZ WYMIENIONE W PRACY

. . . . . .

PODPIS

MASTER OF SCIENCE THESIS

AUTOMATIC CLASSIFICATION, CLUSTERING AND SEARCHING FOR GENERALIZATIONS FOR POLISH LANGUAGE.

SUPERVISOR:

Adrian Horzyk Ph.D

Krakow 2011

Serdecznie dzi¸ekuj¸e promotorowi, rodzinie i przyjaciołom za wsparcie podczas pisania tej pracy.

Spis tresci´

1. Wst¸ep

Komunikacja oraz przepływ informacji droga¸ elektroniczna¸ sa¸ nieodzownym elementem współcze-

snego zycia˙. W wielu sytuacjach niezwykle uzyteczna˙ jest automatyzacja tego procesu, nie tylko pod katem¸ wydajnosci´ i bezpieczenstwa,´ ale takze˙ poprawnosci´ lingwistycznej. O ile w przypadku gotowych szablonów wiadomosci´ poprawnos´c´ nie jest problemem, to gdy wymagana jest wi¸eksza interaktywnos´c,´ pojawiaja¸ si¸e kłopoty.
Rozmaite inteligentne czatboty posiadaja¸ wbudowane mechanizmy, które umozliwiaj˙a¸ im wysoce in-

teraktywna¸ komunikacj¸e z uzytkownikiem,˙ która jednak cz¸esto nie jest pozbawiona rozlicznych bł¸edów zarówno składniowych, jak i logicznych. Niniejsza praca podejmuje tematyk¸ automatycznego grupo-wania i klasyfikacji wyrazów w j¸ezyku polskim w celu modelowania uogólnien´ i przechodniosci´ własci´-wosci´ obiektów w hierarchii słów.

Zakres pracy obejmuje zbudowanie specjalistycznego grafu lingwistycznego, agregujacego¸ słowa w

zalezno¨sci´ od stopnia uogólnienia poj¸ec,´ które reprezentuja,¸ na podstawie ich automatycznej klasyfi-kacji oraz grupowania. Do zbudowania tego grafu powinny byc´ wykorzystane rózne¨ korpusy tekstów pozyskiwane z baz danych oraz Internetu. Celem pracy jest zbudowanie takiego grafu oraz pokazanie mozliwo¨sci´ przenoszenia własciwo´sci´ obiektów przez hierarchi¸ klas tego grafu - modelujac¸ mozliwo¨sci´ uogólniania, jakie zachodza¸ w ramach j¸ezyka.

1.1. Cele pracy

Celem pracy jest zbudowanie specjalistycznego grafu przyzwyczajen´ lingwistycznych (ang. Langu-age Habits Graph, LHG)[9], który ukazuje relacje pomi¸edzy wyrazami. W szczególny sposób skupia si¸e

on na uogólnianiu i na jego podstawie klasyfikowaniu słów oraz wyróznianiu¨ ich wspólnych własciwo´-sci´. Zadanie obejmowało stworzenie pajaka¸ internetowego, który przeszukujac¸ wyniki z wyszukiwarki

internetowej buduje baz¸e wiedzy, wokół okreslonego´ przez uzytkownika¨ słowa. Na podstawie uzyska-nych w ten sposób informacji, program tworzy siec´ relacji dla podanego wyrazu. Siec´ ta reprezentowana jest na grafie LHG w uproszczonej postaci, z uwzgl¸ednieniem siły (cz¸estosci´ wyst¸epowania) okreslonych´ połacze¸n´. Zamodelowana została w ten sposób zhierarchizowana struktura, która obrazuje grupowanie obiektów, jak i przechodnios´c´ cech wspólnych wewnatrz¸ danej klasy.

1.2. Zawartos´c´ pracy

1.W rozdziale 1 znajduje si¸e wst¸ep zawierajacy¸ krótka¸ informacj¸e o tematyce i celu pracy.

2.Rozdział 2 opisuje zagadnienie lingwistyki komputerowej i ukazuje problematyk¸ pracy w szerszej perspektywie.

3.Zagadnienia zwiazanych¸ z relacjami wyst¸epujacymi¸ mi¸edzy wyrazami w j¸ezyku polskim zostały opisane w rozdziale 3.

4.Rozdział 4 zawiera szczegółowy opis własnego rozwiazania¸.

5.Rozdział 5 przedstawia wyniki działania aplikacji na konkretnych przykładach.

6.W rozdziale 6 zamieszczone zostało podsumowanie wyników pracy oraz przemyslenia´ i wnioski.

7.Rozdział 7 jest zakonczeniem,´ w którym zestawione sa¸ wyniki pracy z zamierzonymi celami.

Ł. Wróbel

2. Wprowadzenie

2.1. Czym jest lingwistyka?

Definicja 1 Lingwistyka (j¸ezykoznawstwo) jest to dział nauk humanistycznych badajacy¸ istot¸e, budow¸e i rozwój j¸ezyka[22].

J¸ezyk mówiony i pisany od wieków był przedmiotem badan´ i rozwaza˙n´ naukowych. Wi¸ekszos´c´ wysiłków koncentrowała si¸e jednak na zgł¸ebianiu jego ewolucji poprzez analiz¸e dzieł literackich, czy tez˙

poszukiwaniu podobienstw´ pomi¸edzy róznymi˙ j¸ezykami i dialektami. Dopiero w XX wieku zacz¸eło si¸e kształtowac´ bardziej ustrukturalizowane i sformalizowane podejscie´ do tego zagadnienia[22].

Zasadniczo lingwistyka dzieli si¸e na teoretyczna¸ i stosowana¸. J¸ezykoznawstwo teoretyczne skupia si¸e na badaniu aspektów współczesnej odmiany konkretnego j¸ezyka i wyszukiwaniu oraz opisywaniu jego ogólnych zasad. Przedmiotem rozwaza˙n´ moga¸ byc´ tez˙ ogólne reguły i aspekty wspólne dla wszyst-

kich j¸ezyków. Stosowana lingwistyka podejmuje problemy zwiazane¸ z wykorzystaniem wyników tych rozwaza˙n´ w innych dziedzinach. Bada j¸ezyk w szerszej perspektywie zarówno w porównaniu z innymi, jak i jego własnym kształtowaniem si¸e przez lata[22].
Współczesne j¸ezykoznawstwo korzystajace¸ z najnowszej technologii dało poczatek¸ informatyzacji

tej nauki. Otworzyła ona nie tylko nowe mozliwo˙sci,´ ale i liczne wyzwania zwiazane¸ z elektroniczna¸

komunikacja¸ mi¸edzy lud´zmi, a przede wszystkim człowieka z komputerem. Tak narodziła si¸e lingwistyka komputerowa.

2.2. Lingwistyka komputerowa

Definicja 2 Lingwistyka komputerowa to dział lingwistyki uzywaj¨acy¸ modeli komputerowych w celu te-

stowania hipotez dotyczacych¸ mowy i j¸ezyka oraz tworzenia programów komputerowych przetwarzaja¸-cych j¸ezyk naturalny[23].

Ta ogólna definicja ukazuje jak bardzo lingwistyka komputerowa zakorzeniona jest w rozmaitych dziedzinach j¸ezykoznawstwa. Zakres jej zastosowan´ jest jednak o wiele wi¸ekszy, gdyz¨ obecna jest takze¨ mi¸edzy innymi w filozofii, psychologii, informatyce (np. sztuczna inteligencja) czy matematyce (np. rachunek prawdopodobienstwa,´ teoria automatów, logika matematyczna, statystyka, czy j¸ezyki formalne)[23].

Dziedzin¸ ta¸ mozna¨ podzielic´ na cz¸es´c´ teoretyczna¸ i praktyczna¸. Zagadnienia teoretyczne skupiaja¸

si¸e na badaniu mozliwo¨sci´ automatyzacji procesów zwiazanych¸ z klasyfikacja¸ podzbiorów j¸ezykowych, obliczaniu wydajnosci´ tych procesów, a takze¨ tworzeniu ich formalnych opisów. Praktyczne podejscie´

do lingwistyki komputerowej owocuje natomiast powstawaniem algorytmów i złozonych¨ modeli auto-

matyzujacych¸ i przetwarzajacych,¸ które znajduja¸ zastosowanie w rozmaitych aspektach j¸ezykowych[23].

Rozwój i cele stojace¸ przed lingwistyka¸ komputerowa¸ sa¸ sci´sle´ zwiazane¸ z rozwojem technologii,

jak i z rosnacymi¸ potrzebami automatyzacji i formalizacji procesów j¸ezykowych, które wykraczaja¸ poza mozliwo¨sci´ człowieka. Coraz nowoczesniejsze´ maszyny z duz¨a¸ moca¸ obliczeniowa¸ sa¸ w stanie realizo-wac´ nawet bardzo złozone¨ i jeszcze niedawno zbyt kosztowne algorytmy, co z kolei sprawia, ze¨ zagad-nienia zwiazane¸ ze sztuczna¸ inteligencja¸ sa¸ obecnie bardzo popularne.

2.3. Historia lingwistyki

Aby dobrze zrozumiec´ jak wielki post¸ep dokonał si¸e w dziedzinie lingwistyki komputerowej warto zagł¸ebic´ si¸e w histori¸e jej rozwoju. Daje to mozliwo¨s´c´ zaobserwowania trendów i zagadnien,´ które były, sa¸ i b¸eda¸ przedmiotem badan´ współczesnego j¸ezykoznawstwa informatycznego.

Poczatki¸ lingwistyki jako nauki o j¸ezyku si¸egaja¸ czasów starozytnych¨. Za jej kolebk¸ uwaza¨ si¸e In-

die, gdzie juz¨ w pierwszym tysiacleciu¸ p.n.e. dokonywano tłumaczen´ starozytnych¨ tekstów religijnych. Za twórc¸e pierwszej gramatyki uwaza¨ si¸e Paniniego, który około V-IV w. p.n.e. stworzył szereg poj¸ec´ i definicji oraz około 4 tysiace¸ reguł j¸ezykowych. Niektóre powstałe wtedy terminy takie jak rozróz¨-nienie

mi¸edzy głoskami, a literami, fleksja, słowotwórstwo, struktura wyrazu, czy klasyfikacja cz¸esci´

mowy uzywane˙ sa¸ do dnia dzisiejszego. Wszelkie wyjatki,¸ które licznie pojawiaja¸ si¸e w j¸ezykach, takich jak łacina, czy greka Panini w swojej gramatyce opisał za pomoca¸ bardzo szczegółowych reguł, które opisywały nawet pojedyncze przypadki[14].
W starozytnej˙ Grecji i Rzymie skupiano si¸e bardziej na ogólnych problemach j¸ezyka, takich jak jego istota, stosunek do logiki, retoryki i poetyki. Ukształtowały si¸e dwa odmienne punkty widzenia. Ana-logizm, który zakładał, ze˙ j¸ezyk ma charakter logiczny oraz anomalizm, który uznawał ze˙ j¸ezyk jest zmienny. Z tego okresu wywodza¸ si¸e poj¸ecia etymologii (nauka o pochodzeniu i ewolucji znaczenia oraz formy wyrazów), teorii nominacji (nazywania swiata´ za pomoca¸ słów), onomatopei (d´zwi¸ekonasladow´-nictwa). Szczególnie rozwin¸eło si¸e takze˙ j¸ezykoznawstwo opisowe. Powstał podział na cz¸esci´ mowy, opisano deklinacje, koniugacje i zasady opisu składni[14].

Dopiero w XVIII w. gramatyka j¸ezyka łacinskiego´ przestała byc´ powszechnie uwazana˙ za uniwer-salna,¸ do czego przyczyniły si¸e badania Abu Hayan at Tauhaidi'ego, który postulował, ze˙ na proces

kształtowania j¸ezyka maja¸ wpływ kultura i historia narodu. Zapoczatkowało¸ to pojawianie si¸e filologii narodowych: m.in. francuskiej, niemieckiej, angielskiej i słowianskiej[14]´.

XIX w. to rozwój j¸ezykoznawstwa historycznego, który rozpocz¸eło odkrycie przez William'a Jones'a

sanskrytu (starozytnego˙ j¸ezyka Indii). Jego znaczne podobienstwo´ do j¸ezyków europejskich zainicjowało badania nad komparatywistyka¸ (lingwistyka¸ porównawcza)¸. Okreslone´ zostały regiony i narody, których formy komunikacji maja¸ wspólne korzenie historyczne, tworzac¸ tak zwane rodziny j¸ezykowe[20].

Wiek XX to okres strukturalizmu i formalizacji zasad j¸ezykowych. Zamiast do genezy zacz¸eto przywiazywa¸c´ wi¸eksza¸ wag¸e do opisu struktury. F. de Saussure uwazał,˙ ze˙ kazdy˙ znak zbudowany

jest z dwóch elementów: znaczacego¸ (jego reprezentacja np. obraz, czy słowo) i znaczonego (poj¸e-

cie z nim zwiazane)¸. J¸ezyk zdefiniowano wi¸ec jako dwuklasowy system semiotyczny, który słuzy˙ do komunikacji[19]. Powstał takze˙ pierwszy matematyczny opis j¸ezyka, który za pomoca¸ formalnych na-rz¸edzi matematycznych takich jak dowody, twierdzenia i rachunki stworzył drzewiasta¸ struktur¸e wszyst-kich wyraze˙n´ poprawnych w danej gramatyce. N. Chomsky wprowadził gramatyk¸e transformacyjno-generatywna,¸ która jest systemem okreslaj´acym¸ skonczon´a¸ ilos´c´ reguł. Z ich pomoca¸ rozmówca jest w stanie utworzyc´ poprawne zdanie, którego nigdy wczesniej´ nie słyszał, a odbiorca stwierdzic,´ czy zdanie, które usłyszał jest poprawne w danym j¸ezyku, czy nie[13].

Formalizacja zasad lingwistycznych stworzyła podwaliny pod powstanie j¸ezykoznawstwa informa-

tycznego, które wykorzystujac¸ sci´sle´ okreslone´ reguły umozliwiło˙ automatyczna¸ analiz¸e i przetwarzanie tekstu, wyszukiwanie, a takze˙ budowanie poprawnych form j¸ezykowych przez programy komputerowe. Poczatkowo¸ w lingwistyce komputerowej dominowały dwa podejscia´. Pierwsze oparte było na meto-dach symbolicznych (teoria generatywna, sztuczna inteligencja), natomiast drugie bazowało na metodach statycznych takich jak analiza tekstu, rozpoznawanie liter itp[6]. Powstały modele statystyczne słuz˙ace¸ do syntezy mowy, j¸ezyki programowania oparte o logik¸e formalna¸ i metody wnioskowania (np. Prolog), a takze˙ aplikacje sterowane komendami w j¸ezyku naturalnym.

Lata 90 to dynamiczny rozwój Internetu oraz mocy obliczeniowej i pojemnosci´ pami¸eciowej kom-puterów. Pojawiaja¸ si¸e złozone˙ aplikacje wykorzystujace¸ zaawansowane algorytmy oparte m.in. o sta-tystyk¸e. W

odpowiedzi na zapotrzebowanie na programy zajmujace¸ si¸e ekstrakcja¸ danych z Internetu,

czy korpusów tekstów oraz narz¸edzi przetwarzajacych¸ i analizujacych¸ tekst z wykorzystaniem metod sztucznej inteligencji powstało wiele rozwiaza¸n´ komercyjnych[6].

2.4. Problemy i cele współczesnej lingwistyki komputerowej

Najwi¸ekszym problemem, przed jakim staje współczesne j¸ezykoznawstwo komputerowe jest wielo-

znacznos´c[6]. Dotyczy on fonetyki (homofonia - fonetyczna tozsamo˙s´c´ róznych˙ znaków j¸ezykowych),

morfologii (homonimia - posiadanie róznych˙ znaczen´ przez identyczne formy j¸ezykowe), składni (niejednoznacznos´c´ przy okreslaniu´ cz¸esci´ zdania), semantyki (idiomy, polisemia - posiadanie przez jedno

słowo wielu znaczen),´ pragmatyki (metafory, ironia). O ile człowiek korzystajac¸ z wyzszego˙ poziomu j¸ezyka moze˙ w łatwy sposób rozwiaza¸c´ te problemy, to juz˙ komputer, ma z tym spore kłopoty.

Według Bonnie Webber (2001) dwa podstawowe cele jakie stoja¸ przed lingwistyka¸ informatyczna¸ to[11]:

modelowanie ludzkiego rozumienia i generacji j¸ezyka naturalnego jako systemu procesów prze-

twarzajacych¸ informacj¸e (lingwistyka informatyczna)

wyposazenie˙ komputerów w mechanizmy analizy i generowania j¸ezyka naturalnego w celu do-

starczenia uzytecznej˙ usługi (stosowane przetwarzanie j¸ezyka naturalnego - ang. applied natural language processing, inzynieria˙ j¸ezyka naturalnego - ang. natural language engineering, techno-logia j¸ezykowa - ang. language technology)

Roland Hausser (2001) uwaza,˙ ze˙ lingwistyka komputerowa powinna si¸e skupiac´ na modelowaniu komunikacji człowieka z komputerem, konstruujac¸ modele wyjasniaj´ace¸ naturalny przekaz informacji w sposób spójny funkcjonalnie, precyzyjny matematycznie i efektywny obliczeniowo[11].

Z punktu widzenia współczesnych wymagan´ najintensywniejsze prace prowadzone sa¸ nad[11]:

aplikacja¸ umozliwiaj˙ac¸a¸ wyszukiwanie specyficznych informacji na podstawie dokładnie opisa-nych wymagan´ co do rezultatów

systemem potrafiacym¸ uczyc´ si¸e ze zródeł´ przyswajalnych przez człowieka, a nast¸epnie potrafia¸-

cym rozwiaza¸c´ test sprawdzajacy¸ zdobyta wiedz¸e, jaki rozwiazałby¸ człowiek

systemem do automatycznego tłumaczenia na inne j¸ezyki naturalne, a takze˙ j¸ezyki nieistniejace,¸ ale opisane okreslonym´ zestawem reguł

aplikacja¸ umozliwiaj˙ac¸a¸ komunikacj¸e głosowa¸ człowieka z komputerem

Niewyczerpanym zródłem´ wiedzy dla programów wykorzystujacych¸ sztuczna¸ inteligencj¸ do wyszu-kiwania informacji jest Internet. Ogromu zawartych w nim informacji człowiek nie jest w stanie prze-tworzyc,´ natomiast posiadajaca¸ odpowiedni zestaw reguł oraz sposób wartosciowania´ i ekstrakcji danych maszyna moze˙ podja¸c´ si¸e tego zadania. O ile juz˙ samo stworzenie narz¸edzia do inteligentnego pozyski-wania informacji nie jest zadaniem prostym, to dodatkowe trudnosci´ pojawiaja¸ si¸e podczas przetwarzania danych. System, aby wykorzystac´ pozyskana¸ wiedz¸ w okreslonym´ celu, musi prawidłowo rozumiec´ za-równo posiadane dane, jak i cel ich wykorzystania. W szczególnosci´ stanowi to ogromny problem, gdy zadaniem jest przetłumaczenie tekstu na inny j¸ezyk. Znajomos´c´ reguł tworzenia poprawnych form nie gwarantuje wiernosci´ przekazu. Przykładem moze˙ byc´ napisana przed wieloma laty rosyjska aplikacja, która tłumaczyła wyrazenia˙ z j¸ezyka angielskiego na rosyjski i pó´zniej znów na angielski. Zamieniła ona

wyrazenie˙ „out of sight, out of mind” (co z oczu, to z serca) na „invisible idiot” (niewidzialny idiota)

-out of mind zostało przetłumaczone jako „niespełna rozumu”, gdyz˙ system nie zrozumiał semantyki

całego wyrazenia[1]˙.

Cele bezposrednio´ zwiazane¸ z przedmiotem tej pracy to wyszukiwanie i wyodr¸ebnianie informacji, ich analiza i na jej podstawie ukazanie zasad modelowania wybranej dziedziny j¸ezyka. Nast¸epne pod-

rozdziały to ogólny opis narz¸edzi i technik wykorzystywanych do rozwiazywania¸ tych problemów wraz

z przykładami istniejacych¸ implementacji.

Ł. Wróbel

2.5. Pajak¸ internetowy

2.5. Pajak¸ internetowy

Definicja 3 Pajak¸ (robot) internetowy (ang. Web crawler, spider bot) to program przeszukujacy¸ zawar-tos´c sieci Internet w zorganizowany i zautomatyzowany sposób[24].

Rysunek 2.1: Schemat działania pajaka˛ internetowego

Podstawowym zadaniem wszelkiego rodzaju robotów internetowych jest automatyczna nawigacja po

stronach internetowych w sposób okreslony´ przez zadane reguły. Kazdy˙ pajak¸ ma poczatkowo¸ okreslon´a¸

pul¸e adresów, od której zaczyna przegladanie¸. Odwiedzajac¸ kolejne strony przenosi adresy z tej listy na list¸e adresów juz˙ odwiedzonych. Na witrynie, przez która¸ przechodzi moze˙ szukac´ nowych linków, które

dodaje do listy „do odwiedzenia”, pod warunkiem, ze˙ nie ma ich jeszcze na zadnej˙ z list. Robot działa dopóki na liscie´ „do odwiedzenia” znajduje si¸e przynajmniej jeden adres. Specyficzne pajaki¸ moga¸ oczy-

wiscie´ przechodzic´ przez te same strony wielokrotnie, lub działac´ w nieskonczonej´ p¸etli, aby okresowo patrolowac´ okreslone´ witryny[24].
Po wejsciu´ na stron¸e robot sprawdza znajdujacy¸ si¸e na serwerze plik robots.txt, który okresla´ do ja-kiej cz¸esci´ serwisu ma dost¸ep. Zawartos´c´ pliku to spis plików i katalogów opisana za pomoca¸ protokołu Robot Exclusion Protocol, niedost¸epnych dla pajaka¸. Administrator strony moze˙ w ten sposób ukryc´ cz¸es´c´ serwisu przed wszelkimi automatami, lub tez˙ całkowicie odmówic´ im dost¸epu, aby na przykład nie obcia¸zały˙ łacza¸. Plik ten jest jednak publicznie dost¸epny, wi¸ec zapisywanie w nim informacji na temat niepublicznych danych nie jest dobrym pomysłem. Roboty internetowe (w szczególnosci´ wszel-kiego rodzaju malware, harvestery i spam boty) moga¸ po prostu ignorowac´ plik robots.txt, wi¸ec nie jest on zabezpieczeniem absolutnym. Informacje dla pajaka¸ moga¸ znajdowac´ si¸e takze˙ w meta tagach, które okreslaj´a¸ zasady zachowania robota na danej stronie. Najcz¸esciej´ stosowane tagi to NOINDEX

(uniemozliwia˙ indeksowanie zawartosci´ strony) i NOFOLLOW (uniemozliwia˙ bezposrednie´ poda¸zanie˙

za linkami znajdujacymi¸ si¸e na stronie)[16].

Pajaki¸ internetowe oprócz odwiedzania witryn słuz˙a¸ przede wszystkim do zbierania informacji na nich umieszczonych. Wyszukiwarki internetowe wykorzystuja¸ rozmaite boty do indeksowania i spraw-dzania aktualnej zawartosci´ stron w sieci, aby wykorzystac´ uzyskane w ten sposób dane do zwrócenia jak najlepszej listy wyników. Cz¸esto na potrzeby pobierania danych innych niz˙ linki sprawdzany jest takze˙ j¸ezyk strony, czy kodowanie znaków, opisane w nagłówku, aby niepotrzebnie nie czytac´ całej tresci[24]´.
Kilka przykładowych istniejacych¸ implementacji robotów internetowych:

Googlebot

Jest to pajak¸ wykorzystywany przez popularna¸ wyszukiwark¸e do indeksowania stron w internecie. Działa on w dwóch wersjach: deep crawl (w¸edruje po witrynach gromadzac¸ i odwiedzajac¸ jak najwi¸ecej znalezionych adresów) oraz fresh crawl (odwiedza zmieniajace¸ si¸e cz¸esto strony, w celu

aktualizacji zwiazanych¸ z nimi informacji). Aby Googlebot mógł odwiedzic´ dana¸ stron¸e, musi istniec´ do niej link na innej, przez która¸ juz˙ przechodził. Gromadzi on takze˙ statystyki takie jak ilos´c´ wystapie¸n´ adresu danej strony, która nast¸epnie okresla´ jej pozycj¸e w wynikach wyszukiwa-

nia. Najwi¸ekszym problemem zwiazanym¸ z Googlebotem jest wysokie wykorzystanie transferu, co moze˙

doprowadzic´ nawet do czasowego zawieszenia strony z powodu przekroczenia ograni-czen´ transferowych. Google udost¸epnia administratorom narz¸edzie Webmaster Tools, za pomoca¸ którego mozna˙ ustawic´ m.in. cz¸estotliwos´c´ wizyt pajaka¸ na stronie[22].

WebCrawler

Metawyszukiwarka internetowa agregujaca¸ najlepsze wyniki z google, yahoo, ask.com, bing search i innych popularnych wyszukiwarek. Umozliwia˙ wyszukiwanie takze˙ obrazków, plików au-

dio i wideo. WebCrawler był pierwsza¸ wyszukiwarka¸ umozliwiaj˙ac¸a¸ szukanie w pełnym tekscie´ dokumentów, a nie tylko indeksowanie[25].

Rysunek 2.2: Graficzny interfejs metawyszukiwarki WebCrawler.

IBM Web Fountain

Web Fountain to jeden z pierwszych projektów, których celem jest skatalogowanie i interpreta-cja nieustrukturalizowanych lub cz¸esciowo´ ustrukturalizowanych danych tekstowych z Internetu. Jego głównym zadaniem jest poszukiwanie wzorców i zalezno˙sci´ na potrzeby statystyczne oraz prezentacji trendów w czasie rzeczywistym[21].

Rysunek 2.3: Graficzny interfejs aplikacji Web Fountain.

WGet

Jest to program pobierajacy¸ zasoby z serwerów internetowych b¸edacy¸ cz¸esci´a¸ projektu GNU. Wspiera on obecnie transfer za pomoca¸ protokołów HTTP, HTTPS i FTP. Umozliwia˙ pobiera-

nie rekursywne, konwersj¸ linków umozliwiaj˙ac¸a¸ przegladanie¸ lokalnych plików HTML w try-bie offline oraz wspiera proxy. WGet był jednym z pierwszych programów wykorzystujacych¸ nagłówek HTTP do wznawiania przerwanego transferu. Pajak¸ internetowy wykorzystywany jest przede wszystkim do pobierania rekursywnego, które dokładnie odwzorowuje struktur¸e zasobów na serwerze. Domyslnie´ WGet bierze pod uwag¸e informacje zawarte w pliku robots.txt, ale mozna˙ go tez˙ uruchomic´ z parametrem -e robots=off. Istnieje takze˙ mozliwo˙s´c´ pobierania tylko plików nowszych niz˙ zapisane lokalnie, a takze˙ filtrowania zasobów po nazwach za pomoca¸ wyraze˙n´ regularnych[2].

Ł. Wróbel

2.6. Ekstrakcja danych

Rysunek 2.4: Wynik działania programu WGet.

Crawler4j

Jest to prosty, open source'owy interfejs w javie, który pozwala bardzo łatwo i szybko stworzyc´ własnego pajaka¸ internetowego. Udost¸epnia dwie metody do przeładowania: shouldVisit (okresla´ kryteria definiujace,¸ czy dany URL powinien zostac´ odwiedzony,czy nie) i visit (okresla´ co ma zostac´ zrobione w momencie odwiedzenia strony). Dodatkowo nalezy˙ zdefiniowac´ kontroler, który

okresli´ poczatkowy¸ zbiór adresów, miejsce zapisu danych, czy ilos´c´ równoległych watków[3]¸.

2.6. Ekstrakcja danych

Znalezienie strony, na której moga¸ byc´ zamieszczone poszukiwane informacje, to dopiero połowa

procesu ich pozyskiwania. Sa¸ one zazwyczaj otoczone kodem html, który ułatwia uzytkownikowi˙ ich

przegladanie¸ z uzyciem˙ przegladarki,¸ ale komplikuje proces ekstrakcji własciwego´ tekstu przez program.

Wydobywanie danych moze˙ odbywac´ si¸e na dwa sposoby. Pierwszy zakłada, ze˙ uzytkownika˙ interesuje konkretna informacja, której program poszukuje z wykorzystaniem np. słów kluczowych, czy wyraze˙n´

regularnych. Drugi sposób to ekstrakcja całosci´ tekstu w celu zbudowania przez uzytkownika˙ bazy wie-dzy, z punktu widzenia której istotne jest pozyskanie jak najwi¸ekszej ilosci´ danych. Baz¸e ta¸ wykorzystuje mi¸edzy innymi data mining, który na jej podstawie wyszukuje trendy i formułuje predykcje przyszłych zmian.

Aby uzyskane w ten sposób informacje mogły zostac´ dalej przetwarzane nie moga¸ zawierac´ elemen-tów j¸ezyka html. Niezb¸edny jest zatem parser, który znajdzie i usunie wszystkie tagi html, czy skrypty

j¸ezyka JavaScript zostawiajac¸ tylko istotne informacje. Teoretycznie mozliwe˙ jest wykorzystanie wyra-

zenia˙ regularnego, znajdujacego¸ niepoz˙adane¸ elementy, lecz moze˙ ono nie uwzgl¸ednic´ cz¸estych bł¸edów w strukturze stron, jak np. znak < wewnatrz¸ taga, czy tez˙ fragmenty tekstu przypominajace¸ tagi (równa-

nia matematyczne, kod w róznych˙ j¸ezykach programowania).

Przykłady ekstraktorów danych i parserów html:

Online Data Extractor

Jest to program, który pozwala na wyszukiwanie danych kontaktowych głównie na potrzeby firm, takich jak adresy e-mail, numery telefonu, czy fax, a takze˙ adresy URL i meta tagi). Informacje

te znajdowane sa¸ z pomoca¸ pajaka¸ internetowego przegladaj¸acego¸ okreslone´ strony, a takze˙ ko-

rzystajacego¸ z pomocy popularnych wyszukiwarek. Umozliwia˙ eksport zgromadzonych danych mi¸edzy innymi do plików txt, xls, csv i htm, a takze˙ pozwala na zaawansowana¸ integracj¸ danych do celów data miningu[10].

Rysunek 2.5: Zrzut ekranu obrazujacy¸ działanie aplikacji Online Data Extractor.

HTML Text Extractor

HTML Text Extractor to komercyjne narz¸edzie działajace¸ pod Windowsem, które umozliwia˙ wy-dobywanie kodu html dowolnej strony internetowej, w tym takze˙ tych, w których zablokowany jest podglad¸ zródła,´ czy tez˙ sa¸ zaszyfrowane. Ekstrakcja nast¸epuje w momencie odwiedzenia strony i daje mozliwo˙s´c´ wyodr¸ebnienia, zaznaczenia i skopiowania samego tekstu, czy to z całej witryny, czy tez˙ jej fragmentu[4].

Rysunek 2.6: Zrzut ekranu obrazujacy¸ działanie aplikacji HTML Text Extractor.

Na potrzeby aplikacji zajmujacych¸ si¸e przetwarzaniem i analiza¸ j¸ezyka naturalnego, a takze˙ pogł¸e-biania wiedzy na temat kultury i literatury tworzy si¸e korpusy tekstów, które zawieraja¸ tekst pokazujacy¸ wyszukiwane wyrazy i konstrukcje j¸ezykowe zastosowane w rzeczywistym kontekscie´. Cz¸esto tez˙ do

korpusów udost¸epnionych online dołaczone¸ sa¸ mniej lub bardziej zaawansowane narz¸edzia wyszuku-jace¸. Popularne zródła´ korpusów tekstów to m.in.:

PELCRA

Projekt PELCRA realizowany przez Katedr¸ J¸ezyka Angielskiego Uniwersytetu Łódzkiego przy współpracy z Departamentem J¸ezykoznawstwa i Współczesnego J¸ezyka Angielskiego Uniwersy-tetu w Lancaster opracowuje korpusy tekstów w j¸ezyku polskim i angielskim na potrzeby badan´ m.in. nad leksykografia,¸ translatoryka¸ i lingwistyka¸ (zarówno ogólna¸ jak i komputerowa)[12]¸.

Korpusy wchodzace¸ w skład projektu PELCRA:

– Narodowy Korpus J¸ezyka Polskiego - Jeden z najwi¸ekszych i najpopularniejszych korpu-sów j¸ezyka polskiego współtworzony przez Instytut Podstaw Informatyki PAN, Uniwersytet Łódzki, Wydawnictwo Naukowe PWN, oraz Instytut J¸ezyka Polskiego PAN.

– Korpus referencyjny j¸ezyka polskiego PELCRA - Zawiera on około 93 miliony segmen-tów słów tekstu. Jego cz¸esci´a¸ jest dost¸epny zarówno w wersji tekstowej jak i audio Korpus Polszczyzny Konwersacyjnej, który składa si¸e z nagranych rozmów przypadkowych osób uzupełnionych o dane dotyczace¸ uczestników konwersacji.

– ENHIG - Korpus historyczny zawierajacy¸ teksty staroangielskie, oznakowane pod wzgl¸edem składniowym próbki poezji, prozy i tłumaczen´ stworzony na potrzeby prowadzenia badan´ nad szykiem zdania.

Rysunek 2.7: Kolokacje wyrazu niebo na podstawie Narodowego Korpusu J¸ezyka Polskiego.

IPI PAN

Korpus Polskiej Akademii Nauk zawiera obecnie około 250 milionów segmentów i dost¸epny jest darmowo. Do wyszukiwania służy˙ dost¸epny na licencji GPL, dedykowany program Poliqarp[5].

Wydawnictwo Naukowe PWN

Wydawnictwo PWN udost¸epnia swój korpus j¸ezykowy odpłatnie. Zawiera on ponad 40 milionów słów z fragmentów 386 róznych˙ ksia¸zek,˙ 977 numerów 185 róznych˙ gazet i czasopism, 84 nagra-nych rozmów, 207 stron internetowych oraz kilkuset ulotek reklamowych. Wersja demonstracyjna dost¸epna darmowo składa si¸e z 7.5 miliona słów[26].

3. Relacje mi¸edzy wyrazami w j¸ezyku polskim

3.1. Cz¸esci´ mowy

Głównie ze wzgl¸edów praktycznych wyrazy w j¸ezyku polskim podzielono na klasy j¸ezykowe zwane potocznie cz¸esciami´ mowy. Podział ten zalezny˙ jest od szeregu własciwo´sci´ fleksyjnych, semantycznych

i składniowych leksemów. Z punktu widzenia morfologii istotne jest, czy przyporzadkowane¸ do danej klasy wyrazy sa¸ odmienne i w jaki sposób. Klasyfikacja semantyczna wprowadza natomiast podział słów na samoznaczace¸ (autosemantyczne) i nie majace¸ znaczenia (synsemantyczne)[18].

Definicja 4 Cz¸esci´ mowy to kategorie gramatyczno-leksykalne, które[18]:

maja¸ te same cechy morfologiczne tzn. kazdy˙ leksem nalez˙acy¸ do danej cz¸esci´ mowy ma podobna¸ budow¸e, podobnie si¸e odmienia np. rzeczowniki przez przypadki i liczby, przymiotniki przez ro-dzaje, przypadki i liczby,

pełnia¸ identyczna¸ funkcj¸e składniowa¸ tzn. wyst¸epuja¸ w roli orzeczenia, podmiotu, przydawki itd., maja¸ podobne znaczenie lub niosa¸ podobna¸ tres´c ogólna¸ np. nazywaja¸ zjawiska atmosferyczne, procesy, wielkosci,´ cechy, liczby, stany itd.,

wchodza¸ w okreslone´ relacje (zwiazki)¸ z innymi wyrazami np. przysłówek z czasownikiem tworza¸ zwiazek¸ przynalezno˙sci´.

Tradycyjny podział wyróznia˙ dziesi¸e´c cz¸esci´ mowy[18]:

Rzeczownik Przysłówek Przymiotnik Przyimek Liczebnik Czasownik Spójnik

Zaimek Partykuła

Wykrzyknik

Na potrzeby pracy wykorzystane zostały tylko te cz¸esci´ mowy, które uznane zostały za niosace¸ naj-wi¸ecej istotnych informacji oraz pełniace¸ wazne˙ funkcje w zdaniu, a przede wszystkim bezposrednio´ wykorzystywane w procesie kojarzenia. Z punktu widzenia kognitywistyki najwi¸eksze znaczenie maja¸ wyrazy, z którymi bezposrednio´ mozna˙ skojarzyc´ okreslony´ obraz oraz takie, za pomoca¸ których mozna˙ obraz ten opisac´.

3.2. Relacje mi¸edzy wyrazami

3.2. Relacje mi¸edzy wyrazami

Aby lepiej zrozumiec´ prawa, jakie rzadz¸a¸ relacjami mi¸edzy słowami w j¸ezyku polskim, nalezy˙ naj-pierw wprowadzic´ definicje poj¸ec´ takich jak zdanie i składnik, które sa¸ niezb¸edne z punktu widzenia analizy składniowej.

Definicja 5 Zdanie to odcinek tekstu od wielkiej litery do kropki lub znaku jej równowaznego˙ (pytajnika, wykrzyknika). W odniesieniu do formy mówionej jest to pojedynczy odcinek intonacyjny.[27]

Definicja 6 Składnik to najmniejszy element, jaki da si¸e wyodr¸ebni´c w procesie analizy składniowej, mogacy¸ pełni´c jako całos´c funkcj¸e składniowa¸ i wchodzi´ w zwiazki¸ z innymi takimi elementami.[27]

Najcz¸esciej´ mamy do czynienia z sytuacja,¸ w której składnik pokrywa si¸e z wyrazem. Bywaja¸ jednak

przypadki, gdy ta funkcja składniowa charakteryzuje połaczenie¸ wyrazów np. na pewno, niech b¸edzie,

Jan Kowalski. Takie połaczenie¸ nazywane jest składnikiem złozonym˙.

Wyróżniamy˙ trzy zasadnicze kategorie grup syntaktycznych: podrz¸edne, współrz¸edne i

egzocentryczne.[15] W grupie podrz¸ednej członem nadrz¸ednym jest składnik, którego nie mozna˙

z niej usuna¸c,´ natomiast człon podrz¸edny niesie dodatkowe informacje, ale nie jest niezb¸edny z punktu

widzenia poprawnosci´ składniowej wyrazenia˙ w zdaniu[15]. Przykładowo dla zdania Babcia upiekła pyszne ciasteczka. fraza pyszne ciasteczka jest grupa¸ podrz¸edna,¸ gdzie składnikiem nadrz¸ednym jest wyraz ciasteczka, a podrz¸ednym pyszne, gdyz˙ zdanie Babcia upiekła ciasteczka. jest poprawne, a Babcia upiekła pyszne. juz˙ nie. Ten rodzaj grupy składniowej wykorzystany został w pracy, gdyz˙ niesie on

najwi¸ecej uzytecznych˙ informacji z punktu widzenia relatywnego opisywania rzeczywistosci´.

Grupa współrz¸edna zwana takze˙ szeregiem charakteryzuje si¸e tym, ze˙ składniki sa¸ równorz¸edne

i mozna˙ ja¸ zredukowac´ do jednego składnika (nie b¸edacego¸ spójnikiem)[15]. Przykładowo w zdaniu Babcia upiekła ciasteczka dla mnie i brata. szeregiem b¸edzie fragment mnie i brata, gdzie współrz¸ednymi składnikami sa¸ wyrazy mnie oraz brata, a i jest spójnikiem. Poprawne sa¸ wi¸ec zdania Babcia upiekła ciasteczka dla mnie. oraz Babcia upiekła ciasteczka dla brata., ale Babcia upiekła ciasteczka dla i. juz˙ nie.

Trzecia grupa zwana egzocentryczna¸ składa si¸e z dwóch składników, z których zaden˙ nie jest reduko-walny, a składnik nadrz¸edny wyznacza form¸e podrz¸ednego[15]. Przykładem niech b¸edzie zdanie Babcia upiekła dla nas ciasteczka., gdzie grup¸e egzocentryczna¸ tworza¸ słowa dla nas, gdyz˙ ani zdanie Bab-cia upiekła nas ciasteczka., ani Babcia upiekła dla ciasteczka. nie jest poprawne, a wyraz dla definiuje przypadek zaimka my.

Definicja 7 Akomodacja syntaktyczna (łac. accomodare - przystosowywa´c) oznacza dostosowanie si¸e do siebie jednostek w zdaniu. Dostosowanie to polega najcz¸esciej´ na przyjmowaniu konkretnych wartosci´ kategorii gramatycznych (takich jak rodzaj, liczba, przypadek) przez jednostk¸ wchodzac¸a¸ w konstrukcj¸e z inna¸ jednostka¸.[27]

Akomodacje syntaktyczne podzielic´ mozna˙ na jednostronne, gdy składnik nadrz¸edny wymaga od

podrz¸ednego dostosowania formy fleksyjnej, wzajemne, gdy kazdy˙ ze składników ma wpływ na form¸e drugiego, a takze˙ mi¸edzyfrazowe, w których o formie podrz¸ednika decyduje nadrz¸ednik z innej frazy. Reguły zdefiniowane w ramach akomodacji syntaktycznych morfologicznych definiuja¸ warunki, które musza¸ byc´ spełnione przez formy wyrazowe, aby mogły one istniec´ wspólnie w jednym zdaniu[27]. Przykładowa taka reguła dla leksemów kot i is´c wygladała¸ by jak na diagramie 3.1.

Wyraz nadrz¸edny (kot) wymusza na podrz¸ednym (is´c)´ odpowiednia¸ form¸e gramatyczna:¸ liczb¸e, ro-dzaj i osob¸e. Pozostałe elementy takie jak czas, czy tryb zalez˙a¸ od zródła´ pozaj¸ezykowego, jakim jest kontekst, do którego odniesiona jest wypowied´z. Uogólniajac¸ powyzszy˙ przykład mozna˙ przedstawic´ ogólna¸ reguł¸e kształtujac¸a¸ zalezno˙sci´ mi¸edzy podmiotem a orzeczeniem w zdaniu jak na diagramie 3.2.

Tego typu zalezno˙sci´ akomodacyjnych mozna˙ zaobserwowac´ dla innych cz¸esci´ mowy, jak pokazano na przykładowych diagramach 3.3 i 3.4.

Ł. Wróbel

3.2. Relacje mi¸edzy wyrazami
19

Rysunek 3.1: Przykład oddziaływania akomodacyjnego dla leksemów kot i is´c.

Rysunek 3.2: Schemat uogólnienia oddziaływan´ akomodacyjnych mi¸edzy podmiotem i orzeczeniem.

Rysunek 3.3: Schemat uogólnienia oddziaływan´ akomodacyjnych mi¸edzy czasownikiem i rzeczowni-kiem.

Rysunek 3.4: Schemat uogólnienia oddziaływan´ akomodacyjnych mi¸edzy przymiotnikiem i rzeczowni-kiem.

W gramatyce j¸ezyka polskiego wyst¸epuje równiez˙ akomodacja typu frazy zdaniowej, w której cza-

sownik wymaga zdania podrz¸ednego poprzedzonego spójnikiem lub zdania pytajno-zaleznego˙ np. wyraz sadzi´¸c w znaczeniu przypuszczac´ lub wyraza˙c´ przekonanie wymaga zdania podrz¸ednego rozpoczynaja¸-cego si¸e spójnikiem ze˙ (Sadz¸e, ze˙ gramatyka polska jest trudna.).

Rysunek 3.5: Przykład akomodacji typu frazy zdaniowej.

Zarówno w logice jak i w lingwistyce cz¸esto wykorzystywane jest poj¸ecie konotacji.

Definicja 8 W składni j¸ezyka polskiego konotacja¸ nazywamy zjawisko polegajace¸ na tym, ze˙ wyst¸epujacy¸ leksem niejako „zapowiada” pojawienie si¸e innych leksemów lub konstrukcji składniowych.[27]

Przykładem moze˙ byc´ wypowied´z Wczoraj mój pies..., która pozbawiona dokonczenia´ rodzi pytanie Co si¸e stało z twoim psem?. Naturalnym jest wi¸ec, ze˙ bezposrednio´ po usłyszeniu takiego niepełnego fragmentu słuchacz oczekiwał b¸edzie czasownika, lub innych uzupełnien´ które odpowiedza¸ na jego nie-zadane pytanie. Najcz¸esciej´ konotacja syntaktyczna dotyczy zapowiadania okreslonego´ typu konstrukcji,

a rzadziej wymusza konkretny leksem, lub okreslone´ jego własciwo´sci´. W powyzszym˙ przykładzie wy-raz pies „zapowiada” czasownik lub okreslon´a¸ grup¸e czasownikowa,¸ o strukturze, która wynika z cech akomodacyjnych i konotacyjnych danego czasownika[27].

Istnieje prosta metoda sprawdzenia, czy dany składnik podlega konotacji. Jezeli˙ usuni¸ecie rzeczo-nego składnika sprawi, ze˙ konstrukcja stanie si¸e niepoprawna, czy to w sensie składniowym, czy logicz-nym, oznacza to ze˙ jest on konotowany (wymagany)[27]. Jako przykład usuni¸ecie ze zdania Wczoraj mój pies pogryzł moje ulubione pantofle. składnika pogryzł spowoduje otrzymanie niepoprawnej struktury: Wczoraj mój pies moje ulubione pantofle. Wyraz pies jest równiez˙ konotowany. Zdanie Wczoraj mój pogryzł moje ulubione pantofle. w okreslonym´ kontekscie´ (zapewniajac¸ odpowiedni podmiot domyslny)´ mogłoby zostac´ uznane za poprawne, to jednak bezposrednie´ znaczenie zostało zmienione. Niekonoto-wane sa¸ natomiast składniki: wczoraj, mój, moje i ulubione.

Z. Klemensiewicz w swojej interpretacji budowy zdania, wprowadził nast¸epujacy¸ dychotomiczny

podział zwiazków¸ mi¸edzy składnikami w zdaniu:

zwiazek¸ główny - zwiazek¸ mi¸edzy podmiotem i orzeczeniem zwiazki¸ poboczne - wszystkie pozostałe zwiazki¸ w zdaniu

Powyzszy˙ podział czyni podmiot i orzeczenie najwazniejszymi˙ składnikami zdania. Klemensiewicz

przypisywał dominujac¸a¸ rol¸e podmiotowi, który uznał za „nadrz¸ednik zwiazku¸ orzekajacego”[7]¸. Orze-czenie niesie informacj¸e o stanie podmiotu, lub wykonywanej przez niego czynnosci´. Wymagane jest jednak uprzednie stwierdzenie istnienia podmiotu. Z punktu widzenia czysto strukturalnego czasownik

uznawany jest za nadrz¸ednik zdania, gdyz˙ jest on jego elementem koniecznym i wystarczajacym[27]¸. Zwiazki¸ poboczne kategoryzowane sa¸ nast¸epujaco:¸

zwiazek¸ zgody zwiazek¸ rzadu¸

zwiazek¸ przynalezno˙sci´

Zwiazek¸ zgody zachodzi na przykład mi¸edzy rzeczownikiem i przymiotnikiem lub mi¸edzy dwoma

rzeczownikami. Podrz¸ednik kazdorazowo˙ dostosowuje swoja¸ form¸e fleksyjna¸ do formy nadrz¸ednika[27]. Przykładowo rzeczownik kot narzuca okreslony´ rodzaj, liczb¸e i przypadek przymiotnikowi perski - kota perskiego, kotu perskiemu, kocie perskim. W przypadku dwóch rzeczowników sytuacja wyglada¸ podob-nie: pies bernardyn, psu bernardynowi, psem bernardynem.
W przypadku zwiazku¸ rzadu¸ (głównie z czasownikiem) od rzeczownika wymagany jest sci´sle´ okre-slony´ przypadek[27]. Przykładem niech b¸edzie wyrazenie˙ jecha´ pociagiem¸, lub karmi´c psa. Rzeczow-niki pociag¸ i pies musza¸ wystapi¸c´ odpowiednio w narz¸edniku i bierniku, aby zdanie było składniowo poprawne. Zwiazek¸ rzadu¸ moze˙ równiez˙ zachodzic´ mi¸edzy rzeczownikiem, a przyimkiem: przy drodze, za górami, czy tez˙ mi¸edzy dwoma rzeczownikami: szklanka soku, chwila namysłu. Szczególnym przy-padkiem zwiazku¸ rzadu¸ jest sytuacja, w której oprócz okreslonego´ przypadku rzeczownika wymagany jest równiez˙ odpowiedni przyimek zespalajacy¸ go z czasownikiem, na przykład: idziemy na grzyby, czy tez˙ czytam o lingwistyce.

Zwiazek¸ przynalezno˙sci,´ który charakteryzuje si¸e tym, ze˙ forma wyrazu podrz¸ednego nie jest wymu-szana przez wyraz nadrz¸edny, wyst¸epuje głównie mi¸edzy czasownikiem, a nieodmienna¸ cz¸esci´a¸ mowy, na przykład przysłówkiem[27]. Przykładowo: id¸e szybko, wpadn¸e jutro.

Ł. Wróbel

4. Opis własnego rozwiazania¸

Wraz z narodzinami sztucznej inteligencji powstały nowe mozliwo˙sci´ automatycznej analizy tekstu. Czatboty, które do tej pory mogły jedynie wykonywac´ sci´sle´ zaprogramowane operacje w z góry okreslonych´ przypadkach zyskały nowe mozliwo˙sci,´ które czyniły je nieporównanie bardziej elastycznymi. Dzi¸eki mozliwo˙sci´ wnioskowania, program, któremu dostarczono odpowiednio duz˙a¸ ilos´c´ danych był w stanie sam okresli´c´ reguły stosowane w trakcie rozmowy, a nawet uczyc´ si¸e ich na biez˙aco¸. Wcia¸z˙ jednak nie sa¸ one niezawodne, a z nowymi mozliwo˙sciami´ pojawiły si¸e tez˙ nowe wyzwania i problemy. Od idealnego czatbota oczekuje si¸e, zeby˙ pomyslnie´ przeszedł test Turinga. Oznacza to, ze˙ prowadzacy¸ z nim konwersacj¸ człowiek nie powinien si¸e zorientowac,´ ze˙ rozmawia z programem komputerowym.

Aby sprostac´ temu wyzwaniu czatbot musi spełnic´ szereg warunków poprawnosciowych´. Przede wszystkim, budowane przez program zdania musza¸ miec´ prawidłowa¸ składni¸e w j¸ezyku polskim. Wykorzystywana w tym celu jest tak zwana gramatyka generatywna stworzona przez Noama Chomsky'ego,

która jest skonczonym´ zbiorem reguł umozliwiaj˙acych¸ zbudowanie wszystkich mozliwych,˙ poprawnych zdan´. Czatbot jest dzi¸eki niej w stanie zweryfikowac,´ czy dana forma jest poprawna składniowo, a takze˙

taka¸ form¸e utworzyc´. Jest to mozliwe˙ nawet, gdy zdanie podlegajace¸ weryfikacji zostało napotkane po raz pierwszy. Sama poprawnos´c´ j¸ezykowa nie jest jednak wystarczajaca,¸ aby prowadzic´ rozmow¸e z człowiekiem. Nie mniej wazna,˙ a o wiele bardziej skomplikowana jest warstwa logiczna wypowiedzi. Słowa, które padaja¸ podczas rozmowy cz¸esto maja¸ wi¸ecej niz˙ jedno znaczenie, które nie zawsze program komputerowy jest w stanie wywnioskowac´ z kontekstu zdania, czy nawet wi¸ekszej cz¸esci´ konwersacji. Porównania, przenosnie,´ idiomy, czy nawet pospolite przysłowia lub powiedzenia sa¸ powazn˙a¸ przeszkoda,¸ która wcia¸z˙ stanowi wyzwanie dla programistów.

Aplikacja, która jest przedmiotem tej pracy skupia si¸e na sferze semantycznej rozmowy pomi¸edzy

rzeczonym czatbotem, a uzytkownikiem˙. Dotyka problemu rozumienia znaczenia słów i relacji pomi¸edzy nimi. W szczególnosci´ modeluje struktur¸e, która umozliwi˙ programowi zbadanie kontekstu uzywanych˙

przez uzytkownika˙ słów przez pryzmat posiadanej bazy wiedzy, aby lepiej zrozumiec´ ich znaczenie.

Człowiek, który usłyszy lub przeczyta okreslony´ wyraz natychmiast kojarzy go z odpowiadajacym¸

mu obrazem. Dzi¸eki temu od razu jest w stanie połaczy¸c´ otrzymany obraz z przypisanymi mu (a wi¸ec tez i słowu, które reprezentuje) cechami. Daje mu to mozliwo˙s´c´ opisania tego co zobaczył, a takze˙ porównania z innym słowem (czyli de facto przypisanym mu obrazem). Ta naturalna metoda agregacji jest główna¸ inspiracja¸ przedstawionego dalej rozwiazania¸. Jakkolwiek program komputerowy nie jest w stanie wyobrazic´ sobie obrazu w sposób tak plastyczny jak człowiek, to jednak nie jest to przeszkoda,¸ by

za pomoca¸ grupowania słów z nim zwiazanych¸ był go w stanie z porównywalna¸ dokładnosci´a¸ opisac´.

Napisana przeze mnie aplikacja wykorzystuje pajaka¸ internetowego, z którego pomoca¸ budowana jest baza j¸ezykowa b¸edaca¸ podstawowym zródłem´ wiedzy programu. W zalezno˙sci´ od okreslonych´ przez

uzytkownika˙ słów b¸edacych¸ korzeniami drzewa, na podstawie przechowywanego tekstu powstaje lista słów, które potencjalnie moga¸ byc´ z nimi w relacji. Nast¸epnym krokiem jest wyodr¸ebnienie tych wyra-zów, które łacz¸a¸ si¸e z korzeniami w poprawne j¸ezykowo i logicznie zalezno˙sci´. Ostatnim etapem dzia-łania programu jest zaprezentowanie zdefiniowanych w ten sposób połacze¸n´ na grafie przyzwyczajen´ lingwistycznych. Szczegółowa realizacja wymienionych kroków, a takze˙ ich działanie zostana¸ opisane w kolejnych podrozdziałach.

.

4.1. Załozenia projektowe

4.1. Załozenia˙ projektowe

Niniejszy program zakłada, ze˙ podane przez uzytkownika˙ słowo (korzen)´ jest poprawnym słowem w j¸ezyku polskim napisanym bez bł¸edów. Poprawa bł¸edów i weryfikacja poprawnosci´ nie sa¸ w zakresie

wymagan´ projektu. Ze wzgl¸edu na ograniczenia zakresu pracy aplikacja obsługuje wyłacznie¸ relacje po-mi¸edzy rzeczownikami, czasownikami, przymiotnikami i przysłówkami. Poprawnos´c´ relacji badana jest w oparciu o wyniki z wyszukiwarki Google, która nakłada dodatkowe ograniczenia dotyczace¸ maksy-malnej ilosci´ zapytan´. Problem ten opisany został bardziej szczegółowo w dalszej cz¸esci´ pracy.

4.2. Architektura systemu

Aplikacja składa si¸e z czterech głównych elementów: pajaka¸ internetowego, modułu odpowiedzial-

nego za tworzenie relacji, specjalistycznego grafu LHG oraz interfejsu uzytkownika˙. Pajak¸ jest elemen-

tem opcjonalnym, gdyz˙ uzytkownik˙ ma mozliwo˙s´c´ stworzenia bazy wiedzy na podstawie innego zródła´ niz˙ Internet. Uogólniony diagram klas zaprezentowany został na rysunku 4.1.

Rysunek 4.1: Diagram klas

4.2.1. Interfejs uzytkownika˙

Podstawowym zadaniem interfejsu uzytkownika˙ jest umozliwienie˙ konfiguracji parametrów aplika-cji. Dzi˛eki niemu mozna˙ zdefiniowac´ wejscia´ i wyjscia´ danych, a takze˙ sposób pozyskiwania i wyswietla´-

nia informacji, zaleznie˙ od preferencji uzytkownika,˙ lub ograniczen´ systemu (na przykład brak dost˛epu do Internetu).

Bezposrednio´ po uruchomieniu programu wyswietlone´ zostaje okno połaczenia˛ z baza˛ danych po-kazane na zrzucie ekranu 4.2. Aby nawiaza˛c´ łaczno˛s´c´ nalezy˙ podac´ adres serwera, nazw˛e bazy, a takze˙ login i hasło. Połaczenie˛ z baza˛ danych nie jest wymagane i mozna˙ z niego zrezygnowac´.

Rysunek 4.2: Okno połaczenia˛ z baza˛ danych

Główne okno programu zaprezentowane na zrzucie ekranu 4.3 jest nieco bardziej skomplikowane.

Najwazniejszym˙ elementem jest pole słuz˙ace˛ do podania słów, wokół których budowany b˛edzie specjali-styczny graf przyzwyczajen´ lingwistycznych. Kazde˙ słowo wpisane po przecinku wykorzystane zostanie podczas wyszukiwania stron internetowych, w oparciu o które powstanie baza wiedzy, a takze˙ przy two-

rzeniu relacji. Mozliwe˙ do wpisania sa˛ wyłacznie˛ wyrazy znajdujace˛ si˛e w bazie słownikowej biblioteki CLP opisanej dokładniej w rozdziale 4.2.3.

Rysunek 4.3: Główne okno programu

Posród´ opcji mozliwych˙ do konfiguracji za pomoca¸ głównego okna programu jest takze˙ tryb pracy

aplikacji. Działanie systemu mozliwe˙ jest w jednym z czterech stanów, które definiuja¸ sposób pozy-skiwania informacji, budowania bazy wiedzy, a takze˙ znaczaco¸ wpływaja¸ na wyswietlony´ finalnie graf. Tryby pracy aplikacji dziela¸ si¸e na:

ONLINE - Jest to domyslny´ tryb działania, w którym aplikacja wykorzystuje połaczenie¸ z Inter-

netem w celu zbudowania korpusu tekstu i oceny poprawnosci´ utworzonych relacji. Mozliwe˙ jest takze˙ wykorzystanie plików i bazy danych w celu odczytu i zapisu danych.

OFFLINE - Tryb ten pozwala wyłacznie¸ na zaprezentowanie relacji, które zapisane sa¸ w pliku lub

bazie danych. Nie wykorzystuje on połaczenia¸ z Internetem.

Symulator ONLINE - Symulator pozwala na utworzenie grafu relacji na podstawie korpusu tekstu

bez wykorzystywania Internetu do ich ewaluacji. Umozliwia˙ to zaprezentowanie przykładowego grafu, który jest w stanie zobrazowac´ interesujace¸ informacje, jednoczesnie´ nie korzystajac¸ z ogra-niczonych zasobów wyszukiwarki internetowej Google.

Symulator OFFLINE - Tryb ten umozliwia˙ zasymulowanie pełnego działania programu bez do-

st¸epu do Internetu z wykorzystaniem korpusów tekstów znajdujacych¸ si¸e w plikach lub bazie da-nych.

W głównym oknie programu zdefiniowac´ mozna˙ takze˙ zródła´ danych wejsciowych´. Mozliwe˙ jest wczytanie tekstu lub relacji zarówno z plików jak i bazy danych. Okno prezentujace¸ list¸e plików z da-nymi do wczytania prezentuje zrzut ekranu 4.4. Jezeli˙ plik zaczyna si¸e od taga RELATIONS, aplikacja usiłuje odczytac´ zapisane w nim relacje, w przeciwnym wypadku zostanie on uznany jako korpus tekstu. Informacje te mozna˙ uzyskac´ takze˙ z bazy danych pod warunkiem, ze˙ wczesniej´ ustanowione zostało

połaczenie¸. Mozna˙ je zdefiniowac´ ponownie wybierajac¸ z menu Plik, a nast¸epnie Połacz¸ z baza¸ danych.

Rysunek 4.4: Wybór plików zawierajacych¸ relacje lub korpusy tekstów.

W trybie ONLINE mozliwe˙ jest okreslenie´ liczby linków jaka zostanie przegladni¸eta dla kazdego˙ słowa kluczowego. Adresy stron odwiedzonych przez pajaka¸ zapisywane sa¸ do pliku, aby aplikacja

ponownie nie pobierała informacji z tego samego zródła´. Uzytkownik˙ ma jednak mozliwo˙s´c´ r¸ecznego

opróznienia˙ pliku z linkami.

Waznymi˙ parametrami z punktu widzenia wyswietlania´ informacji na grafie sa¸ parametry definiujace¸

maksymalna¸ liczb¸e połacze¸n,´ jakie zostana¸ wyswietlone´ dla pojedynczego słowa, a takze˙ maksymalna liczb¸e połacze¸n,´ jakie zostana¸ wyswietlone´ dla relacji utworzonych na podstawie pseudo-logicznego rozumowania. Parametry te pozwalaja¸ kontrolowac´ ilos´c´ informacji na grafie i czynia¸ go bardziej czytelnym. Bardziej szczegółowy opis sposobu przedstawiania danych na grafie LHG znajduje si¸e w rozdziale 4.4.

Ł. Wróbel

4.2. Architektura systemu
25

4.2.2. Pajak¸ internetowy i ekstraktor danych

Zdecydowałem si¸e napisac´ własna,¸ uproszczona¸ wersj¸ pajaka¸ internetowego, która wykorzystuje tylko absolutnie podstawowe funkcjonalnosci,´ a jednoczesnie´ w zupełnosci´ zaspokaja potrzeby aplika-

cji. Nie zdecydowałem si¸e na zadne˙ z juz˙ istniejacych¸ rozwiaza¸n,´ takze˙ aby miec´ pełna¸ kontrol¸e nad wykonywanymi przez pajaka¸ operacjami. Podstawowa¸ róznic˙a¸ w stosunku do wi¸ekszosci´ crawlerów jest fakt, ze˙ aplikacja nie poda¸za˙ za napotkanymi linkami, a tylko przechodzi przez adresy z okreslonego´ zródła,´ co znacznie skraca czas budowania bazy wiedzy i ogranicza ilos´c´ pobieranych, choc´ zb¸ednych w danym momencie danych.

Pajak¸ rozpoczyna swoje działanie od znalezienia wszystkich linków na stronie startowej i zapisaniu

ich na list¸e adresów do odwiedzenia. Nast¸epnie tworzona jest okreslona´ liczb¸e watków,¸ które po odwiedzeniu usuwaja¸ kolejne elementy listy. Pajak¸ wykorzystuje framework Htmlparser, który udost¸epnia me-

tod¸e pozwalajaca¸ na pobranie wyłacznie¸ tekstu z danej witryny pozbawionego tagów HTML. Uzyskany

w ten sposób tekst, który wcia¸z˙ jeszcze moze˙ zawierac´ wiele niepoz˙adanych¸ elementów przekazany zostanie do ekstraktora, który zajmie si¸e jego dokładnym przetworzeniem. Po przekazaniu tekstu strony do ekstraktora watek dodaje ja¸ na list¸e odwiedzonych i przechodzi to przetwarzania kolejnego elementu z

listy oczekujacych,¸ lub konczy´ działanie, gdy taki element nie istnieje.

Ekstraktor danych, którego schemat działania widoczny jest na diagramie 4.5, jest integralna¸ cz¸esci´a¸

pajaka¸ internetowego. Jego zadaniem jest usuna¸c´ wszelkie niepoz˙adane¸ elementy pozostałe po usuni¸eciu tagów HTML i wyodr¸ebnic´ tekst, który nast¸epnie b¸edzie mógł byc´ wykorzystany w głównej cz¸esci´ aplikacji. Pierwszym problemem jaki napotkałem było okreslenie´ definicji „czystego tekstu”. Wbrew pozorom pozbycie si¸e elementów j¸ezyka HTML to dopiero poczatek¸ procesu jego pozyskiwania. Podstawowym komponentem, z którego zbudowany jest kazdy˙ tekst jest forma zdaniowa. Przyjałem,¸ ze˙ naj-

mniejsza taka forma musi składac´ si¸e przynajmniej z dwóch wyrazów, aby wnosic´ jakakolwiek¸ wartos´c´ do bazy wiedzy. Z punktu widzenia relacji wyrazenie˙ złozone˙ z pojedynczego słowa jest mało interesu-jace¸.

Dla zmniejszenia stopnia skomplikowania problemu oraz zachowania w miar¸e przewidywalnej bu-

dowy zdan´ przyjałem,¸ ze˙ powinny si¸e one składac´ wyłacznie¸ z okreslonych´ znaków. Dopuszczalny zbiór

znaków zawierał wyłacznie¸ duze˙ i małe litery alfabetu polskiego, a takze˙ białe znaki, kropki, wykrzyk-niki, pytajniki i przecinki. Wszelkie elementy spoza tego zbioru dyskwalifikuja¸ zdanie jako zródło´ in-formacji. To dos´c´ restrykcyjne ograniczenie uzasadnione jest tym, ze˙ znaki takie jak -, :, czy () sa¸ w stanie znaczaco¸ zmienic´ struktur¸e zdania, a takze˙ utrudnic´ automatyczne sprawdzenie jego poprawnosci´. Przykładem moze˙ byc´ znak pomini¸ecia fragmentu tekstu podczas cytowania: (...), czy tez˙ ewentualne pozostałosci´ po tagach HTML, lub fragmentach skryptów które nie zawsze zostana¸ dokładnie usuni¸ete.

Najwi¸ekszym problemem, z jakim zetknałem¸ si¸e podczas tworzenia ekstraktora było okreslenie´ gra-

nicy mi¸edzy formami zdaniowymi w ciagłym¸ tekscie´. Teoretycznie zdanie w j¸ezyku polskim zaczyna si¸e wielka¸ litera,¸ a konczy´ kropka,¸ znakiem zapytania lub wykrzyknikiem. Dla programu komputerowego taka definicja jest jednak niewystarczajaca¸. W poprawnym zdaniu, moga¸ bowiem wystapi¸c´ wielkie litery,

które nie sa¸ poczatkiem¸ nowego, jak i kropki, które tego zdania nie koncz´a¸. Dobrym przykładem moze˙ byc´ wypowied´z: "Pan prof. Kowalski udał si¸e na zasłuzony˙ urlop."Program szukajac¸ nowego zdania znalazłby liter¸ P w słowie Pan, która¸ słusznie uznałby za jego poczatek,¸ ale za koniec uznałby kropk¸e nast¸e-

pujac¸a¸ po skrócie prof. Zaproponowane przeze mnie rozwiazanie¸ wyszukuje co prawda zdania zgodnie z ta¸ intuicyjna¸ definicja,¸ ale odrzuca te, które zakonczone´ sa¸ skrótem jak na przykład prof., gen., gł.,

np., itp., a wyszukiwanie nast¸epnego rozpoczyna si¸e od kolejnej nie b¸edacej¸ elementem zadnego˙ skrótu kropki. Pozyskane w ten sposób zdania zostaja¸ zapisane do bazy danych i plików zdefiniowanych przez

uzytkownika,˙ a takze˙ wykorzystane w procesie budowania grafu relacji.

Ł. Wróbel

Rysunek 4.5: Schemat działania ekstraktora danych

4.2.3. Biblioteka CLP

Niezwykle istotnym z punktu widzenia poprawnosci´ syntaktycznej elementem aplikacji jest słownik fleksyjny CLP. Został on stworzony przez Katedr¸ Informatyki Akademii Górniczo-Hutniczej i Kate-dr¸e Lingwistyki Komputerowej Uniwersytetu Jagiellonskiego,´ a jego autorami sa¸ Wiesław Lubaszew-ski, Henryk Wróbel, Marek Gaj¸ecki, Barbara Moskal, Paweł Pietras, Piotr Pisarek i Teresa Rokicka. Dost¸epny jest on w formie elektronicznej bazy danych SFJP (Słownik Fleksyjny J¸ezyka Polskiego) i biblioteki CLP dla j¸ezyka C.

Podstawowe funkcjonalnosci´ biblioteki to:

rozpoznawanie wyrazu na podstawie jego dowolnej formy fleksyjnej

dostarczanie informacji o wyrazie

wygenerowanie form fleksyjnych dla danego wyrazu dostarczanie informacji o formie wyrazu
Kazdy˙ wyraz w bazie danych reprezentowany jest przez niepowtarzalny numer, dzi¸eki któremu

mozna˙ go jednoznacznie zidentyfikowac,´ wraz z jego etykieta¸ fleksyjna¸. Dla kazdego˙ słowa znajduja¸-cego si¸e w bazie biblioteka jest w stanie zwrócic´ jego numer, lub tablic¸e numerów, jesli´ dopasowanie nie

jest jednoznaczne. Na podstawie etykiety fleksyjnej mozliwe˙ jest okreslenie,´ na przykład cz¸esci´ mowy i wzorca odmiany wyrazu. Identyfikator słuzy˙ do wyznaczenia formy podstawowej danego słowa, a takze˙

wszystkich jego mozliwych˙ form fleksyjnych wraz z jego pozycja¸ wsród´ nich. W bazie słownika znaj-duje si¸e ponad 120 tysi¸ecy wyrazów. Jest ona na poczatku¸ ładowana do pami¸eci RAM, co znacznie

przyspiesza wyszukiwanie. Istniejaca¸ wersja działa wyłacznie¸ pod systemem GNU/Linux.

4.2.4. Specjalistyczny graf przyzwyczajen´ lingwistycznych

Głównym zadaniem uproszczonej postaci grafu LHG, która jest wynikiem działania aplikacji jest

pokazanie uzytkownikowi˙ mozliwie˙ jak najbardziej reprezentatywnego fragmentu powstałych relacji. Do wizualizacji wykorzystałem bibliotek¸ JUNG, gdyz˙ oferuje dos´c´ bogate mozliwo˙sci´ prezentacji i

spora¸ interakcj¸ z uzytkownikiem˙. Wierzchołki grafu reprezentuja¸ formy bazowe słów, które wchodza¸ w

skład relacji, natomiast kraw¸edzie odpowiadaja¸ relacjom mi¸edzy połaczonymi¸ przez nie wyrazami. Kolor

kazdego˙ wierzchołka zalezny˙ jest od tego, jaka¸ cz¸esci´a¸ mowy jest słowo, które reprezentuje. Szczegóły

dotyczace¸ przyporzadkowania¸ kolorów do poszczególnych cz¸esci´ mowy zostały zaprezentowane w tabeli 4.1.

Tablica 4.1: Kolory poszczególnych cz¸esci´ mowy

Cz¸es´c´ mowy
Kolor

rzeczownik
niebieski
czasownik
zółty˙
przymiotnik
zielony
przysłówek
turkusowy
liczebnik
pomaranczowy´

Na grafie nie sa¸ prezentowane wszystkie powstałe i zaimportowane relacje, gdyz˙ stałby si¸e on

nieczytelny. Parametr ustawiony przez uzytkownika˙ w głównym oknie programu okresla´ maksymalna¸ liczb¸e kraw¸edzi, które sa¸ wyswietlane´ dla pojedynczego wierzchołka. Wszystkie mniej lub bardziej po-

prawne połaczenia¸ mi¸edzy słowami zaprezentowane sa¸ na jednej skierowanej kraw¸edzi. Po klikni¸eciu

na nia¸ uzytkownik˙ moze˙ odczytac´ wszystkie relacje, które reprezentuje, ich typ, a takze˙ nadana¸ im wartos´c´. Przykład wyswietlania´ szczegółowych informacji o kraw¸edziach został przedstawiony na zrzutach

ekranu 4.6 i 4.7. Prezentowane sa¸ wyłacznie¸ informacje uznane za najbardziej wartosciowe´ i dajace¸ najlepszy obraz działania systemu. Dokładny opis algorytmu decyzyjnego odpowiedzialnego za rysowanie kraw¸edzi opisany został w sekcji 4.4.

Uzytkownik˙ ma dodatkowo mozliwo˙s´c´ wyswietlenia´ relacji przechodnich. Sa¸ one wynikiem pseudo-

logicznego rozumowania modelujacego¸ naturalna¸ przechodnios´c´ cech w ramach hierarchii. Nie sa¸ one

bezposrednio´ uzyskane z tekstu zródłowego,´ ale powstaja¸ poprzez wyciagania¸ wniosków z otrzymanej struktury, co maksymalizuje ilos´c´ uzyskanych informacji i dokładniej oddaje sposób ludzkiego mysle´-

nia. Dodatkowo mozliwe˙ jest uzupełnienie otrzymanego grafu nowymi słowami kluczowymi i relacjami przez wybór opcji Dodaj wi¸ecej informacji z menu Plik. Nalezy˙ jednak pami¸etac,´ ze˙ zaprezentowany

zostaje zawsze wyłacznie¸ ograniczony wycinek zbudowanej struktury, co sprawia, ze˙ niektóre oczeki-

wane połaczenia¸ moga¸ nie zostac´ wyswietlone,´ a zamiast nich zaprezentowane zostana¸ te, które zostały najlepiej ocenione.

Jezeli˙ uzytkownik˙ chce zobaczyc´ wszystkie utworzone i pobrane z bazy wiedzy relacje, ma taka¸ moz˙-liwos´c´ przez wybór opcji Pokaz˙ list¸e podstawowych relacji z menu Plik. Dodatkowo, jezeli˙ chce zoba-czyc´ relacje, które powstały na skutek wnioskowania, ma mozliwo˙s´c´ wyboru opcji Pokaz˙ list¸e wszystkich relacji. Przykładowa lista, jaka zostanie zaprezentowana została pokazana na zrzucie ekranu 4.8.

Rysunek 4.6: Szczegóły kraw¸edzi - grupowanie, posiadanie

Rysunek 4.7: Szczegóły kraw¸edzi - okreslenie´

Ł. Wróbel

4.3. Opis działania aplikacji
29

Rysunek 4.8: Lista relacji

4.3. Opis działania aplikacji

Celem działania aplikacji jest stworzenie uproszczonej postaci grafu przyzwyczajen´ lingwistycz-nych, który ilustruje hierarchi¸ wyrazów wraz z ich okresleniami´ oraz pokazuje przechodnios´c´ niektó-rych cech w ramach tej struktury. Powstały graf jest wyłacznie¸ uproszczona¸ odmiana¸ grafu LHG, gdyz˙

prezentuje wyłacznie¸ bazowe formy wyrazów, a takze˙ wyłacznie¸ niektóre typy relacji charakterystycz-nych dla LHG[9]. Z punktu widzenia czatbota, moze˙ to byc´ cenne zródło´ wiedzy, które podczas rozmowy mozna˙ wykorzystac´ do analizy kontekstu, w jakim uzyte˙ zostało jakies´ słowo, zadawania pytan´ o okre-slone´ cechy wyrazu, czy tez˙ jego opisywania.

4.3.1. Tryb online

Rysunek 4.9: Wybór trybu online

Po wybraniu trybu online (zrzut ekranu 4.10), do wyszukiwarki Google zostaje wysłane zapytanie

o podane słowa-korzenie. Ze zwróconej listy wyników pobierana jest okreslona´ w parametrach konfi-

guracyjnych przez uzytkownika˙ liczb¸e linków, które nast¸epnie przekazywane sa¸ do kolejnych watków¸ pajaka¸. Po okresleniu,´ czy strona nadaje si¸e jako zródło´ informacji nast¸epuje ekstrakcja tekstu. Powstaje lista zdan,´ które trafiaja¸ do okreslonego´ na poczatku¸ miejsca docelowego bazy wiedzy dla tekstu. Watki¸ działaja¸ dopóki na liscie´ adresów do odwiedzenia znajduja¸ si¸e jakies´ elementy. Adresy, które zostały odwiedzone wypisane sa¸ w logach: 4.3.1.

2011 03 05
14:34:55
[INFO]
Pobieranie
słów:
http://eszukam.pl/

posokowiec bawarski szczenie pies/debnica kaszubska/324579/

2011 03 05
14:34:55
[INFO]
Pobieranie
zdan´:
http://eszukam.pl/

posokowiec bawarski szczenie pies/debnica kaszubska/324579/

2011 03 05
14:34:55
[INFO]
Pobieranie
tekstu:
http://eszukam.pl/

posokowiec bawarski szczenie pies/debnica kaszubska/324579/

2011 03 05
14:34:55
[INFO]
Pobieranie
słów:
http://tanio.pl/Pies

Baskervilleow Artur Conan Doyle Audiobook1 68234 12277036.html

2011 03 05
14:34:55
[INFO]
Pobieranie

z d a n´ :
h t t p : / / t a n i o . p l / P i e s

B a s k e r v i l l e o w A r t u r Conan Doyle Audiobook1 68234 12277036. h t m l

2011 03 05
1 4 : 3 4 : 5 5
[INFO]
P o b i e r a n i e
t e k s t u :
h t t p : / / t a n i o . p l / P i e s

B a s k e r v i l l e o w A r t u r Conan Doyle Audiobook1 68234 12277036. h t m l

2011 03 05
1 4 : 3 4 : 5 5
[INFO]
P o b i e r a n i e
s ł ó w :
h t t p : / / www.

n i e p e l n o s p r a w n i . p l / l e d g e / x / 1 0 6 8 6

2011 03 05
1 4 : 3 4 : 5 5
[INFO]
P o b i e r a n i e
z d a n´ :
h t t p : / / www.

n i e p e l n o s p r a w n i . p l / l e d g e / x / 1 0 6 8 6

2011 03 05
1 4 : 3 4 : 5 5
[INFO]
P o b i e r a n i e
t e k s t u :
h t t p : / / www.

n i e p e l n o s p r a w n i . p l / l e d g e / x / 1 0 6 8 6

2011 03 05
14:34:55
[INFO]
Pobieranie
słów:
http://klarka.blog.onet.

pl/jaki pan taki pies,2,ID419994422,n

2011 03 05
14:34:55
[INFO]
Pobieranie
zdań:
http://klarka.blog.onet.

pl/jaki pan taki pies,2,ID419994422,n

2011 03 05
14:34:55
[INFO]
Pobieranie
tekstu:
http://klarka.blog.onet

.pl/jaki pan taki pies,2,ID419994422,n

2011 03 05
14:34:55
[INFO]
Pobieranie
słów:
http://pies.org.pl/

2011 03 05
14:34:55
[INFO]
Pobieranie
zdań:
http://pies.org.pl/

2011 03 05
14:34:55
[INFO]

P o b i e r a n i e
t e k s t u :
h t t p : / / p i e s . o r g . p l /

2011 03 05
1 4 : 3 4 : 5 5
[INFO]
P o b i e r a n i e
s ł ó w :
h t t p : / / www. f i l m w e b . p l /

f i l m / P i e s + a n d a l u z y j s k i 1929 906
z d a n´ :

2011 03 05
1 4 : 3 4 : 5 5
[INFO]
P o b i e r a n i e

h t t p : / / www. f i l m w e b . p l /

f i l m / P i e s + a n d a l u z y j s k i 1929 906

2011 03 05
1 4 : 3 4 : 5 5
[INFO]
P o b i e r a n i e
t e k s t u :
h t t p : / / www. f i l m w e b . p l /

f i l m / P i e s + a n d a l u z y j s k i 1929 906

W nast¸epnym kroku wyniki działania pajaka¸ oraz wybrane przez uzytkownika˙ zródła´ tekstu sa¸ ana-

lizowane i wybierane sa¸ z nich zdania zawierajace¸ podane przez uzytkownika˙ słowa w ich dowolnej for-mie fleksyjnej. Wyszukiwane sa¸ wszystkie zwrócone przez bibliotek¸ CLP formy dla korzeni. Powstaje w ten sposób nowa lista zdan´. Wykorzystujac¸ bibliotek¸ CLP wszystkie słowa sa¸ nast¸epnie sprowadzane do formy bazowej, dzi¸eki czemu ich formy fleksyjne nie b¸eda¸ traktowane jako niezalezne˙ słowa. Sa¸ one

nast¸epnie zliczane i sortowane malejaco¸ w zalezno˙sci´ od cz¸estosci´ wyst¸epowania. Dla kazdego˙ korze-nia powstaje oddzielna lista, a sam korzen´ oczywiscie´ znajduje si¸e na jej szczycie, jako ze˙ wystapił¸ on

w kazdym˙ przetworzonym zdaniu. Za nim zazwyczaj umiejscowione sa¸ popularne przyimki np. z, do, w, na i spójniki np. i, a, lub. Program filtruje jednak z pomoca¸ biblioteki CLP otrzymane wyniki, tak

aby pozostały wyłacznie¸ okreslone´ cz¸esci´ mowy. Aplikacja skupia si¸e na najbardziej istotnych i niosa¸-cych najwi¸ecej informacji, a zatem na rzeczownikach, czasownikach, przymiotnikach, przysłówkach i

liczebnikach. Rozróznienia,˙ jaka¸ cz¸esci´a¸ mowy jest dane słowo dokonuje biblioteka CLP.

Ł. Wróbel

4.3. Opis działania aplikacji
31
Jako, ze˙ biblioteka CLP dopasowujac¸ wyraz porównuje go z wszelkimi mozliwymi˙ formami, nie

zawsze zwrócony wynik odpowiada kontekstowi zdania. Przykładowo cz¸esto wyst¸epujace¸ słowo kilka identyfikowane jest w pierwszej kolejnosci´ jako rzeczownik (jest to nazwa ryby), w drugiej jako przy-miotnik, a dopiero w trzeciej kolejnosci´ jako zaimek. Podobna sytuacja wyst¸epuje w przypadku bardzo

cz¸esto uzywanego˙ słowa jest, którego forma bazowa by´c równiez˙ uznawana jest przez CLP jako rzeczow-nik (dopełniacz liczby mnogiej rzeczownika bycie), a dopiero kolejny zwrócony wynik dopasowania jest

czasownikiem. Problem ten rozwiazany¸ został przez rozróznienie˙ na etapie przetwarzania tekstu nie tylko

słów, ale takze˙ ich cz¸esci´ mowy, co sprawia, ze˙ homonimy sa¸ rozróznialne˙ i traktowane jako oddzielne wyrazy.

W tak posortowanej liscie´ wyrazy znajdujace¸ si¸e wysoko maja¸ duz˙a¸ szans¸e byc´ w relacji ze słowem kluczowym, jako ze˙ cz¸esto były uzyte˙ z nim w jednym zdaniu. Takie statystyczne podejscie´ utrudnia

co prawda stwierdzenie koneksji z rzadziej uzywanymi˙ słowami, ale omija problemy zwiazane¸ z za-wiłosciami´ składni j¸ezyka polskiego. Nie jest konieczne budowanie skomplikowanego parsera zdan´ z

wyszukanymi regułami uwzgl¸edniajacymi¸ niuanse j¸ezykowe. Zwiazki¸ mi¸edzy wyrazami ustalane sa¸ w ramach jednego zdania. Frazy, w których klucz został zamieniony zaimkiem lub wystapił¸ jako podmiot domyslny´ w ogóle si¸e tu nie znajda,¸ gdyz˙ zostana¸ odfiltrowane jako niezawierajace¸ korzenia.

Elementy z posortowanej listy potencjalnych kandydatów do relacji ze słowem kluczowym sa¸ ko-

lejno w odpowiedniej formie łaczone¸ z korzeniem (takze˙ w odpowiedniej formie) z wykorzystaniem szablonów relacji. W zalezno˙sci´ od typu relacji do utworzonej w ten sposób formy moze˙ zostac´ dodany

spójnik. Szablony relacji definiuja¸ typowe zalezno˙sci´ pomi¸edzy dwoma wyrazami wyst¸epujacymi¸ w j¸e-zyku polskim i okreslaj´a¸ cz¸esci´ mowy jakie wchodza¸ w ich skład, ich form¸e fleksyjna¸ oraz kolejnos´c´. Nie musza¸ byc´ one skomplikowane, ani brac´ pod uwag¸e innych słów, które mogłyby zaburzyc´ ustalony schemat. Wazne˙ jest, aby sam układ i forma dwóch składowych wyrazów był poprawny.

Dla kazdego˙ schematu utworzona przez niego fraza jest przekazywana do wyszukiwarki Google, po czym

przypisywana jest jej wartos´c´ zalezna˙ od ilosci´ zwróconych wyników zapytania. Dopóki wi¸ec dana relacja mi¸edzy dwoma wyrazami wyst¸epuje w niezmienionej, dokładnie takiej jak okreslona´ for-mie odpowiednio cz¸esto na stronach indeksowanych przez wyszukiwark¸e, program zakłada, ze˙ jest ona

poprawna. Kazda˙ para wyrazów (korzen´ i słowo z listy wyrazów cz¸esto z nim wyst¸epujacych¸ w jednym zdaniu) moze˙ byc´ w potencjalnie dowolnej zalezno˙sci,´ a wi¸ec dla odpowiednich cz¸esci´ mowy spraw-

dzane sa¸ wszystkie mozliwe˙ szablony relacji. Te, dla których Google zwróci odpowiednio duz˙a¸ liczb¸e wyników zostaja¸ uznane za poprawne. Szablony powinny odzwierciedlac´ najcz¸esciej´ wyst¸epujace¸ struktury, gdyz˙ kazdy˙ kolejny zwi¸eksza liczb¸e zapytan´ o poprawnos´c´ utworzonej relacji, co bezposrednio´ przekłada si¸e na maksymalna¸ ilos´c´ uzyskanych wartosciowych´ informacji.

Przykładowo dla korzenia pies i słowa z listy łapa, jako ze˙ oba wyrazy sa¸ rzeczownikami sprawdzone

zostana¸ mozliwe˙ relacje pomi¸edzy dwoma rzeczownikami. Tak wi¸ec dla relacji posiadania o szablonie MIANOWNIK1 + ma + BIERNIK2 ułozona˙ zostanie fraza pies ma łap¸e, dla której Google zwróci X wy-ników. Sprawdzona zostanie tez˙ relacja odwrotna, a zatem łapa ma psa. Kolejna¸ potencjalna¸ relacja¸ po-mi¸edzy dwoma rzeczownikami jest grupowanie, czyli przypadek, gdy jeden wyraz nalezy˙ do podzbioru (czyli jest przykładem) wyrazu drugiego. Szablon tej zalezno˙sci´ wyglada¸ nast¸epujaco:¸ MIANOWNIK1 +

jest + NARZEDNIK2¸, a zatem pies jest łapa¸, czy tez˙ łapa jest psem nie zwróca¸ zbyt wielu wyników. Dla pary kot i zwierz¸ fraza kot jest zwierz¸eciem oceniona zostanie wysoko, a odwrotnos´c´ zwierz¸ jest kotem juz˙ nie.

Algorytm sprawdza poprawnos´c´ relacji wysyłajac¸ kolejne zapytania do wyszukiwarki do momentu,

az˙ nie wyczerpia¸ si¸e wyrazenia,˙ które wymagaja¸ oceny, lub nie zostanie zwrócony bład¸ oznaczajacy¸ przekroczenie maksymalnej ilosci´ zapytan´. Wynikiem jego działania jest lista relacji z przyporzadkowanymi¸

im ocenami. Kazda˙ uzyskana w ten sposób informacja jest cenna. Otrzymane wyniki daja¸ informacj¸e

nie tylko o tym, które wyrazenia˙ sa¸ poprawne, ale takze˙ o tym, które poprawne nie sa¸. Jako, ze˙ nigdy

nie ma absolutnej pewnosci,´ czy dana relacja ma sens, czy nie, przyporzadkowana¸ jej ocena pozwala

ustalic´ wyłacznie¸ prawdopodobienstwo´. Informacje o utworzonych relacjach wypisywane sa¸ w logach aplikacji: 4.3.1.

Ł. Wróbel

4.3. Opis działania aplikacji

Listing 4.1: Fragment logów wyswietlanych´ podczas tworzenia relacji.

```
2011 03 05
1 3 : 3 3 : 2 8
```

[INFO]
o s o b a
l u b i :
139000

2011 03 05
1 3 : 3 4 : 3 6
[INFO]
o s o b a ma
r a s e˛ :
0

2011 03 05
1 3 : 3 4 : 3 8
[INFO]
c a ł y
j a k
o s o b a : 0

2011 03 05
1 3 : 3 4 : 4 0
[INFO]
o s o b a
wie :
93500

2011 03 05
1 3 : 3 4 : 4 2
[INFO]
o s o b a
j e s t
s a˛ s i a d e m :
0

2011 03 05
1 3 : 3 4 : 4 4

[INFO]
o s o b a
t o
s w o j e : 0

2011 03 05
1 3 : 3 4 : 4 5
[INFO]
l u d
ma
o s o b e ̨ :
0

2011 03 05
1 3 : 3 5 : 0 4
[INFO]
o s o b a
z r o b i :
19400

2011 03 05
1 3 : 3 5 : 3 7
[INFO]
c z ł o w i e k
j e s t
o s o b a ̨ : 1550000

2011 03 05
1 3 : 3 6 : 4 4
[INFO]
·

j a k
c z ł o w i e k :
26800

k az d y

2011 03 05
13:37:15
[INFO]
swoje
jest
zwierzęciem:
0

2011 03 05
13:37:16
[INFO]
swoje
jest
rasą: 0

2011 03 05
13:37:18
[INFO]
swoje
jest
sąsiadem:
0

2011 03 05
13:37:20
[INFO]
swoje
jest
człowiekiem:
0

2011 03 05
13:37:33
[INFO]
człowiek
wie:
398000

2011 03 05
13:37:47

[INFO]
oko
t o
z w i e r z e ̨ : 0

2011 03 05
1 3 : 3 7 : 4 9
[INFO]
oko
t o
r a s a : 0

2011 03 05
1 3 : 3 7 : 5 0
[INFO]
oko
t o
s a ̨ s i a d :
0

2011 03 05
1 3 : 3 8 : 2 7
[INFO]
nowy
s a ̨ s i a d : 50500

2011 03 05
1 3 : 3 8 : 2 8
[INFO]
nowy
c z ł o w i e k :
29200

2011 03 05

1 3 : 3 8 : 3 0
[INFO]
oko
j e s t
z w i e r z e ̨ c i e m :
0

2011 03 05
1 3 : 3 8 : 3 2
[INFO]
oko
j e s t
r a s a ̨ :
0

2011 03 05
1 3 : 3 8 : 3 4
[INFO]
oko
j e s t
s a ̨ s i a d e m : 0

2011 03 05
1 3 : 3 8 : 3 5
[INFO]
oko
j e s t
c z ł o w i e k i e m :
0

2011 03 05
1 3 : 4 0 : 1 7
[INFO]
c z ł o w i e k
musi : 1280000

2011 03 05
1 3 : 4 1 : 1 4
[INFO]
oko
c z ł o w i e k a :

50500

2011 03 05
1 3 : 4 1 : 1 6
[INFO]
p a s t e r s k i
j a k
z w i e r z e¸ :

0

2011 03 05
1 3 : 4 1 : 1 8
[INFO]
p a s t e r s k i
j a k
r a s a :
0

2011 03 05
1 3 : 4 1 : 4 8
[INFO]
c z ł o w i e k
j e s t
c a ł y :
342000

4.3.2. Tryb offline

Rysunek 4.10: Wybór trybu offline

Główna¸ róznic˙a¸ mi¸edzy trybem offline, a trybem online jest brak koniecznosci´ łaczenia¸ si¸e z Internetem. W tym stanie aplikacja nie wykorzystuje pajaka¸ internetowego i nie wylicza ocen relacji, a jedynie wyswietla´ te, które zostały pobrane z bazy danych lub plików. Działanie programu w tym trybie jest

naturalnie o wiele szybsze. Jako, ze˙ nie jest mozliwe˙ okreslenie´ poprawnosci´ relacji, uzytkownik˙ nie musi

podawac´ słowa kluczowego. Graf zostanie zbudowany na podstawie informacji, które juz˙ zostały zebrane. W trybie tym istnieje takze˙ mozliwo˙s´c´ wczytania relacji z plików i ich zapis do bazy danych.

4.3.3. Symulator

Tryb symulatora zaimplementowany został wyłacznie¸ na potrzeby prezentacji idei działania apli-kacji. Nadaje on nowo utworzonym relacjom stałe oceny poprawnosci,´ aby zostały one wyswietlone´ na

grafie. Ustalanie poprawnosci´ w ten sposób oczywiscie´ nie ma nic wspólnego z rzeczywistosci´a,¸ pozwala jednak przedstawic´ sposób działania programu i wszystkie jego funkcje bez koniecznosci´ korzystania z mechanizmu sprawdzania poprawnosci´. Zrzut ekranu 4.11 przedstawia wybór symulacji sprawdzania poprawnosci´ odpowiednio dla trybu offline i online.

Rysunek 4.11: Wybór trybu pracy z symulatorem

4.4. Generacja specjalistycznego grafu LHG

Ostatnim etapem działania aplikacji jest wygenerowanie grafu przyzwyczajen´ lingwistycznych w uproszczonej postaci, który zaprezentuje znalezione relacje z przypisanymi do nich wartosciami´. Po zakonczeniu´ przetwarzania danych powstaje lista relacji złozona˙ zarówno z wczytanych przez uzytkow˙-nika, jak i uzyskanych na podstawie analizowanego tekstu. Słowa, które wchodza¸ w skład wyraze˙n´ staja¸ si¸e w¸ezłami i zapisane sa¸ w formie bazowej. W zalezno˙sci´ od tego, jaka¸ cz¸es´c´ mowy reprezentuja,¸ przy-

porzadkowany¸ jest im odpowiedni kolor zgodnie z tabela:¸ 4.1. Słowa z tym samym zródłosłowem,´ lecz

b¸edace¸ róznymi˙ cz¸esciami´ mowy sa¸ tez˙ róznymi˙ wierzchołkami. Przykład grafu obrazujacego¸ zrózni˙-cowanie kolorów w zalezno˙sci´ od cz¸esci´ mowy, jaka¸ reprezentuje słowo w wierzchołku pokazuje zrzut ekranu 4.12.

Rysunek 4.12: Przykładowy graf z wierzchołkami b¸edacymi¸ róznymi˙ cz¸esciami´ mowy.

Poprzez klikni¸ecie na wierzchołku grafu uzytkownik˙ ma mozliwo˙s´c´ odczytania, jaka¸ cz¸esci´a¸ mowy jest słowo, które dany wierzchołek reprezentuje. Informacja ta nie tylko okreslona´ jest za pomoca¸ koloru zgodnego z tabelka¸ 4.1, ale takze˙ jest wyswietlana´ w panelu informacyjnym, jak pokazuje zrzut ekranu 4.13.

Rysunek 4.13: Informacja o konkretnym wierzchołku wyswietlona´ na grafie.

Kraw¸edzie grafu definiuja¸ połaczenia¸ mi¸edzy dwoma wyrazami. Nie odpowiadaja¸ one bezposred´-nio relacji, ale raczej grupie relacji mi¸edzy wierzchołkami. Zwrot kraw¸edzi okresla´ rol¸e jaka¸ dany w¸ezeł

pełni w ich połaczeniu¸. Zawsze jest on skierowany od nadrz¸ednego wyrazu w wyrazeniu˙ do podrz¸ednego.

Mi¸edzy dwoma wierzchołkami moze˙ zatem wystapi¸c´ co najwyzej˙ dwie kraw¸edzie o róznych˙ zwrotach. Relacje, które uznane zostały za niepoprawne (maja¸ ocen¸ 0) nie sa¸ wyswietlane´ na grafie, co znacznie poprawia jego czytelnos´c´. Decyzja, czy dane relacje utworza¸ kraw¸ed´z, która nast¸epnie zostanie wyswie´-tlona na grafie zalezy˙ od sredniej´ oceny relacji mi¸edzy słowami reprezentowanymi przez wierzchołki. Ich ilos´c´ jest ograniczona przez parametry opisane w sekcji 4.2.1. Tylko kraw¸edzie o najwi¸ekszej sredniej´ ocenie relacji zostana¸ wyswietlone´.

Relacje utworzone na podstawie analizy hierarchicznej struktury istniejacych¸ powiaza¸n´ sa¸ wyswie´-

tlane tylko na z˙adanie¸ uzytkownika,˙ aby nie zaciemniac´ grafu. Powstałe w ten sposób kraw¸edzie ozna-czone sa¸ kolorem czerwonym, aby łatwo mozna˙ było je odrózni˙c´ i przesledzi´c´ scie´zk¸˙e rozumowania aplikacji. Liczba dorysowanych w ten sposób kraw¸edzi zalezy˙ od zdefiniowanego przez uzytkownika˙

w głównym oknie programu parametru Maksymalna ilos´c relacji spropagowanych opisanego w sekcji 4.2.1. Graf z widocznymi spropagowanymi relacjami został zaprezentowany na zrzucie ekranu 4.14.

Rysunek 4.14: Widok pokazujacy¸ spropagowane relacje na grafie.

5. Prezentacja działania aplikacji

Aby dokładnie zaprezentowac´ działanie aplikacji przesled´zmy przykład jej wykorzystania w celu wygenerowania struktury relacji dla zadanych słów. Załózmy,˙ ze˙ czatbot rozmawiajacy¸ własnie´ z uzyt˙-kownikiem chce przeanalizowac´ otrzymana¸ wypowied´z pod katem¸ okreslonych´ słów kluczowych. Jest to konieczne, jezeli˙ chce nawiaza¸c´ do cech charakterystycznych tematu rozmowy, lub tez˙ sformułowac´ inteligentne pytanie. To, czy czatbot „zrozumie” niektóre elementy rozmowy zaleze˙c´ b¸edzie od tego jakie b¸eda¸ jego skojarzenia, czyli relacje, które posiada w bazie danych.

Przykładowo dla wyrazu gepard dane wejsciowe´ aplikacji moga¸ wyglada¸c´ jak na zrzucie ekranu 5.1.

Oczywiscie´ mozliwe˙ jest wykorzystanie wi¸ekszej ilosci´ zasobów (zarówno stron internetowych, jak i korpusów tekstów).

Rysunek 5.1: Przykładowe dane wejsciowe´ dla programu.

Ze 100 wyników zwróconych przez wyszukiwark¸e Google dla słowa gepard pobierany jest tekst, który nast¸epnie dzielony jest na zdania, a te z kolei na wyrazy, które pojawiaja¸ si¸e w kontekscie´ słowa gepard. Najcz¸esciej´ pojawiajace¸ si¸e wyrazy sa¸ nast¸epnie podstawiane do szablonów relacji wraz ze sło-wem gepard i ich poprawnos´c´ jest sprawdzana za pomoca¸ wyszukiwarki Google. Im wi¸ecej wyników zostanie zwróconych dla okreslonej´ frazy, tym cz¸esciej´ wyst¸epuje ona w Internecie, co posrednio´ prze-kłada si¸e

takze˙ na jej poprawnos´c´ j¸ezykowa¸. Gdy do wyszukiwarki trafi zbyt wiele zapytan,´ tworzenie nowych relacji zostaje przerwane i wyswietlone´ zostaja¸ uzyskane do tej pory wyniki. Sytuacj¸e taka¸ ob-razuje wycinek logów 5.

2011 04 09
1 1 : 3 2 : 5 6
[INFO]
˙
g e p a r d : 17

uzywany

2011 04 09
1 1 : 3 2 : 5 8
[INFO]
w i e l k i
j a k
g e p a r d : 0

2011 04 09
1 1 : 3 2 : 5 8
[INFO]
Z a k on´ c z o n o
t w o r z e n i e r e l a c j i po ds taw ow ych .

2011 04 09
1 1 : 3 2 : 5 9
[INFO]
f i r m a
ma
z d j e¸ c i e :
0

2011 04 09
1 1 : 3 3 : 0 1
[INFO]
f i r m a
ma
z w i e r z e¸ :
0
z a p y t a n´

2011 04 09
1 1 : 3 3 : 0 9
[ERROR]
P r z e k r o c z o n o l i m i t

do g o o g l e .

2011 04 09
1 1 : 3 3 : 0 9
[INFO]
c z ł o w i e k
t o f i r m a :
0

2011 04 09
1 1 : 3 3 : 0 9
[INFO]
Z a k on´ c z o n o t w o r z e n i e
r e l a c j i
p r z e c h o d n i c h .

Jak widac´ na zrzucie ekranu 5.2 utworzony został graf przyzwyczajen´ lingwistycznych, który obra-zuje wycinek relacji uzyskanych dla wyrazu gepard przy wykorzystaniu zdefiniowanej wczesniej´ bazy wiedzy. Zgodnie z parametrem, jaki został podany, maksymalnie 10 kraw¸edzi łaczy¸ si¸e z pojedynczym wierzchołkiem, co przy ocenach utworzonych relacji pozwala wyswietli´c´ 8 wierzchołków reprezentuja¸-cych słowa zwiazane¸ relacjami z korzeniem. Popularne czasowniki takie jak by´c, móc i mie´c niebezpod-stawnie łacz¸a¸ si¸e z wieloma rzeczownikami. Rzeczownik firma pojawia si¸e, gdyz˙ GEPARD TRANS jest nazwa¸ dobrze rozreklamowanej w Internecie polskiej firmy przewozowej. Kolor czarny został przypisany gepardowi, choc´ w rzeczywistosci´ odnosi si¸e on raczej do c¸etek niz˙ do samego geparda.

Rysunek 5.2: Graf relacji dla słowa gepard.

Na pełnej liscie´ utworzonych relacji zaobserwowac´ mozna˙ relacje, które nie zostały zaprezentowane

na grafie ze wzgl¸edu na zbyt niska¸ wartos´c´ kraw¸edzi, takie jak uzywany˙ gepard (GEpard to nazwa dealera samochodów), czy gepard zostanie. Lista ta widoczna jest na zrzucie ekranu 5.3.

Rysunek 5.3: Przykładowa lista relacji wygenerowanych dla słowa gepard.

Dla wyrazu strus´ baza wiedzy zbudowana została na podstawie 300 wyników z wyszukiwarki Go-ogle, co nieco poprawiło jakos´c´ utworzonych relacji. Graf LHG został zaprezentowany na zrzucie ekranu 5.4. Znów pojawiły si¸e na nim czasowniki by´c i mie´c, ale wyswietlone´ zostały takze˙ czasowniki bardziej charakterystyczne dla strusia, jak chowa´, czy biega´c. Warto zauwazy˙c,´ ze˙ biblioteka CLP wyra´znie rozróznia˙ słowa biega´c i biec. Wyswietlone´ przymiotniki, jakie charakteryzuja¸ strusia to afryka´nski, szybki i

polski. Brak rozróznienia˙ cz¸esci´ mowy w momencie analizy zdania i koniecznos´c´ wykorzystania wszystkich dopasowan´ zwróconych przez bibliotek¸ CLP skutkuje tym, ze˙ jako poprawna relacja uznane zostało wyrazenie˙ strus´ szybki, gdzie szybki jest dopełniaczem wyrazu szybka. Google zwróciło wiele wyników, pomimo iz˙ wydawałoby si¸e, ze˙ kolejnos´c´ wyrazów nie jest poprawna z gramatycznego punktu widzenia. Okazuje si¸e jednak, ze˙ gra Strus´ Szybki Lopez jest na tyle popularna w Internecie, ze˙ algorytm sprawdzania poprawnosci´ uznaje ta¸ relacj¸ za poprawna¸.

Rysunek 5.4: Graf relacji dla słowa strus´.

Interesujace¸ informacje mozna˙ takze˙ odczytac´ z pełnej listy relacji zaprezentowanej na zrzucie ekranu 5.5. Mimo, iz˙ relacja mówiaca,¸ ze˙ strus´ jest ptakiem zwróciła kilkaset wyników, to jednak nie została zaprezentowana na grafie LHG ze wzgl¸edu na limit 10 kraw¸edzi dla jednego wierzchołka. Prze-widzenie, czy okreslony´ limit ograniczy wyswietlenie´ wartosciowych´ informacji, gdy b¸edzie zbyt mały,

czy pozwoli na wyswietlenie´ niepotrzebnych relacji, zaciemniajacych¸ graf, gdy b¸edzie zbyt duzy,˙ jest trudne do przewidzenia. Na liscie´ widoczne sa¸ takze˙ z pozoru niezrozumiałe relacje takie jak kojot stru-sia, czy p¸edziwiatr to strus´. Nawiazuj¸a¸ one jednak do popularnej bajki „Strus´ P¸edziwiatr” studia Warner Bros.

Rysunek 5.5: Przykładowa lista relacji wygenerowanych dla słowa strus´.

Aby wygenerowac´ wi¸ecej relacji przechodnich na podstawie relacji juz˙ istniejacych¸ wystarczy wczy-tac´ te relacje, czy to z plików, czy bazy danych. Nie jest konieczne wykorzystywanie pajaka¸ interneto-wego, wi¸ec ilos´c´ linków do pobrania mozna˙ ustawic´ na 0. Jezeli˙ z bazy danych lub z pliku nie zostana¸ pobrane zdania, na podstawie których powstac´ moga¸ nowe relacje, aplikacja tworzyc´ b¸edzie wyłacz¸-

nie relacje na podstawie analizowania istniejacej¸ hierarchii i wyciagania¸ pseudo-logicznych wniosków.

Przykładowa konfiguracja takich danych wejsciowych´ zaprezentowana została na zrzucie ekranu 5.6.

Rysunek 5.6: Przykładowe dane wejsciowe´ dla generacji dodatkowych relacji przechodnich.

Po pobraniu relacji z bazy danych lub plików, graf przyzwyczajen´ lingwistycznych rozrasta si¸e, a słowa, które wczesniej´ charakteryzowały pojedyncze wyrazy łacz¸a¸ si¸e z kilkoma innymi tworzac¸ bar-dziej rozbudowana¸ struktur¸e. Nadal obowiazuj¸a¸ jednak reguły dotyczace¸ maksymalnej ilosci´ kraw¸edzi dla jednego wierzchołka, które ustalone zostały w momencie definiowania danych wejsciowych´. Taki bardziej rozbudowany graf zawierajacy¸ zarówno relacje z dwóch poprzednich przykładów, jak i kilka innych pokazany został na zrzucie ekranu 5.7.

Rysunek 5.7: Przykładowy graf relacji dla kilku słów.

Wi¸eksza struktura tego typu pozwala na podejmowanie przez aplikacj¸e prób utworzenia relacji na podstawie analizy hierarchii. Na zrzucie ekranu 5.8 widac´ owoce tego procesu. Wyswietlone´ zostały nowe kraw¸edzie, które powstały na podstawie nowych propozycji relacji. Jak widac´ na zrzucie ekranu 5.2 poprawna jest relacja gepard jest zwierz¸eciem. Choc´ nie została ona wyswietlona´ na bardziej rozbudowanym grafie, to jednak została uwzgl¸edniona w procesie wnioskowania. Jako, ze˙ poprawna jest takze˙ relacja zwierz¸ lubi, aplikacja uznała łacz¸ac¸ te dwa fakty, ze˙ rzeczownik gepard i czasownik lubic´ takze˙

moga¸ byc´ połaczone¸ relacja¸. Weryfikacja przeprowadzona na podstawie ilosci´ wyników zwróconych

przez wyszukiwark¸e Google potwierdziła poprawnos´c´ takiego połaczenia,¸ co zaowocowało powstaniem nowej czerwonej kraw¸edzi na grafie, obrazujacej¸ spropagowana¸ relacj¸. Dokładna lista wszystkich relacji została zaprezentowana na zrzucie ekranu 5.9.

Rysunek 5.8: Przykładowy graf relacji wraz z relacjami przechodnimi dla kilku słów.

Rysunek 5.9: Przykładowa lista relacji wygenerowanych dla kilku róznych˙ słów.

W przykładach zaprezentowanych powyzej˙ starałem si¸e pokazac´ sposób działania aplikacji i uzy-skiwane przez nia¸ wyniki. Na efekt działania programu silnie wpływaja¸ trendy panujace¸ aktualnie w Internecie, a takze˙ reklamy rozmaitych produktów, których nazwy cz¸esto łacz¸a¸ słowa w nieoczekiwany

sposób. Homonimy rozrózniane˙ sa¸ poprzez wykorzystanie wszystkich cz¸esci´ mowy zwróconych przez bibliotek¸ CLP i przekazanie ich jako zapytanie do wyszukiwarki w odpowiedniej formie. Dwa identycz-

nie zapisane wyrazy w formie bazowej, b¸edace¸ jednak róznymi˙ cz¸esciami´ mowy, rozróznione˙ zostana¸ na podstawie reguł akomodacyjnych.

Ł. Wróbel

6. Podsumowanie wyników pracy i wnioski

6.1. Podsumowanie

Współczesna technologia wcia¸z˙ jest daleka od skonstruowania inteligentnego robota, który byłby w stanie idealnie komunikowac´ si¸e z lud´zmi. Taka maszyna musi znac´ nie tylko gramatyk¸e danego j¸ezyka,

ale takze˙ rozumiec´ znaczenie wypowiadanych słów na poziomie wyzszym˙ niz˙ prosta poprawnos´c´ j¸ezykowa. Problem badania znaczenia słów, aby nast¸epnie poprawnie uzy˙c´ ich podczas konwersacji nie

moze˙ sprowadzac´ si¸e wyłacznie¸ do analizowania ich kontekstu w pojedynczym zdaniu lub wypowiedzi. Mimo, iz˙ rozszyfrowanie formy frazeologicznej słowa nie jest zadaniem skomplikowanym, to sformu-łowanie nie tylko poprawnej gramatycznie, ale i logicznie wypowiedzi zalezy˙ juz˙ od wi¸ekszej liczby czynników. Podobnie jak człowiek, który po raz pierwszy usłyszał dane słowo, jest w stanie wywnio-

skowac´ jego znaczenie z kontekstu zdania i nast¸epnie uzywa˙c´ go poprawnie, tak i mozliwe˙ jest to dla czatbota. Niezb¸ednym jednak elementem, który warunkuje ta¸ umiej¸etnos´c´ jest posiadanie odpowied-

niej wiedzy, na podstawie której stosujac¸ logiczne rozumowanie mozliwe˙ staje si¸e uzyskanie nowych,

poprawnych informacji. Stworzona przeze mnie aplikacja jest propozycja¸ rozwiazania¸ tego problemu. Dostarcza ona danych, które wykorzystane przez czatbota pozwola¸ mu zarówno na znacznie lepsze rozumienie analizowanych przez niego wypowiedzi, jak i formułowanie własnych logicznie poprawnych wyraze˙n´.

Najwi¸ekszym atutem mojego rozwiazania¸ jest fakt, ze˙ nie ogranicza si¸e ono do informacji zawartych

wyłacznie¸ w korpusach tekstów. Jakkolwiek sa¸ one i długo pozostana¸ nieocenionym zródłem´ danych dla czatbotów, to ilos´c´ słów zawarta nawet w najwi¸ekszych tego typu repozytoriach jest nieporównanie mniejsza niz˙ mnogos´c´ form wyrazowych w Internecie. Do osiagni¸ecia zamierzonego celu niezb¸edna

była baza wiedzy, która nie ogranicza si¸e wyłacznie¸ do pojedynczych wyrazów, ale takze˙ ich mozliwych˙ połacze¸n´. Podczas, gdy korpusy tekstów licza¸ dziesiatki¸ lub setki tysi¸ecy wystapie¸n´ wielu popularnych

wyrazów, zasoby oferowane przez Google słowa takie licza¸ w dziesiatkach,¸ czy nawet setkach milionów. Dodatkowym argumentem przemawiajacym¸ na korzys´c´ korzystania bezposrednio´ z wiedzy zawartej w Internecie jest fakt ciagłej¸ zmiennosci´ umieszczonych tam tekstów. J¸ezyk ewoluuje, a zabytki pismien´-nictwa, które cz¸esto sa¸ integralna¸ cz¸esci´a¸ korpusów nie zawieraja¸ wielu współczesnie´ uzywanych˙ form i wyraze˙n´. Z drugiej strony zawieraja¸ formy, które rzadko pojawiaja¸ si¸e w mowie potocznej dzisiejszych czasów. Ale nawet znalezienie rzadkich, czy archaicznych sformułowan´ jest łatwiejsze, gdy ma si¸e do dyspozycji cała¸ wiedz¸ zgromadzona¸ w Internecie.

Spore niebezpieczenstwo´ dla poprawnosci´ j¸ezykowej zgromadzonych danych stanowi sposób ich za-

pisu na stronach internetowych. W szczególnych przypadkach niepoz˙adane¸ znaki, lub fragmenty kodu

moga¸ znale´c´ si¸e w tworzonej bazie wiedzy. Był to główny powód, dla którego zdecydowałem si¸e dos´c´

restrykcyjnie podejs´c´ do filtrowania niepoz˙adanych¸ informacji. Nie wszystkie zagrozenia˙ dla poprawnosci´ przechowywanych danych maja¸ podłoze˙ opierajace¸ si¸e o nietypowe znaki lub ich kombinacje. Bardzo wiele portali internetowych zawiera tresci´ niepozbawione bł¸edów. Nie chodzi tylko o literówki w artykułach, czy esejach. Prawdziwa¸ plaga¸ staja¸ si¸e wpisy na rozmaitych forach internetowych i blo-gach, a takze˙ komentarze do wiadomosci´ na portalach informacyjnych. Zarówno poziom ortografii, jak

i stylistyki, a nawet logiki wielu wypowiedzi pozostawia wiele do zyczenia˙. Wsród´ wpisów tego typu cz¸estym zjawiskiem sa¸ boty reklamujace,¸ które na wielu forach automatycznie tworza¸ posty składajace¸

si¸e wyłacznie¸ z pisanych jednym ciagiem¸ haseł. Majac¸ do dyspozycji wiele zródeł´ informacji pojedyncze

.

6.2. Dalsze mozliwosci´ rozwoju

przypadki tego typu byłyby z pewnosci´a¸ niezauwazalne˙
. Proceder ten jest jednak powszechny i automa-

tycznie propagowany przy pomocy złosliwych´ pajaków¸
internetowych na bardzo wiele stron. Ilos´c´ tego

typu zdegenerowanych danych moze˙ miec´ wpływ na proces automatycznego decydowania o poprawno-sci´ wypowiedzi.

Podstawowa wersja pajaka¸ internetowego, która¸ zdecydowałem si¸e zaimplementowac´ spełnia swoje zadanie. Wyszukiwarka Google dostarcza wyniki dos´c´ dobrej jakosci,´ co bezposrednio´ przekłada si¸e na jakos´c´ stron odwiedzanych przez robota. Nawet jezeli˙ pomijane sa¸ czasem zasoby, które potencjalnie

mogłyby zostac´ wykorzystane do budowania bazy wiedzy, to najwazniejszym˙ aspektem działania pajaka¸
´ ´
jest zebranie reprezentatywnych zródeł´. Zródła te maja¸ posłuzyc zgromadzeniu informacji o słowach, które jak najcz¸esciej´ pojawiaja¸ si¸e w kontekscie´ wyrazu, który w danej chwili jest obiektem zaintereso-wania programu.

Ekstrakcja danych i tworzenie relacji przebiega bardzo sprawnie. Waskim¸ gardłem aplikacji jest

zdecydowanie sprawdzanie poprawnosci´ relacji. Jako, ze˙ Google nie udost¸epnia API pozwalajacego¸ na dost¸ep do wyników wyszukiwania, konieczne jest pozyskiwanie ich bezposrednio´ parsujac¸ otrzymana¸ stron¸e wynikowa¸. Cz¸este automatyczne zapytania sa¸ jednak bardzo skutecznie rozpoznawane przez algorytmy wyszukiwarki, co skutkuje blokada¸ mozliwo˙sci´ korzystania z niej na kilka godzin, lub konieczno-sci´ uzupełnienia formularza zwanego CAPTCHA (ang. Completely Automated Public Turing test to tell

Computers and Humans Apart), którego zadaniem jest weryfikacja, czy uzytkownik˙ jest człowiekiem. Zaproponowana przeze mnie metoda oceny poprawnosci´ relacji nie jest niezawodna. Jej główna niedoskonałos´c´ wia¸ze˙ si¸e z algorytmem wyszukujacym¸ Google, który mozliwie˙ jak najbardziej poszerza

zadane przez uzytkownika˙ kryteria, aby zwrócic´ jak najwi¸eksza¸ liczb¸e wyników. Ignorowane sa¸ zatem wszelkie znaki interpunkcyjne (zarówno w wyszukiwanej frazie, jak i potencjalnych wynikach), które moga¸ miec´ znaczacy¸ wpływ na jakos´c´ rezultatu. Prostym przykładem jest sprawdzanie poprawnosci´ wy-razenia˙ kot to samochód. Grupowanie takie wydaje si¸e niedorzeczne, ale Google znajdzie to wyrazenie˙ w tekscie´ zawierajacym¸ kolejno zdania: Obok przebiegł przerazony˙ kot. To samochód przejezd˙zaj˙acy¸ obok tak go przestraszył. Fakt, ze˙ mi¸edzy kot, a to samochód jest kropka zostanie całkowicie zignorowany.

Problem stanowia¸ takze˙ tytuły ksia¸zek,˙ filmów lub artykułów, które cz¸esto celowo formułowane sa¸ w sposób syntaktycznie lub semantycznie niepoprawny. Mozna˙ by pomysle´c,´ ze˙ nie ma w tym nic złego. O rozmaitych komentarzach pod artykułami na stronach informacyjnych lub wpisach na forach, czy blogach mozna˙ powiedziec´ to samo. Niestety tytuły dzieł publicznie dost¸epnych sa¸ nieporównanie szerzej rozpowszechnione, co sprawia, ze˙ Google podczas wyszukiwania natrafi na nie o wiele cz¸esciej´ przeszukujac¸ rozmaite strony zawierajace¸ reklamy, zapowiedzi, czy recenzje.

6.2. Dalsze mozliwo˙sci´ rozwoju

W aplikacji istnieje wcia¸z˙ wiele obszarów, które mozna˙ poprawic,´ aby polepszyc´ zarówno jakos´c´ otrzymywanych wyników, jak i wydajnos´c´ działania. Główne elementy, które mozna˙ rozwina¸c´ w przyszłosci´ to:

1.Algorytm sprawdzania poprawnosci´ relacji.

Ograniczenia narzucone przez Google sa¸ najwi¸ekszym mankamentem zaproponowanego przeze mnie rozwiazania¸. Koniecznos´c´ oczekiwania pomi¸edzy zapytaniami, a takze˙ mozliwo˙s´c´ blokady zapytan´ na kilka godzin znacznie obniza˙ wydajnos´c´ aplikacji. Niestety inne wypróbowywane przeze mnie wyszukiwarki internetowe zwracaja¸ nieporównanie mniej wyników, co w przypadku sprawdzania poprawnosci´ metoda¸ statystyczna¸ ma spore znaczenie. Byc´ moze˙ w przyszłosci´ znów dost¸epne b¸edzie API, które pozwoli na zautomatyzowane korzystanie z wyszukiwarki Google, co

z pewnosci´a¸ znacznie poprawi wydajnos´c´ programu. Akceptowalnym rozwiazaniem¸ byłby takze˙ znacznie bardziej zaawansowany mechanizm wysyłajacy¸ zapytania wykorzystujacy¸ cz¸este zmiany

przegladarek¸ podawanych w nagłówku wysyłanego zapytania, a takze˙ korzystajacy¸ z serwerów proxy.

2.Szablony relacji

Jezeli˙ problem skuteczniejszego mechanizmu sprawdzania poprawnosci´ relacji zostałby rozwia¸-zany, to nic nie stałoby na przeszkodzie, aby wprowadzic´ wi¸ecej rodzajów relacji, które znacznie wzbogaciłyby wartos´c´ lingwistyczna¸ tworzonej bazy wiedzy. Kolejnym krokiem w rozwoju tego elementu systemu mogłoby byc´ wyodr¸ebnienie spójników jako oddzielnych cz¸esci´ mowy, a nie jak
obecnie traktowanie ich jako „sztywnych” elementów zespalajacych¸ połaczenie,¸ mi¸edzy bardziej

znaczacymi¸ cz¸esciami´ mowy. Na podstawie analizy składniowej tekstu mozliwe˙ byłoby dodanie automatycznego tworzenia szablonów relacji na podstawie cz¸esto spotykanych struktur grama-tycznych.

3.Słownik fleksyjny

Kolejna¸ mozliwo˙sci´a¸ ulepszenia aplikacji jest wzbogacenie słownika fleksyjnego o nowe formy wyrazowe, a takze˙ bardziej przejrzysty sposób okreslania´ i definiowania odmiany słów. Mozli˙-wos´c´ dokładnego okreslenia,´ odmiana której cechy wyrazu nas interesuje byłaby znacznie prak-tyczniejsza niz˙ wybór po indeksie w tablicy. Przykładowo majac¸ wyraz kot w formie bazowej, chcac¸ uzyskac´ dopełniacz liczby mnogiej musimy odwołac´ si¸e do okreslonego´ indeksu, który od-
powiada z˙adanej¸ formie. O wiele wygodniejsza byłaby mozliwo˙s´c´ podania parametrów formy, które nas interesuja¸ np. zwrocForme("kot", DOPELNIACZ, L_MN).

6.3. Porównanie z rozwiazaniami¸ o podobnej tematyce

Przykładowa, znaleziona w Internecie implementacja czatbota[17] napisana przez Tomasza J¸edrze-

jewskiego wykorzystuje prosty algorytm tworzenia wypowiedzi. Na podstawie rozmowy z uzytkowni˙-kiem, zapami¸etuje pojawiajace¸ si¸e zdania w postaci skierowanego grafu przedstawionego na rysunku 6.1.

Graf ten łaczy¸ wierzchołki odpowiadajace¸ uzytym˙ wyrazom skierowanymi kraw¸edziami, które odpowia-

daja¸ połaczeniom¸ mi¸edzy nimi. Wierzchołki $ i # oznaczaja¸ odpowiednio poczatek¸ i koniec zdania. Generowanie wypowiedzi polega na wyborze losowej scie´zki˙ pomi¸edzy elementami grafu od wierzchołka $ do #.

Rysunek 6.1: Graf obrazujacy¸ połaczenia¸ mi¸edzy wyrazami dla czatbota zaimplementowanego przez Tomasza J¸edrzejewskiego.

Algorytm tego typu jest bardzo prosty, a generowane przez czatbota wypowiedzi bardzo cz¸esto sa¸ niepoprawne nie tylko semantycznie, ale równiez˙ składniowo, gdyz˙ nie wykorzystywany jest zaden˙ słownik morfologiczny. Baza wiedzy, na podstawie której tworzone sa¸ zdania jest ograniczona wyłacznie¸ do danych zebranych podczas rozmowy z uzytkownikiem˙. Zebranie duzej˙ ilosci´ danych z wielu konwer-

sacji nie poprawi jakosci´ działania programu, gdyz˙ losowy wybór scie´zki˙ w grafie moze˙ powodowac´ zap¸etlenia, lub powstawanie długich pozbawionych sensu zdan´.

Rozwiazanie,¸ które zaproponowałem korzysta ze zdecydowanie bogatszej bazy wiedzy, co pozwala na generowanie wypowiedzi z wykorzystaniem słów, które nie pojawiły si¸e w trakcie rozmowy. Połacze¸-nia mi¸edzy wyrazami sa¸ poprawne syntaktycznie, a takze˙ odfiltrowywane sa¸ te, uznane za nie poprawne z punktu widzenia semantyki. O poprawnosci´ decyduje w głównej mierze popularnos´c´ fraz w Interne-cie, zatem bł¸edy choc´ sa¸ nieuniknione, pojawiac´ si¸e b¸eda¸ znacznie rzadziej. Baza wiedzy budowana jest poprzez wyszukiwanie informacji zwiazanych¸ z konkretnym tematem. Bezposrednio´ przekłada si¸e to na jakos´c´ otrzymywanych wyników, które sa¸ tym dokładniejsze, im wi¸eksza jest baza wiedzy, na podstawie której zostały wygenerowane.

Marcin Gadamer w swojej pracy[8] wykorzystał algorytm, który buduje graf przyzwyczajen´ lingwi-stycznych na podstawie korpusów tekstów. Generowanie wypowiedzi odbywa si¸e w interakcji z uzyt˙-kownikiem, który wskazuje kolejne słowo z listy zaproponowanych mozliwo˙sci´. Przykładowy fragment grafu zaprezentowany został na zrzucie ekranu 6.2. W bazie danych przechowywane sa¸ trójki wyrazów, z zachowaniem kolejnosci,´ w jakiej wyst¸epowały w analizowanych zdaniach. Na podstawie dwóch pierw-szych form wyrazowych z trójki prezentowane sa¸ mozliwe˙ poprawne formy kolejnej, wraz z okresleniem´ cz¸estosci´ wystapie¸n´.

Rysunek 6.2: Fragment grafu LHG z aplikacji Marcina Gadamera prezentujacy¸ cztery słowa wypowiedzi i proponowane nast¸epniki.

Algorytm ten dobrze sprawdza si¸e podczas automatycznej korekcji tekstu, jednak słabiej spisywałby si¸e w przypadku automatycznej generacji zdan´. Wrazliwo˙s´c´ na zap¸etlenia, a takze˙ powstawanie zbyt

długich wypowiedzi zostały rozwiazane¸ przez interakcj¸ z uzytkownikiem˙. Problemem jest takze˙ ilos´c´

danych przechowywanych w bazie wiedzy. Kazde˙ przeanalizowane zdanie rozkładane jest na n-2 trójek, gdzie n jest ilosci´a¸ wyrazów w zdaniu.

Generowana przez moja¸ aplikacj¸e uproszczona postac´ grafu LHG skupia si¸e w wi¸ekszym stopniu

na relacjach mi¸edzy konkretnymi parami wyrazów. Kazda˙ informacja odnosnie´ takiego połaczenia¸ jest istotna i posiada wartos´c´ dla czatbota. Nawet jezeli˙ do bazy danych zapisywane sa¸ niepoprawne relacje, to jest to takze˙ wazny˙ element wiedzy programu. Na podstawie zebranych danych nie da si¸e co prawda utworzyc´ pełnych zdan´ w j¸ezyku polskim, ale mozna˙ je wykorzystac´ podczas procesu „rozumienia”

sensu wypowiedzi uzytkownika,˙ a takze˙ generowania odpowiedzi zgodnej z tematem rozmowy.

Ł. Wróbel

7. Zakonczenie´

Cel jaki został postawiony w tej pracy to utworzenie uproszczonej wersji grafu przyzwyczajen´ lingwistycznych (LHG), który obrazuje relacje zachodzace¸ mi¸edzy wyrazami w j¸ezyku polskim, w szczególnosci´ modelujac¸ grupowanie wokół okreslonych´ cech i budow¸e hierarchicznej struktury w obr¸ebie danej grupy. Osiagni¸ecie tego celu sprowadzone zostało do utworzenia trzech głównych modułów aplikacji oraz interfejsu uzytkownika¨.

1.Pajak¸ internetowy

Aby wykorzystac´ wiedz¸ zgromadzona¸ w Internecie niezb¸edne było wykorzystanie pajaka¸ inter-netowego, który analizujac¸ tres´c´ odwiedzanych stron pozyskuje informacje, które nast¸epnie wy-korzystywane sa¸ do utworzenia bazy wiedzy. Utworzony przeze mnie robot wykorzystujac¸ wyniki z wyszukiwarki Google pobiera poprawne zdania w j¸ezyku polskim i zapisuje je w bazie danych stale powi¸ekszajac¸ dost¸epne do wykorzystania repozytorium wiedzy. Jego zadaniem jest takze¨ rozkład zdan´ na pojedyncze słowa, które nast¸epnie wykorzystywane sa¸ w mechanizmie tworzenia relacji. Oprócz pajaka¸ dodałem takze¨ mozliwo¨s´c´ wykorzystania korpusów tekstów znajdujacych¸ si¸e w plikach tekstowych, aby Internet nie był jedynym zródłem´ wiedzy, z którego aplikacja moze¨ korzystac´.

2.Kreator relacji

Zasadniczy mechanizm działania aplikacji miał za zadanie z wyników dostarczonych przez pajaka¸ utworzyc´ struktur¸e, która modeluje relacje mi¸edzy słowami w j¸ezyku polskim. Zastosowany przeze mnie algorytm zlicza ilos´c´ wystapie¸n´ wyrazów w kontekscie´ słowa, wokół którego tworzony jest graf relacji. Najcz¸esciej´ wyst¸epujace¸ sa¸ nast¸epnie podstawiane do pre-definiowanych szablonów,
które zawieraja¸ oba elementy relacji w okreslonych´ formach fleksyjnych tworzacych¸ poprawne

syntaktycznie wyrazenie˙. Poprawnos´c´ takiego wyrazenia˙ jest nast¸epnie sprawdzana za pomoca¸ wyszukiwarki Google. Ilos´c´ zwróconych wyników okresla´ „popularnos´c”´ frazy w Internecie, co przekłada si¸e na jej poprawnos´c´ zarówno syntaktyczna,¸ jak i semantyczna¸.

Wazn˙a¸ funkcja¸ kreatora relacji jest takze˙ proces pseudo-logicznego rozumowania. Wykorzystujac¸ znane juz˙ poprawne relacje i ich hierarchiczna¸ struktur¸e sprawdzane sa¸ potencjalne połaczenia¸ mi¸edzy słowami. Elementy nalez˙ace¸ do wspólnej grupy zestawiane sa¸ z cechami składników tejze˙ grupy. Podobnie wyrazy, które posiadaja¸ wspólne cechy sprawdzane sa¸ pod katem¸ przynalezno˙sci´ do wspólnej grupy.

3.Specjalistyczny graf przyzwyczajen´ lingwistycznych

Zadaniem uproszczonego grafu LHG było zilustrowanie modelu relacji w sposób czytelny dla

uzytkownika˙. Zostały na nim pokazane rózne˙ rodzaje połacze¸n´ mi¸edzy wyrazami oraz wartos´c´

liczbowa okreslaj´aca¸ ich poprawnos´c´. Relacje zapisane zostały jako wyrazenia˙ składajace¸ si¸e z

dwóch wyrazów w odpowiedniej formie połaczonych¸ ewentualnym spójnikiem i zaprezentowane

na skierowanych kraw¸edziach grafu. Na z˙adanie¸ uzytkownika˙ wyswietlone´ moga¸ zostac´ takze˙ relacje uzyskane z procesu tak zwanej „przechodniosci”,´ czyli wykonanego wczesniej´ pseudo-
logicznego procesu rozumowania i analizy istniejacych¸ połacze¸n´.

Popularnos´c´ tematyki sztucznej inteligencji jest obecnie bardzo duza,˙ a zainteresowanie urzadze¸-niami i aplikacjami zdolnymi porozumiewac´ si¸e z człowiekiem jest szczególnie wysokie. Roboty, które

komunikuja¸ si¸e z uzytkownikiem˙ i na biez˙aco¸ ucza¸ si¸e poszerzajac¸ swoja¸ baz¸e wiedzy przestaja¸ byc´ wytworem filmów science fiction, a zaczynaja¸ pojawiac´ si¸e juz˙ jako gotowe produkty. Wcia¸z˙ jeszcze charakteryzowane sa¸ raczej jako nowinki techniczne, lub modne gadzety,˙ ale wzrastajace¸ zapotrzebowa-nie moze˙ sprawic,´ ze˙ juz˙ niedługo napotykac´ je b¸edziemy w codziennym zyciu˙.

Zaproponowane przeze mnie rozwiazanie¸ jest zaledwie małym krokiem na drodze do stworzenia inteligentnego czatbota. Daje ono jednak w miar¸e zadowalajace¸ rezultaty, które moga¸ zostac´ wykorzy-stane przy tworzeniu nie tylko poprawnych gramatycznie, ale i logicznie zdan´. Wizja robota, który nie

tylko formułuje poprawne zdania, ale takze˙ rozumie kontekst poszczególnych słów, jakich uzywa,˙ jak i tych, które otrzymuje od rozmówcy, staje si¸e coraz bardziej realna. Pomimo wielu problemów, jakie wcia¸z˙ stoja¸ przed programistami, powstanie czatbota, który pozytywnie przejdzie test Turinga jest tylko kwestia¸ czasu.

Ł. Wróbel

Spis rysunków

SPIS RYSUNKÓW
53

Ł. Wróbel

Bibliografia

[1]Geofferey Landis. Out of Sight, Out Of Mind. http://www.geoffreylandis.com/sight.htp, 1999.

[2]GNU. WGet. http://www.gnu.org/software/wget/, 2010.

[3]Google Code. crawler4j. http://code.google.com/p/crawler4j/, 2010.

[4]ICONICO. HTML Text Extractor. http://www.iconico.com/htmlextractor/, 2010. [doste¸p: 2010-05-08].

[5]IPI PAN. Korpus IPI PAN. http://korpus.pl/, 2008.

[6]M. Junczys-Dowmunt. Zastosowanie automatów skonczonych´ stanów w przetwarzaniu je¸zyków naturalnych, 2008.

[7]Z. Klemensiewicz. Zarys składni polskiej. Warszawa, 1961.

[8]M. Gadamer. Automatyczna kontekstowa korekta tekstów na podstawie Grafu Przyzwyczajen´ Lingwistycznych (LHG) zbudowanego przez robota internetowego dla j¸ezyka polskiego, Kraków, 2009.

[9]M. Gadamer, A. Horzyk. Automatyczna kontekstowa korekta tekstów z wykorzystaniem grafu LHG. Computer Science AGH, Vol. 10, pp. 39-57, Kraków, 2010.

[10]Online Data Extractor. Online data extractor tool. http://www.onlinedataextractor. com, 2010.

[11]M. Piasecki. Cele i zadania lingwistyki informatycznej. Metodologie je¸zykoznastwa. Współczesne tendencje i kontrowersje, 2007.

[12]Piotr Pe¸zik, Narodowy Korpus Je¸zyka Polskiego . PELCRA. http://www.nkjp.uni.lodz. pl/, 2010. [doste¸p: 2010-03-10].

[13]Portal naukowy.pl. Gramatyka generatywna. http://www.naukowy.pl/encyklopedia/ Gramatyka_generatywna, 2010.

[14]Portal naukowy.pl. Historia lingwistyki. http://www.naukowy.pl/encyklopedia/ Historia_lingwistyki, 2010.
´´

[15]Z. Saloni and M. Swidzinski. Składnia współczesnego je¸zyka polskiego. PWN, 1985.

[16]The Web Robots Pages. About /robots.txt. http://www.robotstxt.org/robotstxt. html, 2010.

[17]Tomasz J¸edrzejewski. Piszemy czatbota. http://www.eioba.pl/a/1nrd/ piszemy-chatbota, 2007.

[18]Wapedia. Klasyfikacja cz¸esci´ mowy. http://wapedia.mobi/pl/Klasyfikacja_cz% C4%99%C5%9Bci_mowy, 2010.

BIBLIOGRAFIA
55

[19]Wikipedia. Ferdinand de saussure. http://pl.wikipedia.org/w/index.php?title= Ferdinand_de_Saussure&oldid=24025783, 2010.

[20]Wikipedia. Historia je¸zykoznastwa. http://pl.wikipedia.org/w/index.php? title=Historia_j%C4%99zykoznawstwa&oldid=23236068, 2010.

[21]Wikipedia. IBM WebFountain. http://en.wikipedia.org/w/index.php?title= IBM_WebFountain&oldid=391584874, 2010.

[22]Wikipedia. Je¸zykoznastwo. http://pl.wikipedia.org/w/index.php?title=J%C4% 99zykoznawstwo&oldid=23891961, 2010.

[23]Wikipedia. Lingwistyka komputerowa. http://pl.wikipedia.org/w/index.php? title=Lingwistyka_komputerowa&oldid=23235735, 2010.

[24]Wikipedia. Web crawler. http://en.wikipedia.org/w/index.php?title=Web_ crawler&oldid=399429057, 2010.

[25]Wikipedia. Web crawler. http://en.wikipedia.org/w/index.php?title=Web_ crawler&oldid=398058393,

2010.

[26]Wydawnictwo Naukowe PWN. Korpus Je˛zyka Polskiego Wydawnictwa Naukowego PWN. http://korpus.pwn.pl/, 2010.

[27] P. Zmigrodzki. Zwiazki˛ mie˛dzy składnikami w zdaniu polskim. Szkoła je˛zyka i kultury polskiej

Uniwersytet Slaski,˛ Katowice.

Ł. Wróbel