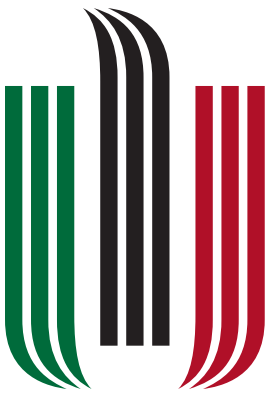


Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie

Wydział Informatyki, Elektroniki i Telekomunikacji

Katedra Informatyki

Praca dyplomowa magisterska

Automatyczne uzgadnianie wartości kategorii gramatycznych w polskich tekstach

Automatic reconciliation of morphosyntactic categories in Polish texts

Autor:	Marta Tomzik
Kierunek studiów:	informatyka
Opiekun pracy:	dr inż. Aleksander Smywiński-Pohl

Kraków 2016

Oświadczenie autora pracy

Uprzedzony o odpowiedzialności karnej na podstawie art. 115 ust. 1 i 2 ustawy z dnia 4 lutego 1994 r. o prawie autorskim i prawach pokrewnych (t.j. Dz.U. z 2006 r. Nr 90, poz. 631 z późn. zm.): „Kto przywłaszcza sobie autorstwo albo wprowadza w błąd co do autorstwa całości lub części cudzego utworu albo artystycznego wykonania, podlega grzywnie, karze ograniczenia wolności albo pozbawienia wolności do lat 3. Tej samej karze podlega, kto rozpowszechnia bez podania nazwiska lub pseudonimu twórcy cudzy utwór w wersji oryginalnej albo w postaci opracowania, artystyczne wykonanie albo publicznie zniekształca taki utwór, artystyczne wykonanie, fonogram, wideogram lub nadanie.", a także uprzedzony o odpowiedzialności dyscyplinarnej na podstawie art. 211 ust. 1 ustawy z dnia 27 lipca 2005 r. Prawo o szkolnictwie wyższym (t.j. Dz. U. z 2012 r. poz. 572, z późn. zm.) Za naruszenie przepisów obowiązujących w uczelni oraz za czyny uchybiające godności studenta student ponosi odpowiedzialność dyscyplinarną przed komisją dyscyplinarną albo przed sądem koleżeńskim samorządu studenckiego, zwanym dalej „sądem koleżeńskim", oświadczam, że niniejszą pracę dyplomową wykonałem osobiście i samodzielnie i że nie korzystałem ze źródeł innych niż wymienione w pracy.

.....

podpis

Spis treści

1. Wstęp

1.1. Definicja problemu

Jak podaje źródło [1], blisko 40 milionów osób używa języka polskiego jako ojczystego, co czyni go 26. najczęściej używanym językiem na świecie. Uchodzi on także za jeden z najtrudniejszych do nauki ze względu na skomplikowaną gramatykę - siedem przypadków, nagromadzenie form nieregularnych i wyjątków, zwłaszcza w odmianie rzeczowników, a także występowanie form dokonanych i niedokonanych w przypadku odmiany czasowników. Poprawne posługiwanie się polskim jest niełatwym zadaniem nie tylko dla cudzoziemców, ale także dla rodzimych użytkowników języka. Z tego powodu stworzenie poprawnego językowo tekstu jest złożonym i skomplikowanym procesem, szerzej opisanym w artykule [2], który wymaga między innymi:

- korekty, czyli sprawdzenia tekstu pod względem poprawności ortograficznej i interpunkcyjnej,
- adjustacji, czyli sprawdzenie poprawności leksykalno-stylistycznej zdań oraz układu graficznego całego tekstu,
- redakcji, czyli przygotowania tekstu pod względem stylistycznym i merytorycznym.

W dzisiejszych czasach cały czas dąży się do jak największej automatyzacji tych procesów. Część z wymienionych powyżej działań, jak na przykład korekta, została już z powodzeniem zautomatyzowana, a na rynku można znaleźć wiele różnych rozwiązań. Jednakże wciąż istnieją obszary, takie jak redakcja, w których narzędzi jest albo niewiele, albo mają one bardzo wąski zakres funkcjonalności. Niniejsza praca powstała w celu zbadania możliwości rozszerzenia automatycznej redakcji tekstu pisanego o kilka nowych funkcjonalności, szczegółowo opisanych w kolejnych rozdziałach.

1.2. Cel pracy

W języku polskim nie tylko gramatyka może sprawiać trudność - reguły ortografii i interpunkcji potrafią być równie zawiłe i pełne nieregularności. Dobrym przykładem jest choćby pisownia słów „Grzegorz” i „gżegżółka” albo różnica w postawieniu przecinków w wyrażeniach „uważam, że” oraz „chyba że”. Obecnie powszechne jest sprawdzanie poprawności interpunkcyjnej i ortograficznej w edytorach tekstu czy przeglądarkach internetowych. Dla przykładu - znane narzędzie korektorskie LanguageTool[1] pozwala na identyfikację i poprawę między innymi następujących błędów i potknięć językowych:

[1] Strona projektu: https://languagetool.org/pl/

- błędy fonetyczne - np. „we Łodzi” zamiast „w Łodzi”,
- błędy frazeologiczne, np. „nie dać sobie w kotlet dmuchać” zamiast „nie dać sobie w kaszę dmuchać”,
- błędy interpunkcyjne, np. „ten który” zamiast „ten, który”,
- błędy leksykalne, np. „bajońskie ceny” zamiast „wysokie ceny”,
- błędy odmiany, np. „z powoda suszy” zamiast „z powodu suszy”,
- błędy ortograficzne, np. „wydażyło” zamiast „wydarzyło”,
- błędy rodzaju gramatycznego, np. „zjadł pomarańcz” zamiast „zjadł pomarańczę”,
- błędy składniowe, np. „Mieszkam na ulicy Dworcowej” zamiast „Mieszkam przy ulicy Dworcowej”,
- błędy typograficzne, np. „mały , biały domek” zamiast „mały, biały domek”,
- błędy w szyku wyrazów,
- pisownia małą i wielką literą,
- pleonazmy,
- prawdopodobne literówki,
- wyrazy modne i nadużywane.

Użycie takich narzędzi to dobre rozwiązanie w przypadku, gdy błąd językowy faktycznie został popełniony, a użytkownik albo nie jest tego świadomy, albo nie ma czasu i chęci na jego ręczną poprawę. Zdarzają się natomiast sytuacje, gdy tekst pomimo zachowania poprawności językowej wymaga dalszej redakcji użytkownika. Dzieje się tak chociażby wtedy, gdy zachodzi potrzeba zamiany występującego w tekście słowa na jego synonim albo słowo o podobnym znaczeniu. Jeśli autor tekstu zdecyduje się użyć wyrazu o innym rodzaju gramatycznym, musi także ręcznie pozamieniać końcówki wszystkich zależnych wyrazów. Przykładowo w stosunkowo krótkim tekście:

„Ala ma **małego burego** kota. **Ten** kot bywa **agresywny** i drapie.”

po zamianie słowa „kot” w rodzaju męskim na słowo „kotka” w rodzaju żeńskim bez ponownego uzgodnienia końcówek ten sam tekst będzie wyglądał następująco:

„Ala ma *małego burego* kotkę. *Ten* kotka bywa a*gresywny* i drapie.”

Takie działanie generuje poważne błędy fleksyjne wymagające kolejnej poprawy. Po ich usunięciu tekst będzie miał postać:

„Ala ma **małą burą** kotkę. **Ta** kotka bywa **agresywna** i drapie.”

W przypadku wielu wystąpień tego samego słowa w dłuższym tekście ręczna poprawa każdej jego formy staje się czasochłonna i nużąca. Ławo także o przeoczenia pojedynczych wystąpień. Nie istnieje bowiem narzędzie pozwalające na automatyczne uzgadnianie tych właśnie form. Dokładnym celem niniejszej pracy jest stworzenie narzędzia redakcyjnego pozwalającego na rozwiązanie automatycznego dopasowywania wartości wspomnianych końcówek fleksyjnych w tekstach napisanych w języku polskim.

W kolejnych rozdziałach zostanie dokonana analiza głównego problemu oraz mniejszych podproblemów oraz przedstawione rozwiązanie wraz z przykładowymi wynikami.

1.3. Struktura pracy

Rozdział drugi traktuje o badaniu struktury tekstu i problemie podziału go na zdania. Rozdział trzeci zawiera omówienie wybranych klas i kategorii gramatycznych występujących w języku polskim, wyjaśnia znaczenie tagów oficjalnie stosowanych przez IPI PAN (Instytut Podstaw Informatyki Państwowej Akademii Nauk) [2]. Rozdział czwarty przedstawia problem niejednoznaczności kategorii gramatycznych, a także prezentuje stworzoną implementację detekcji kongruencji w zdaniu. W rozdziale piątym został przedstawiony problem występowania koreferencji w tekście oraz algorytm prezentujący przykładowe jego rozwiązanie. Rozdział szósty zawiera szczegóły implementacyjne oraz przedstawia etapy przyjętego rozwiązania. W rozdziale siódmym znajduje się analiza i uzasadnienie wyboru źródeł danych, a także przykładowy wynik działania całości zaimplementowanego rozwiązania. Rozdział ósmy zawiera podsumowanie dotychczasowych prac oraz możliwe kierunki rozwoju narzędzia.

[2] Strona: https://ipipan.waw.pl/

2. Tokenizacja tekstu

2.1. Definicja problemu

Analiza tekstu i jego budowy to jeden z kluczowych problemów postawionych w tej pracy. Ze względu na konieczność rozpatrywania relacji między poszczególnymi słowami w zdaniu, surowy tekst jako sekwencja symboli nie może być poddany bezpośredniemu działaniu algorytmów wykrywania kongruencji i koreferencji. W związku z powyższym zachodzi konieczność tokenizacji wprowadzanego tekstu. Tokenizacja [3] [17] jest procesem, w wyniku którego ciągły teskt zostaje podzielony na ciąg pojedynczych tokenów – znaków ograniczonych dowolnymi separatorami. Separatorem może być nie tylko pojedynczy znak, ale także dowolny ciąg znaków dający się opisać przy pomocy wyrażenia regularnego. Najprostszym i najbardziej elementarnym przykładem tokenizacji jest podział na słowa. Tekst jest wówczas dzielony przy użyciu wyrażenia regularnego „\s", oznaczającego dowolny biały znak – na przykład tabulator, spację, znak nowej linii itp. Opisaną operację implementuje w języku Python metoda `split()`. Dla przykładu tekst

„Dziś jest ładna pogoda."

po wywołaniu na nim wspomnianej metody będzie miał postać:

[„Dziś", „jest", „ładna", „pogoda."].

Jednakże w poruszanym w pracy przypadku zachodzi konieczność bardziej zaawansowanej tokenizacji, a mianowicie podziału na zdania lub równoważniki zdań. W językoznastwie zdanie [4] określa się jako zespół wyrazów powiązanych zależnościami gramatycznymi i zawierający orzeczenie. Równoważnik zdania [4] z kolei stanowi wyraz lub grupa wyrazów, formalnie nie tworzące zdania (ze względu na brak orzeczenia), pełniące jednak tę samą funkcję co zdanie. W zależności od celu ich wypowiadania, zdania można podzielić [13] na:

- oznajmujące - zawiera wiadomości, które mówiący chce przekazać słuchaczowi, np. „Wyszedłem na pole.",
- pytające - informuje rozmówcę, że mówiący czegoś nie wie, ale chce się tego dowiedzieć, np. „Czy wyjdziesz na pole?",
- rozkazujące - wyrażają wolę mówiącego, zawierają polecenia, rozkazy i zakazy, np. „Wyjdź na pole!".

Dla każdego z tych typów charakterystyczny jest inny znak interpunkcyjny – kropka, znak zapytania lub wykrzyknik. Reguły języka polskiego wyraźnie stanowią, iż każde poprawne zdanie zawiera orzeczenie,

rozpoczyna się wielką literą i kończy jednym ze wspomnianych wyżej znaków interpunkcyjnych. Problemy jednak pojawiają się, gdy w zdaniu słowo zaczynające się od wielkiej litery, kropka, znak zapytania oraz wykrzyknik występują więcej niż jeden raz. Między innymi skróty, nazwy własne, listy, cytaty i dialogi przesądzają o trudności procesu podziału na zdania. Prosty tekst

„Ala ma kota. Basia ma psa."

zostałby podzielony poprawnie na dwa zdania

„Ala ma kota."
i
„Basia ma psa.".

Z kolei poprawne, choć bardziej skomplikowane zdanie

„Pani mgr inż. Anna Kowalska ma kota.",

zostałoby niepoprawnie rozdzielone na

„Pani mgr inż." i „Anna Kowalska ma kota.".

2.2. Prezentacja rozwiązania problemu

Rozwiązaniem rozważanego problemu jest wykorzystanie gotowego tokenizera z zaimplementowanymi regułami segmentacji uwzględniającymi bardziej złożone wzorce zdań w języku polskim. Użyty w pracy tokenizer Toki [1] dokonuje segmentacji tekstu w oparciu o polskie reguły SRX [2] autorstwa Marcina Miłkowskiego [16]. Przyjmując na wejściu plik z oryginalnym tekstem Toki dzieli go na zdania i zapisuje w podanym przez uzytkownika pliku. Każde zdanie oddziela znak „|" wybrany ze względu na małe prawdopodobieństwo znalezienia go w tekście. Wejście – surowy tekst [3]:

> *Korpus językowy to zbiór danych tekstowych dostępnych w formie elektronicznej, stanowiący materiał do badań. Korpusy stanowią obecnie jedno z podstawowych narzędzi w badaniach nad językiem, literaturą i kulturą. Od lat są nieodzownym narzędziem autorów słowników i podręczników do nauki języka, a coraz częściej używane są na co dzień również przez tłumaczy, nauczycieli oraz osoby pragnące pogłębić swoją znajomość języka obcego.*

Wyjście – ten sam tekst podzielony na zdania:

[1] Strona projektu: http://nlp.pwr.wroc.pl/narzedzia-i-zasoby/narzedzia/toki
[2] Segmentation Rules eXchange
[3] Tekst pochodzi ze strony: http://korpusy.net/

| Korpus językowy to zbiór danych tekstowych dostępnych w formie elektronicznej , stanowiący materiał do badań .| Korpusy stanowią obecnie jedno z podstawowych narzędzi w badaniach nad językiem , literaturą i kulturą .| Od lat są nieodzownym narzędziem autorów słowników i podręczników do nauki języka , a coraz częściej używane są na co dzień również przez tłumaczy , nauczycieli oraz osoby pragnące pogłębić swoją znajomość języka obcego .

Jak widać na przedstawionym przykładzie poprawnie została dokonana tokenizacja nie tylko prostych zdań pojedynczych, ale także zdań złożonych i równoważników zdań.

3. Klasy i kategorie gramatyczne

Każdy użytkownik języka polskiego intuicyjnie wie czym jest np. rzeczownik; potrafi wskazać czasownik w zdaniu oraz umie odróżnić przymiotnik od przysłówka. Natomiast zapytany o definicję słownikową chociażby liczebnika prawdopodobnie nie podałby odpowiedzi. Nie jest to przecież potrzebne do używania języka w mowie ani do pisania tekstów. Jednakże zrozumienie struktury języka, jego kategorii i klas gramatycznych jest niezbędne w procesie tagowania, a także późniejszych detekcji kongruencji i koreferencji. Z kolei do przetwarzania i analizy danych potrzebna jest znajomość oznaczeń stosowanych w narzędziach IPI PAN, które są szeroko wykorzystywane w tej pracy. Poniższy rozdział zawiera omówienie wybranych klas i kategorii gramatycznych występujących w języku polskim i mających znaczenie dla rozwiązania problemu. Nie omawiano kategorii typu aspekt czy tryb czasownika, ponieważ nie są one brane pod uwagę podczas tworzenia rozwiązania postawionego w pracy problemu.
Wyjaśnione także jest znaczenie tagów stosowanych w oficjalnym tagsecie IPI PAN.

3.1. Omówienie wybranych kategorii gramatycznych

3.1.1. Przypadek

Przypadek [4] określa składniową funkcję rzeczownika, przymiotnika, zaimków oraz liczebników i imiesłowów odmiennych. W języku polskim występuje siedem przypadków odpowiadających na następujące pytania[13]:

- mianownik (M.) *kto?, co?*,
- dopełniacz (D.) *kogo?, czego?*,
- celownik (C.) *komu?, czemu?*,
- biernik (B.) *kogo?, co?*,
- narzędnik (N.) *kim?, czym?*,
- miejscownik (Msc.) *o kim?, o czym?*,
- wołacz (W.) *o! - bezpośredni zwrot do kogoś lub czegoś [13].*

3.1.2. Liczba

Liczba [15] jest kategorią gramatyczną dotyczącą rzeczowników, czasowników, przymiotników, zaimków przymiotnych, imiesłowów przymiotnikowych oraz niektórych gerundiów. Kategoria liczby

przyjmuje dwie wartości i zasadza się na różnicy semantycznej:

- liczba pojedyncza - odnosi się do jednego obiektu,
- liczba mnoga - do większej liczby obiektów.

3.1.3. Rodzaj

Rodzaj gramatyczny [14] to właściwość rzeczowników, która jest dziedziczona po nich przez przymiotniki (a także przymiotne formy zaimków, liczebników czy imiesłowów), a która determinuje według jakiego sposobu dane wyrazy będą się odmieniały. Gramatyka języka polskiego [5] wyróżnia możliwe rodzaje:

- żeński,
- nijaki zbiorowy,
- nijaki zwykły,
- męski zwierzęcy,
- męski osobowy,
- męski rzeczowy,
- przymnogi osobowy,
- przymnogi zwykły.

Rodzaj wskazuje przede wszystkim, które formy wyrazowe są ze sobą połączone związkiem składniowym. Dodatkowo kategoria rodzaju decyduje w języku polskim o podziale na deklinacje.

3.1.4. Deprecjatywność

Deprecjatywność [15] rzeczownika dotyczy tylko mianownika i biernika liczby mnogiej w rodzaju męskoosobowym. Deprecjatywne formy rzeczowników wymagają formy deprecjatywnej od zależnych czasowników, przymiotników, zaimków przymiotnych, imiesłowów przymiotnikowych i liczebników.

3.1.5. Stopień

Jako kategoria gramatyczna przyimków i przysłówków stopień [4] określa natężenie cechy jakiegoś przedmiotu lub czynności. W języku polskim występują trzy stopnie:

- równy,
- wyższy,
- najwyższy.

3.1.6. Strona

Formy strony [13] informują o wykonawcy i odbiorcy czynności. W języku polskim wyróżniamy dwie strony :

- strona bierna - używana podczas łączenia czasownika z osobą lub przedmiotem, który czynność wykonuje, np. „Pacjent jest leczony przez lekarza.”,

– strona czynna - używana podczas łączenia czasownika z osobą lub przedmiotem, który danej czynności doznaje, np. „Lekarz leczy pacjenta.”.

3.1.7. Czas

Formy czasu [13] informują, w jakim czasie dana czynność się odbywa. Formy czasu wyrażają stosunek czasu czynności do czasu, w którym się o niej mówi.

– czasu teraźniejszego używa się do opisu czynności, która odbywa się w tym samym czasie, w którym się o niej mówi,
– czasu przeszłego używa się do opisu czynności, która odbywa się przed czasem, w którym się o niej mówi,
– czasy przyszłego używa się do opisu czynności, która nastąpi w czasie późniejszym niż ten, w którym się o niej mówi,

3.1.8. Osoba

Kategoria osoby [15] jest określona tylko dla form osobowych i przybiera następujące wartości [13]:

– pierwsza osoba liczby pojedynczej – informuje, że o czynności mówi ta sama osoba, która ją wykonuje,
– druga osoba liczby pojedynczej – informuje, że czynność wykonuje osoba, do której bezpośrednio zwraca się podmiot,
– trzecia osoba liczby pojedynczej - informuje, że osoba wykonująca czynność jest oddalona i nie bierze udziału w rozmowie, a o jej poczynaniach mówi jedna osoba do drugiej,
– pierwsza osoba liczby mnogiej – informuje, że o czynności kilku osób mówi jedna z nich kilka lub wszystkie wykonujące daną czynność,
– druga osoba liczby mnogiej – informuje, że o czynności osób mówi inna osoba lub grupa osób, która w danej czynności nie bierze udziału,
– trzecia osoba liczby mnogiej - informuje, że grupa osób wykonująca czynność jest oddalona i nie bierze udziału w rozmowie.

3.2. Omówienie klas gramatycznych

3.2.1. Rzeczownik

Rzeczownik [4] jest kategorią gramatyczną obejmującą wyrazy oznaczające osobę, zwierzę, przedmiot lub zjawisko. Rzeczowniki [13] odmienne są przez liczbę i przypadek, a także mają ustaloną wartość kategorii rodzaju. Nie mają z kolei przypisanej kategorii osoby.

3.2.2. Przymiotnik

Przymiotnik [13] określa cechy ludzi, zwierząt i przedmiotów. Każdy przymiotnik opisuje tylko jedną, oddzielną cechę - ma więc szczegółowe znaczenie. Ze względu na znaczenie przymiotniki można podzielić na zmysłowe, czyli takie, które opisują cechy spostrzegane zmysłami oraz umysłowe - wnioskowane z postępowania ludzi. Przymiotniki mogą także określać przedmioty pod względem ich stosunku do pewnych ludzi i miejsc.

3.2.3. Przysłówek

Przysłówki [13] należą do słów nieodmiennych, a ich zadaniem jest określanie jakości danej czynności, okoliczności i stopnia natężenia cech. W przypadku określania czasowników, przysłówki odpowiadają za sposób określania czynności oraz jej miejsce, czas i okoliczności. W języku polskim można wyróżnić następujące rodzaje przysłówków:

- przysłówki sposobu,
- przysłówki miejsca,
- przysłówki czasu,
- przysłówki stopnia,
- przysłówki miary,
- przysłówki przeczące.

3.2.4. Liczebnik

Liczebniki [13] określają ilościowe cechy przedmiotów. Używa się ich do opisywania liczby i kolejności. Język polski wyróżnia między innymi następujące rodzaje liczebników:

- liczebniki główne,
- liczebniki ułamkowe,
- liczebniki ilościowe,
- liczebniki mnożne i wielorakie,
- liczebniki nieokreślone,
- liczebniki porządkowe.

3.2.5. Czasownik

Czasownik [13] jest nazwą czynności, które wykonują istoty żywe lub maszyny oraz nazwą stanów, w których te istoty się znajdują, a także zmian zachodzących w przyrodzie. Czasowniki odmieniają się przez osoby, liczby, czasy, tryby, a także przez rodzaje - w przypadku czasów: przeszłego i przyszłego złożonego. W języku polskim, poza osobowymi formami, występują także następujące nieosobowe formy:

- bezokolicznik [15] - forma czasownika, która pełni w zdaniu funkcję mniej istotną niż formy finitywne: jest z reguły wymagana (rządzona) przez jakiś wyraz z kontekstu, najczęściej nadrzędną

formę innego czasownika. Bezokolicznik jest formą nazywającą czynność bez względu na to, kto ją wykonuje, kiedy była wykonywana itp. [13] ,
- imiesłów przysłówkowy [15] - jest podporządkowany formie finitywnej zajmującej w niej miejsce centralne i samodzielnie nie wyraża trybu, czas natomiast – tylko w sposób pośredni - w stosunku do czasu wyrażanego przez formę finitywną w strukturze wyższego rzędu,
- gerundium - inaczej rzeczownik odczasownikowy [4] będący nazwą czynności lub stanu.

3.2.6. Zaimek

Zaimek [13] może zastępować dowolny samodzielny wyraz za wyjątkiem czasownika. Znaczenie zaimka jest bardzo ogólne, dlatego też zauważyć można bardzo szeroki zakres jego użycia. Zaimkiem można zastąpić rzeczownik lub przymiotnik dowolnego rodzaju, a także liczebnik czy przysłówek. Wśród zaimków można wyróżnić między innymi:
- zaimki osobowe,
- zaimki dzierżawcze,
- zaimki zwrotne,
- zaimki wskazujące,
- zaimki pytające,
- zaimki anaforyczne.

O konkretnym znaczeniu zaimka decyduje kontekst, czyli wypowiedzi poprzedzające jego użycie.

3.2.7. Leksemy nieodmienne

Kryteria czysto fleksyjne nie pozwalają na podzielenie leksemów nieodmiennych na takie klasy jak przyimki, spójniki i partykuło-przysłówki. Rozróżnienie między klasami nieodmiennymi odbywa się więc na zasadzie różnic własności składniowych. Jako przyimki wyróżnia się leksemy pełniące w wypowiedzeniu funkcję łączącą i wymagające określonego przypadka. Za spójniki uznaje się leksemy pełniące funkcję łączącą, ale nie wymagające określonego przypadka. Pozostałe leksemy nieodmienne [5] należące do systemu leksykalnego polszczyzny zalicza się do resztkowej klasy kublików.

3.2.8. Ciała obce

Ciała obce [5] to elementy nie należące do systemu fleksyjnego polszczyzny, pojawiające się w polskich tekstach. Należą do tej klasy w szczególności elementy pochodzące z innych języków, a także nazwy własne. Ciało obce wykazujące szczątkową formę odmiany to ciała obce nominalne; wszystkie pozostałe zalicza się do ciał obcych luźnych.

3.3. Znaczenie tagów z oficjalnego tagsetu IPI PAN

W tabelach 1 i 2 znajduje się pełne wyjaśnienie znaczenia tagów z oficjalnego tagsetu IPI PAN[5].

3.3.1. Tagi klas gramatycznych

Klasa gramatyczna	Tag w tagsecie IPI PAN
rzeczownik	subst
przymiotnik	adj
niesamodzielna forma przymiotnika	adj0
przysłówek stopniowalny	adv
przysłówek przyprzyimkowy	adv0
liczebnik	num
zaimek osobowy on	ppron
zaimek trzecioosobowy	ppron3
zaimek nietrzecioosobowy	ppron12
nieprzeszła forma finitywna czasownika BYĆ	fin
przyszła forma finitywna czasownika BYĆ	bedzie
forma aglutynacyjna czasownika BYĆ	aglt
pseudoimiesłów	praet
bezosobnik	imps
bezokolicznik	inf
rozkaźnik	impt
imiesłów przysłówkowy współczesny	pcon
imiesłów przysłówkowy uprzedni	pant
odsłownik	ger
imiesłów przymiotnikowy czynny	pact
imiesłów przymiotnikowy bierny	ppas
przyimek	prep
spójnik	conj
partykuło - przysłówek	part
ciało obce luźne	xxs
ciało obce nominalne	xxx
kublik	qub
predykatyw	pred
czasownik typu “winien”	winien

Tabela 1: Wartości tagów klas gramatycznych w tagsecie IPI PAN

3.3.2. Tagi kategorii gramatycznych

Kategoria gramatyczna	**Zestaw wartości**	**Zestaw tagów w tagsecie IPI PAN**
liczba	pojedyncza	sg
	mnoga	pl
przypadek	mianownik	nom
	dopełniacz	gen
	celownik	dat
	biernik	acc
	narzędnik	inst
	miejscownik	loc
	wołacz	voc
rodzaj	męski osobowy	m1
	męski zwierzęcy	m2
	męski rzeczowy	m3
	żeński	f
	nijaki zbiorowy	n1
	nijaki zwykły	n2
	przymnogi osobowy	p1
	przymnogi zwykły	p2
	przymnogi opisowy	p3
osoba	pierwsza	pri
	druga	sec
	trzecia	tri
stopień	równy	pos
	wyższy	comp
	najwyższy	suc
aspekt	niedokonany	imperf
	dokonany	perf
negacja	niezanegowana	aff
	zanegowana	neg
deprecjatywność	niedeprecjatywna	ndepr
	deprecjatywna	depr
akcentowość	akcentowana	akc
	nieakcentowana	nakc
poprzyimkowość	poprzyimkowa	praep
	niepoprzyimkowa	npraep
akomodacyjność	uzgadniająca	congr
	rządząca	rec
aglutynacyjność	nieaglutynacyjna	naglt
	aglutynacyjna	aglt
wokaliczność	wokaliczna	wok
	niewokaliczna	nwok

Tabela 2: Wartości tagów kategorii gramatycznych w tagsecie IPI PAN

4. Wykorzystanie mechanizmu tagowania do wykrywania kongruencji w zdaniach

4.1. Definicja problemu niejednoznaczności kategorii gramatycznych

Dla rodzimych użytkowników języka polskiego fleksja wydaje się być intuicyjna i oczywista. Pomimo wielu potocznie występujących błędów wiadomym jest chociażby, że dopełniacz liczby pojedynczej słowa „punkt" to „punktu", a nie „punkta". Z kolei ze względu na swoją wieloznaczność słowo „powód" może w dopełniaczu mieć pisownię „powodu" (znaczenie powód-przyczyna, np. „Z powodu deszczu wycieczka została odwołana."), ale też „powoda" (znaczenie powód-osoba pozywana do sądu, np. „Sąd oddalił wniosek o ukaranie żony powoda grzywną."). Odmiana słów zaczyna się komplikować, gdy słowo ma wiele znaczeń, z czego każde z tych znaczeń ma inną odmianę. Dodatkowo zdarza się, że jedno z tych znaczeń występuje w języku powszechnie, reszta natomiast bardzo rzadko. Przykładowo odmiana przez przypadki słowa „taśma" w liczbie pojedycznej i mnogiej prezentuje się w następujący sposób:

przypadek	a) liczba pojedyncza	b) liczba mnoga
M. (*kto? co?*)	taśma,	**taśmy**,
D. (*kogo? czego?*)	**taśmy**,	taśm,
C. (*komu? czemu?*)	**taśmie**,	taśmom,
B. (*kogo? co?*)	taśmę,	**taśmy**,
N. (*kim? czym?*)	taśmą,	taśmami,
Msc. (*o kim? o czym?*)	**taśmie**,	taśmach,
W. (*o!*)	taśmo.	**taśmy**.

Odmiana słowa „taśma" - ze względu na tylko jedno jego znaczenie - nie sprawia większych problemów. W języku polskim występują zjawiska polisemii i homonimii [4]. W przypadku polisemii, pomimo różnych znaczeń słowa dają się sprowadzić do wspólnego źródła. Homonimy, pomimo identycznego brzmienia mają różne znaczenie jak i różne pochodzenie – rozpatruje się je zatem jako odrębne słowa. W języku polskim występowanie homonimii w formie fleksyjnej stwarza problem niejednoznaczności odmiany. Jak widać na powyższym przykładzie forma „taśmy" może oznaczać dopełniacz liczby pojednynczej albo mianownik liczby mnogiej, albo wołacz liczby mnogiej. Bardziej skomplikowanym przykładem jest forma „dam", która może oznaczać osobową formę czasownika „dać", ale także dopeł-

niacz liczby mnogiej rzeczownika „dama". Z kolei odmiana przez przypadki wspomnianego wcześniej słowa „powód" znacznie różni się w zależności od znaczenia.

przypadek	a) powód - przyczyna	b) powód - osoba pozwana do sądu
M. (*kto? co?*)	powód,	powód,
D. (*kogo? czego?*)	**powodu**,	**powoda**,
C. (*komu? czemu?*)	powodowi,	powodowi,
B. (*kogo? co?*)	powód,	powoda,
N. (*kim? czym?*)	powodem,	powodem,
Msc. (*o kim? o czym?*)	powodzie,	powodzie,
W. (*o!*)	powodzie.	powodzie.

Z przedstawionych przykładów wyciągnąć można wniosek, iż nie jest możliwe jednoznaczne rozstrzygnięcie kategorii gramatycznej słowa bez podanego kontekstu, w jakim ono występuje. Istnieją narzędzia do analizy morfologicznej, na przykład Morfologik[1] - analizator morfologiczny, słownik morfologiczny i korektor gramatyczny. Jednakże ze względu na to, że dla wprowadzonego słowa Morfologik poda wszystkie możliwe jego formy, wykorzystanie go w pracy byłoby niefektywne. Rozwiązaniem przedstawionego problemu było wykorzystanie procesu tagowania, a co za tym idzie - programu zwanego taggerem. Jak podaje publikacja [6], tagowanie to proces wyboru najbardziej właściwego opisu morfosyntaktycznego dla każdego słowa w analizowanym zdaniu. Ilustracją różnicy w wynikach wykorzystania analizatora morfologicznego i taggera może być następujące zdanie:

„Ten płyn jest bardzo **drogi**."

Dla słowa „drogi" analizator morfologiczny zwróci następujący wynik:

drogi	drogi	adj:sg:acc:m3:pos+**adj:sg:nom.voc:m1.m2.m3:pos**+ subst:pl:acc:p3+subst:pl:nom:p3+subst:pl:voc:p3
drogi	droga	subst:pl:acc:f+subst:pl:nom:f+subst:pl:voc:f+subst:sg:gen:f

Zgodna z kontekstem zdania wersja (przymiotnik, rodzaj męski rzeczowy, mianownik) została podana jako jedna z wielu możliwości. Z kolei tagger poda jednoznaczny wynik:

drogi	drogi	**adj:sg:nom:m3:pos**

W pracy zostały użyte następujące taggery: Concraft-pl [2] - zainstalowany na lokalnej maszynie oraz zestaw narzędzi udostępnianych przez Multiservice [3] - WCRTF, WMBT, Pantera, Polita oraz Concraft. Szczegółowe wykorzystanie wyżej wymienionych taggerów zostało opisane w dalszych rozdziałach.

[1] Strona projektu: http://morfologik.blogspot.com/ oraz repozytorium w serwisie Github: https://github.com/morfologik

[2] Repozytorium projektu: https://github.com/kawu/concraft-pl

[3] Online demo można znaleźć pod adresem: http://multiservice.nlp.ipipan.waw.pl/

4.2. Definicja problemu detekcji kongruencji w zdaniu

W językoznastwie kongruencją (związkiem zgody) [4] nazywa się relację między elementami zdania, która wymaga, aby jeden miał taką samą formę jak drugi. Reguły kongruencji w języku polskim wyglądają następująco [13]:

- rzeczownik + czasownik – związek główny – zgodność liczby i rodzaju,
- rzeczownik + przymiotnik – zgodność liczby, rodzaju i przypadka,
- rzeczownik + zaimek przymiotny – zgodność liczby, rodzaju i przypadka,
- rzeczownik + imiesłów przymiotnikowy – zgodność liczby, rodzaju i przypadka.

Przykładowo w zdaniu

„Mój jutrzejszy męczący lot do Berlina został odwołany"

wyrazy pozostające w związku zgody z wyrazem „lot" to:

„mój", „jutrzejszy", „męczący", „został", „odwołany"

co zgodnie z przytoczonymi wyżej regułami kongruencji można łatwo udowodnić.

- „lot": część mowy: rzeczownik, przypadek: **mianownik**, rodzaj: **męski rzeczowy**, liczba: **pojedyncza**,
- „mój": część mowy: przymiotnik, przypadek: **mianownik**, rodzaj: **męski rzeczowy**, liczba: **pojedyncza**,
- „jutrzejszy": część mowy: przymiotnik, przypadek: **mianownik**, rodzaj: **męski rzeczowy**, liczba: **pojedyncza**,
- „męczący": część mowy: imiesłów przymiotnikowy, przypadek: **mianownik**, rodzaj: **męski rzeczowy**, liczba: **pojedyncza**,
- „odwołany", część mowy: **imiesłów przymiotnikowy**, przypadek: **mianownik**, rodzaj: **męski rzeczowy**, liczba: **pojedyncza**,
- „został": część mowy: czasownik, rodzaj: **męski rzeczowy**, liczba: **pojedyncza**.

Gdyby poddać analizie tagi poszczególnych wyrazów wynik byłby taki sam.

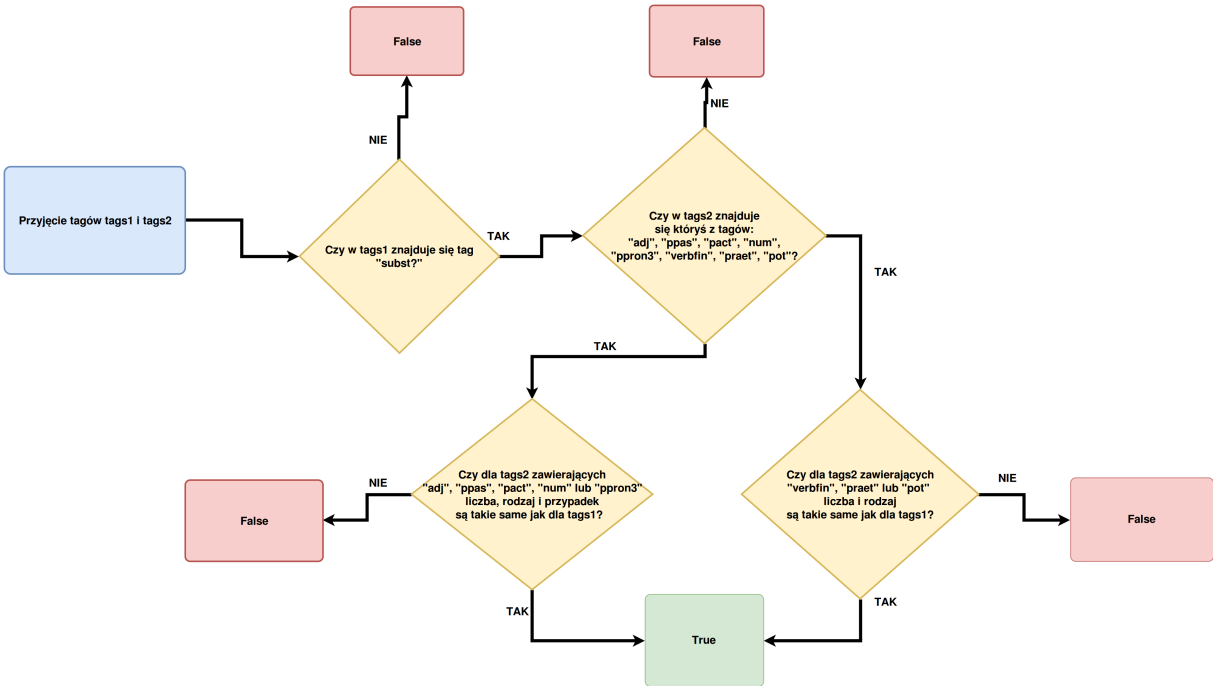
- „lot": subst:**sg:nom:m3**,
- „mój": adj**:sg:nom:m3**:pos,
- „jutrzejszy": adj:**sg:nom:m3**:pos,
- „męczący": pact:**sg:nom:m3**:pos,
- „odwołany", ppas:**sg:nom:m3**:pos,
- „został": praet:**sg:m3**:perf.

4.3. Prezentacja rozwiązania problemów

Rozwiązanie opisywanych w tym rozdziale problemów opiera się na analizie odpowiednich tagów z wcześniej przetworzonego tekstu. Sprawdzić należy przede wszystkim:

– Czy wyraz główny (ten, dla którego poszukuje się kongruencji) jest rzeczownikiem?
– Czy analizowany wyraz należy do poprawnej kategorii gramatycznej (rzeczownik, przymiotnik, zaimek przymiotny lub imiesłów przymiotnikowy) gramatycznej może pozostawać w kongruencji z wyrazem głównym?
– Czy została zachowana zgodność liczby, rodzaju i przypadka dla przymiotników, zaimków przymiotnych oraz imiesłowów przymiotnikowych albo liczby i rodzaju dla czasowników?

Podczas tworzenia algorytmu okazało się, iż z punktu widzania detekcji kongruencji nie jest istotne rozróżnienie rodzajów: męskiego osobowego, męskiego zwierzęcego, męskiego rzeczowego, a także: nijakiego zbiorowego, nijakiego zwykłego, przymnogiego osobowego, przymnogiego zwykłego i przymnogiego opisowego. Na ryskunku 1 przedstawiono główną istotę algorytmu wykrywania kongruencji w zdaniach. Podczas wywoływania funkcji zawsze zachowana jest kolejność podawania argumentów: najpierw podawane są tagi słowa głównego, a potem słowa sprawdzanego. Pomimo możliwości zastosowania różnych taggerów kolejność występowania istotnych tagów (liczba, rodzaj, przypadek) pozostaje niezmienna.



Rysunek 1: Schemat algorytmu wykrywania kongruencji

Jak widać, detekcja potencjalnego związku zgody odbywa się w oparciu o porównanie konkretnych tagów reprezentujących liczbę i rodzaj dla czasowników lub liczbę, rodzaj i przypadek dla przymiotników, zaimków przymiotnych i imiesłowów przymiotnikowych.

5. Wykrywanie koreferencji w tekście

5.1. Definicja problemu

Opisany powyżej algorytm wykrywania kongruencji jest skuteczny tylko dla pojedynczego zdania. Zdarza się jednak, że szukane słowo powtarzane jest w więcej niż jednym miejscu w tekście. Dla przykładu:

„**Chłopiec poszedł** do szkoły. Nie **umiał** nic na klasówkę. **Uznał on**, że spróbuje ściągać."

zauważyć można, że słowa pozostające w związku zgody ze słowem „chłopiec" znajdują się w więcej niż jednym zdaniu. Nie zawsze jednak szukane słowo występuje dosłownie w zdaniu. Na powyższym przykładzie widać konieczność brania pod uwagę podmiotu domyślnego oraz zaimków osobowych, a nawet wyrazów bliskoznacznych.

5.2. Prezentacja rozwiązania problemu

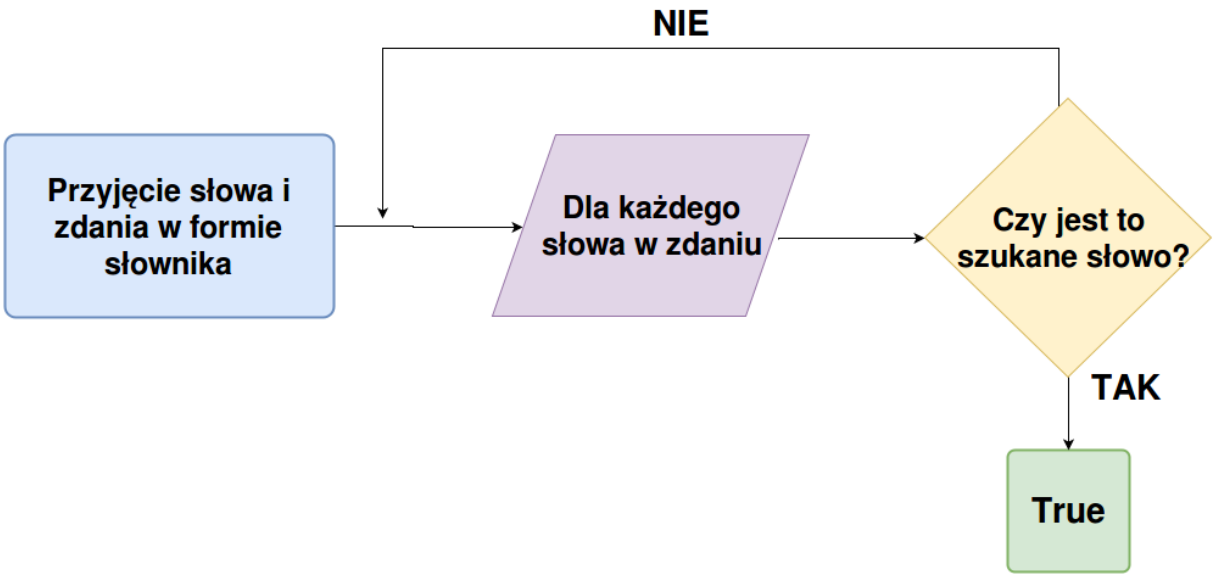
Przeszukiwanie przetworzonego tekstu pod kątem występowania koreferencji opiera się o analizę całego zdania. Wykorzystywane są nie tylko tagi poszczególnych słów, ale także budowa całego zdania. Algorytm przedstawia się następująco:

- **Czy w zdaniu bezpośrednio występuje szukane słowo?**
  W tym przypadku przeszukiwanie opiera się nie o tagi słowa, ale o jego formę podstawową. Przykładowo w tekście:

  „Podeszłam do **okna**. **Okno** było brudne."

  W obu zdaniach występuje słowo *„okno"*, tak więc każde z nich zostanie przeszukane pod kątem występowania kongruencji. Schemat algorytmu przeszukiwania zdania pod kątem wystąpienia konkretnego słowa przedstawia rysunek 2.

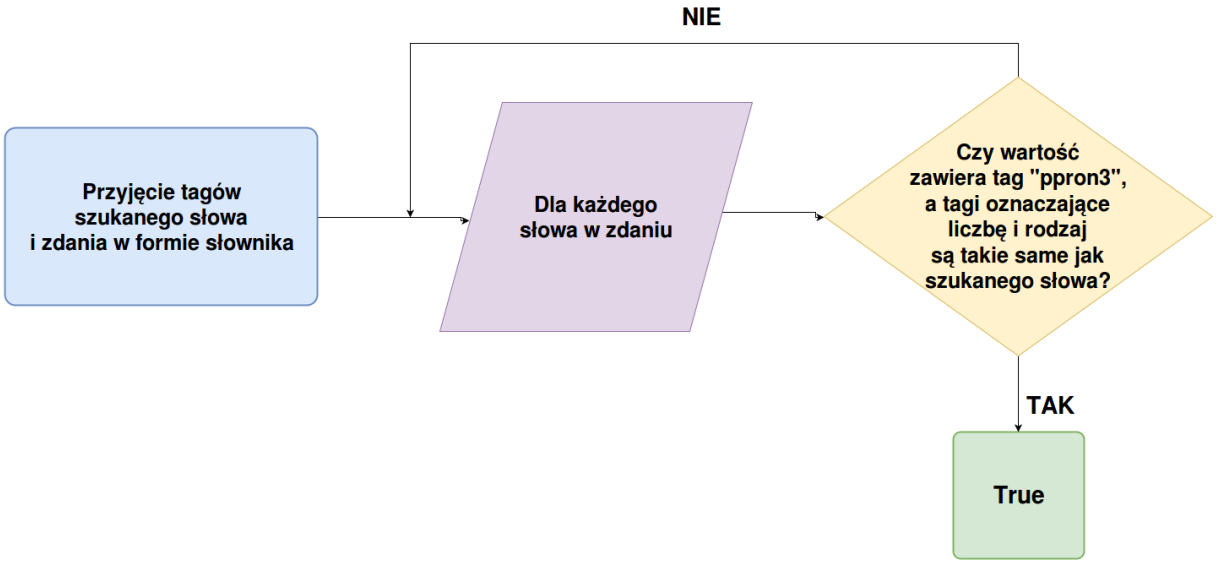
- **Czy w zdaniu występuje zaimek osobowy o takiej samej liczbie i rodzaju jak szukane słowo i jednocześnie czy w tym samym zdaniu nie występuje słowo o innym znaczeniu, ale takiej samej liczbie i rodzaju?**
  Sprawdzenie tego warunku pozwala na skuteczne wykrywanie koreferencji w tekstach typu:



Rysunek 2: Schemat algorytmu przeszukiwania zdania pod kątem wystąpienia danego słowa

„Podeszłam do **okna**. **Ono** było brudne."

W powyższym przypadku zdanie *„Ono było brudne"* także zostanie wzięte pod uwagę podczas szukania kongruencji, mimo iż słowo *„okno"* bezpośrednio w nim nie występuje. Schemat algorytmu przeszukiwania zdania pod kątem wystąpienia konkretnej formy zaimka osobowego przedstawia rysunek 3.



Rysunek 3: Schemat algorytmu przeszukiwania zdania pod kątem wystąpienia konkretnej formy zaimka osobowego

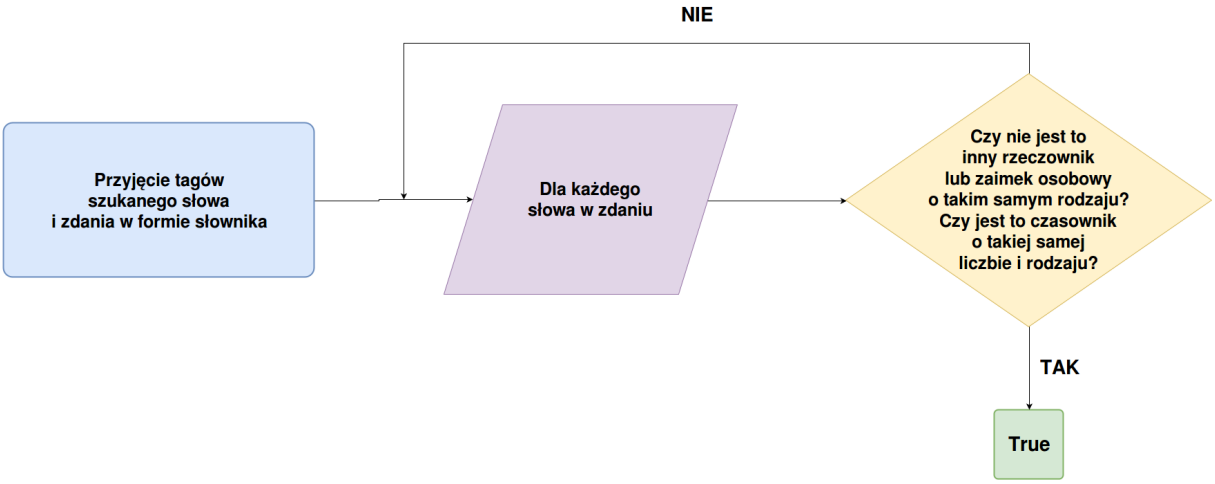
W przeciwieństwie do sprawdzania czy w zdaniu występuje szukane słowo, w tym przypadku sprawdzanie poprawności zaimka osobowego opiera się o sprawdzanie tagów.

– **Czy w zdaniu nie występuje ani szukane słowo, ani słowo o takiej samej liczbie i rodzaju, ale jednocześnie liczba i rodzaj orzeczenia zgadza się z liczbą i rodzajem szukanego słowa?**

Wspomniany warunek pozwala na wykrycie koreferencji w przypadku wystąpienia w zdaniu podmiotu domyślnego, czyli takiego, który nie jest wyrażony oddzielnym wyrazem w zdaniu, ale można go zidentyfikować za pomocą formy orzeczenia. Dobrą ilustracją może być tekst:

„Podeszłam do **okna**. **Było** bardzo brudne. ”

W drugim zdaniu słowo *„okno”* nie występuje ani bezpośrednio, ani w postaci zaimka osobowego (*„ono”*). Jednakże liczba i rodzaj orzeczenia zgadzają się z liczbą i rodzajem słowa *„okno”*, a ponadto w zdaniu nie występuje żadne inne słowo z takimi cechami. Schemat algorytmu przeszukiwania zdania pod kątem wystąpienia konkretnej formy podmiotu domyślnego przedstawia rysunek 4.



Rysunek 4: Schemat algorytmu przeszukiwania zdania pod kątem wystąpienia podmiotu domyślnego

W powyższym przypadku najpierw następuje sprawdzanie czy zdanie nie zawiera innych rzeczowników ani zaimków osobowych, a następnie czy liczba i rodzaj orzeczenia zgadzają z tymi, które ma szukane słowo. Pod uwagę wzięto jedynie orzeczenia będące czasownikami w czasie przeszłym, gdyż tylko te mogą pozostawać w związku zgody.

6. Szczegóły implenentacyjne

6.1. Wybór języka programowania

Całość rozwiązania została zaimplementowana przy użyciu języka Python ze względu na przejrzystość rozwiązania oraz łatwość integracji zewnętrznych narzędzi. Podczas implementacji korzystano z IDE pyCharm na licencji edukacyjnej [1]. Narzędzie zostało zaimplementowane i przetestowane na systemie Ubuntu 14.04. System ten został wybrany ze względu na łatwość integracji narzędzi służących do przetwarzania języka naturalnego oraz wygodny interfejs konsolowy. Niemal wszystkie zewnętrzne narzędzia mają stworzone instrukcje instalacji i użytkowania tylko dla systemów z rodziny Linux. Próba dostosowania ich do warunków systemu Windows byłaby czasochłonna i niezwiązana z głównym problemem pracy, a w niektórych przypadkach może nawet niemożliwa. Ze względu na obecność w języku polskim znaków narodowych i konieczność kodowania ich przy pomocy standardu UTF-8, a także fakt, iż zewnętrzne narzędzia uruchamia się przy pomocy odpowiedniego polecenia w konsoli systemu operacyjnego, szeroko używano następujących modułów:

- `codecs` [7] - moduł dostaraczający interfejsów umożliwiających konwersję danych między różnymi formatami, zazwyczaj używany do danych przechowywanych w formacie Unicode. W pracy zastosowano go w celu ułatwienia operacji na plikach kodowanych w formacie UTF-8 zawierających teksty w języku i dane na różnych etapach rozwiązywania problemu,
- `subprocess` [7] - moduł pozwający na tworzenie i komunikację z innymi procesami w systemie. Wszystkie integrowane narzędzia wykorzystywały interfejs konsolowy do komunikacji. Wykorzystanie modułu `subprocess` konieczne było do wywoływania z poziomu skryptu poleceń uruchamiających konkretne narzędzie oraz przyjęcia danych zwracanych przez nie,
- `requests` [8] - prosty i przyjazny użytkownikowi moduł umożliwiający komunikację przy pomocy protokołu HTTP. Pozwala m.in. na wysłanie konkretnego zapytania (np. GET) oraz analizę odpowiedzi na nie. Użycie `requests` umożliwiło komunikację z narzędziami Słowosieci służącymi do odnajdywania wyrazów bliskoznacznych.
- `json` [7] - moduł odpowiedzialny za analizę i przetwarzanie danych przechowywanych w formacie JSON [2]. `Json` użyty został podczas analizy odpowiedzi z serwisów Multiservice i Słowosieć,
- ze względu na niezgodność języka systemu operacyjnego (angielski) i tego używanego w pracy (polski) konieczne stało się użycie metody `setdefaultencoding(„utf-8”)` z modułu `sys` w

[1] Strona narzędzia: https://www.jetbrains.com/pycharm-edu/

[2] JavaScript Object Notation

każdym z plików projektu.

6.2. Wykorzystanie zewnętrznych narzędzi

Poniżej znajduje się lista zewnętrznych narzędzi do przetwarzania i analizy tekstu w języku polskim wraz z ich rolą podczas rozwiązywania postawionego w pracy problemu.

- `Toki` - polski tokenizer stworzony przez grupę NLP Politechniki Wrocławskiej [3] służący do podziału na tokeny i zdania w oparciu o reguły SRX autorstwa Marcina Miłkowskiego. Więcej informacji o projekcie Toki można znaleźć w publikacji [9]
- `Multiservice` - zbiór narzędzi do analizy i przetwarzania tekstów napisanych w języku polskim. W pracy zostały wykorzystane taggery: Concraft, WCRTF, Polita, Pantera oraz WMBT. Komunikacja odbywa się poprzez wysłanie odpowiedniego zapytania do serwera i odebranie odpowiedzi w formacie JSON. Wymaga zainstalowania klienta dostępnego na stronie projektu [4] oraz połączenia z internetem. Wiecej informacji o Multiservice można znaleźć w artykule [10],
- `Concraft-pl` [5] - tagger dla języka polskiego. Jest on dostępny jako jedna z opcji narzędzia Multiservice, ale także jako samodzielny program. Do działania wymaga pobrania pliku z korpusem językowym nkjp-model-02.tgz [6]. Przyjmuje na wejściu plik z tekstem bądź sam tekst; na wyjściu zwraca dane w formacie plain text z możliwością przekierowania ich do wybranego pliku. Stosowany jako jedna z opcji dla użytkownika. Nie wymaga połączenia z internetem, więc z powodzeniem wykorzystywany jest jako wersja offline aplikacji. Więcej informacji o projekcie Concraft-pl można znaleźć w artykule [11],
- `Morfeusz SGJP` [7] [12] - analizator morfologiczny dla języka polskiego. Dla przyjętego słowa zwraca wszystkie jego dostępne formy. Pozwala na końcową „odmianę" słowa przez rodzaje (np. „piękny" - przymiotnik, mianownik, rodzaj męski, liczba pojedyncza -> „piękna", przymiotnik, mianownik, rodzaj żeński, liczba pojedyncza) podczas końcowej analizy tekstu i zamiany słów zgodnie z ich nowym rodzajem. Na wejściu przyjmuje wskazane przez użytkownika słowo, a zwraca listę możliwych jego form wraz z zestawem tagów dla każdego z nich. Nie wymaga połączenia z internetem.
- `Słowosieć` [8] - słownik semantyczny, który odzwierciedla system leksykalny języka polskiego. W pracy został użyty podczas badania zagadnienia koreferencji w zdaniach. Dostęp do zasobów odbywa się przez API [9] webservice'u, dlatego też do działania wymaga połączenia z internetem.

[3] Strona grupy: http://nlp.pwr.wroc.pl/
[4] Strona projektu: http://zil.ipipan.waw.pl/Multiservice
[5] Dostępny pod adresem: https://github.com/kawu/concraft-pl
[6] Do pobrania: http://clip.ipipan.waw.pl/NationalCorpusOfPolish
[7] Strona projektu: http://sgjp.pl/morfeusz/
[8] Strona projektu: http://plwordnet.pwr.wroc.pl/wordnet/
[9] Dostępne pod adresem: http://api.slowosiec.clarin-pl.eu:8080/docs/index.html

6.3. Struktura danych

W celu usprawnienia i przyspieszenia pracy na tekście zostały zaprojektowana i stworzone specjalne struktury danych.

6.3.1. Reprezentacja pojedynczego zdania

Struktura `sentence` reprezentuje pojedyncze zdanie występujące w analizowanym tekście i składa się z następujących pól:

- `sentence` - zdanie wejściowe w dokładnie takiej formie, jak w tekście,
- `sentence_dictionary` – słownik, którego kluczami są poszczególne słowa w zdaniu dokładnie w takiej formie i pisowni jak w tekście; wartości natomiast to krotki o strukturze `(tagi słowa, forma podstawowa słowa)`,
- `needs_refactor` - flaga o wartości True, jeśli w zdaniu wystąpiło szukane słowo lub jego koreferencja lub False w przeciwnym przypadku.

6.3.2. Reprezentacja całego tekstu

Struktura `data_structrure` reprezentuje cały tekst, wstępnie obrobiony i gotowy do dalszego przetwarzania. Jest to lista, której elementami są struktury typu sentence stworzone dla każdego zdania. Ze względów optymalizacyjnych tagowanie całego tekstu, a co za tym idzie – tworzenie jego struktury odbywa się tylko raz.

6.3.3. Prezentacja modelu przykładowego tekstu

Dla tekstu

„Idę do szkoły. Ta szkoła jest w mieście. Lubię do niej uczęszczać. Poszłam do placówki oświatowej.”

struktura `data_structure` prezentuje się następująco:

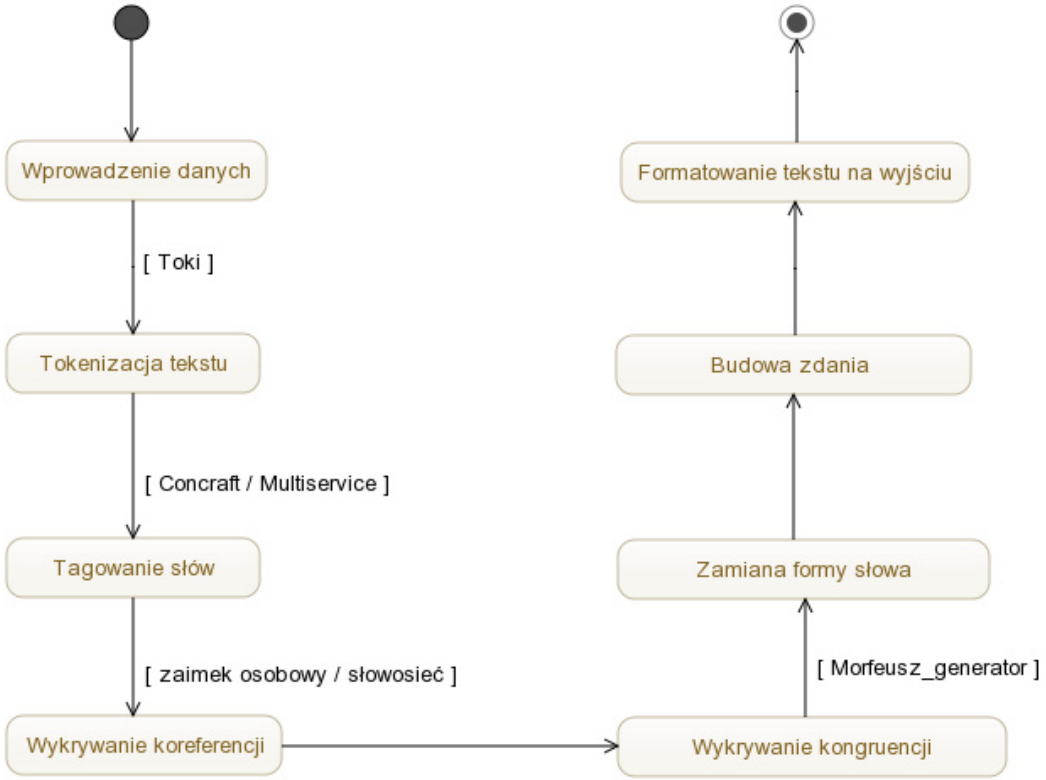
Zdanie	Wynik tagowania	Wartość `needs_refactor`
„Idę do szkoły .”	„do”: („prep:gen”, „do”), „szkoły”: („subst:sg:gen:f”, „szkoła”), „Idę”: („fin:sg:pri:imperf”, „iść”), „.”: („interp”, „.”)	True
„Chodzę do szkoły podstawowej .”	„Chodzę”: („fin:sg:pri:imperf”, „chodzić”), „do”: („prep:gen”, „do”), „podstawowej”: („adj:sg:gen:f:pos”, „podstawowy”), „szkoły”: („subst:sg:gen:f”, „szkoła”), „.”: („interp”, „.”)	True
„Lubię do niej uczęszczać .”	„Lubię”: („fin:sg:pri:imperf”, „lubić”), „niej”: („ppron3:sg:gen:f:ter:akc:praep”, „on”’), „uczęszczać”: („inf:imperf”, „uczęszczać”), „.”: („interp”, „.”), „do”: („prep:gen”, „do”)	True
„Ta szkoła jest w mieście .”	„jest”: („fin:sg:ter:imperf”, „być”), „mieście”: („subst:sg:loc:n”, „miasto”), „szkoła”: („subst:sg:nom:f”, „szkoła”), „.”: („interp”, „.”), „w”: („prep:loc:nwok”, „w”), „Ta”: („adj:sg:nom:f:pos”, „ten”)	True
„Poszłam do placówki oświatowej .”	„Poszła”: („praet:sg:f:perf”, „pójść”), „.”: („interp”, „.”), „placówki”: („subst:sg:gen:f”, „placówka”), „oświatowej”: („adj:sg:gen:f:pos”, „oświatowy”), „do”: („prep:gen”, „do”)	False

6.4. Etapy rozwiązywania problemu

1. **Przyjęcie od użytkownika stosownych danych wejściowych**
   Użytkownik podaje pełną ścieżkę do pliku z tekstem do analizy, wyraz, którego poszukuje w tekście, wyraz, którym chce zastąpić wyżej wspomniany oraz rodzaj i nazwę taggera. Możliwe jest użycie jednego z taggerów usługi Multiservice albo narzędzia Concraft-pl. Przykładowy poprawny zestaw danych wejściowych wygląda następująco:
   [„/home/user1/Desktop/teksty/tekst.txt”, „kot”, „biedronka”, „online”, „WCRFT”]
2. **Sprawdzenie poprawności podanych danych**
   Jeżeli format którychkolwiek danych jest niepoprawny program natychmiastowo przerywa działanie i wyświetla użytkownikowi stosowny komunikat o błędzie, np.
   „Błędna ścieżka. Podany plik nie istnieje.”

3. **Tokenizacja**
   Program odczytuje z podanego pliku cały tekst. Następnie uruchamia się tokenizer Toki, który wczytany tekst dzieli na zdania i zapisuje do tymczasowego pliku.
4. **Tagowanie i budowa struktury danych**
   Zgodnie z preferencjami podanymi na początku przez użytkownika zostaje uruchomiony odpowiedni tagger. Dla każdego wydzielonego we wcześniejszym etapie zdania zostaje stworzona odpowiednia struktura danych. W początkowej fazie obróbki zmienna logiczna informująca o konieczności przekształcenia danego zdania zostaje ustawiona na false.
5. **Przeszukanie każdego zdania pod kątem wystąpienia szukanego słowa lub jego koreferencji**
   Następuje sprawdzanie w których konkretnie zdaniach należy poszukiwać słowa podanego przez użytkownika. W przypadku znalezienia wystąpienia danego słowa bądź jego koreferencji, zmienna logiczna `needs_refactor` zostaje ustawiona na wartość `true`.
6. **Dla każdego zdania z wystąpieniem szukanego słowa przeszukanie zdania pod kątem słów pozostających w kongruencji z szukanym słowem**
   W zdaniach, w których zmienna logiczna `needs_refactor` ma wartość `true` zostaje uruchomiony algorytm wykrywania słów pozostających w kongruencji z szukanym słowem. Słowa te zostają dodane do listy.
7. **Dopasowanie form gramatycznych każdego słowa**
   Następuje potrzeba wykrycia rodzaju słowa, które zastąpi to szukane w tekście. Dla każdego słowa z uprzednio utworzonej listy zostaje utworzony nowy tag z uaktualnionym już rodzajem. Przykładowo aby zamienić słowo „piękna” na „piękny” należy w tagach zamienić oznaczenie rodzaju z żeńskiego na męski, czyli w tym przypadku z „f” na „m1”. Przy pomocy narzędzia Morfeusz generator zostaje pobrana forma słowa z oznaczeniem rodzaju odpowiednim dla uaktualnionej wersji.
8. **Budowa i końcowa obróbka poprawionego tekstu**
   Odpowiednio odmienione słowa zostają podmienione w tekście. Następuje także dodatkowe formatowanie tekstu w celu oczyszczenia go ze zbędnych białych znaków, będących pozostałością po procesie tagowania i tokenizacji. Tekst w nowej, zmienionej formie zostaje zwrócony użytkownikowi na standardowe wyjście.

Rysunek 5 pokazuje schemat rozwiązania postawionego problemu.



Rysunek 5: Schemat rozwiązania problemu

7. Działanie

7.1. Źródła danych

W celu przeprowadzenia testów zaproponowanego rozwiązania został zebrany zbiór trzydziestu tekstów. Teksty są zróżnicowane pod względem długości, przeznaczenia i poziomu językowego. Zasadniczo zebrane teksty pochodzą z trzech niezależnych źródeł:

- Portal internetowy Onet [1] - polski portal internetowy, 260. najczęściej odwiedzana strona na świecie, 6. najczęściej odwiedzana witryna w Polsce (pozycje wg rankingu Alexa [2]). Portal ten wybrany został ze względu na popularność, a także różnorodność zamieszczanych tekstów - od krótkich notatek prasowych informujących o wydarzeniach w Polsce i na świecie, po długie artykuły,
- Strona internetowa Polskiej Agencji Prasowej [3]- jedyna państwowa agencja informacyjna w Polsce. Do jej najważniejszych zadań należy przekazywanie rzetelnych, obiektywnych i wszechstronnych informacji z kraju i z zagranicy. Wybrana jako główne źródło krótkich profesjonalnych i pełnych informacji notatek prasowych,
- Strony z opowieściami użytkowników - Anonimowe [4] oraz Piekielni [5] - serwisy internetowe pozwalające na anonimową publikację krótkich własnych historii z życia codziennego. Strony te są źródłem historii pisanych przez zwyczajnych obywateli posługujących się na co dzień językiem polskim. Nie są to jednak profesjonalne teksty poddawane redakcji i korekcie, ale często krótkie notki pełne błędów językowych i niejednoznaczności koreferencji. Wybrane właśnie ze względu na wyrażenia potoczne i zamieszczane pod wpływem emocji treści.

7.2. Wynik działania zaimplementowanego rozwiązania na średniej długości tekście

- szukane słowo: korpus,
- zamieniane słowo: podstawa,
- rodzaj i nazwa taggera: online, WCRFT.

[1] Dostępny pod adresem: http://www.onet.pl
[2] wg danych ze strony: http://www.alexa.com/siteinfo/onet.pl
[3] Dostępna pod adresem: http://www.pap.pl
[4] Dostępna pod adresem: http://www.anonimowe.pl
[5] Dostępna pod adresem: http://www.piekielni.pl

Przykładowy tekst [6] średniej długości na wejściu wygląda następująco:

> **Korpus językowy** to zbiór danych tekstowych dostępnych w formie elektronicznej, **stanowiący** materiał do badań. **Korpusy stanowią** obecnie jedno z podstawowych narzędzi w badaniach nad językiem, literaturą i kulturą. Od lat są nieodzownym narzędziem autorów słowników i podręczników do nauki języka, a coraz częściej używane są na co dzień również przez tłumaczy, nauczycieli oraz osoby pragnące pogłębić swoją znajomość języka obcego.**Korpusy** przeszukuje się za pomocą specjalnie stworzonych do tego programów o różnym stopniu skomplikowania - najprostsze z łatwością obsługiwać może nawet zupełnie początkujący użytkownik. Witryna **korpusy.net** powstała w Instytucie Anglistyki UW pod redakcją Błażeja Gałkowskiego. Ma ona stanowić przystępne wprowadzenie do pracy z **korpusami**. Przedstawione zostały istniejące **korpusy** różnego typu, z których wiele dostępnych jest nieodpłatnie w sieci. Można tu również znaleźć porady dla tych, którzy chcieliby stworzyć **własny korpus**, porównanie programów do analizy danych językowych, przykładowe artykuły omawiające różne zastosowania **korpusów**, odnośniki do opublikowanych źródeł i innych stron internetowych oraz słowniczek wyjaśniający podstawowe terminy stosowane w językoznawstwie korpusowym. **Korpus językowy** to zbiór tekstów, w którym szukamy typowych użyć słów i konstrukcji oraz innych informacji o ich znaczeniu i funkcji. Bez dostępu do **korpusu** nie da się dziś prowadzić badań językoznawczych, pisać słowników ani podręczników języków obcych, tworzyć wyszukiwarek uwzględniających polską odmianę, tłumaczy komputerowych ani innych programów zaawansowanej technologii językowej. **Korpus** jest **niezbędny** do pracy językoznawcom, ale korzystają zeń często także informatycy, historycy, bibliotekarze, badacze literatury i kultury oraz specjaliści z wielu innych dziedzin humanistycznych i informatycznych.

Ten sam tekst na wyjściu ma następującą postać:

[6] Tekst pochodzi ze strony http://korpusy.net/

Podstawa językowa to zbiór danych tekstowych dostępnych w formie elektronicznej, **stanowiąca** materiał do badań. **Podstaw**y stanowią obecnie jedno z podstawowych narzędzi w badaniach nad językiem, literaturą i kulturą. Od lat są nieodzownym narzędziem autorów słowników i podręczników do nauki języka, a coraz częściej używane są na co dzień również przez tłumaczy, nauczycieli oraz osoby pragnące pogłębić swoją znajomość języka obcego. **Podstawy** przeszukuje się za pomocą specjalnie stworzonych do tego programów o różnym stopniu skomplikowania - najprostsze z łatwością obsługiwać może nawet zupełnie początkujący użytkownik. Witryna `korpusy.net` powstała w Instytucie Anglistyki UW pod redakcją Błażeja Gałkowskiego. Ma ona stanowić przystępne wprowadzenie do pracy z **podstawami**. przedstawione zostały istniejące **podstawy** różnego typu, z których wiele dostępnych jest nieodpłatnie w sieci. Można tu również znaleźć porady dla tych, którzy chcieliby stworzyć **własną podstawę**, porównanie programów do analizParty danych językowych, przykładowe artykuły omawiające różne zastosowania **podstaw**, odnośniki do opublikowanych źródeł i innych stron internetowych oraz słowniczek wyjaśniająca podstawowe terminy stosowane w językoznawstwie **podstawowym**. **Podstawa językowa** to zbiór tekstów, w której szukamy typowych użyć słów i konstrukcji oraz innych informacji o ich znaczeniu i funkcji. Bez dostępu do **podstawy** nie da się dziś prowadzić badań językoznawczych, pisać słowników ani podręczników języków obcych, tworzyć wyszukiwarek uwzględniających polską odmianę, tłumaczy komputerowych ani innych programów zaawansowanej technologii językowej. **Podstawa** jest **niezbędna** do pracy językoznawcom, ale korzystają zeń często także informatycy, historycy, bibliotekarze, badacze literatury i kultury oraz specjaliści z wielu innych dziedzin humanistycznych i informatycznych.

8. Podsumowanie

8.1. Wnioski

W wyniku realizacji pracy powstało narzędzie pozwalające na uzgadnianie wartości kategorii gramatycznych w tekstach. Jak pokazują wyniki zaprezentowane w rozdziale 7. stworzony system poprawnie dokonuje zamiany końcówek wyrazów w przypadku zmiany rodzaju głównego słowa. W tekście nie zostało jedynie słowo „korpusy.net", choć zgodnie z założeniami powinno było zostać zastąpione wyrażeniem „podstawy.net". Wynika to z faktu, że w języku naturalnym takie słowo nie istnieje - jest to reprezentacja adresu internetowego strony. Nie wystąpiła natomiast sytuacja, by zamienione zostało słowo, które nigdy nie pozostawało z związku zgody ze słowem głównym.
Podczas tworzenia pracy zrealizowano szereg działań prowadzących do powstania końcowego narzędzia. Zintegrowane zostały zewnętrzne narzędzia do przetwarzania języka naturalnego. W celach testowych zebrano zbiór różnorodnych tekstów. Zaimplementowano algorytmy detekcji kongruencji i koreferencji. W związku z powyższym można wysnuć wniosek, iż główny cel postawiony w pracy został osiągnięty.

8.2. Kierunki rozwoju pracy

Kod źródłowy stworzonego narzędzia jest otwarty i znajduje się w publicznym repozytorium w serwisie Github. [1] W celu zmniejszenia czasu pracy programu proponowana jest analiza kodu źródłowego pod kątem możliwości jego optymalizacji. Istotnym i niestety niemożliwym do całkowitego wyeliminowania ograniczeniem jest czas przetwarzania tekstu przez tagger i tokenizer, dlatego też optymalizacja powinna skupić się głównie na algorytmach detekcji kongruencji i koreferencji.
Inna możliwość rozwoju pracy to ulepszenie algorytmu detekcji koreferencji oraz lepsza integracja z narzędziem Słowosieć. Na chwilę obecną nie w każdym przypadku wykrywana jest koreferencja przy użyciu podmiotu domyślnego.
Jednym z proponowanych kierunków rozwoju pracy jest także zbadanie możliwości dostosowania narzędzia do systemu Windows oraz Mac OS X, a także przetestowanie stworzonej pracy na innych dystrybucjach systemu Linux takich jak Fedora czy CentOS. Dobrym rozwiązaniem byłoby także stworzenie graficznego interfejsu użytkownika, gdyż na chwilę obecną zaimplementowana jest tylko wersja konsolowa.

[1] Adres repozytorium: https://github.com/mtomzik/automatic-reconciliation

Kolejne możliwe działanie to rozszerzenie pracy o wykrywanie błędów powtórzeń. W chwili obecnej takiej możliwości nie zapewnia nawet wspominany we wcześniejszych rozdziałach LanguageTool. Potencjalnie przydatnym w szerokim zastosowaniu pracy byłaby jej integracja z narzędziem LanguageTool, a także dodanie obsługi innych języków charakteryzujących się silną fleksją, jak na przykład czeskiego lub słowackiego.

Bibliografia

[1] Iwona Dominiak. „Dlaczego język polski jest tak diabelnie trudny?". W: Newsweek Polska, dostęp pod adresem: http://www.newsweek.pl/polska/polski-najtrudniejszy-jezyk-swiata-jezykoznawca-odpowiada-newsweek,artykuly,352233,1.html (dostęp 07.09.2016).

[2] Piotr Müldner-Nieckowski. „Trzy pojęcia - redakcja, adiustacja, korekta". Dostęp pod adresem: http://aula.home.pl/edytorstwo/trzy_pojecia.htm (dostęp 07.09.2016).

[3] Marcin Mirończuk, Tadeusz Maciak. „System informacyjny na temat sieci hydrantów dla krajowego systemu ratowniczo-gaśniczego: metoda segmentacji tekstu i jej ocena", W: *Metody Informatyki Stosowanej Nr 4/2011 (29)*, s.45-46.

[4] *Zdanie, Równoważnik zdania, Homonimia, Polisemia, Przypadek, Stopień, Gerundium, Kongruencja*. W: Słownik Języka Polskiego PWN, dostęp pod adresem: http://sjp.pwn.pl (dostęp 07.09.2016).

[5] Marcin Woliński. „System znaczników morfosyntaktycznych w korpuse IPI PAN", dostęp pod adresem: http://nlp.ipipan.waw.pl/CORPUS/znakowanie.pdf (dostęp 07.09.2016).

[6] Daniel Jurafsky, James H. Martin. „Speech and Language Processing". Dostęp pod adresem: https://web.stanford.edu/~jurafsky/slp3, Rozdz. Part-of-Speech Tagging (dostęp 07.09.2016).

[7] *codecs, subprocess, json*. Python Module of The Week. Dostep pod adresem: https://pymotw.com/3/ (dostęp 07.09.2016).

[8] Requests: HTTP for Humans, dostęp pod adresem: http://docs.python-requests.org/en/master/ (dostęp 07.09.2016).

[9] Adam Radziszewski, Tomasz Śniatowski. „Maca - a configurable tool to integrate Polish morphological data". Dostęp pod adresem: http://mt-archive.info/FreeRBMT-2011-Radziszewski.pdf (dostęp 07.09.2016).

[10] Maciej Ogrodniczuk, Michał Lenart. „Web Service integration platform for Polish linguistic resources." W: *Proceedings of the Eighth International Conference on Language Resources and Evaluation*, LREC 2012, s. 1164–1168.

[11] Jakub Waszczuk. „Harnessing the CRF complexity with domain-specific constraints. The case of morphosyntactic tagging of a highly inflected language." W: *Proceedings of COLING 2012*.

[12] Marcin Woliński. „Morfeusz reloaded.” W: *Proceedings of the Ninth International Conference on Language Resources and Evaluation*, LREC 2014, s. 1106–1111.

[13] Piotr Bąk. „Gramatyka języka polskiego”, W: Wiedza powszechna, 2010. Rozdz.: Rzeczowniki, Przymiotniki, Liczebniki, Przysłówki, Zaimki, Czasowniki, Odmiana czasowników, Odmiana wyrazów przez przypadki, Podział wypowiedzeń ze względu na znaczenie, Rodzaje związków określających.

[14] Mirosław Koziarski, Adrian P. Krysiak. „Rodzaj gramatyczny rzeczownika jako nośnik informacji pozagramatycznej. Przegląd literatury dotyczącej rodzaju gramatycznego.”. Dostęp pod adresem: http://www.staff.amu.edu.pl/inveling/pdf/Koziarski-26.pdf (dostęp 07.09.2016).

[15] Zygmunt Saloni. „Podstawy teoretyczne „Słownika gramatycznego języka polskiego”'. Dostęp pod adresem: http://sgjp.pl/static/pdf/Wst%C4%99p%20do%20II%20wydania%20SGJP.pdf (dostęp 07.09.2016).

[16] Marcin Miłkowski, Jarosław Lipski. „Using SRX Standard for Sentence Segmentation”. W: Lecture Notes in Artificial Intelligence 6562, s. 172–182. Springer-Verlag, Berlin, Heidelberg, 2011.

[17] Mariusz Lipiński. „SCJP - Tokenizacja tekstu”. Dostęp pod adresem: http://www.mariuszlipinski.pl/2009/04/scjp-tokenizacja-tekstu.html (dostęp 07.09.2016).