Politechnika
Wrocławska

Politechnika Wrocławska
Wydział Informatyki i Zarządzania
Instytut Informatyki

ROZPRAWA DOKTORSKA

Ekstrakcja informacji o relacjach semantycznych między jednostkami identyfikacyjnymi z dokumentów tekstowych

mgr inż. Michał Marcińczuk

Promotor: prof. Zbigniew Huzar

Wrocław 2012

Spis treści

Spis rysunków

Spis tablic

Streszczenie

Tematem rozprawy jest zagadnienie ekstrakcji informacji, które jest jednym z zadań przetwarzania języka naturalnego. W pracy przedstawiona została teza, że wyszukiwanie informacji określonej klasy w tekstach ciągłych w języku polskim może być efektywniej realizowane przy pomocy nadzorowanego systemu ekstrakcji informacji niż tradycyjnych wyszukiwarek internetowych. Efektywność wyszukiwania informacji jest rozumiana jako czas dotarcia do informacji poszukiwanej przez użytkownika. W ramach rozprawy rozpatrywana jest określona klasa zadań wyszukiwania informacji ograniczona do pytań o nazwy obiektów będących w określonej relacji względem obiektu o zadanej nazwie.

W celu weryfikacji tezy zostały wyznaczone i zrealizowane trzy cele. Pierwszym celem było opracowanie nadzorowanej metody rozpoznawania jednostek identyfikacyjnych w tekście ciągłym w języku polskim. Zakres jednostek identyfikacyjnych był ograniczony do nazw i nazw własnych 56 kategorii obiektów. Jednostki były rozpoznawane przy pomocy hybrydowej metody łączącej metody nadzorowanego uczenia (metoda warunkowych pól losowych; CRF) z metodami słownikowymi i regułowymi. Do rozpoznawania jednostek zostały wykorzystane informacje ortograficzne, morfologiczne, semantyczne oraz słowniki.

Drugim celem było opracowanie nadzorowanej metody rozpoznawania relacji semantycznych zadanego typu pomiędzy wcześniej rozpoznanymi jednostkami identyfikacyjnymi. Zakres rozpoznawanych kategorii relacji został ograniczony do ośmiu (*autorstwo*, *kompozycja*, *narodowość*, *pochodzenie*, *położenie*, *przynależność*, *sąsiedztwo*, *tożsamość*). Relacje te zachodzą między jednostkami występującymi w obrębie jednego zdania. Cel został osiągnięty dzięki opracowaniu dwufazowej, w pełni nadzorowanej metody rozpoznawania relacji. W pierwszej fazie został wykorzystany paradygmat indukcyjnego programowania logicznego do konstrukcji reguł do rozpoznawania relacji. Do konstrukcji reguł zostały opracowane trzy modele reprezentacji danych, wykorzystujące informację ortograficzną, morfologiczną, składniową i semantyczną. W drugiej

fazie wygenerowane reguły zostały użyte jako cechy do konstrukcji zbioru klasyfikatorów binarnych.

Ostatnim celem była konstrukcja systemu wyszukiwania odpowiedzi na pytania zadane w języku naturalnym. Cel został zrealizowany poprzez konstrukcję dwumodułowego prototypu systemu ekstrakcji informacji. Pierwszy moduł odpowiedzialny jest za przetwarzanie tekstów i indeksowanie rozpoznanych informacji w relacyjnej bazie danych. Drugi moduł odpowiedzialny jest za transformację pytań w języku naturalnym do postaci zapytań SQL, za pomocą których poszukiwana informacja może być wyciągnięta bezpośrednio z bazy danych. Transformacja pytań do zapytań SQL odbywa się w oparciu o ręcznie opracowane reguły transformacji, których pokrycie zostało zwiększone przy użyciu Słowosieci i częściowego dopasowania.

Opracowany prototyp został przetestowany na zbiorze przykładowych zapytań podzielonych na dwie grupy: pytania o listę nazw obiektów oraz pytania o nazwę pojedynczego obiektu. Otrzymane wyniki zostały porównane z wynikami działania istniejących wyszukiwarek internetowych oraz systemów ekstrakcji informacji dla języka polskiego. Otrzymane wyniki pokazują, że zaproponowana metoda ekstrakcji informacji, pomimo niskiego pokrycia, może być skutecznie wykorzystana do strukturalizowania informacji zawartych w tekstach ciągłych, a tym samym wyszukiwania odpowiedzi na pytania przeglądowe (m.in. takie, których odpowiedzią jest lista nazw obiektów). W porównaniu do tradycyjnych wyszukiwarek internetowych opracowany system może pozyskiwać informacje z nieustrukturalizowanego tekstu. Natomiast wyszukiwarki internetowe muszą mieć dostęp do dokumentu zawierającego gotową odpowiedź na zadanie pytanie, aby mieć możliwość zwrócenia właściwej odpowiedzi. Z kolei dla pytań o nazwę konkretnego obiektu opracowany system potrafi wskazać dokładną odpowiedź, gdy wyszukiwarki internetowe wskazywały tylko dokument (w pewnych przypadkach akapit) zawierający właściwą odpowiedź.

Rozdział 1

Wstęp

1.1. Przedmiot rozprawy

Ekstrakcja informacji (ang. *Information Extraction*) jest jednym z zadań w ramach dziedziny przetwarzania języka naturalnego (ang. *Natural Language Processing*; NLP) (Indurkhya i Damerau, 2010). Nie istnieje jedna, ugruntowana definicja tego zadania, przez co w różnych pracach można spotkać podobnie brzmiące definicje różniące się między sobą w drobnych szczegółach. W 1987 w ramach pierwszej konferencji Message Understanding Conference (MUC) zadanie ekstrakcji informacji zostało zdefiniowane bardzo ogólnie jako „ekstrakcja lub wyciąganie właściwej informacji z dużych kolekcji dokumentów". Z kolei Hobbs i Riloff (2010) zdefiniowali zadanie ekstrakcji informacji bardziej szczegółowo jako proces analizy tekstu w poszukiwaniu informacji istotnych dla odbiorcy w określonym kontekście, takich jak jednostki identyfikacyjne (fragmenty tekstu odnoszące się do obiektów ze świata), relacje semantyczne między jednostkami (zależności między tymi obiektami) i zdarzenia (akcje zmieniające stan świata, umiejscowione w czasie i przestrzeni oraz angażujące obiekty). W praktyce oznacza to wskazanie zbiorów fragmentów tekstu posiadających określoną przez warunki zadania interpretację, np. „rozpoznanie informacji o wypadkach drogowych na terenie Polski składających się z numeru drogi, na której doszło do wypadku, daty, liczby poszkodowanych i liczby ofiar śmiertelnych".

Appelt i Israel (1999) opisali zadanie ekstrakcji informacji jako zagadnienie mieszczące się pomiędzy wyszukiwaniem informacji (ang. *Information Retrieval*), a rozumieniem tekstu (ang. *Text Understanding*). Wyszukiwanie informacji koncentruje się na znajdowaniu całych dokumentów lub ich fragmentów (wydzielonych akapitów lub zdań) zawierających informacje interesujące użytkownika w oparciu o słowa kluczowe występujące w treści dokumentów lub metadane powiązane z dokumentem. W realizacji pomija się poziom semantyczny analizowanych dokumentów i skupia wyłącznie na poziomie znakowym — w taki właśnie sposób działają popularne wyszukiwarki in-

ternetowe (Google.com, netsprint.pl itp.). Ekstrakcja informacji, w odróżnieniu od wyszukiwania informacji, uwzględnia poziom semantyczny tekstu, dzięki czemu możliwe jest pełniejsze zrozumienie informacji w nim zawartych. Analiza semantyczna wiąże się z wykorzystaniem narzędzi do przetwarzania języka naturalnego i zasobów językowych, m.in. analizatory morfologiczne, tagery, listy nazw własnych, powierzchniową i pełna analizę składniową zdań (Appelt i Israel, 1999). Dzięki wykorzystaniu tych dodatkowych narzędzi i zasobów możliwe staje się precyzyjne wskazanie w dokumencie fragmentu tekstu zawierającego informację poszukiwaną przez użytkownika oraz późniejsze jej ustrukturalizowanie (np. opracowanie listy nazw własnych obiektów zadanego typu).

Z drugiej strony ekstrakcja informacji jest pewnym uproszczeniem zadania rozumienia tekstu, które to zakłada pełną analizę semantyczną i pragmatyczną treści dokumentu. Celem zadania rozumienia tekstu jest strukturalizacja semantyki tekstu. Uchwycenie i formalizacja każdej informacji wyrażonej przy pomocy języka naturalnego jest bardzo trudnym zadaniem, ponieważ język naturalny umożliwia opis bardzo wielu pojęć — prawdziwych i zmyślonych, pochodzących ze świata rzeczywistego lub fikcyjnego, będących faktem lub przypuszczeniem itd. Zatem, ekstrakcja informacji, w porównaniu do zadania rozumienia tekstu, jest jego uproszczeniem — ograniczeniem się wyłącznie do pewnej wąskiej grupy elementów.

Zakres informacji, jaki może być wyciągany z dokumentów jest bardzo szeroki. W literaturze wyróżnia się trzy główne podzadania, które zostały już wspomniane w pierwszym akapicie, a są to: (1) rozpoznawanie jednostek identyfikacyjnych, (2) rozpoznawanie relacji między jednostkami oraz (3) rozpoznawanie zdarzeń (Hobbs i Riloff, 2010). Zakres poszczególnych zadań jest także bardzo zróżnicowany i zależny od docelowego zastosowania. W bardzo ogólnym znaczeniu, rozpoznawanie jednostek identyfikacyjnych sprowadza się do rozpoznawania fragmentów tekstu reprezentujących określone przez warunki klasy obiektów. Mogą to być nazwy własne określonych obiektów, wyrażenia liczbowe, nazwy symboliczne itd. Następnie, między parami jednostek mogą zachodzić związki semantyczne (relacje) oraz jednostki mogą być elementami złożonych struktur informacyjnych (zdarzenia; ang. *event templates*; zob. Hobbs i Riloff (2010)).

Tekst jest jedną z najpowszechniejszych form, w jakiej są przechowywane i wymieniane informacje na temat otaczającego nas świata. Szczególnie w obecnych czasach, kiedy komputery i dostęp do Internetu są masowo powszechne, co sprzyja generowaniu ogromnych ilości elektronicznych dokumentów tekstowych. Drugim ważnym czynnikiem jest sam język naturalny, który cały czas ewoluuje — pojawiają się nowe terminy i zwroty, a istniejące nabierają nowego znaczenia, przez co nie można go zamknąć w sztywne ramy. Zapotrzebowanie na narzędzia, które pozwolą na automatyczną analizę publikowanych treści, a następnie ich agregację staje się coraz większe. Przykładem może być wykorzystanie metod do ekstrakcji informacji na potrzeby identyfikacji aktywności terrorystycznych po zamachu z 11 września 2001 roku (Tang *et al.*, 2003) lub systemy do śledzenia groźnych sytuacji pogodowych, drogowych i innych związanych z bezpieczeństwem wydarzeń (Piskorski *et al.*, 2011).

1.2. Motywacje rozprawy

Zainteresowanie tematem ekstrakcji informacji z tekstu sięga roku 1987, kiedy to amerykańska agencja ds. obrony Stanów Zjednoczonych o nazwie DARPA (ang. *Defense Advanced Research Projects Agency*) po raz pierwszy zorganizowała konferencję poświęconą temu zagadnieniu o nazwie MUC[1] (ang. *Message Understanding Conference*). Celem konferencji było zwrócenie uwagi zespołów badawczych na zagadnienie ekstrakcji informacji z tekstu i możliwości wykorzystania tej technologii jako narzędzia wspomagającego akcje prewencyjne związane z obronnością kraju. Pierwsze dwie konferencje MUC-1 (1987) i MUC-2 (1989) były poświęcone przetwarzaniu wojskowych wiadomości dotyczących działań morskich i obserwacji terenów morskich (Grishman i Sundheim, 1996). W kolejnych latach (1991–1997) zostało zorganizowanych następnych pięć konferencji z tej serii poświęconych przetwarzaniu wiadomości prasowych dotyczących kolejno: aktywności terrorystycznej w Ameryce Łacińskiej (MUC-3 i MUC-4), przedsiębiorstw kapitałowych z przemysłu mikroelektronicznego (MUC-5), negocjacji dotyczących sporów pracowniczych i zmian na pozycjach kierowniczych w korporacjach (MUC-6), wypadków lotniczych, wystrzeleń rakiet i pocisków odrzutowych (MUC-7).

Głównym celem ekstrakcji informacji jest strukturalizacja informacji zawartych w dokumentach tekstowych. Informacje przedstawione w sposób uporządkowany i sformalizowany mogą być wykorzystane w wielu różnych zastosowaniach, m.in.:

- systemy odpowiedzi na pytania — „odpowiadanie na pytania, zanim zostaną zadane" (Fleischman *et al.*, 2003) poprzez indeksowanie faktów, a także agregację faktów w celu odpowiedzi na pytania przeglądowe (np. *Jaka firma wyemitowała najwięcej akcji w 2011 roku?*) lub jako dodatkowe źródło informacji przy ekstrakcji odpowiedzi z dokumentu (Walas i Jassem, 2010),
- monitorowanie wiadomości — analizowanie wiadomości pochodzących z różnych źródeł w celu śledzenia pewnych obiektów lub zjawisk masowych, np. epidemie chorób (Grishman *et al.*, 2002) czy wypadków drogowych i katastrof naturalnych (Piskorski *et al.*, 2011),
- obsługa klientów (Sarawagi, 2008) — zbieranie i przetwarzanie elektronicznych dokumentów (listy elektroniczne, zamówienia, faktury itp.) związanych z obsługą klientów biznesowych, m.in. strukturalizacja faktur (Zhu *et al.*, 2007), wniosków o odszkodowanie (Popowich, 2005), dokumentacji medycznych (Marciniak *et al.*, 2005) itp.,
- systemy dialogowe — automatyzacja telefonicznych centrów informacyjnych, np. automatyczne odpowiadanie na pytania związane z komunikacją miejską (Marciniak, 2010),
- porządkowanie danych w hurtowniach danych (Sarawagi, 2008) — wiele danych w dużych bazach przechowywanych jest w postaci tekstowej, np. adresy. Użycie takich danych do analizy w hurtowniach danych wymaga ich wcześniejszego ustrukturalizowania i znormalizowania poprzez wydzielenie atomowych elementów

1. Strona domowa: `http://www-nlpir.nist.gov/related_projects/muc/`

(w przypadku adresu może to być nazwa ulicy, miasta, kod pocztowy itp.) (Borkar *et al.*, 2001; Sarawagi i Bhamidipaty, 2002),

- katalogowanie produktów i usług (Sarawagi, 2008) — analiza i strukturalizacja ogłoszeń i reklam w celu automatycznego generowania katalogów produktów i usług (Muslea *et al.*, 1999; Soderland, 1999),
- wsparcie wyszukiwarek internetowych — rozpoznawanie i indeksowanie jednostek identyfikacyjnych, relacji i zdarzeń umożliwia wyszukiwanie dokumentów po określonych kategoriach semantycznych (Strzalkowski *et al.*, 2000; Suchanek *et al.*, 2006; Cafarella *et al.*, 2007); jednym z przykładów jest także polska wyszukiwarka semantyczna `www.ktoco.pl`, która bazuje na wynikach wyszukiwarki Google.com,
- systemy wnioskowania — ustrukturalizowana informacja może być wykorzystana jako baza wiedzy dla systemów wnioskowania, np. odpowiedzi na pytania o lokalizację z wykorzystaniem ontologii o obiektach geopolitycznych jako bazy wiedzy do wnioskowania (Walas, 2012),
- bazy cytowań (Sarawagi, 2008) — narzędzia ekstrakcji informacji są wykorzystywane do automatyzacji procesu tworzenia indeksów cytowań, a jednym z najbardziej znanych działających przykładów jest portal CiteSeer[2] (Lawrence *et al.*, 1999),
- badanie opinii publicznej — badanie opinii na temat produktów, firm i organizacji na podstawie wiadomości i komentarzy umieszczanych na portalach informacyjnych, np. badanie popularności partii politycznych oferowane przez Barometr Polityczny[3],
- porównywarki produktów — strony umożliwiające porównywanie cen produktów w różnych sklepach cieszą się bardzo dużym zainteresowanie (Doorenbos *et al.*, 1997). Obecnie na rynku polskim istnieje już wiele takich portali, m.in. `ceneo.pl`, `nokaut.pl`, `skapiec.pl`.

Największy postęp w dziedzinie ekstrakcji informacji został osiągnięty dla języka angielskiego. Mimo to temat ten jest nadal przedmiotem prac badawczych, ponieważ nie zostały opracowane na tyle uniwersalne metody, aby dla dowolnego zadania pozwalały osiągnąć zadowalające wyniki. Technologie opracowane dla języka angielskiego nie mają bezpośredniego przełożenia na język polski, m.in. dlatego, że język polski jest typologicznie odmienny od języka angielskiego. Przepiórkowski (2007) wskazuje szereg cech języków słowiańskich, które powodują, że ekstrakcja informacji jest trudniejsza niż dla języków germańskich czy romańskich. Są to m.in. rozbudowana odmiana frazy nominalnej, różna odmiana form homonimicznych słów pospolitych i nazw własnych, złożona odmiana nazw obcojęzycznych, rozmiar zbioru znaczników używanych do opisu morfologii (tzw. *tagset*), synkretyzm form fleksyjnych[4] bądź przypadków[5], złożoność

2. Strona domowa: `http://citeseer.ist.psu.edu/index`
3. Strona domowa: `http://www.zetema.pl/barometr/`
4. Ta sama forma wyrazowa może oznaczać różne formy gramatyczne, np. *dam* jako rzeczownik lub czasownik.
5. Takie same formy fleksyjne dla różnych przypadków, np. *pani* może być mianownikiem, dopełniaczem, celownikiem, miejscownikiem lub wołaczem.

fraz liczbowych oraz słabo ograniczony szyk zdania (dzięki rozbudowanej morfologii elementy zdania mogą być złożone ze sobą na wiele sposobów). Innymi czynnikami utrudniającymi adaptację istniejących rozwiązań do języka polskiego jest brak przenaszalności zasobów (np. reguły pisane są dla konkretnego języka oraz dostosowane są do użytych narzędzi, np. zestaw znaczników użytych do opisu morfologii), brak narzędzi lub niezadowalająca ich jakość dla języka polskiego (bardziej zaawansowane metody wymagają konkretnych narzędzi, np. pogłębiona analiza składniowa zdania lub ujednoznacznianie sensów słów).

1.3. Teza i cele pracy

Teza pracy została sformułowana następująco:

Wyszukiwanie informacji ograniczone do wyszukiwania jednostek identyfikacyjnych będących w określonej relacji semantycznej z zadaną jednostką może być realizowane bardziej efektywnie przy użyciu nadzorowanych metod ekstrakcji informacji niż przy użyciu tradycyjnych wyszukiwarek internetowych.

W celu uzasadnienia tezy zostały zdefiniowane trzy cele:

1. **Opracowanie nadzorowanej metody do rozpoznawania wybranych kategorii jednostek identyfikacyjnych w tekstach w języku polskim.**

   Realizacja tego celu pozwoli na automatyczne rozpoznawanie jednostek identyfikacyjnych określonych kategorii, między którymi będą rozpoznawane relacje semantyczne. W momencie rozpoczęcia prac nie istniało uniwersalne narzędzie ani ogólnodostępne zasoby do rozpoznawania jednostek identyfikacyjnych. Obecnie istnieje już szereg dostępnych zasobów i narzędzi dla języka polskiego, m.in. korpus NKJP[6] zawierający m.in. oznakowane jednostki identyfikacyjne (Przepiórkowski *et al.*, 2012), leksykony nazw własnych (Savary i Piskorski, 2011)[7]; narzędzie i modele do rozpoznawania nazw własnych NERF[8]. Narzędzia te i zasoby powstały równolegle do prac realizowanych w ramach rozprawy.
   Pomimo udostępnienia narzędzia NERF, wykorzystanie tego narzędzia okazało się być niewystarczające ze względu na niską kompletność rozpoznawania jednostek (zob. sekcję 4.9). NERF wykorzystuje model statystyczny bazujący wyłącznie na formach ortograficznych słów bez wykorzystania informacji z zewnętrznych źródeł (np. analiza morfologiczna, słowniki, wordnet). Konieczne było opracowanie

6. Strona www: `http://nkjp.pl/`
7. Dostępne na stronie: `http://clip.ipipan.waw.pl/Gazetteer`
8. Dostępne na stronie: `http://clip.ipipan.waw.pl/Nerf`

narzędzia pozwalającego na osiągnięcie wyższej kompletności.

2. **Opracowanie nadzorowanej metody do rozpoznawania relacji semantycznych określonych kategorii między jednostkami identyfikacyjnymi w tekście w języku polskim.**

   Opracowana metoda będzie wykorzystywała wieloaspektową analizę tekstu na różnych poziomach szczegółowości oraz będzie wykorzystywała dodatkowe, rozszerzalne, zewnętrzne zasoby językowe. W odróżnieniu od istniejących prac w tej dziedzinie dla języka polskiego (Piskorski *et al.*, 2004; Marciniak i Mykowiecka, 2007; Mykowiecka *et al.*, 2009), zaproponowana metoda będzie oparta na metodach maszynowego uczenia, które pozwolą na automatyzację procesu adaptacji do nowych zadań i dziedzin tekstów. Uniwersalność metody została także wsparta poprzez wykorzystanie istniejących zasobów (m.in. Słowosieci[9] (Piasecki *et al.*, 2009)) i ogólnie dostępnych narzędzi dla języka polskiego. Osiągnięcie tego celu pozwoli na rozpoznawanie i indeksowanie faktów w tekstach ciągłych, które posłużą jako baza wiedzy dla systemu odpowiedzi na pytania.

3. **Opracowanie prototypu systemu odpowiedzi na pytania wykorzystującego bazę wiedzy stworzoną przy użyciu narzędzi do rozpoznawania jednostek identyfikacyjnych i relacji między nimi.**

   Zostanie skonstruowany system ekstrakcji informacji pozwalający na udzielenie odpowiedzi na pytania o nazwy obiektów będących w określonej relacji względem zadanego obiektu. Pytania w języku naturalnym będą transformowane do postaci zapytań SQL, za pomocą których informacje poszukiwane przez użytkownika będą wyciągane z bazy danych. Dane do bazy danych zostaną pozyskane przy użyciu opracowanych metod rozpoznawania jednostek identyfikacyjnych i relacji semantycznych w tekście ciągłym.

Do osiągnięcia postawionych celi zostały zdefiniowane następujące zadania:

1. Opracowanie wytycznych znakowania jednostek identyfikacyjnych dla języka polskiego w oparciu o istniejące prace dla innych języków.
2. Opracowanie nadzorowanej metody rozpoznawania określonych przez wytyczne jednostek identyfikacyjnych dla języka polskiego.
3. Przegląd metod rozpoznawania relacji semantycznych.
4. Opracowanie wytycznych znakowania relacji semantycznych między jednostkami identyfikacyjnymi dla języka polskiego w oparciu o istniejące prace dla innych języków.

9. Strona domowa: `http://nlp.pwr.wroc.pl/wordnet`

5. Opracowanie korpusu tekstów w języku polskim znakowanych jednostkami identyfikacyjnymi i relacjami semantycznymi.
6. Opracowanie nadzorowanej i w pełni zautomatyzowanej metody rozpoznawania określonych przez wytyczne relacji semantycznych między jednostkami identyfikacyjnymi dla języka polskiego.
7. Opracowanie prototypu systemu odpowiedzi na pytania wykorzystującego metody rozpoznawania jednostek identyfikacyjnych i relacji między nimi.
8. Porównaniu prototypu z istniejącymi wyszukiwarkami internetowymi i systemami odpowiedzi na pytania. Ocena wyników.

1.4. Zakres oraz zawartość pracy

1.4.1. Struktura

Praca została podzielona na 7 rozdziałów. Poniżej znajduje się krótki opis zawartości poszczególnych rozdziałów:

Rozdział 1 zawiera opis przedmiotu pracy badawczej, motywacje, cele i zakres badań oraz streszczenie poszczególnych rozdziałów pracy, a także listę symboli i skrótów użytych w pracy.

Rozdział 2 omawia zastosowanie systemów ekstrakcji informacji w kontekście obecnego rozwoju technologicznego. Ponadto w rozdziale znajduje się szczegółowa definicja zadania ekstrakcji informacji oraz jego głównych podzadań, a także przegląd istniejących prac o tej temacie, ze szczególnych naciskiem na osiągnięcia dla języka polskiego oraz języka angielskiego (najbardziej zaawansowane prace).

Rozdział 3 zawiera charakterystykę dziedziny, na której będą prowadzone badania. W rozdziale znajduje się opis i statystyki danych testowych, schemat rozpoznawanych jednostek identyfikacyjnych oraz relacji semantycznych.

Rozdział 4 przedstawia opracowaną metodą do rozpoznawania jednostek identyfikacyjnych opartą o model sekwencyjny i metodę CRF. Zaprezentowane są osiągnięte wyniki dla dziedziny giełdowej oraz efekty wykorzystania jej na tekstach spoza dziedziny ekonomicznej.

Rozdział 5 przedstawia metodę rozpoznawania relacji między jednostkami identyfikacyjnymi w tekście w oparciu o nadzorowane uczenie, w szczególności indukcyjne programowanie logiczne (ILP) i klasyfikatory.

Rozdział 6 przedstawia prototyp systemu wyszukiwania informacji. Rozdział zawiera opis architektury systemu, omówienie procedury transformacji pytań w języku naturalnym do postaci zapytań SQL oraz przykładowe wyniki dla wybranych pytań wraz z porównaniem z innymi wyszukiwarkami internetowymi.

Rozdział 7 podsumowuje osiągnięte wyniki, przedstawia analizę błędów, zawiera dyskusję na temat uniwersalności przedstawionych metod oraz ich ograniczeń.

1.4.2. Załączniki

Załącznik A Pełny schemat jednostek identyfikacyjnych z podziałem na kategorie wraz z krótką definicją i przykładami.

Załącznik B Pełny schemat anotacji relacji zawierający słownik podkategorii wraz z przykładowymi zdaniami pochodzącymi z korpusu KPWr.

Załącznik C Przykłady automatycznie wygenerowanych reguł do rozpoznawania relacji semantycznych między jednostkami identyfikacyjnymi.

Załącznik D Opis formalizmu języka WCCL.

Załącznik E Opis dostępności narzędzi i zasobów powstałych w ramach rozprawy doktorskiej.

Załącznik F Słownik ważniejszych pojęć.

1.4.3. Oznaczenia i skróty stosowane w pracy

P — precyzja wyrażona w % (ang. *precision*),

R — kompletność wyrażona w % (ang. *recall*),

F — średnia harmoniczna precyzji i kompletności wyrażona w % (ang. *F-measure*),

TP — liczba przykładów prawidłowo zaklasyfikowanych jako pozytywne (ang. *true positive*),

FP — liczba przykładów nieprawidłowo zaklasyfikowanych jako pozytywne (ang. *false positive*),

FN — liczba przykładów nieprawidłowo zaklasyfikowanych jako negatywne (ang. *false negative*),

KPWr — Korpus Politechniki Wrocławskiej (zob. 3.4.1),

IKW — fragment korpusu KPWr użyty do redukcji cech (zob. 4.7.4),

CSER — korpus raportów giełdowych (ang. *Corpus of Stock Exchange Reports*; zob. 3.4.2),

CEN — korpus wiadomości gospodarczych (ang. *Corpus of Economic News*; zob. 3.4.4),

CPR — korpus raportów policyjnych (ang. *Corpus of Police Reports*; zob. 3.4.3),

SIL — uczenie na podstawie pojedynczych wystąpień elementów (ang. *Single Instance Learning*; zob. 2.2.2),

MIL — uczenie na podstawie wystąpień tych samych elementów w różnych kontekstach (ang. *Multiple Instance Learning*; zob. 2.2.2),

ILP — indukcyjne programowanie logiczne (ang. *Inductive Logic Programming*; zob. 5.4),

CRF — warunkowe pola losowe, (ang. *Conditional Random Fields*; zob. 4.3)

Rozdział 2

Ekstrakcja informacji

Ekstrakcja informacji jest to zadanie polegające na identyfikacji fragmentów tekstu spełniających określone ograniczenia semantyczne. W ramach niniejszej pracy ekstrakcja informacji została ograniczona do identyfikacji fragmentów tekstu w obrębie pojedynczych zdań. Założenie to wynika z braku dostępnych narzędzi do rozwiązywania koreferencji dla języka polskiego. Dopiero niedawno Kopeć i Ogrodniczuk (2012) podjęli pierwsze próby stworzenia takich narzędzi. Ograniczenia semantyczne są z góry ustalane na potrzeby konkretnego zadania ekstrakcji informacji i wynikają z planowanego zastosowania. Na przykład w poniższym zdaniu:

> *W dniu 20 stycznia 2011 roku pan Jan Nowak został wybrany na stanowisko prezesa spółki Markson S.A. z siedzibą we Wrocławiu.*

mamy szereg fragmentów tekstu reprezentujących pewne obiekty ze świata rzeczywistego lub ich cechy. Fraza *20 stycznia 2011 roku* to jednostka czasu reprezentująca pewien punkt (w ogólności przedział, w zależności od przyjętego poziomu szczegółowości) czasu na osi czasu, *Jan Nowak* jest nazwą osoby, *prezesa* (jako forma odmieniona słowa „prezes") jest nazwą stanowiska, *Markson S.A.* jest nazwą firmy i *Wrocławiu* jest nazwą miasta (jako forma odmieniona nazwy *Wrocław*). Pomiędzy wskazanymi elementami zachodzą pewne zależności, np. *Wrocław* jest nazwą miasta, w którym ma swoją siedzibę firma o nazwie *Markson S.A*, osoba *Jan Nowak* sprawuje funkcję zarządczą w firmie *Markson S.A*. Wskazane elementy są także atrybutami zdarzenia *wybór na stanowisko — 20 stycznia 2011 roku* jest datą wyboru osoby na stanowisko, *Jan Nowak* jest nazwą osoby wybranej na stanowisko, *prezes* jest nazwą stanowiska i *Markson S.A* jest nazwą firmy, w której wskazana osoba objęła wskazane stanowisko.

Ekstrakcja informacji jest zatem uproszczoną formą pełnej analizy semantycznej tekstu (Califf, 1998) i polega na rozpoznawaniu tylko wybranych informacji. Dlatego też każdy system ekstrakcji informacji tworzony jest pod kątem konkretnego zastosowania, np. analizy wiadomości prasowych pod kątem zamachów terrorystycznych (MUC-3 i

MUC-4), monitorowania i identyfikacji zdarzeń o wypadkach, zamachach i katastrofach naturalnych (Piskorski *et al.*, 2011), analizy dokumentów medycznych (Mykowiecka *et al.*, 2009).

W ramach ekstrakcji informacji wyróżnia się trzy główne podzadania (Hobbs i Riloff, 2010): rozpoznawanie jednostek identyfikacyjnych, rozpoznawanie relacji i rozpoznawanie zdarzeń (poszczególne podzadania zostały opisane w sekcji 2.1). Każde z tych trzech podzadań może być zdefiniowane na wiele sposobów, co jest uzależnione od przyjętych założeń i przyjętego kryterium sukcesu. Można wyróżnić kilka wspólnych kryteriów, które wpływają na ostateczny kształt zadania. Mogą to być: stopień strukturalizacji dokumentów, powtarzalność informacji między dokumentami (możliwość agregacji informacji) oraz stopień jednoznaczności danych. Wymienione kryteria zostały szczegółowo omówione w sekcji 2.2.

2.1. Definicja zadań

2.1.1. Rozpoznawanie jednostek identyfikacyjnych

Jednostki identyfikacyjne (w literaturze znane też jako *byty nazwane*; ang. *Named Entities*) są to fragmenty tekstu odnoszące się do pewnych obiektów ze świata rzeczywistego lub fikcyjnego, o których jest mowa w tekście. Nie istnieje jedna, ogólnie przyjęta formalna definicja jednostek identyfikacyjnych. Najbardziej rozpowszechniona jest klasyfikacja zaproponowana przez Linguistic Data Consortium (2008a) w ramach programu ACE[1] (ang. *Automatic Content Extraction*). Zgodnie z tymi wytycznymi jednostki identyfikacyjne zostały podzielone ze względu na dwa kryteria: językowe i semantyczne.

Kryterium semantyczne dzieli jednostki identyfikacyjne ze względu na klasy obiektów, do których odnoszą się jednostki. Najczęściej wyróżnia się takie klasy obiektów jak: osoba (ang. *person*), miejsce (*ang. location*), organizacja (*organization*), budynek (ang. *facility*), przedmiot (ang. *artifact*), wyrażenia liczbowe (ang. *numex*), wyrażenia czasowe (ang. *timex*). W klasyfikacji ACE podział jest gruboziarnisty i obejmuje pięć głównych kategorii (m.in. osoby, organizacje, obiekty geopolityczne, lokalizacje i budynki). Bardziej szczegółowe klasyfikacje uwzględniają kryteria podziału semantycznego, np. Sekine (2009) opracował dwupoziomową hierarchię obiektów zawierającą ponad 100 kategorii. Z kolei wielopoziomowa klasyfikacja obiektów może być zaczerpnięta z ontologii ogólnych, np. SUMO[2] (Niles i Pease, 2001).

W przetwarzaniu tekstów dziedzinowych ten podstawowy zestaw klas obiektów jest rozszerzany o klasy dziedzinowe, np. broń (ang. *weapon*) z dziedziny antyterrorystycznej (Patwardhan i Riloff, 2006), środki transportu i nazwy przystanków w dziedzinie

1. Strona domowa ACE: `http://www.itl.nist.gov/iad/mig/tests/ace/`
2. Dostępna w formie elektronicznej na stronie: `http://www.ontologyportal.org/`

transportu publicznego (Marciniak, 2010) lub jednostki miar, części ciała człowieka w dziedzinie medycznej (Marciniak i Mykowiecka, 2007).

Kryterium językowe, zgodnie z wytycznymi Linguistic Data Consortium (2008a), uwzględnia klasyfikację ze względu na:

- **rodzaj deskrypcji** — nazwa własna, deskrypcja określona, fraza nominalna,
- **liczność i jednoznaczność denotowanego zbioru obiektów** — zbiór pusty, jednoznaczny obiekt, klasa obiektów, nieokreślony podzbiór obiektów,
- **rodzaj odniesienia** — odwołanie wprowadzające nowy obiekt do dyskursu lub odwołujące się do wcześniej wspomnianego obiektu.

Podział leksykalno-gramatyczny jest bardziej ujednolicony i obejmuje cztery kategorie: nazwy własne, deskrypcje określone, frazy nominalne oraz zaimki osobowe.

2.1.2. Rozpoznawanie relacji semantycznych

W literaturze poświęconej przetwarzaniu języka naturalnego pojęcie *relacja semantyczna* najczęściej odnosi się do jednej z dwóch definicji. Pierwsza wiąże się z relacjami między jednostkami leksykalnymi, które definiowane są w obrębie wordnetu, takimi jak hiperonimia (słowo nadrzędne o szerszym znaczeniu, np. *pojazd* jest hiperonimem słowa *samochód*), hiponimia (relacja odwrotna do hiperonimii), meronimia, synonimia itd. (Fellbaum, 1998; Piasecki *et al.*, 2009). Druga definicja odnosi się do relacji semantycznych między jednostkami identyfikacyjnymi (zob. punkt 2.1.1). W niniejszej pracy termin *relacje semantyczne* będzie zawsze odnosił się do relacji semantycznych między jednostkami identyfikacyjnymi.

Zadanie rozpoznawania relacji semantycznych polega na identyfikacji par jednostek identyfikacyjnych występujących w obrębie jednego dokumentu, między którymi zachodzi określona relacja semantyczna, np. położenia (miasto X znajduje się w państwie Y, firma Z ma swoją siedzibę w mieście M), przynależności (osoba A pracuje w firmie B) itp.

Podobnie jak w przypadku jednostek identyfikakcyjnych nie istnieje jedna i wyczerpująca klasyfikacja relacji semantycznych dla wszystkich możliwych zastosowań z dziedziny ekstrakcji informacji. Jedną z najbardziej rozpowszechnionych klasyfikacji z dziedziny ogólnej (relacje zachodzące między podstawowymi jednostkami identyfikacyjnymi takimi jak osoby, organizacje, miejsca, przedmioty itd.) jest przewodnik relacji semantycznych opracowany przez Linguistic Data Consortium (2008b). Poza szeroko rozumianą dziedziną ogólna, można wyróżnić wytyczne dostosowane do konkretnej dziedziny, na przykład interakcje między białkami w dziedzinie medycznej.

Przykładowe zadania rozpoznawania relacji:

- relacje między osobami, organizacjami, jednostkami geopolitycznymi i lokalizacją (Linguistic Data Consortium, 2008b),
- miejsce i czas organizacji zdarzeń (np. imprez sportowych) (Brun i Hagège, 2009),
- interakcja genów i białek (Nédellec, 2005)[3] — jakie białka i geny wchodzą ze sobą

3. Strona projektu: `http://genome.jouy.inra.fr/texte/LLLchallenge/`

w interakcję, np. ze zdania "Expression of the sigma(K)-dependent cwlH gene depended on gerE." wynikają dwie zależności: (*sigma(K)*, *cwl(H)*) i (*gerE*, *cwlH*) (przykład pochodzi z Giuliano *et al.* 2006),
- charakterystyka białek, m.in. powiązanie z chorobami, występowanie białek w komórkach i tkankach (Craven i Kumlien, 1999),
- interakcje między białkami (Ono *et al.*, 2001).

2.1.3. Rozpoznawanie zdarzeń

Rozpoznawanie zdarzeń jest pewnego rodzaju rozszerzeniem relacji binarnych na relacje n-narne. Oznacza to, że zdarzenie opisane za pomocą zbioru jednostek identyfikacyjnych przypisanych do atrybutów tego zdarzenia. O ile w przypadku relacji mamy do czynienia z dwoma rolami przypisywanymi jednostkom (jednostka źródłowa i jednostka docelowa), tak w opisie zdarzeń mamy do czynienia z wieloma rolami, których liczba i znaczenie zależy od kategorii zdarzenia. Na przykład, transakcja kupna-sprzedaży może składać się z dwóch jednostek reprezentujących firmy lub osoby, jednej jednostki liczbowej oraz jednej jednostki czasu. Jedna z firm lub osób będzie miała przypisaną rolę *sprzedający*, a druga *kupujący*. Jednostka czasu będzie oznaczała datę podpisania umowy, a jednostka liczbowa wartość podpisanej umowy.

Drugim czynnikiem odróżniającym relacje od zdarzeń jest stanowość. W przypadku relacji mamy do czynienia z pewnym stanem, w jakim znajdują się dwa obiekty względem siebie. W przypadku zdarzeń odnosimy się przede wszystkim do zmiany stanów lub pewnych aktywności, które miały miejsce. Dlatego też zdarzenia przeważnie powiązane są z czasem (pewnym punktem w czasie, kiedy zaszła zmiana, lub też interwałem czasowym). Wynikiem zaistnienia zdarzenia może być także zmiana stanu, np. zaistnienie relacji. Na przykład w poniższym zdaniu:

> *W 2009 roku firma Oracle przejęła MySQL Development za 20 milionów dolarów.*

mamy opisane zdarzenie przejęcia jednej firmy przez drugą, z dodatkowymi atrybutami zdarzenia (rok przejęcia i kwota transakcji). Wynikiem tego zdarzenia jest zaistnienie relacji między MySQL Development i Oracle typu *jest częścią*.

2.2. Kryteria złożoności zadania

Poza zakresem semantycznym zadań ekstrakcji informacji opisanych w sekcji 2.1 można wyróżnić jeszcze szereg innych kryteriów wpływających na złożoność danego zadania. W tabeli 2.1 została przestawiona zbiorcza lista czynników, które mają istotny wpływ na realizację zadania. Szczegółowy opis poszczególnych kryteriów znajduje się w kolejnych punktach tej sekcji.

	Kryterium		**Kategorie**
A.	Strukturalizacja dokumentu	1.	Tekst ciągły, narracyjny
		2.	Częściowo ustrutkuralizowany tekst (np. znaczniki HTML, SGML)
		3.	Dokumenty ustrukturalizowane (tabele, listy itp.)
B.	Jednoznaczność informacji	1.	Informacje są jednoznaczne
		2.	Informacje są wieloznaczne
C.	Natężenie informacji	1.	Informacje występują pojedynczo
		2.	Wiele informacji w jednym dokumencie
D.	Modalność komunikatu	1.	Fakt
		2.	Negacja
		3.	Stopnie pewności

Tabela 2.1. Klasyfikacja kryteriów złożoności zadania ekstrakcji informacji

2.2.1. Strukturalizacja dokumentu

Czynnik *strukturalizacji dokumentu* świadczy o tym, w jakim stopniu semantyka elementów tekstu wynika ze struktur językowych użytych do zapisu informacji, a w jakim z przyjętego formatowania i ułożenia przestrzennego tych elementów. Z jednej strony mamy do czynienia z **tekstem ciągłym**, w którym informacja zapisana jest przy pomocy pełnych zdań i zgodnie z gramatyką użytego języka. W tym podejściu znaczenie poszczególnych elementów tekstu wynika przede wszystkim z użytych słów i zależności składniowych między nimi. Z drugiej strony mamy do czynienia z **tekstami ustrukturalizowanymi**, w których część informacji wynika z układu przestrzennego, w jakich informacja została zapisana. W tym podejściu trudność problemu została przeniesiona z interpretacji znaczenia słów w kierunku interpretacji rozmieszczenia tekstu względem siebie i w przestrzeni oraz interpretacji pewnych symboli użytych do zapisu formatowania dokumentu. Przykładem jest analiza struktury dokumentów HTML (Kushmeric, 1997).

Zapisanie informacji, że Jan Nowak jest prezesem firmy XYZ w postaci tekstu ciągłego może przyjąć postać następującego zdania: *Jan Nowak, który zajmuje stanowisko prezesa spółki Software Sp. z o.o. od 1996, poinformował o podpisaniu umowy.* W tym przykładzie osoba czytająca musi wiedzieć, że zwrot *zajmuje stanowisko prezesa* oznacza, że *jest prezesem.* Ta sama informacja może być zapisana w postaci podpisu w mailu:

```
1  Jan Nowak
2  Prezes Zarządu
3  Software Sp. z o.o.
```

w którym występują tylko trzy elementy: nazwa osoby, nazwa stanowiska i nazwa firmy.

Ponieważ jest to podpis listu, wiemy, że w pierwszej linijce przeważnie znajduje się imię i nazwisko osoby, w drugiej nazwa stanowiska, a w trzeciej nazwa firmy.

2.2.2. Jednoznaczność informacji

Agregacja informacji bierze pod uwagę powtarzalność tej samej informacji w różnych źródłach. Założenie, że ta sama informacja może wystąpić wielokrotnie, w różnych dokumentach pochodzących z różnych źródeł, pozwala na pominięcie nietypowych sposobów przedstawienia informacji i skupieniu się wyłącznie na częstych konstrukcjach. Z jednej strony problem różnorodności zapisu informacji został w tym podejściu odłożony na bok, jednak jego miejsce zajął problem grupowania informacji, tj. rozstrzygnięcia, czy pewien zbiór dokumentów opisuje to samo zdarzenie, czy kilka różnych. Przykładem może być analiza wiadomości prasowych pochodzących z różnych agencji opisujące te same zdarzenia ze świata (Piskorski *et al.*, 2011). Z drugiej strony zakłada się, że każdy dokument jest analizowany z osobna i rejestrowane jest każde wystąpienie informacji z osobna. Przy tym założeniu nacisk kładziony jest na analizę każdego dokumentu pod kątem wydobycia każdej istotnej informacji, nawet przedstawionej w nietypowy sposób. Przyjmuje się, że każdy dokument opisuje unikalne informacje, na przykład rejestr pacjentów (Marciniak i Mykowiecka, 2007; Mykowiecka *et al.*, 2009).

Bunescu (2007) wyróżnił dwa scenariusze rozpoznawania relacji: SIL (ang. *Single Instance Learning*) i MIL (ang. *Multiple Instance Learning*). Scenariusz SIL ukierunkowany jest na rozpoznanie każdego wystąpienia relacji między dwoma elementami w zadanym tekście. Z kolei MIL zakłada wydobycie listy par elementów, między którymi zachodzi pewien rodzaj relacji na podstawie dostarczonego zbioru dokumentów. Podejście MIL zakłada masowe przetwarzanie dużych ilości tekstów, w których poszukiwane pary elementów występują wielokrotnie w różnych kontekstach. Istnienie relacji między daną parą jest wnioskowane na podstawie wszystkich wystąpień tych elementów w różnych dokumentach. W SIL każde wystąpienie jest klasyfikowane niezależnie od innych wystąpień danej pary elementów w tekście, która w szczególności może być jedynym wystąpieniem w całym dostępnym tekście.

W scenariuszu MIL przyjmuje się, że każda para (lub przeważająca większość) tych samych elementów powiązana jest co najwyżej jedną kategorią relacji. Dzięki temu założeniu można zredukować nakład prac związany z przygotowaniem danych do uczenia. Zamiast znakować wszystkie wystąpienia elementów będących w zadanej relacji wybiera się pary, które uznaje się za odzwierciedlające daną relację. To założenie niesie za sobą ograniczenie dotyczące różnorodności relacji między parą takich samych elementów. Na przykład, rozważając parę nazw własnych (*Jan Nowak*, *Kraków*) możemy zdefiniować kilka typów relacji semantycznych, jakie zachodzą między tymi elementami, takie jak: *pochodzenie* (osoba A urodziła się w mieście B), *praca* (osoba A pracuje w mieście B) i *zamieszkanie* (osoba A mieszka w mieście B). W związku z tym wybór podejścia uzależniony jest od charakterystyki realizowanego zadania. Jeżeli dla rozpatrywanego problemu nie można przyjąć założenia, że między wszystkimi wystąpieniami

danej pary jednostek w tekście zawsze zachodzi jeden typ relacji, to niemożliwe jest wykorzystanie podejścia MIL. Wiąże się to z tym, że różne typy relacji zostaną zredukowane do pojedynczej relacji.

2.2.3. Natężenie informacji

Natężenie informacji wiąże się z liczbą różnych informacji występujących w analizowanym fragmencie tekstu. W przypadku niektórych dokumentów dokumentów można założyć, że pojedynczy dokument opisuje jedno wystąpienie jakiegoś zdarzenia lub wiele zdarzeń tego samego typu. Drugi przypadek jest oczywiście bardziej ogólny, ale jednocześnie bardziej złożony, ponieważ wymaga analizy dyskursu, aby prawidłowo przypisać różne cząstkowe informacje do odpowiednich zdarzeń. Przy założeniu wystąpienia tylko jednego zdarzenia na dokument nie jest to wymagane, ponieważ można przyjąć, że każdy rozpoznany atrybut jest elementem opisu tego samego zdarzenia. Na przykład akta medyczne opisują pojedyncze diagnozy choroby. Podobnie informacje prasowe opisują konkretne zdarzenia, ale z drugiej strony mogą odnosić się do podobnych lub powiązanych zdarzeń.

Analogicznie, w przypadku relacji w jednym zdaniu może wystąpić jedna instancja relacji lub większa ich ilość. W pierwszym przypadku wystarczy wykryć obecność relacji danej kategorii w zdaniu oraz jednostki identyfikacyjne. Natomiast w drugim przypadku konieczne jest poprawne połączenie jednostek w pary, ponieważ nie każda para jednostek w takim zdaniu tworzy relację.

2.2.4. Modalność komunikatu

Ostatnim rozważanym czynnikiem wpływającym na złożoność zadania ekstrakcji informacji jest modalność komunikatu, która dotyczy prawdziwości lub stopnia pewności nadawcy co do prawdziwości komunikatu. Struktura leksykalno-składniowa komunikatów będących stwierdzeniem faktu jest bardzo podobna do innych komunikatów, w których nadawca nie jest pewny prawdziwości komunikatu. Często problem modalności komunikatu traktuje się jako kolejny poziom klasyfikacji wypowiedzi. Poniżej znajdują się przykłady zdań pokazujące różne rodzaje modalności komunikatu:

- **fakt** — stwierdzenie istnienia pewnej relacji między elementami. Intencją autora jest zakomunikowanie pewnej informacji bez określania jej zgodności z rzeczywistością, np. *Wieża Eiffla znajduje się w Paryżu.*;
- **negacja** — zdanie zawiera zaprzeczenie istnienia pewnej relacji między elementami, np. *Statua Wolności nie znajduje się w Paryżu.*;
- **przypuszczenie** — nadawca nie wie, czy komunikat jest zgodny z rzeczywistością, np. *Wydaje mi się, że Andrzej mieszka w Krakowie.*;
- **życzenie** — zdanie wyrażające życzenie, aby coś miało miejsce, np. *Chciałbym, aby Andrzej mieszkał w Krakowie.*;
- **warunek** — pewna relacja zachodzi, ale tylko w określonych warunkach *Andrzej mieszka w Krakowie, ale tylko w czasie wakacji.*.

2.3. Metody ekstrakcji informacji

W tej części pracy znajduje się przegląd istniejących metod ekstrakcji informacji. Przegląd skupia się przede wszystkim na pracach poświęconych rozpoznawaniu relacji między jednostkami, ale także uwzględnia metody typowe dla rozpoznawania zdarzeń, które przy pewnych założeniach mogłyby być wykorzystane do zadania rozpoznawania relacji.

2.3.1. Ręczna konstrukcja reguł

Pierwsze systemy ekstrakcji informacji opierały się na ręcznie konstruowanych regułach[4]. Czynnikiem decydującym o sukcesie tego rozwiązania jest, po pierwsze, dostęp do ekspertów dziedzinowych, którzy będą w stanie zakodować swoją wiedzę dziedzinową w postaci reguł wyrażonych w pewnym języku formalnym, po drugie, ekspresyjność przyjętego formalizmu zapisu reguł oraz jego przejrzystość i czytelność dla eksperta dziedzinowego. Reguły mogą operować na wielu poziomach analizy tekstu, np. na poziomie znakowym, liniowej sekwencji tokenów lub nieliniowych zależności między elementami zdania po zastosowaniu analizy zależnościowej.

Ręcznie tworzone reguły charakteryzują się bardzo niewielką przenaszalnością miedzy zadaniami i dziedzinami. Są także bardzo silnie uzależnione od użytych narzędzi do przetwarzania tekstu, np. zbioru znaczników opisujących morfologię. Z drugiej strony reguły wsparte odpowiednim środowiskiem testowym dają dużą kontrolę nad procesem rozpoznawania informacji i pozwalają na prześledzenie procesu podejmowania decyzji. Jednakże adaptacja do nowego zadania lub dziedziny wymaga stałego nakładu pracy. Podejście czysto regułowe może być natomiast punktem wyjścia do automatyzacji procesu tworzenia reguł.

2.3.2. Automatyczne generowanie reguł

Głównym założeniem metod należących do tej grupy jest automatyczne generowanie reguł ekstrakcji informacji wyrażonych w pewnym formalizmie.

RAPIER (1998)

Califf (1998) przedstawił metodę rozpoznawania atrybutów zdarzeń, w której każdy atrybut zdarzenia (będący jednostką identyfikacyjną określonego typu i kategorii) rozpoznawany jest niezależnie przy pomocy reguł składających się z trzech wzorców opisujących:

- kontekst lewostronny,
- jednostkę do rozpoznania — ten fragment tekstu zostaje oznaczony jako rozpoznana nazwa własna określonej kategorii,
- kontekst prawostronny.

4. *Reguła* (tj. *reguła ekstrakcji informacji*) rozumiana jest jako zbiór ograniczeń wyrażonych w pewnym formalizmie pozwalających na identyfikację i klasyfikację istotnych fragmentów tekstu.

```
Lewy kontekst:          Element:                    Prawy kontekst:
1) word: in             1) list: max length: 2      1) word: ,
   tag: in                 tag: nnp                    tag: ,
                                                    2) tag: nnp
                                                       semantic: state
```

Rys. 2.1. Przykładowa reguła wygenerowana przez system RAPIER rozpoznająca lokalizację obiektu. Przykład pochodzi z pracy Califf (1998) i jest przedstawiony w niezmienionej postaci.

Proces generowania reguł odbywa się dwuetapowo. W pierwszym etapie dla każdego przykładu generowany jest zbiór najbardziej szczegółowych reguł w oparciu o dane uczące. Na tym etapie reguły składają się z ograniczeń na formę bazową i klasę gramatyczną słów. Ograniczenia na klasę semantyczną nie są nakładane na tym etapie ze względu na dużą niejednoznaczność słów. W drugiej fazie każdy zbiór reguł poddawany jest generalizacji w oparciu o zestaw heurystyk. Dla każdej pary reguł ze zbioru dokonuje się próby wygenerowania bardziej ogólnej reguły poprzez rozluźnienie dotychczasowych ograniczeń. Na tym etapie generowane są także ograniczenia na klasę semantyczną słów w oparciu o wordnet.

Na rysunku 2.1 przedstawiona jest przykładowa reguła, jaka została wygenerowana dla atrybutu lokalizacja na podstawie dwóch przykładów: *located in Atlanta, Georgia* i *offices in Kansas City, Missouri.*. Zgodnie z tą regułą element jest rozpoznany jako lokalizacja, jeżeli jest poprzedzony słowem *in* (pl. w) z klasą gramatyczną *in* (przymiotnik), składa się z maksymalnie dwóch słów oznaczonych klasami gramatycznymi *nnp* (nazwa własna) i występuje przed przecinkiem, po którym występuje słowo rozpoznane jako nazwa stanu (*state*).

Metoda ta była przetestowana na dwóch hermetycznych zbiorach danych: ogłoszeniach o pracę i ogłoszeniach o wykładach (Califf, 1998). W obu przypadkach problem wieloznaczności atrybutów zdarzeń nie występował, ponieważ każde ogłoszenie opisywało jedno zdarzenie. Wyniki dla poszczególnych atrybutów wynosiły od 80% do 99% precyzji i od 31% do 87% kompletności.

Dużym ograniczeniem tego rozwiązania jest niezależne rozpoznawanie elementów zdarzeń, które może prowadzić do niejednoznaczności w przypadku wystąpienia wielu zdarzeń w obrębie jednego dokumentu.

Espresso (2006)

Pantel i Pennacchiotti (2006) przedstawili częściowo nadzorowaną metodę do rozpoznawania relacji między jednostkami w oparciu o zbiór przykładowych par jednostek i nieoznakowany korpus. Proces ekstrakcji par odbywa się iteracyjnie, gdzie każda iteracja składa się z dwóch faz. Pierwsza faza polega na wygenerowaniu zbioru wzorców występowania relacji. W tym celu z nieoznakowanego korpusu wyciągane są wszystkie zdania, w których występuje para słów z zadanego zbioru przykładów. Na podstawie

wydobytych zdań tworzone są wzorce relacji, które przyjmują postać ograniczeń leksykalnych na sekwencję tokenów występujących pomiędzy jednostkami z pary. Następnie tworzony jest ranking wzorców, na bazie którego wybierane są tylko najlepsze wzorce. W drugiej fazie wybrane wzorce służą do wydobycia nowych par słów. Podobnie jak w przypadku wzorców wydobyte pary są oceniane i tworzony jest ranking. Najwyżej ocenione pary zostają dodane do początkowego zbioru przykładów i proces zostaje powtórzony.

Pomimo że ta metoda nie jest nastawiona na generowanie wzorców do rozpoznawania relacji, a koncentruje się na rozpoznaniu nowych par jednostek będących w relacji, to została ona tu uwzględniona, ponieważ możliwe jest niebezpośrednie pozyskanie wzorców. Nastawienie na rozpoznawanie nowych par jest odzwierciedlone w sposobie oceny, w której główną miarą jest precyzja. Skuteczność tej metody została przebadana m.in. dla relacji *następstwa*, np. *Benedykt XVI* był następcą *Jana Pawła II*, *George Bush* był następcą *Billa Clintona* (Pantel i Pennacchiotti, 2006) na angielskim zbiorze testowym TREC. Espresso osiągnęło precyzję na poziomie 49%. Kompletność dla tej relacji nie została podana.

Częściowo nadzorowane generowanie wzorców leksykalno-syntaktycznych (2009)

Brun i Hagège (2009) przedstawili częściowo nadzorowaną, heurystyczną procedurę generowania reguł ekstrakcji informacji w oparciu o zbiór przykładowych instancji i nieoznakowany korpus. Zbiór przykładów ma postać krotek, np. data, miejsce i nazwa zdarzenia. Generowane reguły wykorzystują analizę składniową zdania oraz rozpoznawanie jednostek identyfikacyjnych. Proces generowania wzorców rozpoczyna się od wyszukania w nieoznaczonym korpusie zdań zawierających wszystkie elementy krotek z początkowego zbioru lub część. Wybrane zdania poddawane są analizie zależnościowej i rozpoznawaniu jednostek identyfikacyjnych. Następnie dla każdego zdania generowany jest zbiór ograniczeń identyfikujących elementy krotki. Ograniczenia dotyczą klasy semantycznej elementów opisu oraz powiązań predykatowo-argumentowych. W kolejnym kroku ograniczenia poddawane są uogólnieniu, które polega na podmianie wartości argumentów zmiennymi określonej klasy jednostek. Na wydruku 5.2 znajduje się przykładowa reguła rozpoznająca nazwę olimpiady oraz rok i miejsce jej organizacji. Reguła oznacza: jeżeli X jest nazwą miejsca (`PLACE`) i podmiotem (`SUBJ`) słowa *accueillir*, Y jest nazwą zdarzenia (`EVENT`) i obiektem (`OBJ`) słowa *accueillir* oraz Z jest datą (`DATE`) i modyfikatorem (`VMOD`) słowa *accueillir* to X jest miejscem, a Z datą zdarzenia Y.

Przedstawiona metoda została przetestowana na korpusie 1500 zdań zawierających informacje o organizacji igrzysk olimpijskich. Celem było rozpoznanie roku i miejsca organizacji poszczególnych igrzysk. Jednoczesne rozpoznanie roku i miejsca osiągnęło ponad 90% precyzji i 49% kompletności (średnia harmoniczna wyniosła 63%). Mocną stroną zaproponowanego podejścia jest generowanie bardzo specyficznych reguł osiągających wysoką precyzję. Mimo to dla bardzo ograniczonego i relatywnie prostego

zadania (opis składał się z jednostki czasu, miejsca oraz określonej nazwy *Igrzyska Olimpijskie*) aż 10% wyników była błędna.

```
SUBJ(accueillir , PLACE(X))
& OBJ(accueillir , EVENT(Y))
& VMOD(accueillir , DATE(Z))
==>DATE-and-PLACE-of-EVENT(Z,X,Y)
```

Wydruk 2.1. Przykładowa reguła rozpoznająca atrybuty zdarzenia. Reguła pochodzi z pracy Brun i Hagège (2009).

2.3.3. Klasyfikacja z wykorzystaniem wektorów cech

Metody z tej grupy polegają na konstrukcji klasyfikatora, który na podstawie wektora cech opisującego parę jednostek identyfikacyjnych w pewnym kontekście (np. kontekst zdania lub dokumentu) wyznacza istnienie relacji semantycznej lub jej brak. Główną trudnością tego podejścia jest definicja właściwych cech, które pozwolą na podkreślenie elementów wskazujących na istnienie relacji oraz pozwalające na rozróżnienie między kategoriami relacji.

Kambhatla (2004)

Zbiór podstawowych cech do opisu pary jednostek identyfikacyjnych został przedstawiony m.in. przez Kambhatla (2004). Cechy uwzględniały informację morfologiczną, leksykalną i składniową. Zbiór cech zawierał następujące elementy:

- **słowa** — formy ortograficzne słów wchodzących w skład jednostek identyfikacyjnych oraz słów występujących pomiędzy jednostkami identyfikacyjnymi w zdaniu. W pracy nie zostało sprecyzowane, w jaki sposób zbiór o nieokreślonej liczbie symboli jest reprezentowany jako stała liczba cech,
- **kategoria semantyczna jednostki** — kategoria jednostki źródłowej i docelowej,
- **rodzaj deskrypcji** — rodzaj deskrypcji jednostki źródłowej i docelowej (zob. 2.1.1),
- **względne położenie jednostek** obejmujące liczbę słów występujących pomiędzy rozważaną parą jednostek, liczbę jednostek występujących pomiędzy rozważaną parą jednostek oraz cechy binarne określające, czy rozważana para jednostek znajduje się w tej samej frazie rzeczownikowej, czasownikowej lub przyimkowej,
- **cechy elementów nadrzędnych** — forma ortograficzna i klasa gramatyczna słowa będącego predykatem nadrzędnym jednostki oraz nazwa frazy składniowej, w której to słowo się znajduje (osobno dla jednostki źródłowej i docelowej),
- **ścieżka w drzewie rozbioru składniowego** — ścieżka pomiędzy rozważaną parą jednostek w postaci sklejenia nazw fraz składniowych z usunięciem powtórzeń.

Zaproponowany zestaw cech został wykorzystany do konstrukcji klasyfikatora opartego o model maksymalnej entropii (ang. *Maximum Entropy Model*; MEM). Model

został przetestowany na korpusie ACE. Model uwzględniający wszystkie zaproponowane cechy osiągnął wynik na poziomie 63% precyzji, 45% kompletności i 52% średniej harmonicznej.

Chan i Roth (2010)

W wielu pracach podejmowano próby wykorzystania bardziej złożonych cech wykorzystujących różne zasoby zewnętrzne. Jedną z takich prac jest praca Chan i Roth (2010), w której wprowadzono następujące cechy:

- **hierarchia relacji** — w przypadku hierarchicznej kategoryzacji[5] możliwe jest wielostopniowe klasyfikowanie relacji na różnych poziomach szczegółowości. Wychodząc od najwyższego poziomu hierarchii kategorii relacji wynik klasyfikacji może być wykorzystany przy klasyfikacji na kolejnych, bardziej szczegółowych poziomach hierarchii.
- **informacja o koreferencji** — jeżeli między dwoma jednostkami zachodzi koreferencja[6], to nie powinna zachodzić między nimi relacja semantyczna. Cecha jest wyrażona jako liczba rzeczywista z zakresu od 0 do 1 określająca prawdopodobieństwo istnienia relacji koreferencji między jednostkami.
- **informacja z zasobów zewnętrznych (Wikipedia)** — jeżeli para jednostek współwystępuje w tym samym dokumencie w pewnej bazie wiedzy, np. Wikipedii, to zwiększa się szansa, że jednostki są połączone jakąś relacją semantyczną. Cecha przyjmuje wartość 1, jeżeli w zbiorze dokumentów znajduje się zdanie, w którym para jednostek współwystępuje, 0 w przeciwnym wypadku.
- **grupowanie słów** — grupowanie słów zostało wykorzystane do uogólnienia znaczenia słów. W tym podejściu został wykorzystany wynik grupowania dokumentów z dziennika New York Times, który został zaprezentowany jako binarne drzewo grup. Każdy węzeł w drzewie ma przypisany unikalny identyfikator będący sekwencją „0” i „1”, odzwierciedlający ścieżkę od korzenia. Jeżeli dana forma bazowa występuję w drzewie grupowania, to przypisywany jest do niej kod odzwierciedlający grupę na określonym poziomie szczegółowości (odległości od korzenia).

Przedstawiony zestaw cech został przetestowany na korpusie ACE w drobnoziarnistym rozpoznawaniu relacji semantycznych (23 podkategorie relacji). Testy zostały wykonane przy użyciu pięciokrotnej walidacji krzyżowej w dwóch wariantach. W pierwszym do uczenia był wykorzystany cały zbiór testowy, a w drugim tylko 10% zbioru uczącego. Ograniczenie do 10% zbioru uczącego miało na celu sprawdzenie jakości rozpoznawania w sytuacji, kiedy zbiór uczący jest bardzo ograniczony.

Dla pierwszego wariantu relacje były rozpoznawane z precyzją 51,4% i kompletnością 57,7% (średnia harmoniczna wyniosła 54,4%). Dla drugiego wariantu, z zredukowanym zbiorem uczącym, relacje były rozpoznane z precyzją 37,9% i kompletnością 39,2% (średnia harmoniczna wyniosła 38,6%). To porównanie pokazuje, jak ważne jest

5. Kategorie relacji uporządkowane w wielopoziomową hierarchię.
6. Koreferencja to zjawisko językowe polegające na odniesieniu dwóch lub więcej wyrażeń językowych (fraza nominalna, nazwa własna, zaimek itp.) do tego samego obiektu pozatekstowego.

posiadanie odpowiednio dużego zbioru uczącego w stosunku do różnorodności zawartych w nich danych. Widać także, że zwiększenie zbioru dziesięciokrotnie pozwoliło na zwiększenie precyzji i kompletności zaledwie 1,5 raza.

Chan i Roth (2011)

W kolejnej pracy Chan i Roth (2011) przedstawili cechy odzwierciedlające struktury syntaktyczno-semantyczne, w których występuje para jednostek identyfikacyjnych. Wykorzystanie tych cech opiera się na spostrzeżeniu, że ponad 80% przykładów relacji zalicza się do 4 podtypów o dobrze określonej strukturze syntaktyczno-semantycznej. Zaliczają się do nich: modyfikatory (np. [Politechnika [Wrocławska]]), formy dzierżawcze, struktury przyimkowe (np. [Wieża Eiffla] w [Paryżu]) i struktury symboliczne (np. [Gdańsk] ([Polska])).

Wzorce były pozyskiwane w sposób automatyczny na podstawie części korpusu uczącego poprzez wygenerowanie wszystkich możliwych struktur dla par jednostek będących w relacji. Wzorce opisywały sekwencję słów ograniczoną jednostkami będącymi w relacji. We wzorcach wykorzystywana była informacja o klasie gramatycznej słów i frazach składniowych. Cechy oparte o wzorce były reprezentowane jako wartość logiczna, przyjmująca wartość 1 w przypadku dopasowania pary jednostek do wzorca lub 0 w przeciwnym wypadku.

Wykorzystanie wzorców pozwoliło na poprawę precyzji rozpoznawania wspominanych typów relacji z 51,6% do 56,4% przy nieznacznym spadku kompletności z 68,4% do 67,4%.

2.3.4. Klasyfikacja z wykorzystaniem funkcji jądrowych

Wspólnym mianownikiem tej grupy podejść jest wykorzystanie funkcji $K(X, X')$ zwanej funkcją jądrową, która, w dużym uproszczeniu, oblicza podobieństwo między obiektami X i X'. W odróżnieniu od klasyfikacji z wykorzystaniem wektorów cech (zob. sekcja 2.3.3) funkcja jądrowa operuje na całych obiektach i definiuje ich cechy poprzez bezpośrednie porównanie z innymi obiektami. Posiadając odpowiednią funkcję jądrową można sprowadzić proces klasyfikacji obiektu X przy użyciu np. metody SVM (ang. *Support Vector Machine*) do wyliczenia następującego równania:

$$\hat{r} = \arg\max_{r \in Y} \sum_{i=1}^{N} \alpha_{ir} K(X_i, X) \tag{2.1}$$

gdzie $\hat{r}$ to przewidywana klasa obiektu, X_i obiekt ze zbioru uczącego, α_{ir} waga obiektu i dla klasy r. Wagi α_{ir} zostają oszacowane w trakcie procesu uczenia.

Funkcja jądrowa oparta na podciągach znaków (2007)

Funkcja jądrowa oparta na podciągach znaków (ang. *Subsequence Kernel*; SSK) zaproponowana przez Bunescu (2007) liczona jest na podstawie liczby wspólnych podciągów dla określonych wzorców.

Zdanie zawierające parę anotacji powiązanych relacją zostaje podzielone na pięć części: sekwencja tokenów występująca przed pierwszą anotacją (s_f), pierwsza anotacja (x_1), sekwencja tokenów pomiędzy pierwszą i drugą anotacją (s_b), druga anotacja (x_2) oraz sekwencja tokenów występująca po drugiej anotacji (s_a). Dla takiej definicji podziału zdania zostały określone 4 kategorie wzorców zdań opisujących relacje, są to: FB — elementy wskazujące na wystąpienie relacji znajdują się w części s_f i s_b, B — w części s_b, BA — w częściach s_b i s_a oraz M — anotacje x_1 i x_2 występują po sobie i jedna jest modyfikatorem drugiej.

Dodatkowo przyjęto założenie, że wystarczą maksymalnie cztery słowa do rozpoznania relacji, w związku z czym przyjęto ograniczenie na długość sekwencji $s_f s_b$, s_b i $s_b s_a$ równe 4.

Funkcja jądrowa oparta na ścieżce zależności (2007)

Bunescu (2007) przedstawił funkcję jądrową opartą na ścieżce zależności (ang. *Dependency Path Kernel*), wykorzystującą pełną analizę składniową zdania. Pary jednostek porównywane są poprzez sprawdzenie najkrótszych ścieżek w rozbiorze zdania łączących dwa argumenty relacji (jednostkę źródłową i jednostkę docelową). Para jednostek reprezentowana jest jako zbiór wierzchołków (tokeny) i krawędzi (powiązań między tokenami). Każdy token reprezentowany jest jako zbiór atrybutów (forma ortograficzna, forma bazowa i klasa gramatyczna). Krawędź opisana jest przez kierunek zależności ($\leftarrow$ lub $\rightarrow$). Rodzaje zależności nie są brane pod uwagę. Funkcja jądrowa $K(x, y)$ jest równa liczbie różnych ścieżek składających się z cech słów i powiązań między słowami wspólnych dla obu relacji i jest obliczana ze wzoru (2.2) przy założeniu, że w każdym węźle brany jest pod uwagę tylko jeden warunek.

$$K(x, y) = \mathbb{1}(m = n) \cdot \prod_{i=1}^{n} c(x_i, y_i) \tag{2.2}$$

gdzie:

- m to długość ścieżki x,
- n to długość ścieżki y,
- x_i to i-ty token ścieżki x,
- y_i to i-ty token ścieżki y,
- $c(x_i, x_i)$ to funkcja zwracającą liczbę wspólnych cech dla tokenów x_i i y_i.

$\mathbb{1}(m = n)$ we wzorze 2.2 oznacza, że dla ścieżek o różnej długości funkcja zwróci wartość 0. Funkcja jądrowa przyjmuje także wartość 0, jeżeli dla dowolnego i zachodzi $c(x_i, y_i) = 0$, czyli para tokenów nie ma żadnych wspólnych cech.

Dużym ograniczeniem tej metody jest konieczność porównywania ścieżek o tej samej długości, co zmniejsza możliwości uogólniania. Na rysunku 2.2 znajduje się kilka przykładowych zdań zawierających informację o relacji *lokalizacja* pomiędzy jednostkami *Kowalski* i *Kraków* wraz z najkrótszą ścieżką między tymi jednostkami — wyniki uzyskane przy użyciu parsera zależnościowego MaltParser[7] z modelem danych skon-

7. Dostępny na stronie: `http://www.maltparser.org/`.

- **Kowalski** mieszka w **Krakowie**.
  (**Kowalski** → mieszka ← w ← **Krakowie**)
  długość 4
- Pan **Kowalski** mieszka w **Krakowie**.
  (**Kowalski** → Pan → mieszka ← w ← **Krakowie**)
  długość 5
- Pan Prezes **Kowalski** mieszka w **Krakowie**.
  (**Kowalski** → Prezes → Pan → mieszka ← w ← **Krakowie**)
  długość 6
- Pan Prezes **Kowalski** mieszka w mieście **Krakowie**.
  (**Kowalski** → Prezes → Pan → mieszka ← w ← mieście ← **Krakowie**)
  długość 7
- Pan Prezes **Kowalski** mieszka na ulicy Długiej w **Krakowie**.
  (**Kowalski** → Prezes → Pan → mieszka ← na ← ulicy ← w ← mieście ← **Krakowie**)
  długość 9

Rys. 2.2. Ścieżki zależności między parą jednostek identyfikacyjnych dla przykładowych zdań zawierających relację *lokalizacja*.

struowanym na bazie części korpusu NKJP (Wróblewska i Woliński, 2012). Jak widać, dla każdego zdania ścieżka jest innej długości, co oznacza, że dla każdego wariantu musi istnieć przykład w zbiorze uczącym, aby dany wariant mógł być rozpoznany. Na rysunku 2.3 została przedstawiona pełna analiza zależnościowa dla przytoczonych przykładów. Każdy wiersz reprezentuje jedno słowo ze zdania i zawiera następujące informacje (w kolejności wystąpienia):

1. Indeks słowa w zdaniu.
2. Forma ortograficzna słowa.
3. Forma bazowa słowa.
4. Kategoria gramatyczna.
5. Powtórzenie kategorii gramatycznej (dla języka polskiego te pole nie dostarcza nowej informacji).
6. Pozostałe atrybuty morfologiczne: przypadek, liczba, rodzaj, osoba, czas.
7. Indeks słowa nadrzędnego. 0 oznacza, że słowo jest predykatem nadrzędnym.
8. Kategoria relacji łączącej dane słowo ze słowem nadrzędnym.

2.3.5. Porównanie wyników

W ramach pracy zostały zebrane i przedstawione w tabeli 2.2 wyniki oceny omówionych metod. Ponieważ metody te były oceniane na różnych zbiorach danych, to niemożliwe jest ich bezpośrednie porównanie. Najczęściej stosowanym zbiorem testowym dla języka angielskiego jest zbiór ACE[8]. Zbiór ACE podzielony jest na dwie części: część

8. Strona www: `http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2011T08`

```
# Kowalski mieszka w Krakowie.
#-----------------------------
1       Kowalski        Kowalski        subst   subst   sg|nom|m1       2       subj    _       _
2       mieszka         mieszkać        verb    fin     sg|ter|imperf   0       pred    _       _
3       w               w               prep    prep    loc|nwok        2       comp    _       _
4       Krakowie        Kraków          subst   subst   sg|loc|m3       3       comp    _       _
5       .               .               interp  interp  _               2       punct   _       _


# Pan Kowalski mieszka w Krakowie.
#---------------------------------
1       Pan             Pan             subst   subst   sg|nom|m1       3       subj    _       _
2       Kowalski        Kowalski        subst   subst   sg|nom|m1       1       app     _       _
3       mieszka         mieszkać        verb    fin     sg|ter|imperf   0       pred    _       _
4       w               w               prep    prep    loc|nwok        3       comp    _       _
5       Krakowie        Kraków          subst   subst   sg|loc|m3       4       comp    _       _
6       .               .               interp  interp  _               3       punct   _       _


# Pan Prezes Kowalski mieszka w Krakowie.
#----------------------------------------
1       Pan             Pan             subst   subst   sg|nom|m1       4       subj    _       _
2       Prezes          prezes          subst   subst   sg|nom|m1       1       app     _       _
3       Kowalski        Kowalski        subst   subst   sg|nom|m1       2       app     _       _
4       mieszka         mieszkać        verb    fin     sg|ter|imperf   0       pred    _       _
5       w               w               prep    prep    loc|nwok        4       comp    _       _
6       Krakowie        Kraków          subst   subst   sg|loc|m3       5       comp    _       _
7       .               .               interp  interp  _               4       punct   _


# Pan Prezes Kowalski mieszka w mieście Krakowie.
#-------------------------------------------------
1       Pan             Pan             subst   subst   sg|nom|m1       4       subj    _       _
2       Prezes          prezes          subst   subst   sg|nom|m1       1       app     _       _
3       Kowalski        Kowalski        subst   subst   sg|nom|m1       2       app     _       _
4       mieszka         mieszkać        verb    fin     sg|ter|imperf   0       pred    _       _
5       w               w               prep    prep    loc|nwok        4       comp    _       _
6       mieście         miasto          subst   subst   sg|loc|n        5       comp    _       _
7       Krakowie        Kraków          subst   subst   sg|loc|m3       6       app     _       _
8       .               .               interp  interp  _               4       punct   _       _


# Pan Prezes Kowalski mieszka na ulicy Długiej w Krakowie.
#---------------------------------------------------------
1       Pan             Pan             subst   subst   sg|nom|m1       4       subj    _       _
2       Prezes          prezes          subst   subst   sg|nom|m1       1       app     _       _
3       Kowalski        Kowalski        subst   subst   sg|nom|m1       2       app     _       _
4       mieszka         mieszkać        verb    fin     sg|ter|imperf   0       pred    _       _
5       na              na              prep    prep    loc             4       comp    _       _
6       ulicy           ulica           subst   subst   sg|loc|f        5       comp    _       _
7       Długiej         długi           adj     adj     sg|loc|f|pos    6       adj     _       _
8       w               w               prep    prep    loc|nwok        6       adj     _       _
9       Krakowie        Kraków          subst   subst   sg|loc|m3       8       comp    _       _
10      .               .               interp  interp  _               4       punct   _       _
```

Rys. 2.3. Analiza zależnościowa uzyskana przy użyciu narzędzia MaltParser z modelem danych skonstruowanym na bazie części korpusu NKJP (Wróblewska i Woliński, 2012) dla przykładowych zdań zawierających relację *lokalizacja* między jednostkami *Kowalski* i *Kraków*.

uczącą zawierającą 422 dokumenty oraz część testową zawierającą 97 dokumentów. Dokumenty są oznakowane pięcioma kategoriami jednostek identyfikacyjnych (osoby, organizacje, budynki, lokalizacje i obiekty geopolityczne) oraz pięcioma kategoriami relacji semantycznych (rola, część, lokalizacja, otoczenie i społeczne). Zbiór uczący zawiera 7.646 relacji wewnątrzzdaniowych oraz 1.490 w zbiorze testowym. Metody K4, SSK (2.3.4), SPK-CCG i SPK-CFG (2.3.4) zostały przetestowane na części testowej. Z kolei metoda Espresso została przetestowana tylko na jednej kategorii relacji i wyłącznie pod kątem precyzji, a kolejne dwie metody zaprezentowane przez Chan i Roth (2010) i Chan i Roth (2011) zostały ocenione przy użyciu walidacji krzyżowej. Ostatnia metoda (Brun i Hagège, 2009) była oceniana na korpusie dziedzinowym opracowanym przez autorów na własne potrzeby.

Z analizy wyników przedstawionych metod można wysunąć kilka wniosków. Po pierwsze, rozpoznawanie ogólnych relacji semantycznych w różnorodnych tekstach nie jest problemem definitywnie rozwiązanym. Od momentu publikacji zbioru testowego ACE w 2004 do dzisiaj osiągnięto wyniki na poziomie 61% średniej harmonicznej, co jest jeszcze odległe od oczekiwanych 100%. Należy także zwrócić uwagę, że od momentu sformułowania zadania i udostępnienia danych testowych upłynęło już 8 lat.

Wprowadzenie ograniczeń na analizowane dane i kategorie rozpatrywanych relacji, a także ograniczenia na rozważane jednostki, jak to miało miejsce w pracy Brun i Hagège (2009), pozwala na osiągnięcie znacząco lepszych wyników nawet na poziomie powyżej 83% średniej harmonicznej. To wskazuje, że im więcej ograniczeń można nałożyć oraz im bardziej spójne są dokumenty, tym lepsze wyniki można osiągnąć.

Pomimo że najlepsze wyniki na zbiorze ACE zostały osiągnięte dla metody wektorowej, to nie można jednoznacznie wskazać dominującej metody. Najlepsza metoda oparta na funkcji jądrowej osiągnęła 50,5% średniej harmonicznej, a najlepsza metoda wykorzystująca wektory cech 61,5%. Należy mieć na uwadze, że nie można dokonać bezpośredniego porównania obu podejść, ponieważ procedury oceny były różne.

Ostatnim spostrzeżeniem jest to, że im więcej różnorodnych informacji jesteśmy w stanie dostarczyć i zakodować w postaci cech, tym lepsze wyniki możemy osiągnąć. Można to zaobserwować u Chan i Roth (2010, 2011), gdzie rozszerzenie zbioru cech pozwoliło na poprawę wyników o ok. 4 punkty procentowe średniej harmonicznej dla pierwszej metody i ok. 3 punkty procentowe dla drugiej metody.

W przedstawionych rozwiązaniach zostały wykorzystane uniwersalne metody maszynowego uczenia. Mimo to każde z podejść wymagało znaczącego wkładu pracy w celu dostosowania ogólnych metod do konkretnego zadania, którym jest rozpoznawanie relacji semantycznych. W przypadku metod wektorowych było to opracowanie zbioru cech, za pomocą których opisane zostały pary anotacji. Dla metod jądrowych było to opracowanie miary odległości między przykładami. Dla metod opartych na wzorcach było to opracowanie procedury generowania wzorców. Obsługa nowych kategorii relacji lub redefinicja istniejących może wymagać ponownego ręcznego dostrojenia tych elementów. Możliwość automatyzacji procesu adaptacji pozwoli na zredukowanie czasu potrzebnego na rozszerzenie zakresu obsługiwanych relacji. Biorąc pod uwagę powyższe wnioski, w

Metoda	**Rodzaj**	**Precyzja**	**Kompletność**	**Miara F**
Zbiór ACE (wszystkie relacje)				
Kambhatla (2004)	wektory	63,5%	45,2%	52,8%
K4	-	70,3%	26,3%	38,0%
SSK (2009)	f. jądrowa	73,9%	35,2%	47,7%
SPK-CCG (2009)	f. jądrowa	67,5%	37,2%	48,0%
SPK-CFG (2009)	f. jądrowa	71,1%	39,2%	50,5%
Zbiór ACE (wszystkie relacje; walidacja krzyżowa)				
Chan i Roth (2010) baseline	wektory	49,9%	51,0%	50,5%
Chan i Roth (2010)	wektory	51,4%	57,7%	54,4%
Chan i Roth (2010) 10%	wektory	37,9%	39,2%	38,6%
Zbiór ACE (wszystkie relacje; walidacja krzyżowa; wybrane typy)				
Chan i Roth (2011) baseline	wektory	51,6%	68,4%	58,8%
Chan i Roth (2011)	wektory	56,4%	67,4%	61,5%
Zbiór ACE (relacja *następca*)				
Espresso (2006)	wzorce	49%	n/d	n/d
Zbiór dziedzinowy (czas i miejsce olimpiady)				
Brun i Hagège (2009)	wzorce	90,3%	49,1%	63,6%

Tabela 2.2. Porównanie wyników rozpoznawania relacji na zbiorze ACE (j. angielski).

rozdziale 5 została przedstawiona w pełni zautomatyzowana procedura, wykorzystująca model wektorowy i automatyczne generowanie cech specyficznych dla określonych kategorii relacji przy użyciu indukcyjnego programowania logicznego (ang. *Inductive Logic Programming*; ILP).

Rozdział 3

Materiał badawczy

Celem pracy jest opracowanie nadzorowanej metody rozpoznawania jednostek identyfikacyjnych i relacji między tymi jednostkami, dlatego też koniecznie było zebranie materiału badawczego w postaci korpusów tekstowych znakowanych jednostkami i relacjami. W momencie rozpoczęcia prac nie istniały ogólnodostępne korpusy dla języka polskiego znakowane jednostkami i relacjami, dlatego koniecznie było skonstruowanie takich zasobów od podstaw. W tym rozdziale pracy zostały opisane przyjęte założenia i wytyczne dotyczące znakowania jednostek identyfikacyjnych (punkt 3.1) i relacji semantycznych (punkt 3.2), narzędzie o nazwie Inforex przygotowane na potrzeby zarządzania korpusami tekstowymi i znakowania ich (punkt 3.3), a także zebrane i opracowane korpusy tekstowe (punkt 3.4).

3.1. Wytyczne jednostek identyfikacyjnych

3.1.1. Założenia

Przy znakowaniu jednostek identyfikacyjnych przyjęto następujące założenia:

1. Zakres znakowania jednostek identyfikacyjnych silnie wiąże się z docelowym zastosowaniem, tj. rozpoznawaniem relacji między unikalnymi obiektami, które są identyfikowane przez nazwy własne. W związku z tym jednostki identyfikacyjne zostały ograniczone do nazw własnych (zgodnie z wytycznymi Linguistic Data Consortium (2008a) pozostałe klasy jednostek, które nie zostały uwzględnione, to frazy rzeczownikowe i zaimki). Należy podkreślić, że ograniczenie się wyłącznie do nazw własnych jest także praktycznym ograniczeniem złożoności zadania. Rozpatrywanie wszystkich klas kategorii jednostek znacznie zwiększyłoby nakład pracy na opracowanie potrzebnych zasobów.

2. Drugie ograniczenie wiąże się z poziomem szczegółowości rozpoznania struktury jednostek identyfikacyjnych, tj. zagnieżdżeniami nazw własnych. Ponieważ relacje semantyczne będą wykrywane wyłącznie między jednostkami rozłącznymi (jednostkami, które się w sobie nie zagnieżdżają), to nie ma potrzeby rozpoznawania zagnieżdżonych nazw własnych. Jednostki identyfikacyjne będą rozpoznawane jako sekwencje, a nie struktury zagnieżdżone, na przykład *Wydział Informatyki i Zarządzania Politechniki Wrocławskiej* zostanie zinterpretowany jako dwie następujące po sobie jednostki: *Wydział Informatyki i Zarządzania* jako nazwa instytucji oraz *Politechnika Wrocławska* jako nazwa organizacji.
3. Kolejnym założeniem jest przyjęty zakres kategorii semantycznych jednostek identyfikacyjnych. Schemat anotacji został opracowany w oparciu o wytyczne Linguistic Data Consortium (2008a), hierarchię jednostek opracowaną przez Sekine (2009)[1], wytyczne dotyczące relacji semantycznych między jednostkami identyfikacyjnymi (zob. sekcję 3.2), a także zebrane kolekcje dokumentów wchodzące w skład korpusu testowego. Grupa lingwistów po przejrzeniu kolekcji zebranych dokumentów zidentyfikowała wszystkie istotne kategorie nazw własnych występujące w korpusie.

3.1.2. Grupy i kategorie jednostek

Spośród kilkuset kategorii jednostek identyfikacyjnych zostało wybranych ponad 50 kategorii, które zostały pogrupowane tematycznie w kilka grup (m.in. na podstawie *Słownika nazw własnych*, Grzenia, 1998). Poniżej znajduje się skrócona lista kategorii jednostek podzielonych na grupy. Definicje poszczególnych kategorii i przykłady znajdują się w załączniku A.

- **antroponimy** (5 kategorii) — nazwy odnoszące się do osób lub istot mających cechy ludzkie (dotyczy to na przykład postaci fikcyjnych, baśniowych itp.), w tym imiona, nazwiska i pseudonimy, a także nacji i grup etnicznych,
- **chrematonimy** (22 kategorie) — nazwy obiektów stworzonych przez ludzi. Są to nazwy: zespołów (muzycznych, sportowych itd.), produktów seryjnych, firm, walut, dokumentów, wydarzeń (sportowych, muzycznych, rozrywkowych itd), budynków, instytucji, licencji, mediów (stacje telewizyjne i radiowe), organizacji, prasy (gazety i czasopisma), partii politycznych, technologii, tytułu utworów artystycznych, portali internetowych, stron WWW, oprogramowania komputerowego, systemów, umów, pojazdów i nagród.
- **hydronimy** (6 kategorii) — nazwy obiektów hydrograficznych. Są to nazwy: zatok, zalewów, jezior, oceanów, rzek i mórz.
- **organizmy żywe** (2 kategorie) — połączenie fitonimów i zoonimów, czyli unikalne nazwy roślin i zwierząt,

1. Hierarchia jednostek dostępna także na stronie: `http://nlp.cs.nyu.edu/ene/`

- **kosmonimy** (1 kategoria) — nazwy obiektów kosmicznych. Są to nazwy: planet, gwiazd, konstelacji itp. Ponieważ nazwy obiektów kosmicznych nie były liczne, dlatego została zdefiniowana tylko jedna kolektywna kategoria.
- **toponimy** (16 kategorii) — nazwy obiektów geograficznych i geopolitycznych. Są to nazwy: kontynentów, regionów, państw, miast, aglomeracji miejskich, podziałów administracyjnych pierwszego, drugiego i trzeciego stopnia, obszarów historycznych, wysp, przylądków, półwyspów, regionów, mierzei.
- **urbanonimy** (5 kategorii) — nazwy obiektów topologicznych związanych z miastami. Są to nazwy: dzielnic, ulic, parków, placów i osiedli.

3.2. Wytyczne relacji semantycznych

3.2.1. Założenia

Przy znakowaniu relacji semantycznych między jednostkami przyjęto następujące założenia:

1. **Rozważane są relacje między elementami występującymi w obrębie jednego zdania.** Ograniczenie to częściowo wynika z braku narzędzia do rozwiązywania anafory i koreferencji (wstępne prace nad tym zagadnieniem są dopiero prowadzone (Broda *et al.*, 2013; **?**)). W momencie opracowania narzędzia do rozwiązywania anafory i koreferencji problem rozpoznawania relacji między nazwami własnymi występującymi w różnych zdaniach może być sprowadzony do problemu rozpoznawania relacji w obrębie jednego zdania poprzez propagację nazw własnych zgodnie z powiązaniami anaforycznymi. Np. dla poniższego fragmentu:

   Jan Nowak urodził się w Warszawie. On od dwóch lat mieszka w Krakowie.

   mamy do czynienia z jedną relacją wewnątrzzdaniową *pochodzenie(Jan Nowak, Warszawie)* i jedną relacją międzyzdaniową *lokalizacja(Jan Nowak, Krakowie).* Wiedząc, że *on* odnosi się do nazwy *Jan Nowak* z pierwszego zdania, można by drugie zdanie przetransformować do postaci:

   Jan Nowak od dwóch lat mieszka w Krakowie.

   a tym samym w prosty sposób sprowadzić relację międzyzdaniową do wewnątrzzdaniowej. Oznacza to, że przy pewnym uproszczeniu problem rozpoznawania relacji między jednostkami występującymi w różnych zdaniach można sprowadzić do zadania rozpoznawania relacji wewnątrzzdaniowych i zadania rozwiązywania anafory.
2. **Wystąpienie relacji musi być podparte pewnymi przesłankami w zdaniu.** Wynika to z założenia, że opracowana metoda będzie ukierunkowana na wykrywanie reguł determinujących wystąpienie relacji, a nie na wyciąganiu z korpusów tekstowych par będących w relacji. Na przykład, ze zdania *Polska graniczy*

na zachodzie z Niemcami. wynika relacja sąsiedztwa między Polską i Niemcami. Natomiast ze zdania *Polska i Niemcy są członkami Unii Europejskiej* relacja ta nie wynika.

3.2.2. Kategorie relacji

Kategorie relacji semantycznych zostały ustalone w dużej mierze na podstawie schematu relacji ACE opracowanego przez Linguistic Data Consortium (2008b), w którym zostało opisanych sześć głównych kategorii relacji; są to: położenie (ang. *Physical*), część-całość (ang. *Part-Whole*), relacje międzyludzkie (ang. *Personal-Social*), związek z organizacją (ang. *ORG-Affiliation*), związek z przedmiotem (ang. *Agent-Artifact*) i związek z jednostkami geopolitycznymi (ang. *Gen-Affiliation*).

Na podstawie wytycznych ACE i jednostek identyfikacyjnych w korpusie KPWr zostały wybrane i wyspecyfikowane takie kategorie relacji semantycznych, które występowały w dostępnym korpusie przynajmniej kilkanaście razy. Przyjęty schemat relacji zawiera 8 kategorii relacji semantycznych. Dla każdej kategorii zostały wyspecyfikowane typy, które określają dopuszczalne pary kategorii jednostek identyfikacyjnych, między którymi może zachodzić dana kategoria relacji, oraz jej semantykę. Ustalenia dotyczące interpretacji poszczególnych kategorii relacji zostały skonsultowane z zespołem lingwistów[2]. Poniżej znajduje się zestaw kategorii relacji. Pełny schemat relacji wraz z dopuszczalnymi typami znajduje się w dodatku B.

- **autorstwo** (9 typów) — obiekt A jest autorem, twórcą, założycielem, inicjatorem, fundatorem obiektu B,
- **kompozycja** (7 typów) — obiekt A jest integralną częścią obiektu B; dotyczy to podziału administracyjnego, przestrzennego, strukturalnego,
- **narodowość** (2 typy) — osoba posiada narodowość kraju, jest obywatelem kraju, należy do nacji,
- **pochodzenie** (7 typów) — obiekt A pochodzi z obiektu B; w przypadku osób dotyczy to np. zamieszkania na określonym obszarze; w przypadku przedmiotów i organizacji oznacza powstanie, zainicjowanie na określonym terenie,
- **położenie** (41 typów) — fizyczne położenie jednego obiektu w drugim z wyłączeniem przypadków odpowiadających relacji *kompozycja*,
- **przynależność** (30 typów) — obiekt A przynależy do obiektu B, np. A jest członkiem organizacji B,
- **sąsiedztwo** (17 typów) — obiekt A sąsiaduje z obiektem B; dotyczy tylko bezpośredniego sąsiedztwa,
- **tożsamość** — alternatywne nazwy, formy stare, wycofane z użytku.

2. Szczególne podziękowania dla doktora Marka Maziarza, Joanny Jesionowskiej i Jana Wieczorka.

3.3. Inforex — system do zarządzania korpusami

Przystępując do konstrukcji korpusu treningowo-testowego konieczne było podjęcie decyzji o wyborze systemu do zarządzania i anotacji korpusów. Wybór został poprzedzony sformułowaniem wymagań, jakie powinien spełniać taki system. Do kluczowych wymagań należą m.in.:

- **zdalny dostęp** — dostęp do systemu przez przeglądarkę internetową bez konieczności instalowania dodatkowego oprogramowania na komputerze (z wyjątkiem standardowej przeglądarki internetowej),
- **współdzielenie danych** — natychmiastowy dostęp wielu użytkowników do współdzielonych danych,
- **kontrola dostępu** — przydzielanie użytkownikom praw dostępu do różnych perspektyw (możliwość wykonywania różnych zadań) i dokumentów,
- **śledzenie zmian i postępu prac** — dzięki śledzeniu zmian (dodawanie anotacji, relacji, edycja treści) możliwe jest wykrycie i uniknięcie kolejnych błędów popełnianych przez użytkowników,
- **transparentna aktualizacja systemu** — aktualizacje systemu powinny być niezauważalne dla użytkownika (wyeliminowanie konieczności ciągłej aktualizacji oprogramowania),
- **uniezależnienie od segmentacji** — system powinien umożliwiać znakowanie tekstów niezależnie od segmentacji tekstu (podział na tokeny i zdania). Założenie to wynikało z trwających prac nad ulepszeniem narzędzi do tokenizacji i analizy morfologicznej tekstu.
- **wsparcie bootstrappingu** — automatyczne znakowanie nowych tekstów na podstawie już oznaczonych.

W momencie dokonywania wyboru istniało lub było zapowiadanych kilka systemów do zarządzania korpusami tekstowymi, m.in.:

- **GATE** (Cunningham *et al.*, 2011) — szeroko rozpowszechniony szkieletowy system do przetwarzania dokumentów tekstowych, który jest rozwijany już od ponad 15 lat. Jest to aplikacja okienkowa napisana w Javie, dzięki czemu może być uruchomiona w dowolnym systemem operacyjnym posiadającym implementację wirtualnej maszyny Java. Jednym z komponentów aplikacji jest moduł do anotowania tekstów. Przetwarzane teksty przechowywane są na lokalnym komputerze, przez co tylko jeden użytkownik ma do nich dostęp. System nie wspiera współdzielenia dokumentów i istnieje konieczność zewnętrznego zarządzania procesem współdzielenia danych.
- **Manufakturzysta 2.0 Luna** (Marcińczuk, 2010) — aplikacja napisana w C# na potrzeby anotacji korpusu transkrypcji rozmów telefonicznych w ramach projektu LUNA (Marciniak, 2010). System był zaprojektowany pod kątem znakowania semantycznego tekstów, m.in. jednostkami identyfikacyjnymi, relacjami między tymi jednostkami, a także zdarzeniami. Ze względu na użyty język programowania

i dedykowane zastosowanie, aplikacja była wykorzystywana wyłącznie na platformie Windows. Drugim ograniczeniem, podobnie jak w systemie GATE, jest brak wsparcia współdzielenia danych.

- **GATE Teamware** (LLC, 2010) — internetowa wersja systemu GATE. W momencie dokonywania przeglądu systemów dostępne były tylko bardzo ogólne informacje na temat funkcjonalności tego systemu. Informacja o powstawaniu systemu była dostępna od 2010 roku, ale sama aplikacja została udostępniona w momencie, kiedy prace nad systemem Inforex były zaawansowane.
- **Anotatornia** (Przepiórkowski i Murzynowski, 2009) — internetowy system do znakowania tekstów na czterech poziomach: podział tekstu na tokeny, podział tekstu na zdania, analiza morfosyntaktyczna i sensy słów. System nie wspierał znakowania jednostek identyfikacyjnych i relacji między jednostkami. Implementacja systemu rozpoczęła się w 2009 i została zakończona publikacją kodów w lipcu 2010, czyli już w momencie zaawansowanych prac nad systemem Inforex.

Biorąc pod uwagę zdefiniowane wymagania, żaden z wymienionych systemów nie spełniał wszystkich kryteriów lub ze względu na ograniczony dostęp nie było możliwe ustalenie funkcjonalności systemu. W efekcie została podjęta decyzja o implementacji dedykowanego systemu zarządzania korpusem o nazwie Inforex (na rysunku 3.1 został przedstawiony zrzut ekranu przedstawiający perspektywę do anotacji). W przeciągu 2 lat został zaimplementowany system, który spełniał wszystkie wymagania. Szczegółowy opis funkcjonalności i architektury systemu Inforex został przedstawiony w Marcińczuk *et al.* (2012).

System Inforex został wykorzystany i jednocześnie rozbudowany w trzech innych projektach naukowo-badawczych[3]:

- **NEKST — Adaptacyjny system wspomagający rozwiązywanie problemów w oparciu o analizę treści dostępnych źródeł elektronicznych**[4] — opracowanie korpusu znakowanego jednostkami identyfikakcyjnymi,
- **SyNaT — System Nauki i Techniki**[5] — system został rozszerzony o komponenty do znakowania relacji między anotacjami, edytor znakowania sensów słów, edytor anafory, model bootstrappingu nazw własnych. System został użyty do zarządzania korpusem KPWr znakowanym na poziomie jednostek identyfikacyjnych, relacji semantycznych między jednostkami, powiązań anaforycznych, fraz składniowych i sensów słów,
- **Listy pożegnalne samobójców — lingwistyczne metody ustalania autentyczności tekstu**[6] — system został rozbudowany o moduł do ręcznej transkrypcji skanów listów. System został wykorzystany do zarządzania korpusem listów pożegnalnych.

3. Poza autorem pracy osoby, które przyczyniły się do rozwoju systemu Inforex, to Jan Kocoń i Marcin Ptak.
4. Strona projektu: `http://www.ipipan.waw.pl/nekst/`
5. Strona projektu: `http://www.synat.pl/`
6. Strona projektu: `http://pcsn.uni.wroc.pl/`

Rys. 3.1. Inforex — widok znakowania i przeglądania jednostek identyfikacyjnych.

3.4. Korpusy

3.4.1. KPWr — Korpus Politechniki Wrocławskiej

Korpus KPWr (Broda *et al.*, 2012) jest zbiorem ponad 1200 fragmentów tekstów różnego gatunku, do których należą: blogi, stenogramy, dialogi, proza współczesna, proza dawna, artykuły prasowe, artykuły popularno-naukowe, Wikipedia, teksty urzędowe i techniczne oraz z różnych dziedzin, np. nauki, prawa, religii. Każdy fragment tekstu jest ciągłym fragmentem oryginalnego dokumentu składającym się z nie więcej niż 300 wyrazów.

W korpusie KPWr 732 dokumenty zostały ręcznie oznaczone nazwami własnymi i relacjami między tymi nazwami (zgodnie ze schematami przedstawionymi w punktach 3.1 i 3.2). Korpus został podzielony w sposób losowy na trzy części — część ucząca (57% całości), część pomocnicza (18%) i część testowa (25%). Część ucząca była podstawowym materiałem do uczenia, część pomocnicza została użyta do określania parametrów i konfiguracji uczenia, a część testowa została użyta do porównywania ostatecznych wyników dla różnych podejść. Szczegółowe statystyki dla całego korpusu i poszczególnych nazw własnych znajdują się w tabeli 3.2 (nazwy własne) i tabeli 3.1 (relacje między nazwami własnymi).

3.4.2. CSER — korpus raportów giełdowych

Korpus CSER składa się z 1215 raportów giełdowych pochodzących z portalu GPWInfoStrefa[7]. Na portalu GPWInfoStrefa publikowane są oficjalne raporty roczne i okresowe spółek giełdowych. Zebrane raporty pochodzą z 2004 roku i zostały opracowane przez 185 różnych spółek.

Korpus został oznaczony jednostkami identyfikacyjnymi i został użyty do trenowania i walidacji modelu jednostek identyfikacyjnych. Szczegółowa liczba jednostek identyfikacyjnych została przedstawiona w tabeli 3.2.

3.4.3. CPR — korpus raportów policyjnych

Korpus CPR składa się z 12 zeznań policyjnych złożonych przez oskarżonych i świadków dostarczonych przez lokalny oddział policji. Dokumenty zostały zebrane w ramach projektu poświęconego automatycznej anonimizacji (Graliński *et al.*, 2009). Ponieważ oryginalne dokumenty zawierały poufne dane, ich zawartość została ręcznie zmieniona przez funkcjonariuszy policji. Wszystkie dane osobowe i inne dane mogą służyć identyfikacji opisanych zdarzeń i osób zostały zmienione na fikcyjne.

Korpus został użyty do oceny podstawowego modelu rozpoznawania jednostek identyfikacyjnych. Szczegółowa liczba jednostek identyfikacyjnych została przedstawiona w tabeli 3.2.

7. Strona www: `http://gpwinfostrefa.pl`

Korpus	**KPWr**	**Cz. ucząca**	**Cz. pomocnicza**	**Cz. testowa**
Statystyki dokumentów, słów i relacji				
dokumenty	732	418	135	208
tokeny	204.354	105.613	43.184	55.557
relacje	3.641	2.317	516	808
Szczegółowe statystyki relacji				
autorstwo	187	94	31	62
kompozycja	385	277	50	58
lokalizacja	1.141	816	156	169
narodowość	27	14	6	7
pochodzenie	106	70	12	26
przynależność	446	250	61	135
sąsiedztwo	159	98	28	33
tożsamość	187	87	33	67

Tabela 3.1. Statystyki korpusu KPWr — liczba relacji semantycznych.

3.4.4. CEN — korpus wiadomości gospodarczych

Korpus CEN składa się z 797 wiadomości pochodzących z portalu informacyjnego Wikinews[8] z działu ekonomicznego. Wiadomości pochodzą z okresu od 25 lutego 2005 do 10 czerwca 2010 roku (data pobierania wiadomości z Internetu).

Korpus został użyty do oceny przenaszalności między zbiorami dokumentów podstawowego modelu do rozpoznawania jednostek identyfikacyjnych. Szczegółowa liczba jednostek identyfikacyjnych została przedstawiona w tabeli 3.2.

8. Strona www: `http://pl.wikinews.org`

	CSER	CPR	CEN	KPWr
Ogólne statystyki				
Dokumenty	1.215	22	797	732
Zdania	10.097	1.527	7.305	11.884
Tokeny	282.401	24.772	144.004	204.354
Anotacje	4.011	1.004	4.997	16.591
Szczegóły wszystkich statystyk grupami				
antroponimy	4.002	1.550	2.048	6.692
chrematonimy	6.070	131	4.305	5.567
hydronimy	50	-	-	111
organizmy żywe	-	-	-	93
kosmonimy	35	-	-	15
toponimy	2.764	218	2.517	3.674
urbanonimy	80	42	406	439
Szczegóły wybranych anotacji[9]				
imiona	688	334	1.112	1.745
nazwiska	691	410	1.637	1.982
nazwy państw	474	27	1.755	978
nazwy miast	1.997	191	671	1.904
nazwy ulic	396	42	62	301

Tabela 3.2. Statystyki dokumentów, zdań, tokenów i anotacji jednostek identyfikacyjnych w korpusach CSER, CPR, CEN i KPWr.

Rozdział 4

Rozpoznawanie jednostek identyfikacyjnych

Rozpoznawanie relacji między jednostkami identyfikacyjnymi wymaga wcześniejszego wyróżnienia samych jednostek identyfikacyjnych w analizowanym tekście. Jednoczesne rozpoznawanie jednostek i relacji między nimi jest bardzo złożonym zadaniem, dlatego typowym rozwiązaniem jest niezależne realizowanie tych dwóch zadań. Z uwagi na to, że w momencie rozpoczynania pracy nad zagadnieniem rozpoznawania relacji semantycznych nie istniało uniwersalne narzędzie do rozpoznawania jednostek identyfikacyjnych, konieczne było skonstruowanie takiego narzędzia. W niniejszym rozdziale znajduje się opis modelu rozpoznawania jednostek identyfikacyjnych dla języka polskiego, opracowanego pod kątem rozpoznawania relacji semantycznych. Wynikiem badań przeprowadzonych w ramach rozprawy było skonstruowanie narzędzia o nazwie Liner2[1], pozwalającego na rozpoznawanie jednostek identyfikacyjnych w ciągłym tekście dla języka polskiego. Równolegle z zakończeniem prac nad tym zagadnieniem zostało opublikowane podobne narzędzie do rozpoznawania węższego zbioru kategorii jednostek identyfikacyjnych o nazwie NERF[2]. Narzędzie to zostało opracowane w ramach projektu NKJP[3] (Przepiórkowski *et al.*, 2012).

4.1. Sposób oceny

Do oceny jakości metod rozpoznawania jednostek identyfikacyjnych zostały użyte trzy podstawowe miary: precyzja, kompletność oraz średnia harmoniczna precyzji i kompletności (miara F). Precyzja i kompletność liczone są na poziomie całych anotacji, które są porównywane ze sobą z dokładnością do granicy anotacji (początek i długość anotacji) oraz jej kategorii. Precyzja (P) określona jest jako stosunek poprawnie rozpoznanych jednostek (TP) do wszystkich jednostek, które zostały rozpoznane ($TP+FP$)

1. Dostępne na stronie: `http://nlp.pwr.wroc.pl/liner2`
2. Dostępne na stronie: `http://zil.ipipan.waw.pl/Nerf`
3. Strona domowa: `http://www.nkjp.pl`

(zob. wzór 4.1). Kompletność (R) to stosunek poprawnie rozpoznanych jednostek (TP) do wszystkich jednostek, które powinny być rozpoznane ($TP + FN$) (zob. wzór 4.2). Średnia harmoniczna liczona jest ze wzoru 4.3. Głównym kryterium oceny jest średnia harmoniczna precyzji i kompletności (F).

$$P = \frac{TP}{TP + FP} \tag{4.1}$$

$$R = \frac{TP}{TP + FN} \tag{4.2}$$

$$F = \frac{2 * P * R}{P + R} \tag{4.3}$$

Zakładając, że anotacja ma postać krotki składającej się z trzech wartości *{początek, koniec, kategoria}*, gdzie *początek* i *koniec* to indeksy tokenów, a *kategoria* to nazwa kategorii anotacji, to anotacje a i b są sobie równe jeżeli spełniony jest warunek 4.4.

$$a.poczatek = b.poczatek \wedge a.koniec = b.koniec \wedge a.kategoria = b.kategoria \tag{4.4}$$

Wartości TP, FP i FN liczone są zgodnie z poniższymi wzorami.

$$TP = |WYNIK \cap REF| \tag{4.5}$$

$$FP = |WYNIK \setminus REF| \tag{4.6}$$

$$FN = |REF \setminus WYNIK| \tag{4.7}$$

Gdzie:

- $WYNIK$ to zbiór rozpoznanych jednostek,
- REF to referencyjny zbiór jednostek.

Do oceny modeli rozpoznających jednostki identyfikacyjne zostały wykorzystane dwie metody:

- **walidacja krzyżowa** — n-krotna walidacja krzyżowa polega na losowym podziale zbioru testowego na n równych części. Następnie wykonywanych jest n testów. W i-tym teście (gdzie $i \in \{1, ..., n\}$) oceniany model zostaje wyuczony na *n-1* częściach zbioru z pominięciem i-tej części i przetestowany na i-tej części zbioru. Dla wszystkich n testów zostają zsumowane wartości TP, TN i FP, na podstawie których zostaje wyliczona precyzja (P), kompletność (R) i średnia harmoniczna precyzji i kompletności (F). W przeprowadzonych eksperymentach przyjęto n=10.
- **walidacja międzydziedzinowa** — do oceny zostały wykorzystane dwa różne zbiory danych (zbiory zawierające teksty z odmiennych dziedzin). Model zostaje

wyuczony na pierwszym zbiorze i przetestowany na drugim. Dla otrzymanych wartości TP, TN i FP wyliczone zostają P, R i F. Do walidacji międzydziedzinowej został wykorzystany korpus CSER (zbiór uczący) i CEN (zbiór testowy).

4.2. Złożoność problemu

Rozpoznawanie jednostek identyfikacyjnych jest zadaniem złożonym. Czynniki, które wpływają na złożoność problemu, to:

- **nieograniczony zbiór nazw własnych** — nazwy własne mogą stanowić dowolny ciąg liter, cyfr i symboli,
- **niejednoznaczność nazw własnych**[4] — obiekty różnych kategorii mogą posiadać takie same nazwy, np. *Włochy* to zarówno nazwa państwa w Europie, wsi w Polsce jak i dzielnicy Warszawy,
- **słowa pospolite jako nazwy własne** — na przykład nazwiska pochodzące od słów pospolitych występujące na początku zdania, np. *Kopacz była ministrem zdrowia w rządzie Tuska*,
- **różna długość nazw własnych** — nazwy własne mogą składać się z wielu słów i zawierać słowa pisane małymi literami, np. spójniki w nazwach instytucji (*Wydział Informatyki i Zarządzania*),
- **nazwy własne mogą występować jedna po drugiej**, co utrudnia wykrycie ich granic — np. *Wydział Informatyki i Zarządzania Politechniki Wrocławskiej*, *Marek Jankowi oddał książkę*.

Biorąc pod uwagę powyższe czynniki, można dojść do wniosku, że rozpoznawanie nazw własnych nie jest prostym zadaniem. Do tej pory zadanie rozpoznawania nazw własnych dla języka polskiego było już przedmiotem zainteresowania w kilku pracach, m.in. Piskorski *et al.* (2004) opracował zestaw gramatyk i leksykonów do rozpoznawania nazw osób i firm. Reguły zostały przetestowane na 100 wiadomościach z działu finansowego pochodzących z elektronicznej wersji gazety *Rzeczpospolita* i osiągnęły odpowiednio 90,6% precyzji i 85,3% kompletności dla nazw osób oraz 87,9% precyzji i 56,6% kompletności dla nazw firm. Kolejne regułowe podejście do rozpoznawania nazw osób i firm zaprezentowały Urbańska i Mykowiecka (2005). Ręcznie opracowane reguły zostały przetestowane na „około 100 krótkich tekstach zebranych z Internetu" i osiągnęły odpowiednio 98% precyzji i 89% kompletności dla nazw osób oraz 85% precyzji i 73% kompletności dla nazw organizacji.

Z kolei Abramowicz *et al.* (2006) zaprezentował metodę rozpoznawania rozszerzonego zbioru nazw własnych, który obejmował nazwy miast, państw, ulic, a także osób. Metoda została przetestowana na 156 dokumentach zawierających ponad 19 tys. oznaczonych jednostek. Dokumenty pochodziły z gazety *Rzeczpospolita* (25 dokumentów), magazynu *Tygodnik Finansowy* (100) oraz internetowych portali informacyjnych (31).

4. W tym kontekście *niejednoznaczność* odnosi się do liczy obiektów denotowanych przez nazwę własną. Wiele obiektów może posiadać taką samą nazwę, przez co jednoznaczne wskazanie denotowanego obiektu na podstawie samej nazwy jest niemożliwe.

Dla nazw państw osiągnięto 91% precyzji i 93% kompletności, dla nazw miast 55% precyzji i 73% kompletności, dla nazw ulic 87% precyzji i 70% kompletności, a dla nazw osób 82% precyzji i 66% kompletności.

Kolejne regułowe podejście przedstawił Graliński *et al.* (2009), w którym ręcznie pisane reguły zostały wykorzystane do anonimizacji imion, nazwisk i nazw firm. System został przetestowany na 59 wiadomościach dostarczonych przez Interpol i osiągnął odpowiednio 93.53%, 88.66% i 94.59% precyzji dla imion, nazwisk i nazw firm. Ze względu na poufność danych system był oceniany przez osobę dokonującą anonimizacji. Ponieważ oceniane były wyłącznie elementy wskazane przez system do anonimizacji, kompletność nie została obliczona.

Bezpośrednie porównanie przedstawionych rozwiązań jest niemożliwe z dwóch powodów. Po pierwsze, każdy z eksperymentów został wykonany na różnych zbiorach danych. Po drugie, nie wszystkie z przedstawionych metod i zasobów są dostępne w takim stopniu, który umożliwiałby powtórzenie eksperymentów na dowolnym zbiorze danych. Także zbiory, na których były prowadzone eksperymenty, nie są możliwe do odtworzenia. Z tego względu konieczne było wyznaczenie wyników bazowych dla opracowanych metod maszynowego uczenia. Pierwszą metodą referencyjną jest rozpoznawanie nazw w oparciu o opracowane reguły, natomiast druga metoda bazuje na leksykonie nazw własnych (z wykorzystaniem istniejących zasobów częściowo rozszerzonych w sposób automatyczny) (Marcińczuk i Piasecki, 2010, 2011).

4.2.1. Podejście regułowe

Reguły opierały się na założeniu, że nazwy własne powinny być pisane dużą literą. To może być wystarczające do rozpoznania wystąpienia nazwy własnej, ale jest niewystarczające do kategoryzacji nazwy własnej. Dlatego konieczne było uwzględnienie w regułach kontekstu, w którym wystąpiły potencjalne nazwy własne. Reguły zostały opracowane w oparciu o korpus CSER i przetestowane na korpusach CPR i CEN. Dzięki temu udało się przetestować możliwość zapisu reguł w znanych tekstach, a także sprawdzenie uniwersalności tych reguł na różnych zbiorach danych.

Dla pięciu kategorii nazw własnych (imion, nazwisk, nazw państw, miast i ulic) zostało opracowanych dwadzieścia reguł do ich rozpoznawania. Poniżej znajdują przykładowe reguły zapisane w formalizmie WCCL[5].

1. **Reguły pomocnicze** — ich celem jest uproszczenie zapisu reguł właściwych poprzez znakowanie często używanych sekwencji tokenów symbolami pomocniczymi. Dzięki temu, w regułach właściwych, zamiast powtarzać skomplikowane warunki dopasowujące określone sekwencje możliwe jest odwołanie się do nich przy pomocy symboli pomocniczych.

5. Formalizm WCCL składa się z dwóch podjęzyków. Pierwszy podjęzyk przeznaczony jest do opisu reguł generujących cechy dla pojedynczych tokenów i został opisany przez Radziszewski *et al.* (2011). Drugi podjęzyk przeznaczony jest do pisania reguł dopasowujących sekwencje tokenów i przypisujących etykiety do całości lub fragmentu dopasowania.

- symbolem `UCF` zostają oznaczone słowa pisane dużą literą (forma ortograficzna pasuje do wyrażenia regularnego `"\p{Lu}+\p{Ll}+"`[6]),

```
apply(
  match(
    regex(orth[0], "\p{Lu}+\p{Ll}+")
  ),
  actions(
    mark(:1, "UCF")
  )
)
```

- symbolem `UCF_SEQ` zostają oznaczone sekwencje słów pisanych dużą literą,

```
apply(
  match(
    repeat( is("UCF") )
  ),
  actions(
    mark(:1, "UCF_SEQ")
  )
)
```

- symbolem `ROAD_IND` zostają oznaczone słowa wskazujące na możliwość wystąpienia nazwy ulicy. Są to słowa, których forma bazowa przyjmuje wartość ze zbioru {"*ulica*", "*aleja*", "*wybrzeże*"},

```
apply(
  match(
    interp( base[0], {"ulica", "aleja", "wybrzeże"})
  ),
  actions(
    mark(:1, "ROAD_IND")
  )
)
```

- symbolem `PERSON_IND` zostają oznaczone słowa wskazujące na możliwość wystąpienia nazwy osoby. Są to słowa, których forma bazowa przyjmuje wartość ze zbioru {"*pan*", "*pani*", "*powiedzieć*"}.

```
apply(
  match(
    interp( base[0], {"pan", "pani", "powiedzieć"})
  ),
  actions(
    mark(:1, "PERSON_IND")
  )
)
```

6. Wyrażenia regularne zapisane są zgodnie ze składnią opisaną na stronie `http://userguide.icu-project.org/strings/regexp`.

2. **Reguły właściwe** — służą do znakowania sekwencji słów będących nazwami własnymi występującymi w określonym kontekście. Do opisu sekwencji wykorzystywane są m.in. symbole pomocnicze.
   - reguła znakuje sekwencję słów pisanych dużą literą (`UCF_FIRST`), występującą po słowie wskazującym na wystąpienie nazwy ulicy (`ROAD_IND`) jako nazwę ulicy (`ROAD_NAM`),

```
apply(
  match(
    is( "ROAD_IND" ),
    is( "UCF_SEQ" )
  ),
  actions(
    mark(:2, "ROAD_NAM")
  )
)
```

   - reguła dopasowuje sekwencję trzech słów pisanych dużą literą (`UCF`) i poprzedzonych słowem wskazującym na nazwę osoby (`PERSON_IND`) i znakuje pierwsze dwa słowa jako imiona (`PERSON_FIRST_NAM`), a trzecie jako nazwisko (`PERSON_LAST_NAM`),

```
apply(
  match(
    is( "PERSON_IND" ),
    is( "UCF" ),
    is( "UCF" ),
    is( "UCF" )
  ),
  actions(
    mark(:2, "PERSON_FIRST_NAM"),
    mark(:3, "PERSON_FIRST_NAM"),
    mark(:4, "PERSON_LAST_NAM")
  )
)
```

   - reguła dopasowuje sekwencję słów pisanych dużą literą (`UCF_SEQ`) występującą po kodzie pocztowym i oznacza ją jako nazwę miasta (`CITY_NAM`),

```
apply(
  match(
    regex( orth[0], "[0-9]{2}" ),
    interp( orth[0], {"-"} ),
    regex( orth[0], "[0-9]{3}" ),
    is( "UCF_SEQ" )
  ),
  actions(
    mark(:4, "CITY_NAM")
  )
)
```

- reguła rozpoznaje nazwę państwa, miasta i ulicy występujących we wzorcu „w `COUNTRY_NAM`, `CITY_NAM`, ul. `ROAD_NAM`".

```
apply(
  match(
    interp( base[0], {"w"} ),
    is( "UCF_SEQ" ),
    interp( orth[0], {","} ),
    is( "CITY_NAM" ),
    interp( orth[0], {","} ),
    is( "ROAD_IND" ),
    is( "UCF_SEQ" )
  ),
  actions(
    mark(:2, "COUNTRY_NAM"),
    mark(:4, "CITY_NAM"),
    mark(:7, "ROAD_NAM")
  )
)
```

W tabeli 4.1 znajdują się wyniki oceny opracowanych reguł na wszystkich trzech zbiorach danych. Na zbiorze użytym do przygotowania reguł osiągnęły one ponad 96% precyzji i prawie 55% kompletności. Niska kompletność wynika z faktu, że dla wielu nazw własnych kontekst ich wystąpienia nie wskazywał na kategorię nazwy własnej (np. niemożliwe było rozróżnienie między miastem i państwem). Drugi powód to problem z odróżnieniem nazw własnych od słów pospolitych występujących na początku zdania. W wyniku tego większość nazw własnych występujących na początku zdania nie była rozpoznawana.

Zgodnie z oczekiwaniem dla wszystkich korpusów reguły osiągnęły bardzo wysoką precyzję — 96,5% dla CSER, 97,92% dla CPR i 92,31%. Jest to związane z tym, że reguły bazowały wyłącznie na kontekście, w którym wystąpiła nazwa, przez co opisywały tylko konteksty o jednoznacznej interpretacji. Z drugiej strony to założenie spowodowało utratę ogólności reguł i ich przenaszalności na inne zbiory dokumentów. Dla korpusu CPR reguły osiągnęły zaledwie 4,58% kompletności, a dla CEN tylko 2,4%.

Mimo niskiej kompletności reguły o wysokiej precyzji mogą być uzupełnieniem dla innych metod (np. statystycznych). Dzięki nim możliwe jest zakodowanie „oczywistych" dla czytelnika wzorców.

4.2.2. Podejście wykorzystujące leksykony

Druga metoda, która została użyta do wyznaczenia wyniku bazowego, wykorzystuje leksykony nazw własnych. Metoda polega na oznaczeniu wszystkich sekwencji słów w tekście, które występują w zadanym słowniku nazw, oraz przypisaniu im kategorii zgodnie z tym słownikiem. Metoda słownikowa dobrze sprawdza się w przypadku rozpoznawania jednoznacznych i popularnych nazw własnych, o ile dostępny jest słownik

	imiona	*nazwiska*	*miasta*	*państwa*	*ulice*	**wszystkie**
			CSER			
P	95,51%	96,48%	96,60%	98,20%	96,61%	96,50%
R	58,89%	55,73%	48,22%	39,61%	93,67%	54,92%
F	72,86%	70,65%	64,33%	56,45%	95,12%	70,00%
			CPR			
P	0,00%	0,00%	100,00%	0,00%	97,37%	97,92%
R	0,00%	0,00%	5,24%	0,00%	88,10%	4,68%
F	0,00%	0,00%	9,95%	0,00%	92,50%	8,94%
			CEN			
P	96,36%	94,55%	37,50%	0,00%	100,00%	92,31%
R	4,83%	3,43%	0,46%	0,00%	38,71%	2,40%
F	9,20%	6,62%	0,90%	0,00%	55,81%	4,68%

Tabela 4.1. Wyniki rozpoznawania nazw własnych z wykorzystaniem ręcznie opracowanych reguł.

o dużym pokryciu nazw. Do wyznaczenia wyniku bazowego dla metody słownikowej zostały użyte dwa słowniki (szczegółowe statystyki słowników przedstawione są w tabeli 4.2).

Pierwszy słownik o nazwie PG (Piskorski *et al.*, 2004) zawiera nazwy, które zostały ręcznie zweryfikowane. Dla większości nazw dostępne są wszystkie formy odmiany. Słownik ten charakteryzuje się dużą jakością[7], ale też niskim pokryciem nazw.

Drugi słownik, o nazwie IG, składa się z nazw, które zostały zebrane z różnych stron internetowych, na których były dostępne w częściowo ustrukturalizowanym formacie (tabele, listy, wyliczenia, linki o określonym formacie). Nazwy zostały automatycznie wyciągnięte ze stron internetowych przy pomocy standardowych programów linuksowych (*wget*, *grep*, *cut*, *sed*[8]) i wyrażeń regularnych. Imiona zostały zebrane z kalendarzy imienin i stron zawierających znaczenia imion; nazwiska z bazy PESEL; nazwy miast i ulic z bazy Głównego Urzędu Statystycznego (GUS)[9] oraz nazwy państw z listy państw na stronie polskiej Wikipedii. IG charakteryzuje się dużą kompletnością form bazowych, ale jednocześnie niższą jakością niż słownik PG. Wynika to m.in. z błędów na listach, dodatkowych, niezwiązanych informacji, które zostały błędnie zebrane jako nazwy.

7. Jakość słownika odnosi się do liczby lub procentu błędnych elementów w słowniku, czyli takich, które nie powinny się w nim znaleźć. Obecność niepoprawnych elementów w słowniku może mieć miejsce, kiedy słownik jest tworzony w sposób zautomatyzowany, przez co nie wszystkie elementy zostają ręcznie zweryfikowane.
8. Opis poszczególnych programów można znaleźć na stronie `http://manpages.ubuntu.com`.
9. Dostępne na stronie: `http://www.stat.gov.pl/gus`

	imiona	*nazwiska*	*miasta*	*państwa*	*ulice*	**wszystkie**
			PG*			
Lematy	118	16 997	29 370	201	0	**47 015**
Formy	1 166	44 608	29 699	1 761	0	**77 214**
Jednoznaczne	801	44 341	29 649	1 741	0	**76 532**
			IG			
Lematy	5 288	400 215	58 083	240	29 486	**493 312**
Formy	7 776	456 068	68 243	578	35 211	**567 876**
Jednoznaczne	3 904	432 250	50 726	393	20 530	**507 803**

* tylko wybrane kategorie nazw własnych

Tabela 4.2. Liczba nazw własnych poszczególnych kategorii w leksykonie PG i IG.

Dużą wadą słownika IG jest brak form odmienionych nazw. Aby uzupełnić te braki, została zastosowana automatyczna procedura rozszerzenia słownika IG o możliwe formy odmienione. W tym celu został wykorzystany duży korpus tekstów otagowany przy pomocy narzędzia TaKIPI (Piasecki, 2007) z aktywnym modułem Odgadywacza (Piasecki i Radziszewski, 2007) — moduł ten dla nieznanych form ortograficznych próbuje ustalić potencjalną formę bazową na podstawie statystycznej analizy końcówek. Z korpusu zostały wybrane wszystkie słowa zaczynające się dużą literą nieobecne w słowniku IG. Następnie zostały wybrane te słowa, których forma bazowa znajdowała się w słowniku IG, i dodane do słownika IG. Dodane słowa zostały przypisane do kategorii zgodnie z klasyfikacją ich form bazowych. Stosując tę procedurę, słownik IG został rozszerzony o 14% potencjalnych form odmienionych.

Dla metody słownikowej zostały przetestowane trzy warianty: z wykorzystaniem samego leksykonu PG, samego IG i połączonych PG i IG. Najlepsze wyniki pod względem średniej harmonicznej (**F**) zostały osiągnięte dla wariantu korzystającego z połączonych leksykonów PG i IG. Wyniki dla najlepszego wariantu dla poszczególnych kategorii nazw własnych zostały przedstawione w tabeli 4.3. Metoda słownikowa, dla wszystkich nazw własnych, osiągnęła od 35,27% średniej harmonicznej na korpusie CSER do 48,12% na korpusie CPR. Dla wszystkich trzech korpusów kompletność była znacząco wyższa niż precyzja. Niska precyzja wynika z kilku faktów, m.in. wieloznaczności nazw własnych (zob. tabela 4.2). Kolejnym powodem jest trudność rozróżnienia między nazwą własną a słowem pospolitym występującym na początku zdania. Wiele nazwisk, nazw miast i ulic pochodzi od słów pospolitych, przez co duża liczba słów występujących na początku zdania jest mylnie znakowana jako nazwa własna (potwierdzeniem jest niska precyzja dla nazw ulic od 0,45% do 8,20%, miast od 10,00% do 32,99% i nazwisk od 11,73% do 34,65%). Z kolei imiona i nazwy państw są bardziej jednoznaczne dzięki czemu są rozpoznawane z większą precyzją.

Pod względem kompletności najlepsze wyniki zostały osiągnięte dla imion i nazw

	imiona	*nazwiska*	*miasta*	*państwa*	*ulice*	**wszystkie**
			CSER			
P	44,95%	11,73%	32,99%	69,48%	7,25%	23,49%
R	88,19%	43,54%	72,30%	83,57%	67,59%	70,78%
F	59,55%	18,48%	45,31%	75,88%	13,10%	35,27%
			CPR			
P	83,54%	34,65%	30,73%	100,00%	8,20%	42,19%
R	82,28%	42,58%	35,08%	92,59%	50,00%	55,98%
F	82,90%	38,21%	32,76%	96,15%	14,09%	48,12%
			CEN			
P	52,91%	28,75%	10,00%	90,43%	0,45%	31,65%
R	64,54%	31,84%	36,83%	95,87%	35,48%	61,42%
F	58,05%	30,22%	15,72%	93,07%	0,88%	41,77%

Tabela 4.3. Wyniki rozpoznawania nazw własnych na korpusie CSER, CPR i CEN z użyciem metody słownikowej wykorzystującej połączone leksykony PG i IG.

państw, czyli kategorii o najmniejszej liczbie unikalnych nazw — dzięki temu leksykon dla tych nazw był pełniejszy. Dla imion kompletność wyniosła od 64,54% do 88,19%, a dla nazw państw od 83,57% do 95,87%. Jedną z przyczyn niekompletności leksykonów był brak wszystkich form odmienionych i obcojęzycznych odpowiedników.

4.3. Modele sekwencyjne

Rozpoznawanie jednostek identyfikacyjnych może być potraktowane jako zadanie tagowania sekwencji, którego celem jest przypisanie dla każdego elementu w sekwencji klasy z określonego zbioru. W tym przypadku sekwencja składa się z tokenów. Każdemu tokenowi zostaje przypisana jedna wartość określająca przynależność do pewnej kategorii anotacji. Wyróżnia się kilka formatów kodowania anotacji, m.in. IOB, IOB2, IOE, IOE2, IOBES (Loper, 2008). Najczęściej stosowanym formatem kodowania jest format IOB — skrót pochodzi od ang. ***I**nside*, ***O**utside*, ***B**egin*, co oznacza rozróżnienie między słowami wchodzącymi w skład jednostek (I), występującymi poza jednostkami (O) i rozpoczynającymi jednostki (B). Zbiór klas składa się z $2 * n + 1$ unikalnych elementów, gdzie n to liczba kategorii anotacji. Dla każdej kategorii P zdefiniowane są dwie klasy: `B-P` oznaczająca token rozpoczynający anotację oraz `I-P` oznaczająca kolejne tokeny anotacji. Dodatkowa klasa służy do oznaczenia pozostałych tokenów, nie będących częścią żadnej anotacji.

W ostatnich pracach poświęconych rozpoznawaniu jednostek identyfikacyjnych najlepsze wyniki osiągane są przy użyciu metody warunkowych pól losowych (ang. *Conditional Random Fields*; CRF) — jednej z metod tagowania sekwencji. Metoda CRF została już z powodzeniem zastosowana m.in. do języka angielskiego (McCallum i Li, 2003), polskiego (Marciniak, 2010; Marcińczuk *et al.*, 2011), bułgarskiego (Georgiev *et al.*, 2009), arabskiego (Benajiba *et al.*, 2008) i innych. Przewagą metody CRF nad metodami generacyjnymi, takimi jak np. HMM (ang. *Hidden Markov Models*), jest możliwość wykorzystania wieloelementowego zbioru cech opisujących obserwowane elementy w sekwencji. Z drugiej strony kluczowe jest dobranie optymalnego zbioru cech.

CRF (Lafferty *et al.*, 2001) jest nieskierowanym modelem grafowym zbudowanym w taki sposób, aby zmaksymalizować warunkowe prawdopodobieństwo $p(Y|X)$, gdzie X jest sekwencją klas pochodzących z określonego zbioru, a X to sekwencja obserwacji (w przypadku zadania rozpoznawania nazw własnych jest to sekwencja tokenów).

4.4. Zestaw cech

Pierwszym krokiem do opracowania modelu do rozpoznawania jednostek identyfikacyjnych było opracowanie zestawu cech, które posłużą do opisu tokenów. Na podstawie istniejących prac i własnych pomysłów został opracowany zbiór 15 cech, który został podzielony na 4 grupy: cechy ortograficzne, morfologiczne, słownikowe i oparte na wordnecie. W kolejnych podpunktach zostały opisane poszczególne grupy cech.

4.4.1. Cechy ortograficzne

Cechy ortograficzne bazują wyłącznie na formie ortograficznej tekstu podzielonego na tokeny. W tej grupie znajdują się następujące cechy:

- **orth** — forma ortograficzna, czyli słowo w takiej postaci, w jakiej wystąpiło w tekście,
- **prefiks(n)** — n pierwszych liter formy ortograficznej słowa, gdzie $n \in \{1, 2, 3, 4\}$. Jeżeli słowo jest krótsze niż n znaków, to w miejsce brakujących znaków wstawiany był „_”. Cecha ta wzięła się z obserwacji, że nazwy pewnych kategorii mają takie same prefiksy, np. nazwy organizacji *Prokom*, *Pro-Internet*.
- **sufiks(n)** — n ostatnich liter formy ortograficznej słowa, gdzie $n \in \{1, 2, 3, 4\}$. Jeżeli słowo jest krótsze niż n znaków, to w miejsce brakujących znaków wstawiany był „_”. Ta cecha pozwalała uchwycić powtarzające się końcówki nazw własnych, np. nazwiska (polskie: *Kowalski*, *Malinowski*; obcojęzyczne: *Robertson*, *Carlsson*).
- **wzorzec** — dla formy ortograficznej zostało zdefiniowanych kilka klas wzorców ortograficznych. Są to:
  - ALL_UPPER — słowo pisane dużymi literami, np. „PKP”, „USA”,
  - ALL_LOWER — słowo pisane małymi literami, np. „ulica”,
  - DIGITS — token składa się tylko z cyfr, np. „102”,

  - SYMBOLS — token składa się tylko z symboli, np. „-_-”, w szczególności pojedyncze znaki interpunkcyjne,
  - UPPER_INIT — pierwsza litera jest duża, pozostałe małe, np. „Andrzej”,
  - UPPER_CAMEL_CASE — pierwsza litera jest duża, pozostałe małe i duże, np. „CamelCase”,
  - LOWER_CAMEL_CASE — pierwsza litera jest mała, pozostałe małe i duże, np. „pascalCase”,
  - MIXED — kombinacja liter i cyfr, np. “H1M1”.
- **osiem binarnych cech ortograficznych** — cecha przyjmuje wartość „1”, jeżeli spełniony jest określony warunek, „0” w przeciwnym wypadku. Cechy binarne wzorowane są na obserwacjach, że pewne kategorie nazw własnych muszą spełniać określone warunki lub nie powinny ich spełniać (Marcińczuk *et al.*, 2011). Na przykład imiona i nazwiska powinny zawsze zaczynać się dużą literą. Cechy te są bardzo zbliżone do cechy wzorca formy ortograficznej, ale wprowadzają dodatkową informację w postaci agregowania wspólnych cech między klasami wzorców, np. ALL_UPPER, UPPER_INIT i UPPER_CAMEL_CASE zaczynają się dużą literą. Zdefiniowano następujące warunki:
  1. Słowo zaczyna się dużą literą.
  2. Słowo zaczyna się małą literą.
  3. Słowo zaczyna się od symbolu.
  4. Słowo zaczyna się od cyfry.
  5. Słowo zawiera dużą literę.
  6. Słowo zawiera małą literę.
  7. Słowo zawiera symbol.
  8. Słowo zawiera cyfrę.

4.4.2. Cechy morfologiczne

Cechy morfologiczne są wykorzystywane m.in. w regułach do rozpoznawania jednostek identyfikacyjnych (Piskorski, 2004a), więc mogą być cennym źródłem informacji dla modelu statystycznego. W ramach tej grupy zostały wyróżnione następujące cechy:

- **base** — forma bazowa słowa ustalona przy pomocy analizatora morfologicznego, redukuje zbiór możliwych form odmienionych i sposobu zapisu (np. wersalikami) do jednej formy bazowej,
- **ctag** — pełny tag opisujący morfologię słowa generowany przez analizator morfologiczny,
- **klasa gramatyczna** — np. rzeczownik, czasownik itd.; pełna lista klas gramatycznych znajduje się w Przepiórkowski *et al.* (2012),
- **przypadek** — jeden z siedmiu przypadków (mianownik, dopełniacz, celownik, biernik, narzędnik, miejscownik, wołacz),
- **rodzaj** — jeden z pięciu rodzajów (męski osobowy, męski zwierzęcy, męski rzeczowy, żeński i nijaki),

- **liczba** — pojedyncza lub mnoga.

4.4.3. Cechy oparte na wordnecie

Cechy oparte na wordnecie mogą być wykorzystane do redukcji różnorodności form bazowych występujących w tekście. Wyróżnione zostały dwie cechy:

- **synonim** — synonim słowa wybierany jako pierwsze alfabetycznie słowo występujące w tym samym synsecie. Ze względu na brak ujednoznaczniania znaczeń leksykalnych słów, w przypadku niejednoznaczności słowa, wybierane jest pierwsze słowo w kolejności alfabetycznej ze wszystkich synsetów[10].
- **hiperonim(n)** — hiperonim[11] słowa w odległości n, gdzie $n \in \{1, 2, 3\}$. W przypadku niejednoznaczności stosowana jest ta sama procedura, co dla synonimu.

4.4.4. Cechy słownikowe

Wyróżnione zostały dwa rodzaje cech słownikowych:

- **słownik słów kluczowych** — zawiera słowa charakterystyczne dla poszczególnych kategorii nazw własnych (słowa występujące przed nazwą, po nazwie lub będące częścią nazwy). Jeżeli słowo znajduje się na liście słów kluczowych, to cecha przyjmuje wartość 1, w przeciwnym wypadku 0. Dla każdej rozpoznawanej kategorii nazwy własnej istnieje osobna cech,
- **słownik nazw własnych** — lista nazw własnych jednej kategorii. Cechy kodowane są w formacie IOB, tj. jeżeli sekwencja tokenów zostanie znaleziona w słowniku kategorii K, to pierwszy token dla cechy K przyjmuje wartość `B`, a kolejne tokeny wartość `I`. Pozostałe tokeny, które nie zostały znalezione w słowniku, przyjmują wartość `O`. Dla każdej rozpoznawanej kategorii nazwy własnej istnieje osobna cecha.

4.5. Model bazowy CRF

Pierwszym krokiem w kierunku konstrukcji modelu CRF było wyznaczenie bazowej jakości systemu i porównanie z wynikami dla metod bazowych. W tym celu zostały użyte wszystkie dostępne cechy zdefiniowane w sekcji 4.4.1 i domyślne wartości parametrów CRF dla kilku wariantów kontekstu dla cech. W kolejnych sekcjach znajdują się wyniki walidacji krzyżowej i oceny międzydziedzinowej.

4.5.1. Walidacja krzyżowa

Ocena jakości rozpoznawania jednostek została wykonana przy użyciu dziesięciokrotnej walidacji krzyżowej na korpusie CSER. Walidacja krzyżowa polega na podziale

10. Synset jest zbiorem jednostek leksykalnych posiadających takie samo znaczenie. Jednostki o tym samym znaczeniu są wymienne w obrębie tego samego kontekstu (Piasecki *et al.*, 2009)
11. Hiperonim jest jednostką leksykalną o szerszym znaczeniu niż jednostka podrzędna, np. *zwierzę* jest hiperonimem *kota*.

korpusu na n części i wykonaniu n eksperymentów. W każdym z eksperymentów zbiorem uczącym jest cały korpus z wyłączeniem i-tej części, a zbiorem testowym i-ta część korpusu. W tabeli 4.4 znajdują się wyniki osiągnięte dla poszczególnych kategorii nazw własnych. Dla wąskiego kontekstu (token bieżący, jeden poprzedzający i jeden następujący) model CRF osiągnął 92,07% precyzji i 89,47% kompletności. Dla szerokiego kontekstu (token bieżący, dwa poprzedzające i dwa następujące) model CRF osiągnął znacznie wyższą precyzję na poziomie 95,20% przy zachowaniu kompletności na podobnym poziomie. Wyższa precyzja dla szerokiego kontekstu oznacza, że im szerszy kontekst nazwy własnej, tym większa szansa na wystąpienie cechy wskazującej na kategorię nazwy własnej. Z drugiej strony zwiększanie kontekstu może negatywnie wpłynąć na ogólność modelu.

	imiona	*nazwiska*	*państwa*	*miasta*	*ulice*	**wszystkie**
			Kontekst [-1,0,+1]			
P	94,67%	95,78%	81,22%	92,53%	93,33%	92,07%
R	81,38%	87,60%	86,15%	95,14%	84,15%	89,47%
F	87,52%	91,51%	83,61%	93,82%	88,51%	90,75%
			Kontekst [-2,-1,0,+1,+2]			
P	96,89%	97,85%	89,67%	94,74%	96,67%	95,20%
R	80,23%	87,88%	82,68%	95,35%	95,08%	90,00%
F	87,77%	92,60%	86,04%	95,04%	95,87%	92,53%

Tabela 4.4. Ocena bazowego modelu CRF na korpusie CSER.

4.5.2. Walidacja międzydziedzinowa

Uniwersalność[12] bazowego modelu CRF została przetestowana na korpusach CPR i CEN, po wcześniejszym wyuczeniu modelu na całym korpusie CSER. Podobnie jak przy walidacji krzyżowej zostały przetestowane dwa warianty modelu CRF, tj. dla wąskiego i szerokiego kontekstu. Pod względem precyzji dla obu korpusów model z większym kontekstem osiągnął lepszą precyzję. W przypadku korpusu CPR była to różnica 2 punktów procentowych, a w przypadku CEN była to nieznaczna różnica 0,4 punktu procentowego. Zwiększenie kontekstu spowodowało także zmniejszenie się kompletności o 4 punkty procentowe w przypadku CPR i 2 punkty procentowe dla CEN.

Model z szerokim kontekstem pozwala na rozpoznanie bardzo precyzyjnych kontekstów występowania nazw własnych. Dzięki temu można osiągnąć dużą precyzję modelu, odbywa się to jednak kosztem jej kompletności.

Jakość rozpoznawania nazw własnych osiągnięta przez model bazowy CRF jest znacznie wyższa od wyników osiągniętych przez metodę regułową i słownikową. Dla

12. *Uniwersalność* rozumiana jest jako skuteczność rozpoznawania nazw własnych przez model w tekstach znacząco różniących się od tych, które wystąpiły w zbiorze uczącym.

walidacji krzyżowej na korpusie CSER średnia harmoniczna wyniosła 92,53%, podczas gdy dla podejścia regułowego osiągnięto 70,00%, a dla podejścia słownikowego zaledwie 35,27%. Dla oceny międzydziedzinowej na korpusie CPR osiągnięto 67,72% średniej harmonicznej, a dla metod bazowych odpowiednio 8,94% dla metody regułowej i 48,12% dla metody słownikowej. Podobne wyniki zostały osiągnięte na korpusie CEN.

	imiona	*nazwiska*	*państwa*	*miasta*	*ulice*	**wszystkie**
			Kontekst [-1,0,+1]			
P	93,89%	93,06%	100,00%	89,05%	100,00%	92,88%
R	50,75%	48,91%	81,48%	63,87%	50,00%	53,29%
F	65,89%	64,11%	89,80%	74,39%	66,67%	67,72%
			Kontekst [-2,-1,0,+1,+2]			
P	94,08%	92,82%	100,00%	95,69%	100,00%	94,48%
R	47,75%	44,04%	81,48%	58,12%	54,76%	49,40%
F	63,35%	59,74%	89,80%	72,31%	70,77%	64,88%

Tabela 4.5. Międzydziedzinowa ocena modelu bazowego na korpusie CPR.

	imiona	*nazwiska*	*państwa*	*miasta*	*ulice*	**wszystkie**
			Kontekst [-1,0,+1]			
P	96,57%	93,06%	91,19%	79,91%	71,43%	91,15%
R	58,98%	51,29%	70,86%	55,71%	16,13%	59,98%
F	73,23%	66,13%	79,75%	65,65%	26,32%	72,35%
			Kontekst [-2,-1,0,+1,+2]			
P	97,05%	94,42%	90,31%	80,87%	62,50%	91,41%
R	57,06%	49,11%	68,73%	50,84%	16,13%	57,53%
F	71,87%	64,61%	78,06%	62,43%	25,64%	70,62%

Tabela 4.6. Międzydziedzinowa ocena modelu bazowego na korpusie CEN.

4.6. Rewizja korpusów i zasobów

Po wykazaniu, że bazowy model CRF pozwala na osiągnięcie lepszych wyników od prostych metod regułowych i słownikowych, uzyskane rezultaty zostały poddane szczegółowej ocenie. W jej wyniku zostało wprowadzonych kilka poprawek, które wiązały się z weryfikacją poprawności korpusów, zmianą sposoby tokenizacji i automatycznym uzupełnieniem słowników (Marcińczuk i Janicki, 2012). W tabeli 4.7 zostały przedstawione wyniki walidacji bazowego modelu na korpusach CSER i CEN.

Walidacja	Precyzja	Kompletność	Miara F
krzyżowa na CSER (B1.1)	95,20%	90,00%	92,53%
międzydziedzinowa na CEN (B2.1)	91,41%	57,53%	70,62%

Tabela 4.7. Wynik modelu bazowego na korpusach CSER i CEN.

4.6.1. Weryfikacja poprawności korpusów

Podczas weryfikacji wyników zostało wychwyconych kilkanaście błędów, m.in. brakujących anotacji, anotacji o błędnej kategorii i anotacji o błędnym zakresie. Bazowy model CRF został ponownie przetestowany na poprawionych korpusach (zob. tabela 4.8).

Walidacja	Precyzja	Kompletność	Miara F
krzyżowa na CSER (B1.2)	96,79%	92,38%	94,53%
międzydziedzinowa na CEN (B2.2)	92,02%	57,61%	70,86%

Tabela 4.8. Wynik modelu bazowego na poprawionych korpusach CSER i CEN.

4.6.2. Segmentacja tekstu

Podczas analizy wyników okazało się, że część nazw własnych nie była poprawnie rozpoznana ze względu na sposób tokenizacji tekstu. Niektóre nazwy własne były „sklejone" z sąsiadującymi elementami w zdaniu, przez co ich rozpoznanie było utrudnione, a wręcz niemożliwe. Na przykład fragment *k/Poznania*, będący skróconym zapisem *koło Poznania*, został potraktowany jako jeden token. Inny przykład to łączenie podwójnego nazwiska dywizem, np. *Kowalska-Nowak* jest także traktowane jako jeden token, pomimo że na poziomie nazw własnych jest oznaczone jako dwa nazwiska. W przypadku stosowania cech słownikowych takie przypadki nie były rozpoznawane.

Dotychczasowy potok przetwarzania (segmentacja tekstu, analiza morfologiczna i tagowanie) realizowany przy pomocy tagera TaKIPI (Piasecki, 2007) został zastąpiony przez zestaw narzędzi: tokenizator *maca* (Radziszewski i Śniatowski, 2011a) wykorzystujący analizator morfologiczny Morfeusz SGJP[13] i tager WMBT (Radziszewski i Śniatowski, 2011b). Tokenizator *maca* dzieli tekst po wszystkich znakach interpunkcyjnych. Dzięki temu, we wspomnianych przypadkach, nazwy własne zostaną wydzielone jako osobne tokeny. Wpływ użycia tagera *maca* na rozpoznawanie nazw własnych został przetestowany przy użyciu walidacji krzyżowej na korpusie CSER. Wyniki dla obu konfiguracji zostały przedstawione w tabeli 4.9. Zgodnie z oczekiwaniami zmiana tokenizacji na drobnoziarnistą spowodowała nieznaczny wzrost kompletności, co oznacza, że zostały rozpoznane nazwy, które do tej pory nie mogłyby być rozpoznane. Średnia

13. Narzędzie dostępne na stronie: `http://sgjp.pl/morfeusz/morfeusz.html`.

harmoniczna dla wariantu z tagerem *maca* wzrosła z 94,53% do 94,66%. W kolejnym eksperymentach zostały użyte korpusy otagowane *macą*.

Tokenizator	**Precyzja**	**Kompletność**	**Miara F**
TaKIPI (B1.2)	96,79%	92,38%	94,53%
maca (B1.3)	96,74%	92,68%	94,66%

Tabela 4.9. Ocena dwóch narzędzi do segmentacji tekstu i analizy morfologicznej.

4.6.3. Uzupełnienie słowników

Cechy słownikowe są bardzo pomocnym źródłem informacji. Pomimo zastosowania automatycznej procedury uzupełnienia form odmienionych opisanej w sekcji 4.2.2 wykorzystane słowniki mają duże braki pod względem form odmienionych. Aby częściowo uzupełnić te braki, została zastosowana automatyczna procedura pozyskania form odmienionych na podstawie słownika odmiany *sjp-odm*[14]. Jest to słownik form odmienionych dostępny na licencji GPL, LPGL lub CC SA. Słownik zawiera ok. 190 000 form bazowych słów pospolitych i nazw własnych wraz z ich odmianami. Ze słownika zostały wybrane wszystkie grupy słów zaczynających się dużą literą. Ponieważ słowa nie były przypisane do żadnych kategorii semantycznych, konieczne było opracowanie procedury rzutowania na kategorie nazw własnych. Dla każdej grupy pierwsze słowo było traktowane jako forma bazowa. Jeżeli forma bazowa grupy znajdowała się w dotychczasowym słowniku nazw własnych i była przypisana do dokładnie jednej kategorii, to pozostałe słowa z tej grupy zostały dodane jako formy odmienione tej nazwy. W przypadku niejednoznaczności formy odmienione nie były dodawane do żadnej kategorii, aby uniknąć niepoprawnych form odmiany. W ten sposób zostały dodane nowe formy imion, nazw państw, miast i ulic. Nazwiska zostały pominięte, ponieważ ręczna weryfikacja pokazała, że większość potencjalnych form odmienionych nazwisk było formami innych nazw (np. formy odmienione nazw ulic zostały przypisane do nazwisk) — odmiana nazwisk różni się od odmiany innych nazw.

Słownik	*imiona*	*nazwiska*	*państwa*	*miasta*	*ulice*	**wszystkie**
podstawowy	22 435	371 379	1 867	77 873	40 859	**514 413**
rozszerzony	46 351	371 379	4 086	152 543	62 106	**636 465**

Tabela 4.10. Statystyki słownika nazw własnych po rozszerzeniu o nowe formy.

Po zastosowaniu tej procedury słownik nazw własnych został rozszerzony o ok. 125 tys. nowych form odmienionych. Szczegółowe statystyki słowników przed rozszerzeniem i po nim zostały przedstawione w tabeli 4.10. Procedura rozszerzenia słownika została

14. Strona domowa: `http://www.sjp.pl/slownik/odmiany/`

oceniona pod kątem wpływu na jakość rozpoznawania nazw własnych przez model CRF. Walidacja krzyżowa na korpusie CSER wykazała wzrost kompletności z 92,68% do 93,84% przy zachowaniu tego samego poziomu precyzji, co w efekcie pozwoliło na wzrost średniej harmonicznej z 94,66% do 95,27% (pełne wyniki zostały przedstawione w tabeli 4.11). W kolejnych eksperymentach będą używane rozszerzone słowniki.

Słownik	**Precyzja**	**Kompletność**	**Miara F**
podstawowy (B1.3)	96.74%	92.68%	94.66%
rozszerzony (B1.4)	96.75%	93.84%	95.27%

Tabela 4.11. Ocena rozszerzonego słownika nazw własnych w kontekście jakości modelu CRF do rozpoznawania nazw własnych.

4.7. Usprawnienie modelu bazowego CRF

W kolejnym kroku została podjęta próba doboru parametrów modelu CRF, w celu zwiększenia precyzji i kompletności rozpoznawania jednostek identyfikacyjnych. Usprawnienie uwzględniało modyfikację istniejących cech, konstrukcję nowych cech, selekcję cech, redukcję cech oraz zastosowanie przetwarzania końcowego (Marcińczuk i Janicki, 2012). W kolejnych sekcjach zostały opisane poszczególne usprawnienia.

4.7.1. Modyfikacja cech

Cechy słownikowe

Częstym problemem przy rozpoznawaniu długich nazw jest błędny ich podział na kilka krótszych. Na przykład *Stany Zjednoczone Ameryki Północnej* były błędnie rozpoznawane jako dwie oddzielne nazwy, tj. *Stany Zjednoczone* i *Ameryki Północnej*. Wynikało to ze sposobu kodowania cech słownikowych.

W bazowym modelu cechy słownikowe uwzględniały wszystkie nazwy własne, w tym zagnieżdżenia tej samej kategorii. W wyniku tego wszystkie tokeny rozpoczynające sekwencję znalezioną w słowniku były znakowane symbolem `B`[15]). To powodowało, że sekwencja *Stany Zjednoczone Ameryki Północnej* była oznaczona jako `B I B I`, ponieważ w słowniku istniały 3 formy: *Stany Zjednoczone Ameryki Północnej*, *Stany Zjednoczone* i *Ameryki Północnej*.

Przy takim kodowaniu sekwencja `B I B I` była niejednoznaczna w interpretacji, ponieważ mogła oznaczać dwie następujące po sobie nazwy lub jedną długą składającą

15. Symbol B oznacza, że dane słowo rozpoczyna sekwencję słów znajdującą się w słowniku nazw własnych. Symbol I oznacza słowo będące kolejnym słowem w sekwencji słów będącej w słowniku nazw własnych.

się z 4 tokenów. Aby uniknąć tej niejednoznaczności, sposób kodowania cech słownikowych został zmieniony w taki sposób, aby były znakowane wyłącznie najdłuższe nazwy. Po tej zmianie wspomniana nazwa własna została oznaczona jako `B I I I`.

Po zastosowaniu tej modyfikacji problem podziału długich nazw własnych na krótsze został rozwiązany. Pozwoliło to na poprawienie wyniku z 94,53% średniej harmonicznej do 95,44%.

Konfiguracja	**Precyzja**	**Kompletność**	**Miara F**
Wynik bazowy (B1.4)	96,75%	93,84%	95,27%
Cechy słownikowe	96,92%	94,01%	95,44%
Cechy wordnetowe	96,70%	93,79%	95,23%
Obie modyfikacje (B1.5)	96,83%	94,12%	95,46%

Tabela 4.12. Ocena modyfikacji cech słownikowych i wordnetowych.

Cechy oparte na wordnecie

Dla słów wieloznacznych wybór pierwszego alfabetycznie słowa spośród wszystkich możliwych interpretacji powodował liczne błędy. Na przykład słowo „członek", który może być interpretowany jako osoba należącą do organizacji (*członek rady nadzorczej*) lub część ciała, był uogólniany tylko do części ciała, co wprowadzało dodatkowy szum w danych. Aby wyeliminować ten problem, ale jednocześnie móc korzystać z cechy uogólniania, procedura ta została zastosowana wyłącznie do słów jednoznacznych. Oznacza to, że słowo X zostało zastąpione przez hiperonim lub synonim Y tylko wtedy, jeżeli wszystkie sensy słowa X mogły być uogólnione do tego samego słowa Y.

Zastosowanie tej modyfikacji pogorszyło nieznacznie średnią harmoniczną dla walidacji krzyżowej na korpusie CSER z 95,27% na 95,23%. Szczegółowe wyniki porównania obu wariantów znajdują się w tabeli 4.12.

Wykorzystanie obu cech

Modyfikacje cech słownikowych i wordnetowych testowane niezależnie pozwoliły na nieznaczną poprawę wyników. Zastosowanie obu modyfikacji jednocześnie dało bardzo nieznaczną poprawę wyniku w stosunku do samej modyfikacji cech słownikowych. Ostateczny wynik modyfikacji cech to poprawa precyzji z 96,75% do 96,83% i kompletności z 93,84% do 94,12%.

4.7.2. Konstrukcja cech

Kolejnym elementem dającym możliwość poprawy wyników jest konstrukcja nowych cech w oparciu o istniejące. Nowe cechy zostały zainspirowane gramatykami do rozpoznawania nazw własnych opracowanymi przez Piskorskiego (2004b). Ze zbioru opracowanych reguł zostały wybrane te, które zostały zaobserwowane w korpusie uczącym.

Każda reguła została zapisana jako złożenie istniejących cech. Zostały skonstruowane i ocenione cztery nowe cechy:

- **Ulica #1**: `road_prefix[-1]/road_nam[0]` — nazwa ulicy ze słownika występująca po słowie występującym przed nazwą ulicy (połączenie cechy słownikowej nazw ulic i cechy słownikowej słów kluczowych ulic),
- **Miasto #1**: `city_nam[0]/pattern[1]/pattern[2]/country_nam[3]` — nazwa miasta i państwa oddzielone pewnymi znakami interpunkcyjnymi (połączenie cechy słownikowej nazwy miasta, państwa i klasy ortograficznej),
- **Osoba #1**: `base[-1]/first_nam[0]/last_nam[1]/base[2]/last_nam[3]` — nazwa osoby składająca się z imienia i podwójnego nazwiska występująca po pewnym słowie (połączenie formy bazowej słowa z cechami słownikowymi imion i nazwisk),
- **Osoba #2**: `base[-2]/first_nam[-1]/pattern[0]/base[1]` — nazwa osoby składająca się z imienia znajdującego się w słowniku i potencjalnego nazwiska, tj. słowa pisanego dużą literą (połączenie form bazowych, cechy słownikowej imion i klasy ortograficznej).

Wprowadzenie nowych cech nie miało znaczącego wpływu na wyniki rozpoznawania nazw (zob. tabelę 4.13). Różnica średniej harmonicznej wahała się do 1 punktu procentowego na korzyść lub niekorzyść danej cechy. Niewielki wpływ tych cech może oznaczać, że większość nazw występujących w tych wzorcach jest już rozpoznawana przez obecny model lub też liczba wystąpień tych wzorców jest niewielka w stosunku do liczby wszystkich nazw.

Cecha	**Precyzja**	**Kompletność**	**Miara F**
Wynik bazowy (B1.5)	96,83%	94,12%	95,46%
Ulicza #1	96,83%	94,10%	95,45%
Miasto #1	96,73%	94,05%	95,38%
Osoba #1	96,86%	94,15%	95,48%
Osoba #2	96,76%	94,12%	95,42%

Tabela 4.13. Ocena wpływu nowych cech na korpusie CSER.

4.7.3. Selekcja cech

Celem selekcji cech było wybranie tych cech, które niosą najwięcej informacji w przypadku zadania rozpoznawania nazw własnych. W tym podejściu została przyjęta niezależność cech, co oznacza, że dla każdej cechy z osobna była mierzona jej zdolność dyskryminacyjna. Do wyznaczenia możliwości dyskryminacyjnej została użyta miara przyrostu informacji (ang. *Information Gain*; IG). Pierwotny model CRF wykorzystuje 172 cechy. Miara IG dla poszczególnych cech została obliczona na fragmencie korpusu KPWr przy użyciu narzędzia Weka[16] (Hall *et al.*, 2009), które posiada zestaw gotowych

16. Narzędzie dostępne na stronie: `http://www.cs.waikato.ac.nz/ml/weka/`.

10 cech z najwyższym IG	
IG	**Cech**
0,45929656	orth+0
0,45623956	base+0
0,44374859	prefix-4+0
0,42121654	suffix-4+0
0,40814769	prefix-3+0
0,35655636	suffix-3+0
0,30563528	prefix-2+0
0,28395843	orth-1
0,26671994	orth-2
0,26541514	base-1

Cechy z najmniejszym IG		
	IG	**Cecha**
(3)	0,00094167	road_prefix+2
	0,00072422	person_prefix+2
	0,00060709	country_prefix-1
	0,00059445	person_noun+2
↓	0,00035950	road_prefix+1
(2)	0,00003794	country_prefix+2
	0,00001289	person_suffix+1
	0,00001239	person_suffix+0
↓	0,00001046	person_suffix-1
(1)	0,00000948	person_suffix+2
	0,00000925	person_suffix-2

Tabela 4.14. Lista cech z największym i najmniejszym przyrostem informacji (IG).

modułów implementujących popularne algorytmy selekcji cech. Najwyższa wartość IG została osiągnięta dla cechy `orth[0]` w wysokości 0,46, co oznacza, że ta cecha niesie ze sobą najwięcej informacji. Z kolei najmniej „informatywną" cechą okazała się cecha `person_suffix[-2]`, która osiągnęła wartość 0,000009. Ta cecha określa słowa kluczowe, które mogą występować po nazwie osoby (np. nazwa zgromadzenia zakonnego, *Remigiusz Recław SJ*). W korpusie testowym słowa występujące po nazwach osobowych pojawiały się zaledwie kilka razy, z tego powodu cecha ta osiągnęła niską wartość IG.

Cechy o najniższych wartościach IG są potencjalnymi cechami do usunięcia. Zostało wybranych jedenaście cech o najniższym IG i utworzono z nich trzy zbiory cech: (1) IG $< 10^{-5}$, (2) $IG < 10^{-4}$ and (3) $IG < 10^{-3}$. Następnie zostały przetestowane trzy warianty modelu CRF: z usunięciem pierwszego, drugiego i trzeciego zbioru cech. Każdy wariant został przetestowany na korpusie, na którym liczono wartości IG, oraz na korpusie CSER.

Dla obu korpusów nie została odnotowana znacząca różnica w osiągniętych wynikach. Oznacza to, że na badanym korpusie wskazane cechy nie mają istotnego wpływu na rozpoznawanie nazw własnych. Na przykład w grupie 1 i 2 znalazły się wszystkie cechy `person_suffix`, co potwierdza wcześniejszą obserwację, że słowa kluczowe występujące po nazwie własnej nie występują w tym korpusie.

4.7.4. Redukcja cech

Przy selekcji cech przyjęto założenie, że każda cecha może być dyskryminacyjna w izolacji od pozostałych cech. W praktyce dopiero połączenie kilku cech może być cechą dyskryminacyjną. Redukcja cech odbywała się iteracyjnie. W każdej iteracji dla zadanego zbioru cech były testowane wszystkie podzbiory cech o liczności o jeden mniejsza niż zbiór początkowy (cechy kontekstowe były usuwane parami, tj. cecha[-2]

Konfiguracja	Cechy	IK			CSER		
		P [%]	**R** [%]	**F** [%]	**P** [%]	**R** [%]	**F** [%]
Wynik bazowy	172	73,51	53,10	61,66	96,83	94,12	95,46
Bez grupy cech (1)	170	73,58	52,56	61,32	96,83	94,08	95,43
Bez grupy cech (2)	166	73,76	52,29	61,20	96,88	94,17	95,51
Bez grupy cech (3)	161	73,96	52,83	61,64	96,83	94,08	95,43

Tabela 4.15. Ocena wpływu selekcji cech na korpusie CSER.

i cecha[+2]). Następnie wybierana była konfiguracja, dla której osiągnięto najlepszą poprawę wyniku i powtarzano iterację z nowym zbiorem cech — zmniejszonym o odrzuconą cechę. Iteracje były wykonywane aż do momentu pogorszenia wyników dla wszystkich wariantów lub do osiągnięcia jednoelementowego zbioru cech.

Redukcja cech została wykonana na fragmencie korpus KPWr (IKW) przy użyciu czterokrotnej walidacji krzyżowej. Wyniki redukcji dla poszczególnych iteracji na korpusie IKW zostały przedstawione w tabeli 4.16 (wszystkie iteracje, dla których osiągnięto poprawę wyników). Po wykonaniu dziesięciu iteracji średnia harmoniczna poprawiła się o 5 punktów procentowych.

It.	**Cecha**	**Akcja**	**Precyzja**	**Kompletność**	**Miara F**
0	—	—	73,51%	53,10%	61,66%
1	number	usunięcie	75,66%	54,45%	63,32%
2	case	kontekst [0]	77,69%	54,45%	64,03%
3	ctag	usunięcie	77,78%	54,72%	64,24%
4	class	kontekst [-1:0:1]	78,46%	54,99%	64,66%
5	suffix-2	usunięcie	78,33%	55,53%	64,98%
6	pattern	kontekst [0]	78,79%	56,06%	65,51%
7	city_nam	kontekst [0]	78,95%	56,60%	65,93%
8	suffix-3	kontekst [0]	78,60%	57,41%	66,36%
9	base	kontekst [0]	79,18%	57,41%	66,56%
10	hyp1	usunięcie	79,48%	57,41%	66,67%

Tabela 4.16. Redukcja cech na korpusie IKW.

Wszystkie dziesięć iteracji zostało powtórzonych na korpusie CSER przy użyciu dziesięciokrotnej walidacji krzyżowej. W tabeli 4.17 zostały przedstawione wyniki dla trzech wariantów: o najlepszej precyzji, najlepszej średniej harmonicznej i dziesiątej iteracji. Pomimo że najlepsze wyniki zostały osiągnięte już dla szóstej iteracji, to po usunięciu cech dla wszystkich dziesięciu iteracji osiągnięto wynik lepszy od wyniku bazowego.

It.	Cecha	Akcja	Precyzja	Kompletność	Miara F
Wynik bazowy (B1.5)			96,83%	94,12%	95,46%
3	ctag	usunięcie	**96,84**%	94,29%	95,55%
6	pattern	kontekst [0]	96,72%	94,46%	**95,58**%
10	hyp1	usunięcie	96,67%	94,36%	95,50%

Tabela 4.17. Ocena wpływu redukcji cech na korpusie CSER.

4.7.5. Przetwarzanie końcowe

Jednoznaczny tager słownikowy

Jednoznaczny tager słownikowy bazuje na obserwacji, że istnieje pewien zbiór nazw własnych, które są jednoznaczne w obrębie kategorii nazw własnych i jednocześnie nie są słowami pospolitymi. Jeśli dysponuje się dostatecznie dużym i zróżnicowanym słownikiem, można rozpoznać nazwy własne z wysoką precyzją. Jednoznaczny tager słownikowy wykorzystuje słownik nazw własnych i słów pospolitych. Na bazie tych słowników znakuje wszystkie nazwy własne, które są jednoznaczne w obrębie kategorii nazw własnych (nazwa przypisana jest tylko do jednej kategorii) i nie znajdują się w słowniku słów pospolitych. Skuteczność rozpoznawania poszczególnych kategorii nazw została przedstawiona w tabeli 4.18. Spośród pięciu kategorii nazw najbardziej jednoznaczne okazały się być nazwy miast i państw. Dla pozostałych kategorii tager uzyskał bardzo niską precyzję, dlatego lista kategorii rozpoznawanych przez tager została ograniczona do nazw miast i państw.

	Precyzja	Kompletność	Miara F
imiona	56,63%	32,13%	41,00%
nazwiska	63,52%	14,68%	23,85%
państwa	98,97%	60,97%	75,46%
miasta	93,22%	61,91%	74,41%
ulice	42,46%	54,04%	47,56%
wszystkie	77,15%	48,58%	59,62%

Tabela 4.18. Ocena jednoznacznego tagera słownikowego.

Wyniki zastosowania jednoznacznego tagera słownikowego w przetwarzaniu końcowym zostały przedstawione w tabeli 4.19. Zastosowanie tagera pozwoliło na nieznaczną poprawę wyników pod względem kompletności. Pomimo jednoprocentowego spadku precyzji ostateczny wynik w postaci średniej harmonicznej był nieznacznie lepszy od wyniku bazowego.

	Precyzja	Kompletność	Miara F
wynik bazowy (B1.5)	96,83%	94,12%	95,46%
CRF + tager słownikowy	95,26%	95,74%	95,50%
CRF + tager regułowy	96,66%	94,50%	95,57%

Tabela 4.19. Ocena łączenia metod przetwarzania końcowego z modelem CRF.

Tager regułowy

Zastosowanie przetwarzania końcowego w postaci tagera regułowego opiera się o wykorzystanie gramatyk do rozpoznawania nazw własnych opracowanych przez Piskorskiego (2004b). Gramatyki rozpoznają nazwy występujące w słowniku w określonym kontekście. Wyniki dla samego tagera regułowego zostały przedstawione w tabeli 4.20. Imiona, nazwiska i nazwy miast były rozpoznawane z bardzo wysoką precyzją (ponad 95%), ale też niską kompletnością. Tylko ulice były rozpoznawane ze znacznie niższą precyzją (poniżej 80%). Analiza wyników wykazała, że większość błędnych anotacji była spowodowana częściowym dopasowaniem nazwy. W zdecydowanej większości nazwy te były już poprawnie rozpoznane przez model CRF. Na bazie tej obserwacji wyniki modelu CRF i tagera regułowego zostały połączone oraz zostały odrzucone wszystkie anotacje zagnieżdżone. Wynik zastosowania tagera regułowego łącznie z modelem CRF został przedstawiony w tabeli 4.19. Połączenie obu metod pozwoliło na nieznaczną poprawę średniej harmonicznej z 95,46% do 95,57%. Niewielka poprawa może oznaczać, że większość nazw rozpoznawanych przez reguły jest już rozpoznana przez model statystyczny, ale jednocześnie widać, że reguły mogą być pomocne.

	Precyzja	Kompletność	Miara F
imiona	96,15%	3,62%	6,97%
nazwiska	95,24%	2,91%	5,64%
państwa	0,00%	0,00%	0,00%
miasta	99,56%	11,26%	20,23%
ulice	77,72%	73,99%	75,81%
wszystkie	86,62%	13,26%	22,99%

Tabela 4.20. Ocena tagera regułowego na korpusie CSER.

4.7.6. Ostateczna konfiguracja

Po przetestowaniu wszystkich zaproponowanych modyfikacji w izolacji zostały zaproponowane trzy konfiguracje. Pierwsza uwzględniała bazowy model CRF połączony tylko z przetwarzaniem końcowym (CRF #0). Drugi model uwzględniał wszystkie modyfikacje (wymienione poniżej), które poprawiły średnią harmoniczną (CRF #1). W

ostatnim, trzecim wariancie znalazły się modyfikacje (wymienione poniżej), które pozwoliły na uzyskanie najwyższej precyzji (CRF #2). Dla każdego z trzech wariantów przyjęto także: zmianę tokenizacji (sekcja 4.6.2), rozszerzone słowniki (sekcja 4.6.3) oraz modyfikację cech słownikowych i wordnetowych (sekcja 4.7.1). Ocena wszystkich wariantów konfiguracji została wykonana na korpusie CSER przy użyciu dziesięciokrotnej walidacji krzyżowej.

CRF #0 — uwzględnia tylko jednoznaczny tager słownikowy i tager regułowy,

CRF #1 — uwzględnia: (a) nową cechę Osoba#1, (b) zestaw cech otrzymany po szóstej iteracji redukcji oraz (c) przetwarzanie końcowe.

CRF #2 — uwzględnia: (a) nową cechę Osoba#1, (b) zestaw cech otrzymany po trzeciej iteracji redukcji, (c) przetwarzanie końcowe oraz (d) odrzucenie cech wordnetowych.

Konfiguracja	**Precyzja**	**Kompletność**	**Miara F**
Wynik bazowy (B1.2)	96,79%	92,38%	94,53%
Wynik bazowy (B1.5)	96,83%	94,12%	95,46%
CRF #0	95,08%	96,09%	95,58%
CRF #1	95,02%	96,28%	**95,65%**
CRF #2	95,08%	96,07%	95,57%

Tabela 4.21. Dziesięciokrotna walidacja krzyżowa na korpusie CSER — porównanie różnych konfiguracji.

Wyniki wszystkich trzech wariantów zostały przedstawione w tabeli 4.21. Dla wszystkich trzech wariantów udało się nieznacznie poprawić średnią harmoniczną. Najlepszy wynik pod względem średniej harmonicznej został osiągnięty dla wariantu CRF #1. W tabeli 4.22 zostały przedstawione szczegółowe wyniki walidacji na korpusie CSER.

	TP	**FP**	**FN**	**Precyzja**	**Kompletność**	**Miara F**
imiona	656	13	27	98,06%	96,05%	97,04%
nazwiska	637	15	51	97,70%	92,59%	95,07%
państwa	440	39	33	91,86%	93,02%	92,44%
miasta	1953	129	28	93,80%	98,59%	96,14%
ulice	378	17	18	95,70%	95,45%	95,58%
wszystkie	4064	213	157	95,02%	96,28%	95,65%

Tabela 4.22. Dziesięciokrotna walidacja krzyżowa na korpusie CSER — szczegóły dla konfiguracji CRF #1.

Do zmierzenia istotności statystycznej poprawy otrzymanych wyników został wykorzystany test t-Studenta. Test t-Studenta jest jednym z najczęściej stosowanych testów w dziedzine maszynowego uczenia dla eksperymentów składających się z kilku

powtórzeń (prób) o różnych układach danych uczących i testowych (Dietterich, 1998). W teście hipotezą zerową jest równość wyników otrzymanych przez dwa różne modele. Wartość t jest obliczana ze wzoru 4.8, gdzie $\overline{p}$ to średnia wartość zmiennej dla wszystkich prób obliczana ze wzoru 4.9, n to liczba prób, $p^{(i)}$ to wartość zmiennej w i-tej próbie.

$$t = \frac{\overline{p} * \sqrt{n}}{\sqrt{\frac{\sum_{i=1}^{n}(p^{(i)} - \overline{p})^2}{n-1}}} \tag{4.8}$$

$$\overline{p} = \frac{1}{n}\sum_{i=1}^{n}(p^{(i)} \tag{4.9}$$

Dla dziewięciu stopni swobody ($n-1$, gdzie $n = 10$ to liczba powtórzeń testu) i poziomu istotności $\alpha = 0,05$ hipoteza zerowa zostaje odrzucona jeżeli $|t| > t_{9;0,95} = 2,262$. Istotność statystyczna została sprawdzona dla średniej harmonicznej precyzji i kompletności (miara F). Każda z przedstawionych modyfikacji oceniona niezależnie od pozostałych osiągnęła $t < 2,262$, co oznaczało, że nie ma podstaw do odrzucenia hipotezy zerowej, zgodnie z którą wyniki są sobie równe. Dopiero zestawienie wszystkich modyfikacji z wynikiem bazowym B1.2 (bazowy model CRF wyuczony i oceniony na poprawionym korpusie; zob. sekcję 4.6.1) pozwala na osiągnięcie $t < 2,262$, co oznacza, że różnica (w tym przypadku poprawa) wyników jest statystycznie istotna.

4.7.7. Walidacja międzydziedzinowa

W celu zweryfikowania uniwersalności zastosowanych modyfikacji została dokonana walidacja międzydziedzinowa — dla wybranych konfiguracji model CRF został wyuczony na korpusie CSER i przetestowany na korpusie CEN. Otrzymane wyniki zostały porównane z wynikiem bazowym (zob. tabela 4.23). Dla obu konfiguracji została osiągnięta znacząca poprawa — prawie 9% średniej harmonicznej. Zastosowane modyfikacje pozwoliły na poprawę kompletności o prawie 13 punktów procentowych przy nieznacznym spadku precyzji o zaledwie 0,5%. Szczegółowe wyniki dla najlepszej konfiguracji CRF #2 zostały przedstawione w tabeli 4.24.

Konfiguracja	**Precyzja**	**Kompletność**	**Miara F**
Wynik bazowy (B2.2)	92,02%	57,61%	70,86%
CRF #0	91,32%	69,86%	79,16%
CRF #1	91,55%	70,32%	79,54%
CRF #2	91,44%	70,53%	**79,63%**

Tabela 4.23. Międzydziedzinowa walidacja na korpusie CEN — porównanie różnych konfiguracji.

	TP	**FP**	**FN**	**Precyzja**	**Kompletność**	**Miara F**
imiona	827	27	285	96,84%	74,37%	84,13%
nazwiska	831	31	806	96,40%	50,76%	66,51%
państwa	1524	92	231	94,31%	86,84%	90,42%
miasta	492	186	173	72,57%	73,98%	73,27%
ulice	11	9	45	55,00%	19,64%	28,95%
wszystkie	3685	345	1540	91,44%	70,53%	79,63%

Tabela 4.24. Międzydziedzinowa walidacja na korpusie CEN — szczegóły dla konfiguracji CRF #2.

4.8. Ocena modelu na pełnym schemacie jednostek

Model do rozpoznawania nazw własnych został użyty do wsparcia procesu znakowania 56 kategorii nazw własnych w korpusie KPWr. Najlepsza konfiguracja modelu CRF osiągnięta w rozpoznawaniu pięciu podstawowych kategorii nazw własnych została użyta do wyuczenia rozszerzonego modelu nazw własnych. Model został wyuczony na około 400 ręcznie znakowanych dokumentach. Następnie model został użyty do rozpoznania nazw własnych w kolejnych 400 dokumentach, które nie były wcześniej znakowane. Wyniki automatycznego rozpoznawania nazw zostały następnie przedstawione lingwiście do oceny. Ocena została wykonana za pośrednictwem dedykowanej perspektywy w systemie Inforex (zob. ekran 6.3). Dla każdej propozycji użytkownik mógł wykonać jedną z czterech akcji: oznaczyć propozycję jako poprawną, zmienić kategorię nazwy własnej (zakres anotacji został poprawnie rozpoznany, tylko kategoria nazwy została błędnie oznaczona), odrzucić anotację (zakres anotacji i kategoria nazwy były niepoprawnie rozpoznane) lub zostawić do oceny na później.

Wyniki ręcznej oceny zostały zachowane w celu obliczenia skuteczności rozpoznawania nazw własnych. Stosując procedurę automatycznego rozpoznawania nazw i ręcznej weryfikacji zostało dodanych ponad 1700 nowych anotacji. 71% z zaproponowanych anotacji było poprawnych pod względem granicy i kategorii nazwy własnej (precyzja) i stanowiło 54% wszystkich dodanych anotacji (kompletność). Z pozostałych 29% częściowo błędnie rozpoznanych anotacji 46% było poprawnie rozpoznanych pod względem granic anotacji i wymagało jedynie ręcznej korekty kategorii anotacji. 46% anotacji zostało dodanych ręcznie.

4.9. Podsumowanie

W rozdziale został przedstawiony problem rozpoznawania jednostek identyfikacyjnych ograniczony do nazw własnych w tekstach w języku polskim. W fazie eksperymentów zbiór rozpoznawanych nazw własnych został zawężony do pięciu kategorii nazw, tj. imion, nazwisk, nazw państw, miast i ulic. Na początku zostały wyznaczone wyniki dla dwóch prostych metod referencyjnych: metody słownikowej i regułowej. Wstępne

eksperymenty wykazały, że żadna z tych metod stosowana osobno nie pozwoliła na osiągnięcie zadowalających wyników.

Następnie został skonstruowany model łączący nadzorowane uczenie z wykorzystaniem słowników i reguł. Do opracowania modelu statystycznego została wykorzystana metoda CRF i bogaty zbiór cech zawierający 15 typów cech (w tym cechy ortograficzne, morfologiczne, wordnetowe i słownikowe). Słowniki zostały wykorzystane dwojako: jako cechy dla modelu CRF oraz jako dane do jednoznacznego tagera słownikowego. Następnie bazowy model CRF został poddany serii usprawnień, które uwzględniały: zmianę segmentacji tekstu, automatyczne uzupełnienie słowników, redefinicję kodowania cech słownikowych i wordnetowych, konstrukcję nowych cech, selekcję i redukcję cech oraz wprowadzenie przetwarzania końcowego.

Ostateczny model rozpoznawania pięciu kategorii nazw własnych testowany na tej samej dziedzinie (walidacja krzyżowa na korpusie raportów giełdowych CSER) osiągnął łączny wynik w wysokości 95,02% precyzji, 96,28% kompletności i 95,65% średniej harmonicznej. Dla oceny międzydziedzinowej (uczenie na korpusie CSER i testowanie na korpusie wiadomości gospodarczych CEN) wyniki wyniosły 91,55% precyzji, 70,32% kompletności i 79,54% średniej harmonicznej.

Rozszerzony model nazw własnych do 56 kategorii nazw własnych, użyty do wsparcia procesu znakowania korpusu KPWr, osiągnął wynik na poziomie 71% precyzji i 54% kompletności. Niska kompletność wiązała się przede wszystkim z brakiem słowników o dużym pokryciu dla wszystkich kategorii nazw własnych. Kolejną przyczyną była trudność z rozstrzyganiem między kategoriami nazw własnych reprezentujących obiekty o zbliżonej semantyce (np. partia polityczna, firma i instytucja posiadają cechy organizacji). Ostatnią przyczyną, częściowo związaną z brakiem kompletnych słowników, jest brak analizy dyskursu, który jest wymagany do rozstrzygnięcia wielu przypadków. Na przykład mając zdanie *Michał mieszka w Ankh.* można się domyśleć, że Ankh to nazwa jakiegoś fikcyjnego miejsca, np. miasta lub państwa, ale na podstawie tego zdania nie można wywnioskować, czym dokładnie jest to miejsce. Natomiast natrafiając na kolejne zdanie *Stolicą Ankh jest Morpork.* można wywnioskować, że Ankh to nazwa państwa, a Morpork to nazwa miasta.

Zatem w celu zwiększenia skuteczności rozpoznawania złożonej hierarchii nazw własnych konieczne jest wprowadzenie częściowej analizy dyskursu. Analiza dyskursu łączy się z koniecznością rozpoznawania anafory i koreferencji między zdaniami, a to z kolei wymaga wstępnego rozpoznawania jednostek identyfikacyjnych, bez ustalania szczegółowej ich kategorii. W celu ustalenia cech nazw własnych w obrębie zdania konieczne jest wykorzystanie analizy zależnościowej między słowami, która dostarczając informacji m.in. o zależnościach predykatowo-argumentowych może wskazać, z jakimi predykatami wiąże się dana nazwa.

Jak już na wstępie zostało wspomniane, bezpośrednie porównanie wyników osiągniętych w innych pracach nad rozpoznawaniem nazw własnych nie jest możliwe z powodu braku dostępu do danych, na których były przeprowadzone eksperymenty. Biorąc pod uwagę inne prace przedstawione w punkcie 4.2, tylko w jednej z nich rozpatrywano

bardzo zbliżony zakres kategorii nazw własnych. W rozpoznawaniu nazw osób, państw, miast i ulic Abramowicz *et al.* (2006) uzyskał bardzo zbliżone wyniki do wyników otrzymanych dla walidacji krzyżowej na korpusie CSER i międzydziedzinowej na korpusie CEN. Jedynie dla rozpoznawania nazw ulic w ocenie międzydziedzinowej otrzymane wyniki są znacząco poniżej wyników przedstawionych w Abramowicz *et al.* (2006), co może wynikać z relatywnie niewielkiej liczby anotacji nazw ulic w korpusie CEN w stosunku do pozostałych kategorii.

Dzięki uprzejmości Jakuba Waszczuka, który dokonał oceny działania narzędzia NERF na korpusach CSER i CEN, możliwe jest bezpośrednie porównanie obu rozwiązań. W tabeli 4.25 zostały przedstawione wyniki rozpoznawania pięciu kategorii nazw własnych na korpusie CEN dla modeli wyuczonych na korpusie CSER. Liner2 wykorzystuje najlepszy model opracowany w ramach rozprawy, a NERF to narzędzie opracowane równolegle w ramach projektu NKJP. Należy podkreślić, że NERF nie wykorzystuje żadnych dodatkowych zasobów w postaci słowników, wordnetu i reguł, a bazuje wyłącznie na cechach ortograficznych. Porównanie wyników pokazuje, że zewnętrzne zasoby są bardzo przydatne w rozpoznawaniu nazw własnych oraz że bezpośredni kontekst wystąpienia nazwy własnej nie zawsze jest wystarczający do prawidłowego rozpoznania i kategoryzacji nazwy.

Konfiguracja	**Precyzja**	**Kompletność**	**Miara F**
Liner2 (model CRF #2)	91,44%	70,53%	**79,63%**
NERF	78,89%	27,69%	**40,99%**

Tabela 4.25. Porównanie rozpoznawania pięciu kategorii nazw własnych z narzędziem NERF na korpusie CEN.

Rys. 4.1. Ekran do weryfikacji automatycznie rozpoznanych nazw w systemie Inforex.

Rozdział 5

Rozpoznawanie relacji semantycznych

Głównym celem pracy było opracowanie nadzorowanej metody do rozpoznawania wystąpień relacji semantycznych między jednostkami identyfikacyjnymi w tekstach ciągłych w języku polskim. Rozpoznawanie relacji polega na identyfikacji par jednostek (fragmentów tekstu odnoszące się do pozatekstowych obiektów) połączonych określonym typem relacji (np. położenie, sąsiedztwo, przynależność). Przykładem takiej relacji jest para jednostek *Brama Brandenburska* i *Berlin*, które połączone są relacją *położenie*, co może być wyrażone takim zdaniem: *Brama Brandenburska to zabytkowa budowla w Berlinie.*. Relacje rozpoznawane są w obrębie pojedynczych zdań. Istnienie relacji określonego typu musi wynikać bezpośrednio z treści zdania, tj. muszą występować przesłanki wskazujące na obecność relacji. Dzięki wykorzystaniu nadzorowanego uczenia możliwe będzie pozyskiwanie reguł dla dowolnie zdefiniowanych typów relacji poprzez dostarczenie zbioru uczącego.

5.1. Przyjęte założenia

W punkcie 3.2 zostały przedstawione wytyczne i założenia dotyczące znakowania relacji semantycznych między jednostkami identyfikacyjnymi w tekście. Poniżej znajdują się dodatkowe założenia, które mają wpływ na realizację zadania i opracowaną metodę:

1. **Jednostki identyfikacyjne, między którymi mają być rozpoznawane relacje, są z góry oznaczone w tekście.** Oznacza to, że etap rozpoznawania relacji musi być poprzedzony etapem rozpoznawania jednostek. Rozpoznawanie jednostek jest ostateczne i na poziomie rozpoznawania relacji nie jest w żaden sposób modyfikowane. Takie podejście z jednej strony upraszcza zadanie rozpoznawania relacji do problemu klasyfikacji par jednostek, a z drugiej uzależnia jakość rozpoznawania relacji od jakości narzędzia do rozpoznawania jednostek. W przypadku

badań nad rozpoznawaniem relacji między nazwami własnymi zbiory uczący i testowy zostały ręcznie oznakowane jednostkami identyfikacyjnymi. W przypadku przetwarzania nowych tekstów etap rozpoznawania relacji jest poprzedzony automatycznym rozpoznawaniem jednostek identyfikacyjnych przy pomocy narzędzia Liner2 (zob. rozdz. 4).

2. **Między parą tych samych jednostek może zachodzić wiele relacji różnych kategorii.** Na przykład *Jan Nowak* i *Polska* mogą być połączone relacjami: *narodowość — Jan Nowak jest obywatelem Polski.*, *pochodzenie — Jan Nowak urodził się w Polsce* i *położenie — Jan Nowak od 2 lat mieszka w Polsce.* To założenie odpowiada scenariuszowi SIL (zob. sekcja 2.2.2).

W kolejnych punktach tego rozdziału zostały przedstawione: wyniki bazowe rozpoznawania relacji z wykorzystaniem heurystyki i ręcznie skonstruowanych reguł (sekcja 5.2), wielowarstwowa reprezentacja zdania wykorzystująca różne poziomy analizy tekstu (sekcja 5.3), nadzorowana metoda konstrukcji reguł do rozpoznawania relacji wykorzystująca indukcyjne programowanie logiczne (sekcja 5.4) oraz metoda rozpoznawania relacji w oparciu o klasyfikatory wykorzystujące reguły jako cechy (sekcja 5.5).

5.2. Wyniki bazowe

Z uwagi na to, że dla języka polskiego nie istnieją ogólnodostępne zbiory testowe oraz eksperymentalne wyniki, do których można by się odnieść, konieczne było wyznaczenie wyniku bazowego. Zostało to zrobione dla dwóch podejść: heurystyki znakującej relacje między wszystkimi nazwami własnymi zgodnie ze słownikiem podtypów (sekcja 5.2.1) oraz ręcznie skonstruowanych reguł (sekcja 5.2.2). W kolejnych podpunktach zostały szczegółowo omówione oba podejścia.

5.2.1. Heurystyka

Pierwszym punktem odniesienia dla automatycznego rozpoznawania relacji jest prosta heurystyka, zgodnie z którą zostają oznaczone wszystkie pary jednostek, których typy występują w słowniku podtypów relacji. Słownik ten zawiera wszystkie dopuszczalne kombinacje kategorii jednostek, między którymi mogą istnieć relacje semantyczne (pełna lista podtypów znajduje się w załączniku B). Pary jednostek muszą występować w obrębie jednego zdania.

Skuteczność rozpoznawania relacji została przetestowana na wszystkich trzech częściach korpusu KPWr, a osiągnięte wyniki zostały przedstawione w tabeli 5.1. Dla wszystkich części korpusu zostały rozpoznane wszystkie relacje (100% kompletności) przy bardzo niskiej precyzji na poziomie od 2% dla relacji *tożsamość* do 22% dla relacji *narodowość*. Niska precyzja świadczy o tym, że współwystąpienie pary jednostek w zdaniu nie oznacza, że istnieje między nimi jedna z oczekiwanych relacji semantycznych.

Relacja	Zbiór uczący			Zbiór pomocniczy			Zbiór testowy		
	P [%]	R [%]	F [%]	P [%]	R [%]	F [%]	P [%]	R [%]	F [%]
autorstwo	14,78	100,0	25,75	12,72	100,0	22,57	20,20	100,0	33,61
kompozycja	26,67	100,0	42,11	16,30	100,0	28,04	11,47	100,0	20,58
narodowość	21,74	100,0	35,71	15,15	100,0	26,32	31,58	100,0	48,00
pochodzenie	8,39	100,0	15,49	6,49	100,0	12,19	8,09	100,0	14,97
położenie	14,30	100,0	25,02	7,31	100,0	13,62	10,21	100,0	18,52
przynależność	13,47	100,0	23,74	13,56	100,0	23,89	11,43	100,0	20,52
sąsiedztwo	2,70	100,0	5,25	2,09	100,0	4,09	1,98	100,0	3,88
tożsamość	1,53	100,0	3,01	2,08	100,0	4,08	1,65	100,0	3,25

Tabela 5.1. Wynik bazowy rozpoznawania relacji między jednostkami identyfikacyjnymi przy pomocy heurystyki.

Najlepsze wyniki zostały osiągnięte dla relacji *narodowość*, co może wynikać z małej liczby podtypów (tylko dwóch).

5.2.2. Ręczna konstrukcja reguł

Formalizm zapisu reguł

Do realizacji regułowego rozpoznawania relacji konieczny był wybór języka do zapisu reguł. Rozważane były trzy istniejące i dostępne narzędzia umożliwiające tworzenie reguł syntaktycznych: WCCL[1] (Radziszewski *et al.*, 2011), Spejd (Przepiórkowski, 2008) i JAPE[2] (Cunningham *et al.*, 2011). Wszystkie narzędzia są dostępne publicznie na wolnej licencji, ale niestety żadne z nich nie posiada wsparcia do tworzenia relacji między jednostkami. WCCL jest zaimplementowany w C++, a Spejd[3] i JAPE w Javie. Do zapisu reguł został wybrany język WCCL z dwóch względów. Pierwszym czynnikiem była wydajność — implementacja w C++ jest o wiele bardziej wydajna niż w Javie. Po drugie, WCCL operuje na formatach danych wykorzystywanych w istniejącym potoku przetwarzania tekstu (segmentacja, analiza morfologiczna, tagowanie). WCCL umożliwia bezpośrednie wykonywanie reguł na plikach w formacie CCL bez konieczności dodatkowych transformacji na inne formaty.

Formalizm ReWCCL. Na potrzeby konstrukcji reguł został wykorzystany interpreter WCCL. Aby uprościć postać reguł, został stworzony dialekt języka WCCL[4] o nazwie

1. Narzędzie dostępne jest na stronie: `http://nlp.pwr.wroc.pl/redmine/projects/joskipi/wiki/`
2. Narzędzie dostępne jest na stronie: `http://gate.ac.uk/`
3. W momencie podejmowania decyzji Spejd był zaimplementowany tylko w Javie. W tym momencie dostępna jest implementacja w C++. Narzędzie dostępne jest na stronie: `http://zil.ipipan.waw.pl/Spejd/`
4. Formalizm języka WCCL do zapisu reguł znakujących fragmenty tekstu został przedstawiony w dodatku D.

ReWCCL, który rozszerza WCCL o nowy operator `link`, którego pełna sygnatura ma postać:

$$link(M_1, T_1, M_2, T_2, R)$$

gdzie:

- M_1 — indeks elementu będącego jednostką źródłową (pierwsza jednostka z pary, między którymi zachodzi relacja),
- T_1 — kategoria semantyczna jednostki źródłowej,
- M_2 — indeks elementu będącego jednostką docelową (druga jednostka z pary, między którymi zachodzi relacja),
- T_2 — kategoria semantyczna jednostki docelowej,
- R — kategoria semantyczna relacji.

Reguły w dialekcie ReWCCL są transformowalne do postaci reguł WCCL i wykonywane przy użyciu standardowego interpretera WCCL. Transformacja reguły ReWCCL na WCCL polega na zamianie operatorów `link` na pary operatorów `mark` i wygenerowaniu symbolicznych nazw jednostek w taki sposób, aby możliwe było późniejsze jednoznaczne połączenie jednostek w pary (procedura łączenia anotacji w pary została przedstawiona poniżej). Na wydruku 5.1 znajduje się przykładowa reguła w dialekcie ReWCCL, a na wydruku 5.2 odpowiadająca jest reguła w języku WCCL.

```
apply(
  match(
    is("person_nam"),                    // grupa 1
    equal( base[0], "urodzić"),          // grupa 2
    equal( base[0], "się"),              // grupa 3
    equal( base[0], "w"),                // grupa 4
    is("city_nam"),                      // grupa 5
  ),
  actions(
    link(1, "person_nam", 5, "city_nam", "origin")
  )
)
```

Wydruk 5.1. Przykładowa reguła rozpoznająca atrybuty zdarzenia. Reguła pochodzi z pracy Brun i Hagège (2009).

Rys. 5.1. Przykładowa reguła w dialekcie ReWCCL tworzącą relację *pochodzenie* między *nazwą osoby* i *nazwą miasta*.

Procedura łączenia jednostek w pary

1. Stwórz listę K zawierającą nazwy anotacji pasujące do wzorca `relation.*`.
2. Dla każdej nazwy k z listy K sprawdź, czy istnieje dokładnie jedna anotacja o nazwie k. Więcej niż jedna anotacja oznacza błąd i przerwanie przetwarzania.
3. Każdą nazwę anotacji podziel po kropce otrzymując wartości:
   `relation.{id}.{type}.{role}.{annotation_type}`

```
apply(
  match(
    is("person_nam"),
    equal( base[0], "urodzić"),
    equal( base[0], "się"),
    equal( base[0], "w"),
    is("city_nam"),
  ),
  actions(
    mark(1, "relation.r1.origin.source.person_nam"),
    mark(5, "relation.r1.origin.target.city_nam")
  )
)
```

Wydruk 5.2. Przykładowa reguła rozpoznająca atrybuty zdarzenia. Reguła pochodzi z pracy Brun i Hagège (2009).

Rys. 5.2. Przykładowa reguła WCCL tworzącą relację *pochodzenie* między *nazwą osoby* i *nazwą miasta*.

4. Pogrupuj nazwy anotacji według atrybucie `{id}`.
5. Każda grupa powinna zawierać parę krotek a i b spełniających warunki:
   (1) a.type=b.type
   (2) (a.role="source"$\wedge$ b.role="target") $\vee$ (b.role="source"$\wedge$ a.role="target")
   Naruszenie któregokolwiek z tych warunków powoduje pominięcie danej grupy wraz ze zgłoszeniem odpowiedniego komunikatu.
6. Wyznacz anotacje powiązane z kanałami — niech p będzie indeksem pierwszego tokenu tworzącego anotację w kanale k. Indeksem anotacji kanału k jest wartość `{annotation_nam}[k]`, gdzie `{annotation_nam}` to nazwa kategorii anotacji. Anotacją powiązaną z kanałem k jest anotacja o nazwie kategorii `{annotation_nam}` i indeksie `{annotation_nam}[k]`.
7. Dla każdej grupy nazw anotacji utwórz relację o nazwie `{type}` między anotacjami powiązanymi z kanałami. Relacja prowadzi od jednostki powiązanej z kanałem `{role} = "source"` do anotacji powiązanej z kanałem `{role} = "target"`.

Konstrukcja reguł

Konstrukcja reguł została podzielona na dwa etapy. W pierwszym etapie został stworzony bazowy zbiór reguł pokrywający jak największą liczbę relacji ze zbioru uczącego. Następnie bazowy zbiór reguł został zweryfikowany i uogólniony w oparciu o zbiór pomocniczy.

Etap 1 — Konstrukcja reguł na zbiorze uczącym. Celem pierwszego etapu konstrukcji reguł było sprawdzenie, czy korzystając z wiedzy ogólnej i zbioru przykładowych zdań możliwe jest opracowanie w krótkim czasie zbioru reguł o możliwie dużej precyzji i kompletności. W tym celu zostało przyjęte ograniczenie czasowe, polegające

na opracowaniu jak najlepszego zbioru reguł w czasie dwóch godzin. Oznaczało to, że dla każdej kategorii relacji zostały poświęcone nie więcej niż dwie godziny na opracowanie jak najlepszego zbioru reguł. Wyjątkiem była relacja *pochodzenie*, nad którą spędzono sześć osobodni (dwie osoby pracowały średnio trzy dni). W tym czasie zostało opracowanych łącznie 117 reguł (po 25 reguł dla relacji *alias* i *przynależność*, 22 dla relacji *położenie*, 15 dla relacji *autorstwo*, po 14 dla relacji *kompozycja* i *pochodzenie*, 9 dla relacji *sąsiedztwo* oraz 7 dla relacji *narodowość*).

Relacja	**Zbiór uczący**			**Zbiór pomocniczy**			**Zbiór testowy**		
	P [%]	R [%]	F [%]	P [%]	R [%]	F [%]	P [%]	R [%]	F [%]
autorstwo	85,71	6,59	12,24	0,00	0,00	0,00	100,0	2,33	4,55
kompozycja	61,90	6,34	11,50	33,33	3,03	5,56	90,00	30,00	45,00
narodowość	83,33	23,81	37,04	0,00	0,00	0,00	66,67	20,00	30,77
pochodzenie	86,54	62,50	72,58	88,89	72,73	80,00	83,33	43,48	57,14
położenie	84,95	10,38	18,50	100,0	10,94	19,77	100,0	10,26	18,60
przynależność	69,44	9,77	17,21	28,57	3,39	6,06	80,95	13,18	22,67
sąsiedztwo	100,0	6,25	11,76	0,00	0,00	0,00	100,0	6,25	10,53
tożsamość	100,0	13,79	24,24	60,00	10,34	17,65	100,0	7,78	14,43

Tabela 5.2. Wynik bazowy rozpoznawania relacji między nazwami własnymi przy pomocy ręcznie opracowanych reguł na bazie zbioru uczącego.

Zbiór reguł został przetestowany na wszystkich trzech częściach korpusu KPWr, a wyniki zostały przedstawione w tabeli 5.2. Zgodnie z oczekiwaniem najlepsze wyniki zostały osiągnięte dla relacji *pochodzenie*, jako że na opracowanie reguł dla tej kategorii relacji poświęcono najwięcej czasu. Mimo to nawet na zbiorze uczącym nie udało się osiągnąć 100% skuteczności. Opracowany zbiór reguł pokrywał zaledwie 62,5% przykładów ze zbioru uczącego z precyzją 86,54%. Na zbiorze pomocniczym i testowym relacje typu *pochodzenie* były rozpoznawane z precyzją powyżej 80%. Na zbiorze testowym pokrycie reguł spadło do 43,48%.

Dla pozostałych kategorii relacji osiągnięte wyniki były dużo poniżej oczekiwań. Dla wszystkich relacji z wyjątkiem *kompozycji* opracowane reguły osiągnęły znacząco wyższą precyzję (od 67,90% dla relacji *położenie* do 96,55% dla relacji *tożsamość*) niż kompletność (od 14,06% dla relacji *przynależność* do 33,33% dla relacji *narodowość*). Niska kompletność wynika z dużej różnorodności form zapisu informacji jednego typu, co wiąże się z koniecznością dodania nowych reguł lub uogólnienia istniejących.

Dla relacji *autorstwo*, *narodowość* i *sąsiedztwo* opracowane reguły nie rozpoznały żadnych relacji w zbiorze pomocniczym. Świadczy to o dużej różnorodności wzorców opisujących te kategorie relacji i niskim pokryciu wzorców relacji pomiędzy zbiorem uczącym i testowym. Te kategorie relacji były także słabo rozpoznane w zbiorze testowym, tj. 8,89% średniej harmonicznej dla relacji *autorstwo*, 11,76% dla relacji *sąsiedztwo* i 30,77% dla relacji *narodowość*.

Podsumowując, w ograniczonym czasie i przy wykorzystaniu wiedzy ogólnej oraz zbioru przykładowych relacji nie udało się opracować zbioru reguł o zadowalającej precyzji i kompletności. Dla trzech relacji (*autorstwo*, *narodowość* i *sąsiedztwo*) zbiory uczący, pomocniczy i testowy zawierają bardzo zróżnicowane przykłady, co może wiązać się z potencjalnie niewielką liczbą powtarzających się wzorców. Natomiast wyniki osiągnięte dla relacji *pochodzenie* dają nadzieję, że zwiększenie nakładu pracy pozwoli na znaczącą poprawę wyników. Aby sprawdzić, do jakiego stopnia możliwa jest poprawa skuteczności reguł, została wykonana druga iteracja konstrukcji reguł. Wyniki zostały omówione w następnym punkcie.

Etap 2 — Weryfikacja reguł na zbiorze pomocniczym. W drugim etapie konstrukcji reguł został wykorzystany zbiór pomocniczy do weryfikacji i uogólnienia dotychczasowych reguł. Dodatkowo formalizm reguł WCCL został rozszerzony o nowy operator `isannpart`[5], dzięki czemu możliwe było pominięcie dowolnej sekwencji tokenów niebędących anotacjami. Na tym etapie zbiór reguł nie był rozszerzany o nowe reguły, a jedynie były modyfikowane już istniejące. Modyfikacja reguł polegała przede wszystkim na rozluźnieniu warunków na sekwencję tokenów poprzez wykorzystanie operatora `isannpart` i pomijanie tokenów z ograniczonego zakresu, a także rozszerzeniu zbiorów słów kluczowych i kategorii jednostek identyfikacyjnych. Nad każdą kategorią relacji lingwista mógł poświęcić do 8 godzin pracy.

Relacja	**Zbiór uczący**			**Zbiór pomocniczy**			**Zbiór testowy**		
	P [%]	R [%]	F [%]	P [%]	R [%]	F [%]	P [%]	R [%]	F [%]
autorstwo	76,32	31,87	44,96	83,33	25,00	38,46	60,00	6,98	12,50
kompozycja	26,35	21,46	23,66	21,21	21,21	21,21	37,50	50,00	42,86
narodowość	87,50	33,33	48,28	0,00	0,00	0,00	66,67	20,00	30,77
pochodzenie	86,54	62,50	72,58	88,89	72,73	80,00	83,33	43,48	57,14
położenie	63,01	20,37	30,78	77,08	23,87	36,45	59,09	16,67	26,00
przynależność	69,64	30,47	42,39	70,83	28,81	40,96	56,82	19,38	28,90
sąsiedztwo	70,00	14,58	24,14	0,00	0,00	0,00	33,33	6,25	10,53
tożsamość	86,84	28,45	42,86	31,58	20,69	25,00	56,00	15,56	24,35

Tabela 5.3. Wynik bazowy rozpoznawania relacji między nazwami własnymi przy pomocy ręcznie opracowanych reguł na bazie zbioru uczącego i pomocniczego.

Dla relacji *narodowość* i *sąsiedztwo* nie udało się uogólnić reguł w takim stopniu, aby były w stanie rozpoznać jakiekolwiek relacje na zbiorze pomocniczym. Konieczne byłoby napisanie zupełnie nowych reguł. Wynika to z dużej różnorodności wzorców opisujących te kategorie relacji. W przypadku relacji *autorstwo* dzięki uogólnionemu zbiorowi reguł można było rozpoznać 25% relacji w zbiorze pomocniczym z precyzją 83%. Uogólniony zbiór reguł poprawił także pokrycie na zbiorze testowym z 4% do

5. Operator *isannpart* to operator wykonujący test na przynależność tokenu o wskazanym indeksie do dowolnej anotacji o podanej nazwie.

prawie 7%, ale przy spadku precyzji z 100% do 60% (średnia harmoniczna wzrosła z 9% do 12%).

Brak poprawy został odnotowany także dla relacji *kompozycja* i *pochodzenie*. W przypadku relacji *kompozycja* nierozpoznane relacje były bardzo odmienne od pierwotnego zbioru i wymagały napisania nowych reguł. W przypadku relacji *pochodzenie* pierwotny zbiór reguł był już na tyle rozbudowany, że trudno było dostrzec elementy poprawiające kompletność bez drastycznego spadku precyzji.

Dla pozostałych trzech relacji, tj. *położenie*, *przynależność* i *tożsamość*, uogólnienie wzorców pozwoliło na poprawę kompletności, ale przy każdorazowym spadku precyzji. Dla wszystkich trzech relacji ostateczny wynik w postaci średniej harmonicznej na zbiorze pomocniczym został poprawiony. Niestety ta poprawa nie miała przełożenia na zbiór testowy. Tylko dla relacji *tożsamość* odnotowano wzrost średniej harmonicznej o 1 punkt procentowy. Natomiast dla relacji *położenie* odnotowano spadek o 3 punkty procentowe, a wynik dla relacji *przynależność* utrzymał się na tym samym poziomie.

Wnioski. Podsumowując, można stwierdzić, że rozpoznawanie relacji semantycznych między jednostkami identyfikacyjnymi nie jest prostym zadaniem i ręczna konstrukcja reguł w krótkim czasie nie pozwala na osiągnięcie zadowalających wyników. Pomimo że relacji *pochodzenie* poświęcono cztery razy więcej czasu niż innym relacjom, nie pozwoliło to osiągnąć zadowalających wyników. Biorąc także pod uwagę nieznaczny wzrost skuteczności reguł pomiędzy zbiorem bazowym (zob. tabela 5.2) a zbiorem po weryfikacji i uogólnieniu (zob. tabela 5.6), można wywnioskować, że dalsza poprawa skuteczności reguł będzie wymagała znacznie więcej czasu i wysiłku. Ponadto wraz ze wzrostem liczby reguł dochodzi problem redundancji i wykluczania się reguł, a uwzględnianie coraz bardziej specyficznych przypadków może doprowadzić do spadku precyzji.

5.3. Zastosowanie nadzorowanego uczenia do rozpoznawania relacji

Ideą nadzorowanego uczenia jest automatyzacja procesu konstrukcji modelu na podstawie zbioru ręcznie przygotowanych przykładów odpowiednich dla danego zadania. W punkcie 2.3 zostały przedstawione różne metody nadzorowanego uczenia wykorzystane w zadaniu ekstrakcji informacji. W każdym z przedstawionych podejść można wyróżnić dwa elementy: niezależne i zależne od języka. Elementem niezależnym od języka są narzędzia statystyczne (np. metody maszynowego uczenia, algorytmy dla funkcji jądrowych). Drugi element silnie wiąże się z przetwarzanym językiem i użytymi narzędziami do wstępnego przetwarzania tekstu. Jest to na przykład zbiór cech dla klasyfikatorów wektorowych, funkcja jądrowa odzwierciedlająca odległość (podobieństwo) między dwoma elementami (np. parami jednostek w kontekście zdania) lub zbiór predykatów użytych do konstrukcji reguł. Kolejnym elementem zależnym językowo, wspólnym dla wszystkich metod, jest ręcznie opracowany zbiór danych uczących.

Istniejące metody nadzorowanego uczenia dostarczają uniwersalnych narzędzi, które były wykorzystywane w wielu zadaniach klasyfikacji. Dla każdego problemu klasyfikacji bardzo ważne jest zdefiniowane zbioru cech opisujących istotne informacje dla danego zadania, które pozwolą na uchwycenie znaczących różnic między przykładami reprezentującymi różne klasy. To samo dotyczy problemu rozpoznawania relacji semantycznych między jednostkami.

Na początku rozdziału przedstawiono przyjęte założenie dotyczące zadania rozpoznawania relacji semantycznych, zgodnie z którym istnienie relacji między dwoma jednostkami w zdaniu musi wynikać z treści tego zdania. Oznacza to, że możliwe jest wskazanie zbioru elementów i powiązań między tymi elementami (cech), na podstawie którego można zidentyfikować obecność relacji i sklasyfikować jej typ. Na przykład w zdaniu *Jan mieszka w Warszawie* występuje relacja *położenie* między jednostkami *Jan* i *Warszawa*. Wyznacznikiem istnienia relacji w tym zdaniu jest obecność słowa *mieszkać*. Dodatkową cechą może być porządek liniowy elementów, tj. wystąpienie nazwy osoby bezpośrednio przed słowem *mieszkać* i nazwy miasta z przyimkiem *w* bezpośrednio po nim. Zaproponowany porządek liniowy jest jedną z wielu możliwych interpretacji.

Nadzorowane uczenie polega na identyfikacji pewnych powtarzających się zależności między cechami opisującymi przykłady. Ze względu na dużą różnorodność form kodowania jednej informacji konieczne jest dostarczenie takich cech, które będą powtarzalne w wielu różnych przykładach. Można wyróżnić następujące kryteria różnorodności:

- **różnorodność ortograficzna** — wynika z bogatej morfologii języka polskiego, np. *Jan* ***mieszka*** *...* i *Jan i Agata* ***mieszkają*** *...*, gdzie *mieszka* i *mieszkają* są różnymi formami tego samego czasownika *mieszkać*.
- **różnorodność składniowa** — predykat i jego argumenty mogą występować w różnej kolejności, np. *Jan mieszka w Krakowie*, *W Krakowie mieszka Jan*, *W Krakowie Jan mieszka*. Niektóre z form zapisu są częściej spotykanie niż pozostałe, a mimo to są poprawne i mogą wystąpić w tekście.
- **różnorodność semantyczna** — różne słowa mogą być użyte do przekazanie tej samej informacji, np. *Jan mieszka w Krakowie* i *Jan żyje w Krakowie*.

Różnorodność ta powoduje występowanie wielu rzadkich wzorców determinujących istnienie relacji określonego typu. W celu zwiększenia powtarzalności wzorców, czyli uwidocznienia cech powtarzalnych między różnymi przykładami, istotne jest zredukowanie tej różnorodności. Jest to możliwe dzięki zastosowaniu istniejących narzędzi do przetwarzania tekstu oraz zasobów dla języka polskiego, tj.:

- **analizator morfologiczny i tager** — przeprowadza analizę morfologiczną dla słów (w tym formę bazową) dzięki czemu redukują formy odmienione dostarczając ich formy podstawowe. W tym zadaniu został wykorzystany analizator *maca* (Radziszewski i Śniatowski, 2011a) i tager WMBT (Radziszewski i Śniatowski, 2011b),

- **parser zależnościowy** — dostarcza informacje o zależnościach między predykatami i ich argumentami w obrębie zdania, np. argumenty czasownika. W tym zadaniu został wykorzystany parser MaltParser z modelem danych dla języka polskiego wyuczonym na części korpusu NKJP (Wróblewska i Woliński, 2012),
- **wordnet** — sieć powiązań semantycznych między formami leksykalnymi, w której opisana jest m.in. relacja hiperonimii (słowa o szerszym znaczeniu) lub synonimii (słowa o takim samym znaczeniu). W tym zadaniu został wykorzystany wordnet dla języka polskiego o nazwie Słowosieć (Piasecki *et al.*, 2009).

5.4. Automatyczna identyfikacja cech

Zakładając, że każde zdanie można przedstawić jako graf, w którym wierzchołki reprezentują tokeny i anotacje, a krawędzie zależności między tokenami i anotacjami, to cecha rozumiana jest jako dowolna ścieżka w tym grafie opisana jako zbiór warunków na sekwencję elementów (warunki na atrybuty tokenów, anotacji i relacji). Każda ścieżka w grafie między dwoma dowolnymi elementami jest potencjalnie istotną cechą, czyli taką, która wiąże się z wystąpieniem relacji określonego typu w danym zdaniu.

Do identyfikacji cech został zastosowany paradygmat indukcyjnego programowania logicznego (ang. *Inductive Logic Programming*; ILP), który był już wykorzystany z powodzeniem do rozpoznawania elementów opisu zdarzeń dla języka angielskiego (Ramakrishnan *et al.*, 2007). ILP jest działem maszynowego uczenia, w którym wykorzystywane są techniki programowania logicznego. Dzięki dużej ekspresyjności ILP możliwe jest szukanie wzorców w danych, które reprezentowane są jako grafy, czyli takich, których nie można opisać przy pomocy stałej liczby atrybutów bez konieczności redukcji informacji.

W programowaniu logicznym wyróżnia się dwa elementy: zbiór predykatów opisujących cechy obiektów i relacje między obiektami oraz zbiór reguł wnioskowania. Reguły wnioskowania zapisywane są przy użyciu rachunku predykatów pierwszego rzędu. W programowaniu logicznym cecha będzie odpowiadała jednej regule. ILP daje narzędzia do automatycznej konstrukcji reguł dla zadanej bazy wiedzy oraz dostarczonych pozytywnych i negatywnych przykładów.

Istnieje wiele systemów implementujących paradygmat indukcyjnego programowania logicznego, m.in. Aleph (Srinivasan, 2006), Foil (Quinlan i Cameron-Jones, 1993), Golem (Muggleton i Feng, 1990), Progol (Muggleton, 1995). Systemy te różnią się pod względem zaimplementowanych algorytmów indukcji reguł, możliwości konfiguracji przeszukiwania przestrzeni rozwiązań i kryteriów sukcesu. Ze względu na największe możliwości konfiguracyjne do eksperymentów został wykorzystany system Aleph.

5.4.1. Definicja bazy wiedzy

W sekcji 5.3 zostały wymienione trzy aspekty, które mają wpływ na różnorodność formy zapisu informacji. Jest to różnorodność ortograficzna, składniowa i semantyczna.

Aby zmaksymalizować powtarzalność cech między różnymi przykładami, należy dostarczyć taką informację, która zredukuje różnorodność poszczególnych elementów. Mając to na uwadze, zdefiniowano listę odpowiednich typów danych i predykatów.

Predykaty

- informacja podstawowa:
  - **token_orth(token,word)** — forma ortograficzna tokenu,
  - **token_pattern(token,pattern)** — wzorzec graficzny formy ortograficznej tokenu,
  - **token_after_token(token,token)** — określenie kolejności tokenów w zdaniu,
  - **annotation_range(annotation,token,token)** — granice anotacji, wskazuje pierwszy i ostatni token anotacji,
  - **annotation_type(annotation,annotation_type)** — określa kategorię anotacji,
  - **sentence_has_annotation(sentence,annotation)** — wskazuje na wystąpienie anotacji w zdaniu,
- analiza morfologiczna — różnorodność ortograficzna:
  - **token_base(token,word)** — forma bazowa tokenu,
  - **token_pos(token,pos)** — klasa gramatyczna tokenu,
  - **sentence_has_base(sentence,word)** — wskazuje na wystąpienie formy bazowej słowa w zdaniu,
- analiza semantyczna — różnorodność semantyczna:
  - **token_hypheronym(token,word)** — hiperonim formy bazowej,
  - **sentence_has_hypheronym(sentence,word)** — wskazuje na wystąpienie hiperonimu w zdaniu,
- analiza zależnościowa — różnorodność składniowa:
  - **token_dependency(token,token,dependency_type)** — zależność składniowa między dwoma tokenami.

Typy danych

- **word** — typ słownikowy, lista słów występujących w korpusie treningowym,
- **pos** — typ wyliczeniowy, lista kategorii gramatycznych,
- **sentence** — obiekt reprezentujący zdanie,
- **token** — obiekt reprezentujący token,
- **annotation** — obiekt reprezentujący anotację,
- **annotation_type** — typ słownikowy, lista typów anotacji.

Przykład

Poniżej znajduje się przykładowe zdanie *Pan Jan Nowak mieszka w Warszawie od urodzenia.* zapisane jako zbiór predykatów. W zdaniu oznaczone są dwie jednostki identyfikacyjne: *Jan Nowak* jako nazwa osoby i *Warszawie* jako nazwa miasta, które są połączone relacją *położenie*.

```
word(Pan). word(Jan). word(Nowak). word(mieszka). word(w).
word(Warszawie). word(od). word(urodzenia). word(.).
```

Definicja atrybutów tokenów (forma ortograficzna, bazowa i klasa gramatyczna):

```
token_orth(t0, "Pan").          token_base(t0, "pan").
                                token_pos(t0, "subst").
token_orth(t1, "Jan").          token_base(t1, "Jan").
                                token_pos(t1, "subst").
token_orth(t2, "Nowak").        token_base(t2, "Nowak").
                                token_pos(t2, "subst").
token_orth(t3, "mieszka").      token_base(t3, "mieszkać").
                                token_pos(t3, "fin").
token_orth(t4, "w").            token_base(t4, "w").
                                token_pos(t4, "prep").
token_orth(t5, "Warszawie").    token_base(t5, "Warszawa").
                                token_pos(t5, "subst").
token_orth(t6, "od").           token_base(t6, "od").
                                token_pos(t6, "prep").
token_orth(t7, "urodzenia").    token_base(t7, "urodzenie").
                                token_pos(t7, "subst").
token_orth(t8, ".").            token_base(t8, ".").
                                token_pos(t8, "interp").
```

Hiperonimy i synonimy form bazowych (wybrane przykłady):

```
token_hypheronym(t0, "word_syn_6797_człowiek_ze_względu
                        _na_swoje_zajęcie").
token_hypheronym(t0, "word_syn_28688_osoba").
token_hypheronym(t0, "word_syn_47701_zwrot_grzecznościowy").
...
token_hypheronym(t7, "word_syn_98146_urodzenie").
token_hypheronym(t7, "word_syn_102576_zrobienie").
...
```

Kolejność tokenów w zdaniu:

```
token_after_token(t0, t1). token_after_token(t1, t2).
token_after_token(t2, t3). token_after_token(t3, t4).
token_after_token(t4, t5). token_after_token(t5, t6).
token_after_token(t6, t7). token_after_token(t7, t8).
```

Zależności między tokenami:

```
token_dependency(d0_s1_t0, d0_s1_t3, "subj").
token_dependency(d0_s1_t1, d0_s1_t0, "app").
token_dependency(d0_s1_t2, d0_s1_t1, "app").
token_dependency(d0_s1_t4, d0_s1_t3, "comp").
token_dependency(d0_s1_t5, d0_s1_t4, "comp").
token_dependency(d0_s1_t6, d0_s1_t5, "adj").
token_dependency(d0_s1_t7, d0_s1_t6, "comp").
token_dependency(d0_s1_t8, d0_s1_t3, "punct").
```

Deklaracja anotacji i ich atrybutów:

```
annotation_type(a1,"person_nam"). annotation_range(a1,t1,t2).
token_before_annotation(a1,t0). token_after_annotation(a1,t3).

annotation_type(a1,"city_nam").   annotation_range(a2,t5,t5).
token_before_annotation(a2,t4). token_after_annotation(a1,t6).
```

Deklaracja relacji między parą anotacji:

```
relation(a1,a2,"location").
```

5.4.2. Konfiguracja przeszukiwania przestrzeni rozwiązań

Po przeprowadzeniu wstępnych eksperymentów (Marcińczuk i Ptak, 2012) została przyjęta następująca konfiguracja przeszukiwania przestrzeni:

- **poziom zagnieżdżenia predykatów** (`i=8`) — określa liczbę poziomów zagnieżdżenia dla zmiennych wprowadzanych przez predykaty, np. dla poniższej reguły:

  ```
  relation(A,B,origin) :-
    annotation_first_token(A,C),
    token_after_token(D,C),
    token_orth(D,w).
  ```

  Zmienna C jest zmienną na pierwszym poziomie zagnieżdżenia, zmienna D jest na drugim poziomie, a wartość „w” jest na 3 poziomie.
- **długość reguły** (`clauselength=8`) — ograniczenie na maksymalną liczbę predykatów w regule,
- **limit testowanych reguł** (`nodes=320000`) — określa maksymalną liczbę rozwiązań, jakie będę przetestowane dla pojedynczego przykładu. Po osiągnięciu limitu dalsze przeszukiwanie jest przerywane.

- **liczba pozytywnych przykładów** (`minpos=2`) — minimalna liczba przykładów pozytywnych pokrywanych przez regułę. Wartość >1 gwarantowała, że każda reguła musiała być na tyle ogólna, aby była dopasowana do co najmniej dwóch przykładów pozytywnych.
- **liczba negatywnych przykładów** (`noise=2`) — maksymalna liczna przykładów negatywnych pokrywanych przez regułę. Wartość >1 dopuszczała wystąpienie błędów w danych uczących.

5.4.3. Kontrola przeszukiwania przestrzeni rozwiązań

Dowolny uniwersalny system do indukcji reguł, zanim dotrze do akceptowalnego rozwiązania (lokalnego bądź globalnego), rozważa wszystkie potencjalne rozwiązania. Z punktu widzenia konkretnego zadania duża część tych rozwiązań jest redundantna lub niepożądana. Ogromna liczba możliwych rozwiązań skutkuje bardzo długim czasem poszukiwania optymalnego rozwiązania, co w praktyce wymaga ograniczenia liczby rozważanych rozwiązań w celu otrzymania wyniku w dopuszczalnym czasie. Biorąc pod uwagę specyfikę zadania, dla którego wykorzystywane jest programowanie logiczne, możliwe jest opracowanie zbioru reguł odrzucających pewne obszary przestrzeni poszukiwania. Dzięki takim regułom możliwe będzie przetestowanie większej liczby unikalnych rozwiązań w tym samym czasie.

System Aleph umożliwia definicję reguł odcinania (ang. *prunning*), które określają warunki, przy których dane rozwiązanie wraz z całym poddrzewem (zbiór reguł powstałych po dodaniu nowego predykatu do aktualnej reguły) zostaje pominięte. Na podstawie wstępnych eksperymentów z indukcją reguł (Marcińczuk i Ptak, 2012) został opracowany zbiór reguł odcinania.

Predykaty opisujące atrybuty tokenu (`token_orth`, `token_base`, `token_pattern`, `token_hypheronym`) są częściowo zależne od siebie. Jeżeli predykat `A` jest zależny od predykatu `B` (`B` $\Rightarrow$ `A`) i w regule R istnieje predykat `B` wartościowany listą argumentów X=[x_1,...,x_n], to dodanie predykatu `A` wartościowanego listą argumentów X nie zmienia zbioru pokrywanych przykładów. Z punktu widzenia systemu ILP są to różne reguły, które mają oddzielne drzewa przeszukiwania. W takiej sytuacji można pominąć drzewo przeszukiwania dla reguły z predykatem `B` bez ryzyka pominięcia poprawnego rozwiązania. Poniżej przedstawione są zidentyfikowane zależności między predykatami:

- `token_orth(A, _)` $\Rightarrow$ `token_pattern(A, _)` — dla każdej formy ortograficznej istnieje dokładnie jeden wzorzec formy ortograficznej,
- `token_base(A, _)` $\Rightarrow$ `token_hypheronym(A, _)` — dla każdej formy bazowej istnieje dokładnie jeden zbiór hiperonimów; to założenie wynika z braku ujednoznaczniania sensów słów, przez co dla każdego słowa brane są wszystkie możliwe hiperonimy.

Na przykład reguła składająca się z dwóch predykatów `token_base(A, "miasto")` i `token_hypheronym(A, "teren_zabudowany")` jest równoważna regule składającej się tylko z pierwszego predykatu, tj. `token_base(A, "miasto")`. We wspomnianym po-

wyżej przykładzie dodanie predykatu `token_hypheronym(A, "teren_zabudowany")` nie zmienia zbioru pokrywanych przykładów, ponieważ jeżeli prawdziwy jest predykat `token_base(A, "miasto")`, to zawsze prawdziwy jest predykat `token_hypheronym (A, "teren_zabudowany")`.

Na wydruku 5.3 znajduje się postać reguły odcinania zapisanej zgodnie z konwencją używaną w systemie Aleph. Predykaty `member` i `has_pieces` są predykatami pomocniczymi wyrażonymi za pomocą reguł (zob. wydruk 5.4). Zmienna `Body` reprezentuje regułę w postaci łańcucha znaków składającego się z predykatów oddzielonych przecinkiem. Reguła `has_pieces` dzieli regułę `Body` na tablicę predykatów. Reguła `member` określa element tablicy `LBody`, tj. zmienna `P` jest pewnym predykatem z reguły `Body`.

```
1  prune((_H:-Body)) :-
2    has_pieces(Body,LBody),
3    member(P,LBody),
4    P =.. ["token_base",A1,_],
5    member(P2,LBody),
6    P2 =.. ["token_hypheronym",A2,_],
7    A1==A2.
```

Rys. 5.3. Reguła odcinania zapisana w konwencji systemu Aleph usuwająca redundantne reguły.

```
1  member(T,[T|_]).
2  member(X,[_|Q]) :- member(X,Q).
3
4  has_pieces((A,B),[A|L]) :- has_pieces(B,L), !.
5  has_pieces((A),[A]) :- !.
```

Rys. 5.4. Definicja reguł pomocniczych *member* i *has_pieces*

5.4.4. Modele predykatów

Ze względu na dużą przestrzeń możliwych rozwiązań przy uwzględnieniu wszystkich możliwych predykatów zostało zdefiniowanych kilka modeli predykatów, w których zbiór predykatów został ograniczony do wybranych elementów.

Model słów kluczowych kontekstu zdaniowego

Głównym wyznacznikiem występowania relacji semantycznej w obrębie zdania są pewne słowa kluczowe i frazy. Celem tego eksperymentu było sprawdzenie, w jakim stopniu zestaw słów kluczowych oderwany od pozycji w zdaniu i względem jednostek może być wyznacznikiem zaistnienia relacji. Pod uwagę zostały wzięte formy bazowe i hiperonimy form bazowych określonych klas gramatycznych (rzeczowniki, czasowniki i

przymiotniki). Parametry generowania reguł zostały tak dobrane, aby utworzone reguły mogły definiować do trzech słów kluczowych. Zwiększenie dopuszczalnej liczby słów kluczowych powodowało znaczący wzrost przestrzeni przeszukiwania ze względu na wykładniczą liczbę kombinacji słów występujących w zdaniu.

Dla modelu słów kluczowych zostało wygenerowanych łącznie ponad 300 reguł (153 dla relacji *położenie*, 71 dla relacji *przynależność*, 27 dla relacji *autorstwo*, 26 dla relacji *kompozycja*, 20 dla relacji *pochodzenie*, 15 dla relacji *sąsiedztwo*, 6 dla relacji *narodowość* i 0 dla relacji *tożsamość*). Na wydruku 5.5 zostały przedstawione przykładowe reguły.

```
relation(A,B,creator) :-
  sentence_has_annotation(C,B),
  sentence_has_base(C,"word_organizować").

relation(A,B,affiliation) :-
  annotation_of_type(B,band_nam),
  sentence_has_annotation(C,B),
  sentence_has_hypheronym(C,"word_syn_6797_człowiek_ze_względu
    _na_swoje_zajęcie").
```

Rys. 5.5. Przykładowe reguły rozpoznające relacje na podstawie zbioru słów kluczowych.

Dla relacji *tożsamość* nie została wygenerowana żadna reguła. Prawdopodobnie jest to spowodowane tym, że większość wystąpień tej relacji dotyczy alternatywnych nazw, które podawane są w okrągłych nawiasach tuż za nazwą właściwą. Jedynym wyznacznikiem tej relacji jest wystąpienie nawiasów oraz porządek liniowy elementów. Samo wystąpienie nawiasów jest zbyt słabą przesłanką, ponieważ nawiasy wykorzystywane są w wielu innych celach (np. wskazanie przynależności jednego elementu do drugiego).

Dla pozostałych kategorii relacji główną przyczyną wielu nieprawidłowo rozpoznanych relacji (a tym samym niskiej precyzji) są zdania, w których występuje para jednostek połączonych relacją oraz dodatkowe anotacje niepołączone tą relacją. Ponieważ słowa kluczowe identyfikują relacje na poziomie zdań, nie jest możliwe poprawne wskazanie jednostek połączonych relacją i w efekcie generowane są nadmiarowe połączenia.

Mimo niskiej precyzji cechy słów kluczowych mogą być wykorzystane jako cechy pomocnicze np. do filtrowania zdań.

Model kontekstu jednostek identyfikacyjnych

Najwięcej przesłanek wskazujących na istnienie relacji między konkretną parą jednostek identyfikacyjnych znajduje się w bezpośrednim kontekście tych jednostek (tokeny poprzedzające jednostkę i następujące po niej). W celu zamodelowania wzorców kontekstu został wykorzystany predykat opisujący porządek liniowy tokenów `token_after_token` oraz zbiór predykatów definiujących atrybuty tokenów, tj. `token_base`, `token_orth`, `token_hypheronym` i `token_pattern`.

Relacja	Zbiór uczący			Zbiór pomocniczy			Zbiór testowy		
	P [%]	R [%]	F [%]	P [%]	R [%]	F [%]	P [%]	R [%]	F [%]
autorstwo	66,67	68,49	67,57	31,25	75,00	44,12	22,92	51,16	31,65
kompozycja	78,24	83,89	80,97	38,60	68,75	49,44	69,57	53,33	60,38
narodowość	100,0	50,00	66,67	100,0	16,67	28,57	12,50	10,00	11,11
pochodzenie	72,97	77,14	75,00	42,86	54,55	48,00	14,81	54,55	23,30
położenie	66.35	66,98	66,67	13,87	38,26	20,36	11,19	43,79	17,82
przynależność	64,38	82,07	72,15	20,78	54,24	30,05	18,95	46,03	26,85
sąsiedztwo	59,38	42,70	49,67	0,00	0,00	0,00	50,00	25,00	33,33
tożsamość	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00

Tabela 5.4. Wynik rozpoznawania relacji przy pomocy reguł identyfikujących zbiory słów kluczowych.

W tym modelu możliwe jest zapisanie każdej reguły WCCL, która została ręcznie stworzona na potrzeby rozpoznawania relacji semantycznych. Dzięki temu możliwe będzie bezpośrednie porównanie skuteczności automatycznie wygenerowanych reguł i reguł opracowanych ręcznie.

Dla tego modelu zostało wygenerowanych łącznie 510 reguł, w tym 244 dla relacji *położenie*, 96 dla relacji *przynależność*, 44 dla relacji *kompozycja*, 33 dla relacji *sąsiedztwo*, 30 dla relacji *autorstwo*, 29 dla relacji *pochodzenie*, 24 dla relacji *tożsamość* oraz 10 dla relacji *narodowość*. Na wydruku 5.6 zostało przedstawionych kilka wybranych reguł wygenerowanych dla relacji *położenie*, *sąsiedztwo* i *pochodzenie*.

Wyniki osiągnięte przez wygenerowane reguły zostały przedstawione w tabeli 5.5. Automatycznie skonstruowane reguły w porównaniu do reguł stworzonych ręcznie uzyskały wyższą lub taką samą kompletność dla wszystkich kategorii relacji. Natomiast w przypadku precyzji odnotowano większe zmiany niż dla kompletności — na zbiorze pomocniczym dla 4 kategorii nastąpił wzrost precyzji, w jednej nie było zmian i dla 3 kategorii nastąpiło obniżenie. Pomimo spadku precyzji pod względem średniej harmonicznej dla wszystkich relacji poprawił się wynik. Jedynie w przypadku relacji *pochodzenie*, w przypadku której poświęcono więcej czasu na opracowanie reguł, na zbiorze pomocniczym osiągnięto taki sam wynik. Natomiast na zbiorze testowym odnotowano spadek o 16 punktów procentowych.

Jedną z głównych przyczyn pogorszenia precyzji dla części kategorii w stosunku do ręcznie opracowanych reguł jest generowanie reguł, które opisują konteksty niezależnie dla każdej jednostki bez utworzenia połączenia między nimi, np. ostatnia reguła na wydruku 5.5. Takie reguły wykonane na zdaniu, w którym oprócz pary jednostek połączonych relacją posiadają także inne jednostki, które zgodnie ze słownikiem podtypów mogą być ze sobą połączone, powodują wskazanie nieprawidłowych powiązań.

```
relation(A,B,location) :-
  annotation_first_token(B,C),
  token_after_token(D,C),
  token_after_token(E,D),
  token_base(D,word_w),
  annotation_last_token(A,E),
  annotation_of_type(A,facility_nam).

relation(A,B,location) :-
  annotation_first_token(A,C),
  token_after_token(D,C),
  token_hypheronym(D,"word_syn_4884_miejscowość"),
  token_after_token(E,D),
  token_orth(E,word_w).

relation(A,B,neighbourhood) :-
  annotation_first_token(B,C),
  token_after_token(D,C),
  token_base(D,"word_koło").

relation(A,B,neighbourhood) :-
  annotation_last_token(B,C),
  token_after_token(D,C),
  token_base(D,word_przy),
  token_after_token(C,E),
  token_after_token(E,F)

relation(A,B,origin) :-
  annotation_last_token(B,C),
  token_after_token(C,D),
  token_base(D,"word_urodzić").
```

Rys. 5.6. Przykładowe reguły rozpoznające relacje na podstawie kontekstów wokół jednostek identyfikacyjnych.

Relacja	Zbiór uczący			Zbiór pomocniczy			Zbiór testowy		
	P [%]	R [%]	F [%]	P [%]	R [%]	F [%]	P [%]	R [%]	F [%]
autorstwo	79,78	97,26	87,65	44,12	75,00	55,56	26,83	25,58	26,19
kompozycja	83,64	99,44	90,86	37,70	71,88	49,46	43,86	83,33	57,47
narodowość	86,96	100,00	93,02	55,56	83,33	66,67	35,29	60,00	44,44
pochodzenie	83,13	98,57	90,20	88,89	72,73	80,00	31,71	59,09	41,27
położenie	79,93	95,79	87,14	37,25	61,74	46,46	20,93	35,29	26,28
przynależność	75,73	93,23	83,57	34,91	62,71	44,85	33,22	75,40	46,12
sąsiedztwo	79,61	92,13	85,42	11,54	30,00	16,67	13,33	12,50	12,90
tożsamość	86,90	85,88	86,39	55,56	55,56	55,56	49,15	43,28	46,03

Tabela 5.5. Wynik rozpoznawania relacji przy pomocy reguł opisujących bezpośrednie konteksty jednostek.

Model zależności między tokenami

W celu zredukowania różnorodności składniowej porządek liniowy tokenów (wyrażony przy pomocy predykatu `token_after_token`) został zastąpiony zależnościami między tokenami (wyrażonym przy pomocy predykatu `token_dependency`). Dodatkową zaletą modelu zależnościowego jest możliwość uchwycenia dalekich zależności, które ze względu na ograniczenie na głębokość zagnieżdżenia predykatów nie mogą być odkryte w modelu liniowym. Pozostałe predykaty opisujące tokeny zostały takie same jak w modelu kontekstów.

Dla tego modelu zostało wygenerowanych 300 reguł, w tym: 147 dla relacji *położenie*, 56 dla relacji *przynależność*, 22 dla relacji *kompozycja*, 19 dla relacji *tożsamość*, po 17 dla relacji *autorstwo*, *sąsiedztwo* i *pochodzenie* oraz 5 dla relacji *alias*. Na wydruku 5.7 znajdują się dwie przykładowe reguły rozpoznające relację *pochodzenie* i *sąsiedztwo*.

Wyniki osiągnięte przez reguły wygenerowane dla modelu zależności między tokenami zostały przedstawione w tabeli 5.6. W stosunku do modelu kontekstu jednostek tylko dla dwóch kategorii relacji została osiągnięta zauważalna poprawa wyniku na obu zbiorach testowych, tj. dla relacji *autorstwo* i *kompozycja*. Na zbiorze pomocniczym poprawa wyniku wyniosła 7 punktów procentowych dla relacji *autorstwo* i 8 punktów procentowych dla relacji *kompozycja*. Na zbiorze testowym poprawa wyniosła odpowiednio 6 i 8 punktów procentowych. Jednocześnie liczba reguł dla obu kategorii relacji zmalała o połowę — w przypadku relacji *autorstwo* z 30 do 17 reguł, a w przypadku relacji *kompozycja* z 44 do 22.

Dla kolejnych dwóch kategorii relacji, tj. *położenie* i *przynależność* otrzymany wynik był nieznacznie gorszy. Na zbiorze pomocniczym był to spadek średniej harmonicznej o 4 punkty procentowe dla relacji *położenie* i o 0,1 dla relacji *przynależność*. Na zbiorze testowym spadek wyniósł odpowiednio 3 i 5 punktów procentowych. Dla tych kategorii także nastąpił spadek liczby wygenerowanych reguł: z 244 do 147 dla relacji *położenie* i z 96 do 56 dla relacji *przynależność*.

```
relation(A,B,origin) :-
  annotation_first_token(B,C),
  token_dependency(C,D,comp),
  token_hypheronym(D,word_syn_164585_z),
  token_dependency(D,E,adj),
  token_dependency(E,F,adj),
  token_hypheronym(F,word_syn_4750_miejsce).

relation(A,B,neighbourhood) :-
  annotation_of_type(B,road_nam),
  annotation_first_token(A,C),
  token_dependency(C,D,adj),
  token_hypheronym(D,"word_syn_25121_miejsce_ze_względu
    _na_przeznaczenie"),
  token_hypheronym(D,"word_syn_103155_obiekt_fizykalny
    _którego_części_są_połączone_bezpośrednio").
```

Rys. 5.7. Przykładowe reguły rozpoznające relacje na podstawie zależności między tokenami.

Spadek liczby wygenerowanych reguł dla wspomnianych 4 kategorii relacji wskazuje, że dzięki zastąpieniu porządku liniowego tokenów przez nieliniowe zależności między tymi tokenami udało się częściowo zredukować różnorodność składniową. Zmniejszenie liczby reguł miało negatywny wpływ na kompletność rozpoznawanych relacji, która spadła o 1/3. Jednocześnie znacząco wzrosła precyzja reguł — w przypadku relacji *autorstwo* do 73%, a dla relacji *kompozycja* do 75%.

Dla pozostałych 4 kategorii relacji wyniki znacząco się pogorszyły. W przypadku relacji *narodowość*, *sąsiedztwo* i *tożsamość* wygenerowane reguły nie dopasowały żadnego przykładu ze zbioru pomocniczego i testowego (dla relacji *tożsamość* w zbiorze testowym został dopasowany zaledwie jeden przykład). Pomimo znaczącego wzrostu skuteczności rozpoznawania 2 kategorii relacji semantycznych model zależności między tokenami nie może w całości zastąpić porządku liniowego tokenów. Natomiast informacje o zależnościach pomiędzy tokenami mogą być wykorzystane jako dodatkowa informacja, co zostało przedstawione w sekcji 5.5.

5.4.5. Zestawienie wyników

W tabelach 5.7 i 5.8 zostały przedstawione wyniki osiągnięte przez automatycznie wygenerowane reguły dla modelu słów kluczowych (*ilp-key*), modelu kontekstu jednostek (*ilp-seq*) i modelu zależności między tokenami (*ilp-dep*) wraz z wynikami osiągniętymi dla podejść referencyjnych.

Jak widać w tabeli 5.7, model kontekstu jednostek (*ilp-seq*) osiągnął nie gorsze wy-

Relacja	Zbiór uczący			Zbiór pomocniczy			Zbiór testowy		
	P [%]	R [%]	F [%]	P [%]	R [%]	F [%]	P [%]	R [%]	F [%]
autorstwo	90,77	80,82	85,51	73,33	55,00	62,86	44,00	25,58	32,35
kompozycja	93,79	92,22	93,00	75,00	46,88	57,69	61,76	70,00	65,63
narodowość	100,0	60,00	75,00	0,00	0,00	0,00	0,00	0,00	0,00
pochodzenie	90,00	77,14	83,08	37,50	27,27	31,58	25,00	27,27	26,09
położenie	90,67	85,87	88,21	42,11	42,95	42,52	32,06	27,45	29,58
przynależność	90,38	86,06	88,16	41,43	49,15	44,96	40,15	42,06	41,09
sąsiedztwo	87,50	70,79	78,26	0,00	0,00	0,00	0,00	0,00	0,00
tożsamość	95,65	51,76	67,18	0,00	0,00	0,00	4,17	1,49	2,20

Tabela 5.6. Wynik rozpoznawania relacji przy pomocy reguł wykorzystujących model zależnościowy między słowami.

Zbiór	heur	rules	ilp-key	ilp-seq	ilp-dep
	F [%]	F [%]	F [%]	F [%]	F [%]
autorstwo	22,57	38,46	44,12	55,56	**62,86**
kompozycja	28,04	21,21	49,44	49,46	**57,69**
narodowość	26,32	0,00	28,57	**66,67**	0,00
pochodzenie	12,19	**80,00**	48,00	**80,00**	31,58
położenie	13,62	36,45	20,36	**46,46**	42,52
przynależność	23,89	40,96	30,05	44,85	**44,96**
sąsiedztwo	4,09	0,00	0,00	**16,67**	0,00
tożsamość	4,08	25,00	0,00	**55,56**	0,00

Tabela 5.7. Zestawienie wyników (średnia harmoniczna) dla podejść bazowych i automatycznie konstruowanych reguł na zbiorze pomocniczym.

niki niż obie metody referencyjne. Natomiast biorąc pod uwagę wszystkie trzy modele, dla trzech kategorii relacji model *ilp-seq* nie osiągnął najwyższych wyników.

Natomiast na zbiorze testowym (tabela 5.8) model kontekstu jednostek osiągnął najwyższy wynik tylko dla dwóch kategorii relacji, tj. *przynależność* i *tożsamość*. Dla kolejnych dwóch kategorii relacji, tj. *kompozycja* i *położenie* najwyższy wynik został osiągnięty dla modelu zależności między tokenami (*ilp-dep*).

Biorąc pod uwagę powyższe obserwacje, można uznać, że w celu uzyskania optymalnych wyników dla wszystkich kategorii relacji, konieczne jest uwzględnienie informacji ze wszystkich trzech modeli. Jedną z możliwości połączenia otrzymanych reguł i jednoczesnego utrzymania paradygamtu maszynowego uczenia jest skonstruowanie klasyfikatora wykorzystującego reguły z poszczególnych modeli jako przesłanki. W następnej sekcji zostały omówione dwie metody wykorzystania reguł jako wektorów cech opisujących pary jednostek.

Zbiór	heur	rules	ilp-key	ilp-seq	ilp-dep
	F [%]	F [%]	F [%]	F [%]	F [%]
autorstwo	**33,61**	12,50	31,65	26,19	32,35
kompozycja	20,58	42,86	60,38	57,47	**65,63**
narodowość	**48,00**	30,77	11,11	44,44	0,00
pochodzenie	14,97	**57,14**	23,30	41,27	26,09
położenie	18,52	26,00	17,82	26,28	**29,58**
przynależność	20,52	28,90	26,85	**46,12**	41,09
sąsiedztwo	3,88	10,53	**33,33**	12,90	0,00
tożsamość	3,25	24,35	0,00	**46,03**	2,20

Tabela 5.8. Zestawienie wyników (średnia harmoniczna) dla podejść bazowych i automatycznie konstruowanych reguł na zbiorze testowym.

5.5. Klasyfikator relacji w oparciu o wektory cech

W sekcji 5.4 zostały przedstawione trzy modele predykatów, które były wykorzystane do generowania reguł przy pomocy paradygmatu ILP. Każdy z tych modeli osiągał lepsze wyniki od pozostałych tylko dla niektórych kategorii relacji. W podsumowaniu został sformułowany wniosek, że połączenie wszystkich trzech modeli w jeden pozwoli na poprawę wyników. Wykorzystanie wszystkich predykatów do konstrukcji jednego modelu i wygenerowanie reguł przy użyciu ILP nie było możliwe ze względu na duży wymiar przestrzeni potencjalnych rozwiązań — liczba potencjalnych rozwiązań wyniosłaby $nodes^3$, gdzie parametr *nodes* dla dotychczasowych eksperymentów był ustawiony na wartość 320 tys.

W tej sekcji pracy została zaprezentowana metoda łączenia wielu modeli z wykorzystaniem tradycyjnych klasyfikatorów. Reguły wygenerowane dla poszczególnych kategorii relacji i modeli zostały potraktowane jako cechy binarne. Jeżeli reguła zachodzi dla danej pary jednostek, to generowana jest wartość 1, w przeciwnym wypadku 0. Zatem każda para jest reprezentowana jako n-elementowy wektor $X = [x_1, ..., x_n]$, gdzie $x_i \in \{0, 1\}$.

Do klasyfikacji zostało wykorzystane środowisko LexCSD, które daje możliwość przetestowania zadanego zbioru danych na wielu klasyfikatorach. Do testów zostały wykorzystane klasyfikatory, które obsługują cechy binarne, m.in.: BayesianLogisticRegression (BLogicRegression), NaiveBayes, ComplementNaiveBayes (CNaiveBayes), BFTree, LMT, RandomTree i DecisionTable[6].

6. Przedstawione nazwy metod odpowiadają klasom implementującym te metody w środowisku WEKA. Lista wszystkich dostępnych klasyfikatorów znajduje się na stronie `http://weka.sourceforge.net/doc/weka/classifiers/Classifier.html`.

5.5.1. Klasyfikator dla modelu sekwencyjnego

Celem pierwszego eksperymentu było przetestowanie możliwości identyfikacji przez klasyfikator nowych zależności między pojedynczymi regułami w obrębie jednego modelu. Ponieważ najlepsze wyniki zostały osiągnięte dla modelu kontekstu jednostek identyfikacyjnych, to model ten został przyjęty jako model referencyjny i został wykorzystany do przeprowadzenia eksperymentu.

W tym eksperymencie dla każdej kategorii relacji zostały wykorzystane tylko te reguły, które zostały wygenerowane dla danej relacji. Eksperymenty zostały przeprowadzone na wszystkich wspomnianych klasyfikatorach. Dla każdej kategorii relacji został wybrany klasyfikator, który osiągnął najlepsze wyniki na zbiorze pomocniczym, a następnie został przetestowany na zbiorze testowym.

W tabeli 5.9 zostały przedstawione wyniki dla najlepszych klasyfikatorów. Dla sześciu kategorii relacji z ośmiu klasyfikatory, które uzyskały najlepsze wyniki na zbiorze pomocniczym, osiągnęły także poprawę na zbiorze testowym w stosunku do wyników uzyskanych przez same reguły. Wzrost średniej harmonicznej wyniósł od 2 punktów procentowych dla relacji *pochodzenie* do 10 dla relacji *narodowość*. Dodatkowo dla wszystkich sześciu wspomnianych kategorii relacji została odnotowana poprawa precyzji kosztem kompletności. Może to świadczyć o tym, że klasyfikatory podejmują decyzję o istnieniu relacji dopiero, kiedy prawdziwych jest kilka reguł. Efektem tego jest obniżenie kompletności, co może świadczyć o tym, że wiele par wspartych jest tylko jedną regułą.

Dla pozostałych dwóch kategorii relacji, tj. *autorstwo* i *tożsamość* został odnotowany spadek zarówno precyzji, jak i kompletności. Mimo to wyniki eksperymentu wskazują, że klasyfikatory są w stanie zamodelować zależności między cechami, a tym samym wychwycić dodatkowe wzorce będące złożeniem reguł wygenerowanych przy pomocy ILP.

Relacja	**Klasyfikator**	**Zbiór pomocniczy**			**Zbiór testowy**		
		P [%]	**R** [%]	**F** [%]	**P** [%]	**R** [%]	**F** [%]
autorstwo	NaiveBayes	50.00	60.00	54.55	20.00	9.30	12.70
kompozycja	NaiveBayes	95.00	59.38	73.08	100.00	46.67	63.64
narodowość	RandomTree	80.00	66.67	72.73	50.00	60.00	54.55
pochodzenie	CNaiveBayes	100.00	63.64	77.78	47.62	45.45	46.51
położenie	BLogisticRegression	50.00	46.31	48.08	38.79	29.41	33.46
przynależność	DecisionTable	70.45	52.54	60.19	61.70	46.03	52.73
sąsiedztwo	DecisionTable	100.00	20.00	33.33	20.00	12.50	15.38
tożsamość	RandomTree	81.25	48.15	60.47	38.10	18.60	25.00

Tabela 5.9. Wynik rozpoznawania relacji przy pomocy klasyfikatorów wykorzystujących reguły modelu kontekstów jednostek identyfikacyjnych jako cechy.

5.5.2. Klasyfikator dla modelu łączonego

W modelu łączonym do opisu par jednostek zostały wykorzystane wszystkie reguły wygenerowane dla wszystkich kategorii relacji i wszystkich modeli predykatów. Dało to łącznie ponad 1100 reguł, a tym samym cech opisujących każdą parę jednostek identyfikacyjnych. Połączenie reguł wygenerowanych dla różnych modeli dla danej kategorii relacji powinno umożliwić uchwycenie zależności między np. występowaniem słów kluczowych w zdaniu a wzorcami kontekstów jednostek. Z kolei połączenie reguł dla różnych kategorii relacji powinno pomóc w wyeliminowaniu błędnych klasyfikacji między kategoriami jednostek.

Do wyznaczenia najlepszego klasyfikatora została zastosowana taka sama procedura jak dla klasyfikatora wykorzystującego reguły modelu kontekstu, tj. wszystkie wspomniane klasyfikatory zostały przetestowane na zbiorze pomocniczym. Następnie dla każdej kategorii relacji została wybrana konfiguracja, dla której osiągnięto najlepszy wynik. Wybrane klasyfikatory zostały przetestowane na zbiorze testowym.

W tabeli 5.10 zostały przedstawione wyniki dla wybranych klasyfikatorów dla poszczególnych relacji. Dla relacji, które w poprzednim modelu uzyskały gorsze wyniki niż same reguły, udało się znacząco je poprawić. Dla relacji *autorstwo* ostateczny wynik był lepszy o 9 punktów procentowych od najlepszych wyników osiągniętych przez same reguły. Natomiast dla relacji *tożsamość* wynik był bardzo zbliżony do wyników osiąganych przez reguły, ale mimo to gorszy o 0,16 punktu procentowego.

Relacja	**Klasyfikator**	**Zbiór pomocniczy**			**Zbiór testowy**		
		P [%]	**R** [%]	**F** [%]	**P** [%]	**R** [%]	**F** [%]
autorstwo	CLR+CNB+RT	50.0	65.0	56.52	45.71	37.21	41.03
kompozycja	BLogisticRegression	76.92	62.5	68.97	80.77	70.0	75.0
narodowość	BLogisticRegression	100.0	50.0	66.67	66.67	20.0	30.77
pochodzenie	BFTree	88.89	72.73	80.00	60.87	63.64	62.22
położenie	BLogisticRegression	48.25	46.31	47.26	43.88	28.10	34.26
przynależność	BLogisticRegression	53.85	47.46	50.45	47.02	56.35	51.26
sąsiedztwo	BFTree	16.67	20.0	18.18	16.67	12.5	14.29
tożsamość	BFTree	73.68	51.85	60.87	59.52	37.31	45.87

Tabela 5.10. Wynik rozpoznawania relacji przy pomocy klasyfikatorów wykorzystujących reguły modelu słów kluczowych, kontekstów jednostek identyfikacyjnych i zależności między słowami jako cechy.

5.5.3. Zestawienie wyników

Z uwagi na to, że żaden z dwóch modeli klasyfikatorów nie osiągnął zdecydowanie najlepszych wyników dla wszystkich kategorii relacji, to ostateczna konfiguracja dla obu modeli została wybrana spośród obu modeli. Kryterium wyboru najlepszej konfiguracji dla poszczególnych kategorii relacji była średnia harmoniczna precyzji i

kompletności uzyskana na zbiorze pomocniczym. W tabeli 5.11 zostały przedstawione najlepsze konfiguracje i wyniki osiągnięte na zbiorze pomocniczym łącznie z wynikami bazowymi.

Na zbiorze pomocniczym wybrane klasyfikatory, z wyjątkiem jednej kategorii relacji, osiągnęły wyniki nie gorsze niż wyniki osiągnięte przez metody referencyjne (heurystyka i ręcznie opracowane reguły) oraz reguły ILP. Jedynie dla relacji *autorstwo* żaden z klasyfikatorów nie osiągnął wyników nie gorszych niż reguły ILP. Mimo to na zbiorze testowym ten sam klasyfikator osiągnął wyniki lepsze niż reguły ILP. Może to świadczyć o bardzo dużej różnorodności przykładów w obu zbiorach.

Relacja	**Wynik bazowy**			**Klasyfikator**		
	Heur.	Reguły	ILP			
	F [%]	F [%]	F [%]	Model	Typ	F [%]
autorstwo	22,57	38,46	**62,86**	łączony	CLR+CNB+RT	56,52
kompozycja	28,04	21,21	57,69	konteksty	NaiveBayes	**73,08**
narodowość	26,32	0,00	66,67	konteksty	RandomTree	**72,73**
pochodzenie	12,19	**80,00**	**80,00**	łączony	BFTree	**80,00**
położenie	13,62	36,45	46,46	konteksty	BLR	**48,08**
przynależność	23,89	40,96	44,96	konteksty	DecisionTable	**60,19**
sąsiedztwo	4,09	0,00	16,67	konteksty	DecisionTable	**33,33**
tożsamość	4,08	25,00	55,56	łączony	BFTree	**60,87**

Tabela 5.11. Zestawienie konfiguracji dla najlepszych wyników na zbiorze pomocniczym

Relacja	**Wynik bazowy**			**Klasyfikator**		
	Heur.	Reguły	ILP			
	Model	Typ	F [%]	F [%]	F [%]	F [%]
autorstwo	33,61	12,50	32,35	łączony	CLR+CNB+RT	**41,03**
kompozycja	20,58	42,86	**65,63**	konteksty	NaiveBayes	63,64
narodowość	48,00	30,77	44,44	konteksty	RandomTree	**54,55**
pochodzenie	14,97	57,14	41,27	łączony	BFTree	**62,22**
położenie	18,52	26,00	29,58	konteksty	BLR	**33,46**
przynależność	20,52	28,90	46,12	konteksty	DecisionTable	**52,73**
sąsiedztwo	3,88	10,53	**33,33**	konteksty	DecisionTable	15,38
tożsamość	3,25	24,35	**46,03**	łączony	BFTree	45,87

Tabela 5.12. Wyniki dla wybranych konfiguracji na zbiorze testowym razem z wynikami referencyjnymi

Z kolei na zbiorze testowym poprawa wyników została osiągnięta aż dla pięciu kategorii relacji. Dla kolejnych dwóch kategorii, tj. *kompozycji* i *tożsamości* osiągnięty wynik był nieznacznie gorszy od samych reguł. Spadek średniej harmonicznej wyniósł odpowiednio 1,99 i 0,16 punktu procentowego. Największa różnica została odnotowana dla

relacji *sąsiedztwo*, która jest jedną z dwóch najgorzej rozpoznawanych relacji. Metoda oparta na słowniku podtypów osiągnęła zaledwie 3,88% średniej harmonicznej. Ręcznie opracowane reguły na zbiorze pomocniczym nie rozpoznały żadnej relacji. Natomiast na zbiorze testowym osiągnęły tylko 11,76% średniej harmonicznej. Podobne wyniki zostały osiągnięte po drugiej iteracji tworzenia reguł. Wszystkie te obserwacje mogą świadczyć o dużej różnorodności przykładów tej relacji pomiędzy zbiorem uczącym, pomocniczym i testowym, co przekłada się na słabe wyniki dla metod nadzorowanych.

5.5.4. Ocena jakościowa

Ostatnim elementem oceny jakości rozpoznawania relacji między jednostkami identyfikacyjnymi był test jakościowy. Do testu został użyty korpus CEN (zob. 3.4.3). Ponieważ korpus nie był znakowany relacjami semantycznymi, ocenie została poddana jedynie precyzja rozpoznawania relacji. Do rozpoznania relacji zostały użyte najlepsze modele klasyfikatorów przedstawione w tabeli 5.12. Następnie lista rozpoznanych relacji została przedstawiona lingwiście do oceny. Dla każdej rozpoznanej relacji zostało podane następujące informacje:

- kategoria rozpoznanej relacji,
- treść i typ jednostki źródłowej,
- treść i typ jednostki docelowej,
- zdanie, w którym rozpoznano relację.

Zadaniem lingwisty było odrzucenie tych propozycji, dla których relacja została błędnie rozpoznana. Dodatkowym kryterium był wybór tylko tych propozycji, dla których istnienie relacji wynikało wyłącznie z przedstawionego zdania. Oznacza to, że jeżeli zaproponowana relacja była poprawna, ale nie wynikała z podanego zdania, to taka propozycja była także odrzucana. To kryterium wynika z założenia przyjętego na początku, że wystąpienie relacji musi być poparte pewnymi przesłankami w zdaniu.

Poniżej znajduje się fragment pliku zawierający opis rozpoznanych relacji:

```
1  creator ;
2    Fundacji Wikimedia ; organization_nam ;
3    Wikiźródła         ; event_nam ;
4
5    Podobne przeznaczenie ma projekt Fundacji Wikimedia - Wikiźródła ,
6    które zbierają teksty , do których wygasły prawa autorskie
7    lub są dostępne na wolnej licencji .
8
9  location ;
10   Giełdzie Papierów Wartościowych ; company_nam ;
11   Warszawie                       ; city_nam ;
12
13   Kolejny udany debiut na Giełdzie Papierów Wartościowych w Warszawie .
14
```

```
15 nationality ;
16   Neil Armstrong ; person_nam ;
17   Amerykanie     ; nation_nam ;
18
19   Dokładnie 20 lipca 1969 , czyli dokładnie 40 lat temu Amerykanie
20   Neil Armstrong i Edwin Aldrin dokonali pierwszej udanej próby
21   lądowania na obcym obiekcie kosmicznym .
```

Analiza otrzymanych wyników wykazała, że znaczna część negatywnych relacji pochodzi ze zdań, które powstały ze złożenia w jeden ciąg częściowo uporządkowanych elementów, takich jak wypunktowania. Poniżej znajduje się przykładowe zdanie, które powstało ze złożenia wypunktowania:

> Zamknięte reaktory to : * Phénix - francuski prototypowy reaktor na prędkie neutrony , o mocy elektrycznej 233 MW * litewska Ignalina - 2 , produkująca 1185 MW energii elektrycznej , zamknięta zgodnie z postanowieniami traktatu akcesyjnego Litwy do Unii Europejskiej Nowe reaktory to reaktory elektrowni jądrowych , włączone do sieci energetycznej w grudniu 2009 : * Tomari - 3 , 868 MW mocy elektrycznej , Japonia * Radżastan - 5 , 220 MW mocy elektrycznej , Indie Podwyższenie mocy istniejących reaktorów skutkowało wzrostem łącznej mocy elektrycznej o 808 MW .

Tego typu „zdania" są trudne do analizy i są bardzo podatne na błędy, ponieważ wymagają wcześniejszego rozpoznania ich struktury. Ponieważ jednym z założeń było rozpoznawanie relacji w ciągłym tekście, to tego typu zdania zostały usunięte z wyników. W efekcie precyzja uległa nieznacznej poprawie.

Relacja	**TP**	**FP**	**P**	**TP**	**FP**	**P**
	Wszystkie zdania			**Odfiltrowane zdania**		
autorstwo	25	42	37,31%	24	17	58,54%
kompozycja	16	11	59,26%	15	9	62,50%
narodowość	56	47	54,37%	50	19	72,46%
pochodzenie	26	39	40,00%	25	38	39,68%
położenie	178	229	43,73%	175	182	49,02%
przynależność	250	108	69,83%	241	95	71,73%
sąsiedztwo	9	2	81,82%	9	2	81,82%
tożsamość	44	103	29,93%	44	103	29,93%
Wszystkie	604	581	50,97%	583	465	55,63%

Tabela 5.13. Wynik jakościowej oceny rozpoznawania relacji

Rozważane 8 kategorii relacji semantycznych zostało rozpoznanych z precyzją ponad 55%, co odpowiada wynikom osiąganym przez metody wektorowe dla języka angielskiego oceniane na ogólnych zbiorach danych (od 37,9% do 63,5%; zob. tabela 2.2).

Mimo to trzeba mieć na uwadze, że bezpośrednie porównanie nie daje pełnego obrazu, ponieważ ocenie nie została poddana kompletność.

Najwyższa precyzja została osiągnięta dla relacji *sąsiedztwo* na poziomie 81%, ale jednocześnie zostało rozpoznanych najmniej wystąpień — zaledwie 9, co może świadczyć o potencjalnie niskiej kompletności. Wysoka precyzja przy jednocześnie dużej liczbie rozpoznanych relacji została uzyskana dla relacji *przynależność* i wyniosła 71% dla 241 poprawnych wystąpień.

5.6. Podsumowanie

W rozdziale została przedstawiona nadzorowana metoda rozpoznawania relacji semantycznych między jednostkami identyfikacyjnymi wykorzystująca metody maszynowego uczenia. Zaproponowana metoda składa się z dwóch etapów.

W pierwszym etapie wykorzystując paradygmat indukcyjnego programowania logicznego generuje się zbiór reguł rozpoznających relacje. Reguły tworzone są dla trzech modeli predykatów odzwierciedlających różne informacje istotne dla zadania rozpoznawania relacji, którymi są: słowa kluczowe, konteksty jednostek oraz zależności składniowe między tokenami.

W drugim etapie zbiór wygenerowanych reguł zostaje wykorzystany jako cechy dla klasyfikatora wektorowego. Najlepsza konfiguracja klasyfikatora zostaje wybrana na podstawie zbioru pomocniczego, tj. zostaje wybrana konfiguracja, która osiągnęła najlepszą średnią harmoniczną na zbiorze pomocniczym. Ostateczna jakość wybranych klasyfikatorów została przetestowana na zbiorze testowym.

Wyniki otrzymane na zbiorze testowym zostały porównane z wynikami otrzymanymi dla dwóch metod bazowych, tj. metody opartej na heurystyce znakującej wszystkie relacje zgodnie ze słownikiem podkategorii oraz metody opartej na znakowaniu relacji za pomocą ręcznie opracowanych reguł. Porównanie wyników wykazało, że dla wszystkich kategorii relacji wyniki osiągnięte przez metody nadzorowane były nie gorsze niż wyniki osiągnięte przez metody bazowe. Oznacza to, że metody nadzorowanego uczenia mogą skutecznie zastąpić lub wspomóc proces ręcznego tworzenia reguł. Dodatkowo klasyfikator relacji wykorzystujący automatycznie wygenerowane reguły jako cechy był znacząco lepszy od samych reguł dla pięciu kategorii relacji, dla dwóch kolejnych osiągnął bardzo zbliżony wynik, a tylko dla jednej osiągnął zdecydowanie gorszy wynik. Kategoria relacji, dla której został osiągnięty znacząco gorszy wynik, była bardzo słabo rozpoznawana przez heurystykę i ręcznie opracowane reguły, co może świadczyć o dużej różnorodności przykładów, a tym samym zbyt małym zbiorze uczącym.

Na koniec została przeprowadzona ocena jakościowa wybranych modeli rozpoznawania relacji na korpusie CEN (wiadomości gospodarcze z portalu Wikinews). Ocenie została poddana precyzja rozpoznawania relacji, która dla wszystkich kategorii wyniosła ponad 55%, co jest wynikiem zbliżonym dla metod wektorowych osiąganych dla języka angielskiego na zbiorze ACE. Z kolei pod względem średniej harmonicznej, która

została obliczona na korpusie testowym, dla 4 z 8 kategorii relacji osiągnięto wyniki na poziomie wyników osiąganych dla języka angielskiego, czyli powyżej 45%.

Pomimo że uzyskane wyniki są na poziomie wyników uzyskanych dla języka angielskiego, to nie są one zadowalające i pozostawiają duże pole do ulepszenia metody. Jedną z możliwości jest rozszerzenie modeli predykatów o dodatkowe informacje, które zwiększą powtarzalność wzorców, np. skrócenie ścieżek zależności między elementami w zdaniu (zob. przykład na wydruku 2.3), dodanie informacji o frazach składniowych i relacjach między nimi (jako forma pomijania sekwencji tokenów).

Kolejną możliwością jest opracowanie dodatkowych cech opisujących pary jednostek, które mogą być wykorzystane na etapie klasyfikacji. Na przykład jednym z częstych błędów są niepoprawnie rozpoznane relacje w zdaniach zawierających dużą liczbę anotacji niepowiązanych ze sobą. Takie błędy wynikają z reguł, które mają postać dwóch niepowiązanych ze sobą wzorców jednostek. Jedną z możliwość wyeliminowania tego typu błędów byłoby opracowanie cech pozwalających na uchwycenie odległości między jednostkami, np. jako odległości w drzewie zależności lub odległości we frazach rzeczownikowych.

Dobór i ilość danych uczących jest także bardzo ważnym elementem w metodach nadzorowanego uczenia. Opracowane modele mogłyby być wykorzystane do rozszerzenia istniejących korpusów o nowe przykłady, zarówno pozytywne jak i negatywne, z dużym naciskiem na przykłady dla rzadkich podkategorii.

Rozdział 6

Zastosowanie ekstrakcji informacji w systemie odpowiedzi na pytania

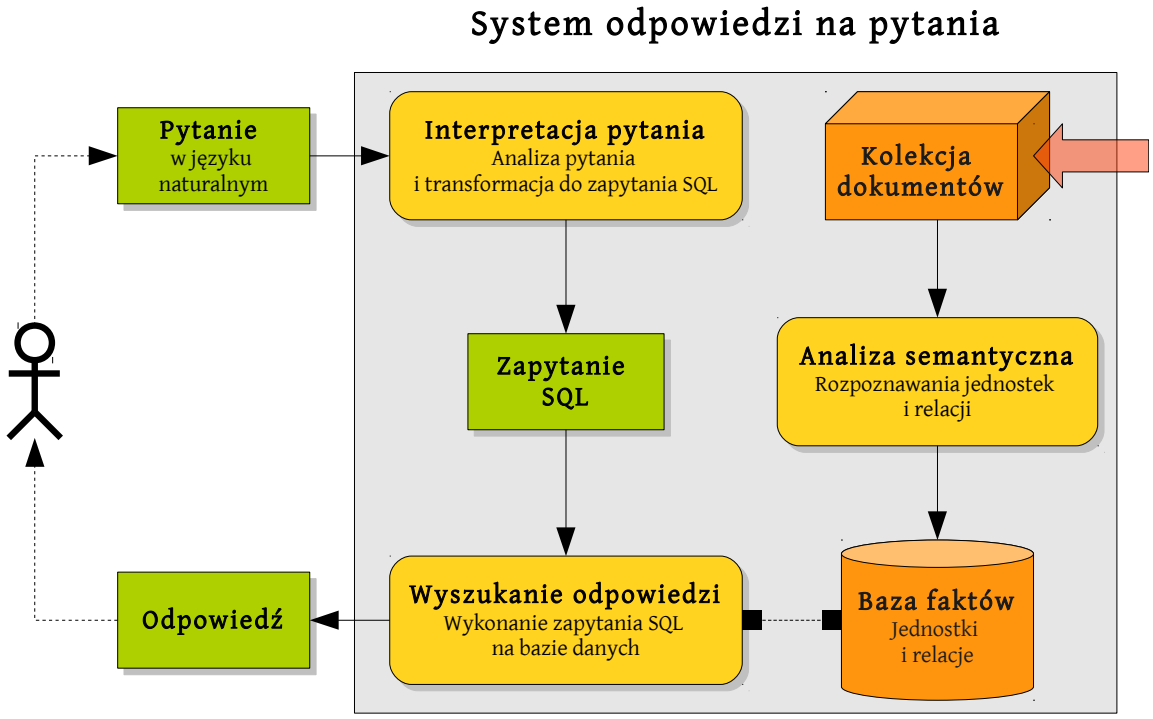
Jednym z praktycznych zastosowań narzędzi do rozpoznawania relacji semantycznych między jednostkami identyfikacyjnymi są systemy odpowiedzi na pytania (ang. *Question Answering*; QA). Systemy QA, w odróżnieniu od wyszukiwarek internetowych takich jak Google czy Bing, pozwalają na udzielenie konkretnej odpowiedzi na pytanie zadane w języku naturalnym. Wyróżnia się dwa podejścia do konstrukcji systemów QA: wykorzystanie treści pytania do znalezienia odpowiedzi lub transformacja pytania do pewnego języka formalnego i wyciągnięcie informacji z bazy wiedzy — koncepcja odpowiedzi na pytania, zanim zostaną zadane (Fleischman *et al.*, 2003). Narzędzia do rozpoznawania relacji mogą być wykorzystane w drugim podejściu do zaindeksowania wystąpień relacji w analizowanych dokumentach.

W tym rozdziale przedstawiony zostanie prototyp systemu odpowiedzi na pytania o relacje semantyczne zachodzące między jednostkami identyfikacyjnymi. W punkcie 6.1 opisana zostanie architektura systemu składającego się z dwóch modułów: moduł rozpoznawania i indeksowania jednostek identyfikacyjnych i relacji między nimi oraz modułu do analizy i transformacji pytań zadanych w języku naturalnym do sformalizowanego zapisu, czyli zapytania SQL pozwalającego na wyciągnięcie szukanej informacji bezpośrednio z bazy danych. W dalszej części rozdziału przedstawiona zostanie procedura analizy pytań składająca się z dwóch etapów: półautomatycznego generowania szablonów transformacji (sekcja 6.3.1) oraz analizy i transformacji pytania do postaci zapytania SQL (sekcja 6.3.2). W ostatniej sekcji 6.6 zostanie omówionych kilka przypadków użycia wraz z porównaniem otrzymanych wyników z wynikami zwracanymi przez istniejące wyszukiwarki wektorowe i semantyczne.

6.1. Architektura systemu

Prototyp systemu odpowiedzi na pytania o relacje semantyczne między jednostkami identyfikacyjnymi składa się z dwóch potoków przetwarzania: rozpoznawania i indekso-

wania relacji (sekcja 6.2) oraz analiza pytania (sekcja 6.3). Poglądowy schemat blokowy systemu został przedstawiony na rysunku 6.1



Rys. 6.1. Schemat blokowy prototypowego systemu odpowiedzi na pytania o relacje semantyczne między jednostkami identyfikacyjnymi.

6.2. Potok rozpoznawania i indeksowania relacji

Moduł rozpoznawania i indeksowania relacji działa niezależnie od modułu analizy pytań. Jego zadaniem jest analiza napływających dokumentów pod kątem występowania jednostek i relacji. Jeżeli w dokumencie zostaną rozpoznane jednostki i relacje między nimi, to dokument zostaje zachowany w bazie danych, a rozpoznane jednostki i relacje zostają zaindeksowane. Do przechowywania informacji o jednostkach i relacjach między nimi wykorzystane zostały dwie tabele: jedna dla jednostek, druga dla relacji, łącząca pary jednostek z pierwszej tabeli.

6.3. Potok analizy pytań

Zadaniem potoku analizy pytań jest transformacja pytania w języku naturalnym do postaci zapytania SQL pozwalającego na wyciągnięcie z bazy danych odpowiedzi na zadane pytanie. Analiza pytania została podzielona na dwa etapy:

1. **Generowanie** szablonów transformacji — etap przygotowawczy wykonywany jednorazowo. Jego wynikiem jest zestaw operatorów WCCL umożliwiających dopasowanie istotnych elementów pytania. Opis tego etapu znajduje się w sekcji 6.3.1.
2. **Interpretacja** pytania i wygenerowanie odpowiedniej kwerendy SQL. Operacja wykonywana jest dla każdego pytania w oparciu o wcześniej wygenerowany zestaw szablonów transformacji. Opis tego etapu znajduje się w sekcji 6.3.2.

6.3.1. Generowanie szablonów

Celem tego etapu jest konstrukcja szablonów transformacji będących podstawą do analizy pytań i generowania odpowiadających im zapytań SQL. Jest to proces częściowo zautomatyzowany, który zaczyna się od ręcznego przygotowania szablonów pytań oraz definicji szablonów kwerend SQL. Następnie, w sposób automatyczny, każdy szablon pytania transformowany jest do postaci reguły WCCL[1] zgodnie z opracowaną procedurą. Podczas transformacji wykorzystywany jest analizator morfologiczny *maca* (Radziszewski i Śniatowski, 2011a), tager WMBT (Radziszewski i Śniatowski, 2011b) oraz Słowosieć (Piasecki *et al.*, 2009). Algorytm transformacji szablonu pytania do postaci reguły WCCL przebiega w następujących krokach:

1. Pytanie zostaje podzielone na tokeny i otagowane przy pomocy WMBT.
2. Dla każdego tokenu, który nie jest meta-symbolem (zob. dalej pkt. 3) zostaje wygenerowany operator WCCL będący warunkiem dla atrybutu `base`, tj.:

```
inter( base[0], ["<base>"] )
```

3. W pytaniu mogą pojawić się następujące meta-symbole:
   - `[PN|nazwa]` — dopasowanie do dowolnej nazwy własnej,
   - `[PN:typ|nazwa]` — dopasowanie do nazwy własnej wskazanego typu,
   - `[SYN:słowo#n]` — synonimy słowa o numerze znaczenia n, np.

```
[SYN:kraj#1] = ["kraj", "państwo"]
[SYN:kraj#2] = ["państwo", "kraj", "kraina", "ziemia"]
```

   - `[SYN:słowo@dziedzina]` — synonimy słowa z synsetów przypisanych do wskazanej dziedziny, np.

```
[SYN:kraj@msc] = ["państwo", "kraj", "kraina", "ziemia",
                  "obrzeże", "skraj"]
```

   Interpretacja meta-symboli:
   - `[PN|nazwa]` — generuje dopasowanie do sekwencji oznaczonej jako `proper_name` w postaci:

```
is( "proper_name" )
```

1. Formalizm języka WCCL do zapisu reguł znakujących fragmenty tekstu został przedstawiony w dodatku D.

Dodatkowo generowany jest operator oznaczający dopasowany element jako `nazwa`.

- `[PN:type|nazwa]` — generuje dopasowanie do anotacji kategorii `type`. Na przykład dla `[PN:person_nam]` wygenerowany operator będzie miał postać:

```
is( "person_nam" )
```

Dodatkowo generowany jest operator oznaczający dopasowany element jako `nazwa`.

- `[SYN:słowo#n]` — generuje regułę pomocniczą, która znakuje wszystkie synonimy danego słowa, np. dla `[SYN:narodowość#1]` zostanie wygenerowana następująca reguła:

```
/* Synonimy słowa narodowość_1 */
apply(
  match(
    inter( base[0], ["narodowość", "nacja"] )
  ),
  actions(
    mark( :1, "syn_narodowość_1" )
  )
);
```

oraz wygeneruje operator w postaci:

```
is( "syn_słowo_n" ),
```

czyli dla powyższego przykładu będzie:

```
is( "syn_narodowość_1" ),
```

- `[SYN:słowo@dziedzina]` — analogicznie do powyższego meta-symbolu, z tą różnicą, że pobierany jest inny zestaw słów, tj. słowa będące synonimami w danej dziedzinie, np. dla `[SYN:kraj@grp]` otrzymamy regułę pomocniczą:

```
/* Synonimy słowa kraj_grp */
apply(
  match(
    inter( base[0], ["kraj", "państwo"] )
  ),
  actions(
    mark( :1, "syn_kraj_grp" )
  )
);
```

i operator:

```
is( "syn_kraj_grp" ),
```

Dla elementów `[PN|nazwa]` i `[PN:type|nazwa]` zostanie wygenerowana akcja znakująca dopasowanie, tj.:

```
actions(
  mark( :n, "nazwa" )
)
```

gdzie n jest indeksem elementu oznaczonego jako `[PN]`. Indeks zależy od liczby warunków, jakie zostaną wygenerowane w regule WCCL.

Na wydrukach 6.1 i 6.2 znajduje się przykładowa reguła WCCL wraz z regułami pomocniczymi, wygenerowanymi dla następującego szablonu pytania:

`Skąd [SYN:pochodzić@cst] [SUBST@os]? [PN:person_nam|arg_person_nam] ?`

```
/* Skąd [SYN:być@cst] [SUBST@os]? [PN:person_nam|arg_person_nam] ?*/
apply(
  match(
    inter(base[0], ["skąd"]),
    is("syn_być_cst"),
    optional(
      is("subst_os")),
    is("person_nam"),
    inter(base[0], ["?"])
  ),
  actions(
    mark(:4, "arg_person_nam")
  )
)
```

Wydruk 6.1. Przykładowa reguła WCCL rozpoznająca argument w pytaniu o pochodzenie osoby.

```
/* Synonimy słowa być */
apply(
  match(
    oneof(
      variant( inter( base[0], ["znajdować"] ),
               inter( base[0], ["się"] ) ),
      variant( inter( base[0], ["odbywać"] ),
               inter( base[0], ["się"] ) ),
      variant( inter( base[0], ["dziać"] ),
               inter( base[0], ["się"] ) ),
      variant( inter( base[0], ["miejsce"] ) ),
      variant( inter( base[0], ["toczyć"] ),
               inter( base[0], ["się"] ) ),
      variant( inter( base[0], ["istnieć"] ) ),
      variant( inter( base[0], ["iść"] ) ),
      variant( inter( base[0], ["egzystować"] ) ),
      variant( inter( base[0], ["to"] ) ),
      variant( inter( base[0], ["przebiegać"] ) ),
      variant( inter( base[0], ["być"] ) ),
      variant( inter( base[0], ["postępować"] ) )
```

```
    )
  ),
  actions(
    mark( :M, "syn_być_cst" )
  )
);

/* Rzeczowniki z dziedziny os (wybrane przykłady)*/
apply(
  match(
    oneof(
      variant( inter( base[0], ["drugi"] ),
               inter( base[0], ["oficer"] ) ),
      variant( inter( base[0], ["kawał"] ),
               inter( base[0], ["chłop"] ) ),
      variant( inter( base[0], ["syn"] ),
               inter( base[0], ["pierworodny"] ) ),
      variant( inter( base[0], ["diakon"] ) ),
      variant( inter( base[0], ["ciotuchna"] ) ),
      variant( inter( base[0], ["ordynat"] ) ),
      variant( inter( base[0], ["patriarcha"] ) )
  ),
  actions(
    mark( M, "subst_os" )
  )
);
```

Wydruk 6.2. Zbiór reguł pomocniczych dla reguły z wydruku 6.1.

6.3.2. Interpretacja pytania

Właściwym etapem analizy pytania jest jego interpretacja w oparciu o wcześniej wygenerowane szablony transformacji. Interpretacja pytania przebiega w następujących krokach:

1. Pytanie zostaje podzielone na tokeny i otagowane przy pomocy WMBT.
2. Przy użyciu narzędzia Liner2 w pytaniu zostają rozpoznane nazwy własne.
3. Na pytaniu zostają wykonane wszystkie operatory WCCL z pliku transformacji. Jeżeli reguła WCCL zostanie uruchomiona, tj. zostanie wykonany operator z sekcji `actions`, to dla danego szablonu transformacji zostaje wygenerowana kwerenda SQL. Proces generowania kwerend został opisany w sekcji 6.3.3.

6.3.3. Wypełnianie szablonu kwerendy SemQL

Wygenerowanie kwerendy SQL polega na wypełnieniu szablonu kwerendy znajdującego się w sekcji `<semql>...</semql>` wartościami oznaczonymi przez operator WCCL. Elementy szablonu, które wymagają podmiany, oznaczone są nawiasami klamrowymi. Wewnątrz nawiasów klamrowych podana jest nazwa argumentu,

np. `{arg_person_nam}` oznacza, że w tym miejscu powinien zostać wstawiony tekst oznaczony przez regułę WCCL jako `arg_person_nam`.

```
<questions>
  <question>
    <meta>
      <name>nationality</name>
      <description>Zapytanie o narodowość osoby.</description>
    </meta>
    <templates>
      <template>
        <q>Jakiej [SYN:narodowość@grp] jest [PN:person_nam]?</q>
        <wccl>
          match_rules(
            apply(
              match(
                inter( base[0], ["jaki"] ),
                inter( base[0], ["narodowość"] ),
                inter( base[0], ["być"] ),
                is( "person_nam" )
              ),
              actions(
                mark( :4, "arg_person_nam")
              )
            )
          )
        </wccl>
      </template>
    </templates>
    <semql>
      SELECT ant.text
        FROM relations r
        JOIN annotations ans
          ON r.annotation_source_id = ans.annotation_id
        JOIN annotations ant
          ON r.annotation_target_id = ant.annotation_id
        JOIN annotation_types ans_type
          ON ans.annotation_type_id = ans_type.annotation_type_id
        JOIN annotation_types ant_type
          ON ant.annotation_type_id = ant_type.annotation_type_id
        JOIN relation_types r_type
          ON r.relation_type_id = r_type.relation_type_id
        WHERE ans.text = "Adam Małysz"
```

```
          AND ant_type.type = "country_nam"
          AND ans_type.type = "person_nam"
          AND r_type.type = "origin"
        GROUP BY ans.text
    </semql>
  </question>
</questions>
```

Wydruk 6.3. Przykładowy szablon transformacji pytania

W powyższym szablonie transformacji znajduje się opis jednego szablonu kwerendy SQL. Szablon ten zostanie wygenerowany, jeżeli dla danego pytania po wykonaniu reguły WCCL zostanie wstawiony znacznik `arg_person_nam`. W miejsce `{arg_person_nam}` zostanie wstawiona forma bazowa anotacji `arg_person_nam`. Ponieważ niedostępne jest narzędzie pozwalające na ustalenie formy bazowej dla dowolnej frazy, to za formę bazową przyjmuje się konkatenację form bazowych dla kolejnych słów z frazy.

Na przykład dla pytania *Skąd pochodzi Adam Małysz?* tekst *Adam Małysz* zostanie oznaczony jako `arg_person_nam` i zostanie wygenerowana następująca kwerenda SQL:

```
SELECT ant.text
  FROM relations r
  JOIN annotations ans ON r.annotation_source_id = ans.annotation_id
  JOIN annotations ant ON r.annotation_target_id = ant.annotation_id
  JOIN annotation_types ans_type
    ON ans.annotation_type_id = ans_type.annotation_type_id
  JOIN annotation_types ant_type
    ON ant.annotation_type_id = ant_type.annotation_type_id
  JOIN relation_types r_type
    ON r.relation_type_id = r_type.relation_type_id
  WHERE ans.text = "Adam Małysz"
    AND ant_type.type = "country_nam"
    AND ans_type.type = "person_nam"
    AND r_type.type = "origin"
  GROUP BY ans.text
```

Wydruk 6.4. Zapytanie SQL zwracające listę nazw miast znajdujących się w Polsce.

Jeżeli szablon kwerendy SQL zawiera kilka argumentów, to wszystkie muszą zostać oznaczone w analizowanym pytaniu, aby możliwe było wygenerowanie kwerendy. Jeżeli będzie brakowało przynajmniej jednego argumentu, to kwerenda nie zostanie wygenerowana.

6.4. Transformacja pytań w oparciu o częściowe dopasowanie

Dotychczas przedstawiona procedura analizy pytań opierała się o pełne dopasowanie szablonów pytań. Takie rozwiązanie pozwala na bardzo precyzyjną analizę pytań, ale jednocześnie nie gwarantuje dużego pokrycia różnych form pytań. Aby rozszerzyć zakres pokrywanych pytań, bez konieczności opracowania kolejnych szablonów pytań, została wdrożona procedura częściowego dopasowania pytań. Dodatkowo dzięki częściowemu dopasowywaniu pytań możliwa będzie częściowa redukcja błędów powstałych na poziomie rozpoznawania nazw własnych. Częściowe dopasowanie uwzględnia następujące sytuacje:

- niepoprawne rozpoznanie kategorii nazw własnych — przypisanie nieprawidłowej kategorii nazwy własnej powoduje brak dopasowania operatora `is("kategoria")`,
- brak rozpoznania nazw własnych — sytuacja analogiczna do powyższej, z tym, że żadna nazwa nie jest rozpoznana,
- różne formy zapisu pytania, który uwzględnia zmienny szyk wyrazów, użycie synonimów, dodatkowe słowa.

Poszczególne aspekty uogólniania zostały omówione w kolejnych podpunktach.

6.4.1. Uogólnienie kategorii nazw własnych

Wszystkie szczegółowe kategorie nazw własnych występujące w regułach dopasowania zostają uogólnione do kategorii `proper_name`. Na etapie dopasowania szablonów, jeżeli żadna z reguł nie zostanie dopasowana, to wykonywana jest kolejna iteracja, w której wszystkie warunki dopasowujące szczegółowe kategorie nazw własnych zostają zamienione na kategorię ogólną `proper_name`.

6.4.2. Rozpoznanie potencjalnych nazw własnych i pełne dopasowanie

Jeżeli moduł Liner2 nie rozpozna żadnej nazwy własnej w analizowanym pytaniu, to warunek dopasowujący nazwy własne zastąpiony jest warunkiem dopasowującym sekwencję będącą potencjalną nazwą własną. Dopasowanie do potencjalnej nazwy własnej ma postać:

```
repeat( is("proper_name_candidate") )
```

Reguła znakująca potencjalne nazwy własne jest umieszczona w pomocniczym pliku z regułami `transformations-common.ccl`, który jest wykonywany przed regułami pytań. Reguła pomocnicza znakująca `proper_name_candidate` została przedstawiona na wydruku 6.5.

```
// Znakuje potencjalne nazwy własne
apply(
  match(
    regex( orth[0], "\\p{Lu}+.*" )
  ),
```

```
    actions(
      mark( M, "proper_name_candidate")
    )
  )
```

Wydruk 6.5. Reguła znakująca fragmenty potencjalnych nazw własnych

6.4.3. Miara podobieństwa między szablonem a pytaniem

Uwzględnienie nadmiarowych elementów w pytaniu zostało obsłużone poprzez częściowe dopasowanie pytania do szablonów pytań. Podobieństwo między pytaniem a szablonem pytania zostaje obliczone przy pomocy modelu wektorowego i miary kosinusowej między wektorami reprezentującymi pytanie i szablon (miara kosinusowa ma praktyczne zastosowanie w wielu zadaniach (zob. Manning i Schütze, 1999). Wymiar wektora jest równy liczbie warunków zdefiniowanych w szablonie powiększony o liczbę elementów w pytaniu niedopasowanych do żadnej reguły w szablonie. Każdy niedopasowany token liczony jest jako jedna przestrzeń wymiarowa wektora.

Analizowane pytanie zostaje porównane ze wszystkimi szablonami pytań. Dla każdego szablonu obliczona zostaje miara podobieństwa, na podstawie której tworzony jest ranking najbardziej podobnych szablonów transformacji. Warunkiem koniecznym do umieszczenia szablonu transformacji w rankingu jest dopasowanie wszystkich argumentów szablonu. W pierwszej iteracji nazwy własne muszą zostać dopasowane z dokładnością do określonego typu. Jeżeli żadna reguła transformacji nie spełnia tego warunku, to warunek na kategorię nazwy własnej zostaje uogólniony do kategorii `proper_name`.

W wyniku znajdują się reguły, których podobieństwo przekroczyło określony próg n. Jeżeli w rankingu znajdzie się tylko jedna reguła, to jest ona traktowana jako reguła właściwa. Jeżeli w wyniku znajdzie się kilka reguł, to wszystkie reguły zostają zwrócone jako możliwe interpretacje pytania.

Procedura częściowego dopasowania

1. Zakładając, że szablon S zawiera regułę R, która składa się z sekwencji warunków W (operatory zdefiniowane bezpośrednio w operatorze match), to dla każdego $w \in W$ tworzona jest reguła pomocnicza składająca się tylko z warunku w. Wynikiem dopasowania jest oznaczenie dopasowanego elementu pomocniczą etykietą e_n, gdzie n to indeks warunku w w regule W.
2. Wykonywane jest drugie dopasowanie analogicznie do punktu 1 z tą różnicą, że wszystkie warunki `is` zostają zastąpione następującym warunkiem:

   ```
   repeat( is("proper_name_candidate") )
   ```

   Jeżeli wybór argumentów jest niejednoznaczny, to argumenty dopasowywane są od końca zdania. Jest to poparte obserwacją, że argumenty pytania częściej pojawiają się na końcu pytania niż na początku.

3. Dla wyniku dopasowania z punktu 1 i 2 wybierane są te dopasowania, dla których zostały ustalone wszystkie argumenty.
4. Dla wszystkich wybranych dopasowań wykonywane są następujące operacje:
   a) Korzystając z etykiet e_n liczone jest, ile warunków z szablonu zostało dopasowanych (p – liczba pasujących), ile nie zostało (nw – liczba niedopasowanych warunków) oraz ile tokenów nie zostało dopasowanych do żadnego warunku (nt – liczba niedopasowanych tokenów).
   b) Dla pytania (v_p) i szablonu (v_s) tworzone są wektory wartości. Każdy z wektorów ma wymiar równy w, gdzie $w = p + nw + nt$. Na każdej pozycji w wektorze znajduje się wartość 0 lub 1, zgodnie z następującymi warunkami:
      - dla pozycji $n = 1, .., p$ ustaw $v_p[n] = 1$ i $v_s[n] = 1$,
      - dla pozycji $n = p + 1, .., p + nw$ ustaw $v_p[n] = 0$ i $v_s[n] = 1$,
      - dla pozycji $n = p + nw + 1, .., p + nw + nt$ ustaw $v_p[n] = 1$ i $v_s[n] = 0$.
   c) Między wektorami v_p i v_s liczony jest cosinus kąta, który jest miarą podobieństwa pytania P do szablonu S.
5. Wybierane są te szablony, które uzyskały najwyższe podobieństwo wyrażone jako cosinus kąta między wektorem pytania a szablonem pytania.

6.4.4. Ocena

Przedstawiona metoda interpretacji pytań została przetestowana na zbiorze 114 pytań. Wyniki oceny zostały przedstawione w tabeli 6.1. Ocena została dokonana na czterech poziomach analizy pytań, tj.:

1. **Kategoria relacji** — poprawność klasyfikacji pytania do jednej z ośmiu kategorii relacji semantycznych (zob. podpunkt 3.2.2).
2. **Podkategoria relacji** — kategorie jednostki źródłowej i docelowej.
3. **Kategoria obiektu** — kategoria jednostki, której dotyczy pytanie.
4. **Argument pytania** — rozpoznanie nazwy własnej będącej argumentem pytania.

Na przykład dla pytania *Z jakiego kraju pochodzi Adam Małysz?* kategoria relacji to *pochodzenie*, podkategoria to *osoba-państwo*, kategoria obiektu to *nazwa państwa* i argument pytania to *Adam Małysz*.

Na poziomie kategorii relacji, analiza pytań osiągnęła wynik prawie 98%, co jest wynikiem bardzo dobrym. Najsłabsze wyniki zostały osiągnięte dla rozpoznawania argumentów pytania. Głównym powodem jest problem z prawidłowym rozpoznawaniem nazw własnych. Drugą przyczyną jest dopasowywanie pytań do niewłaściwych szablonów, co automatycznie uznawane jest za nieprawidłowe dopasowanie argumentu.

6.5. Interfejs

Interfejs prototypowego systemu Serel składa się z trzech elementów (zob. rysunek 6.2). Są to:

1. Pole do wprowadzenia pytania w języku naturalnym,

	P	R	F
Kategoria relacji	96,83	98,39	97,60
Podkategoria relacji	85,71	93,91	89,63
Kategoria obiektu	84,92	93,04	88,80
Wartość argumentu	81,75	89,57	85,48

Tabela 6.1. Ocena skuteczności interpretacji pytań.

Rys. 6.2. Interfejs prototypowego systemu odpowiedzi na pytania o relacje semantyczne.

2. Okno z listą znalezionych odpowiedzi wraz z liczbą wystąpień w nawiasie,
3. Okno ze zdaniem, na podstawie którego udzielono danej odpowiedzi (po wyborze odpowiedzi z okna 2).

Dodatkowo interfejs zawiera ukryte pole zawierające pełną interpretację pytania. Informacja jest dostępna na potrzeby diagnostyczne.

6.6. Porównanie z istniejącymi systemami

Opracowany prototyp systemu ekstrakcji informacji został przetestowany na kilku przykładowych pytaniach, a otrzymane wyniki zostały porównane z wynikami zwracanymi przez istniejące wyszukiwarki internetowe i systemy odpowiedzi na pytania dla języka polskiego. Pytania testowe były pytaniami o nazwy obiektów będących w określonej relacji ze wskazanym obiektem identyfikowanym także nazwą własną.

Interpretacja (pokaż)

Pytanie:	**Jakie miasta leżą w Polsce ?**
Pewność:	0.77151674981
Typ relacji:	location
Podtyp relacji:	city_nam-country_nam
Pytanie o typ obiektu:	city_nam
Argument:	arg_country_nam=Polsce

Zapytanie SQL:	`SELECT ans.text`
	`FROM relations r`
	`JOIN annotations ans ON r.annotation_source_id = ans.annotation_id`
	`JOIN annotations ant ON r.annotation_target_id = ant.annotation_id`
	`JOIN annotation_types ans_type ON ans.annotation_type_id = ans_type.annotation_type_id`
	`JOIN annotation_types ant_type ON ans.annotation_type_id = ant_type.annotation_type_id`
	`JOIN relation_types r_type ON r.relation_type_id = r_type.relation_type_id`
	`WHERE ant.text = 'Polsce'`
	`  AND ant_type.type = 'country_nam'`
	`  AND ans_type.type = 'city_nam'`
	`  AND r_type.type = 'location'`
	`GROUP BY ans.text`

Rys. 6.3. Interpretacja przykładowego pytania.

Oczekiwaną odpowiedzią jest zbiór nazw własnych wraz ze zdaniem potwierdzającym rozpoznanie relacji.

Baza wiedzy wyszukiwarki Serel zawierała 3,5 tys. przetworzonych artykułów z polskiej wersji Wikipedii wraz z zaindeksowanymi jednostkami identyfikacyjnymi i relacjami między nimi. Do rozpoznania relacji został użyty Liner2 z modelem do rozpoznawania 56 kategorii jednostek identyfikacyjnych oraz moduł do rozpoznawania relacji dla najlepszej konfiguracji opisanej w rozdziale 5.

Wyniki zwrócone przez system Serel zostały porównane z następującymi wyszukiwarkami internetowymi i systemami odpowiedzi na pytania:

Google [2] — największa wyszukiwarka internetowa. Prezentowane wyniki zostały uzyskane w dniu 30 sierpnia 2012 roku bez korzystania ze spersonalizowanego wyszukiwania.

Bing [3] — jedna z większych wyszukiwarek internetowych. Prezentowane wyniki zostały uzyskane w dniu 30 sierpnia 2012 roku (wyniki wyszukiwania mogą ulec zmianie w czasie).

KtoCo.pl [4] — wyszukiwarka semantyczna dla języka polskiego bazująca na wynikach zwracanych przez Google.

Hipisek [5] — wyszukiwarka semantyczna dla języka polskiego. Zawiera zaindeksowane dokumenty pochodzące z portali informacyjnych.

6.6.1. Pytanie #1: Jakie miasta znajdują się w Polsce?

Pytanie o nazwy miast znajdujące się na terenie kraju o podanej nazwie. Oczekiwaną odpowiedzią jest lista nazw miast. Tego typu informacja może być w łatwy

2. Strona www: `http://www.google.pl`
3. Strona www: `http://www.bing.com`
4. Strona www: `http://www.ktoco.pl`
5. Strona www: `http://www.hipisek.pl`

sposób wyciągnięta z dedykowanych baz danych, np. Geonames[6]. Jednak na potrzeby tego testu zostało przyjęte, że nie istnieje taka baza wiedzy, a informacje wyciągane są z ciągłego tekstu. To założenie jest umotywowane możliwością zastosowania wyszukiwarki na dowolnym zbiorze dokumentów, np. serii książek fabularnych, w którym mogą występować fikcyjne nazwy miast i państwa.

Poniżej znajduje się ręczna ocena wyników zwróconych przez rozważane systemy.

Serel — system zwrócił 145 nazw będących potencjalnymi nazwami własnymi miast znajdujących się w Polsce. 56% propozycji było prawidłowymi nazwami miejscowości znajdującymi się w Polsce. 44% pozostałych propozycji zostało błędnie rozpoznanych. Najczęstszą przyczyną błędów było nieprawidłowe rozpoznanie kategorii jednostki identyfikacyjnej (23% wszystkich zwróconych propozycji) i niepoprawne wskazanie relacji między jednostkami (13%). Na rysunku 6.2 znajduje się zrzut ekranu przedstawiający wynik zwrócony przez system Serel.

Google — wyszukiwarka zwróciła ponad 106 milionów stron. Na 2 stronie z wynikami znalazła się pozycja wskazująca artykuł w polskiej Wikipedii zawierający ustrukturalizowaną listę miast w Polsce. Na rysunku 6.5 znajduje się zrzut ekranu przedstawiający pierwszą stronę wyników zwróconych przez wyszukiwarkę Google.

Bing — wyszukiwarka zwróciła prawie 2 miliony stron. Analogicznie do wyników zwróconych przez Google ocenie zostało poddanych pierwszych pięć stron, tj. 50 pozycji. Wśród sprawdzonych pozycji wystąpiły tylko dwie, których tytuł sugerował wystąpienie szukanej informacji. Na żadnej ze stron nie wystąpiła ustrutkuralizowana lista miejscowości. Na rysunku 6.6 znajduje się zrzut ekranu przedstawiający pierwszą stronę wyników zwróconych przez wyszukiwarkę Bing.

KtoCo.pl — wyszukiwarka zawęziła wyniki zwrócone przez Google do czterech pozycji, ale żadna z nich nie zawierała poszukiwanych informacji. Na rysunku 6.4 znajduje się zrzut ekranu przedstawiający wynik zwrócony przez wyszukiwarkę KtoCo.pl.

Hipisek — system zwrócił dwie odpowiedzi. Żadna z nich nie była odpowiednia do zadanego pytania. Na rysunku 6.7 znajduje się zrzut ekranu przedstawiający wynik zwrócony przez system Hipisek.

6.6.2. Pytanie #2: Kto należy do PiS?

Pytanie o listę nazw osób należących do partii o podanej nazwie. Oczekiwaną odpowiedzią jest lista nazw osób.

Serel — system zwrócił 22 odpowiedzi i wszystkie z nich były prawidłowe.

Google — wyszukiwarka zwróciła ponad 74 milionów stron. Ocenie zostało poddanych pięć pierwszych stron, tj. 50 pozycji. Wśród nich znalazła się tylko jedna, której tytuł sugerował wystąpienie poszukiwanej informacji. Była to strona z Wikipedii

6. Strona www: `http://www.geonames.org/`

Rys. 6.4. Zrzut ekranu przedstawiający wynik zwrócony przez wyszukiwarkę KtoCo.pl dla pytania *Jakie miasta znajdują się w Polsce?*.

poświęcona partii PiS. Zawierała ona ustrukturalizowaną listę osób należących do partii.

Bing — wyszukiwarka zwróciła ponad 0,7 miliona stron. Ocenie zostały poddanych pięć pierwszych stron, tj. 50 pozycji. Wśród nich znalazły się dwie, których tytuł sugerował poszukiwaną informację. W rzeczywistości okazały się to być listy wyborcze kandydatów partii PiS, które zawierają niepełną listę osób należących do PiS.

KtoCo.pl — wyszukiwarka zawęziła wyniki zwrócone przez Google do 10 pozycji. W przypadku dwóch z nich tytuł sugerował wystąpienie poszukiwanej informacji, ale żadna z propozycji nie zawierała ustrukturalizowanej listy osób.

Hipisek — system nie zwrócił żadnej odpowiedzi.

6.6.3. Pytanie #3: W jakim kraju leży Leeuwarden?

Pytanie o nazwę kraju, w którym znajduje się miasto o podanej nazwie. Oczekiwaną odpowiedzią jest jedna nazwa państwa.

Serel — system zwrócił 9 odpowiedzi, z których tylko jedna była prawidłowa. Pozostałe odpowiedzi wynikały z nieprawidłowego rozpoznania relacji między jednostkami identyfikacyjnymi, co można było zweryfikować przeglądając konteksty relacji.

Google — wyszukiwarka zwróciła ponad 0,7 miliona stron. Na pierwszej pozycji znalazł się link do strony Wikipedii poświęconej mieście Leeuwarden, na której znajdowała się poszukiwana odpowiedź.

Bing — wyszukiwarka zwróciła 189 stron. Wśród pierwszych 50 pozycji nie znalazła się żadna, której tytuł lub załączony fragment tekstu sugerowałby wystąpienie odpowiedzi na pytanie.

KtoCo.pl — zwróciła jedną odpowiedź w formie pojedynczego zdania, które zawierało odpowiedź na pytanie.

Hipisek — system zwrócił pięć odpowiedzi, ale żadna z nich nie była odpowiednia do pytania.

6.6.4. Pytanie #4: Do jakiej partii należy Andrzej Pęczak?

Pytanie o nazwę partii politycznej, do której należy dana osoba. Oczekiwanym wynikiem jest nazwa jednej partii, ew. kilku, jeżeli dana osoba zmieniała przynależność do partii.

Serel — system zwrócił jedną, prawidłową odpowiedź. Odpowiedź była poparta zdaniem, z którego wynikała poszukiwana informacja.

Google — wyszukiwarka zwróciła 62 tys. stron. Na pierwszej stronie z wynikami znalazła się pozycja, która była odpowiedzią na zadane pytanie. Odpowiedzią było pełne zdanie bez zaznaczonej odpowiedzi.

Bing — wyszukiwarka zwróciła 21 tys. stron. Na pierwszej stronie z wynikami znalazła się pozycja, która była odpowiedzią na zadane pytanie. Odpowiedzią było pełne zdanie bez zaznaczonej odpowiedzi.

KtoCo.pl — nie zwrócił żadnej odpowiedzi.

Hipisek — system zwrócił pięć odpowiedzi, ale żadna z nich nie była związana z zadanym pytaniem.

6.6.5. Podsumowanie

Opracowany prototyp systemu odpowiedzi na pytania o nazwie Serel wykorzystujący metody ekstrakcji informacji został przetestowany na dwóch kategoriach pytań: (1) pytania o listę nazw własnych obiektów będących w określonej relacji względem zadanego obiektu oraz (2) pytania o nazwy pojedynczych obiektów będących w określonej relacji względem zadanych nazw własnych. Wyniki otrzymane z systemu Serel zostały porównane z wynikami otrzymanymi z dwóch wyszukiwarek internetowych (Google i Bing) oraz z dwóch systemów odpowiedzi na pytania dla języka polskiego (KtoCo.pl i Hipisek).

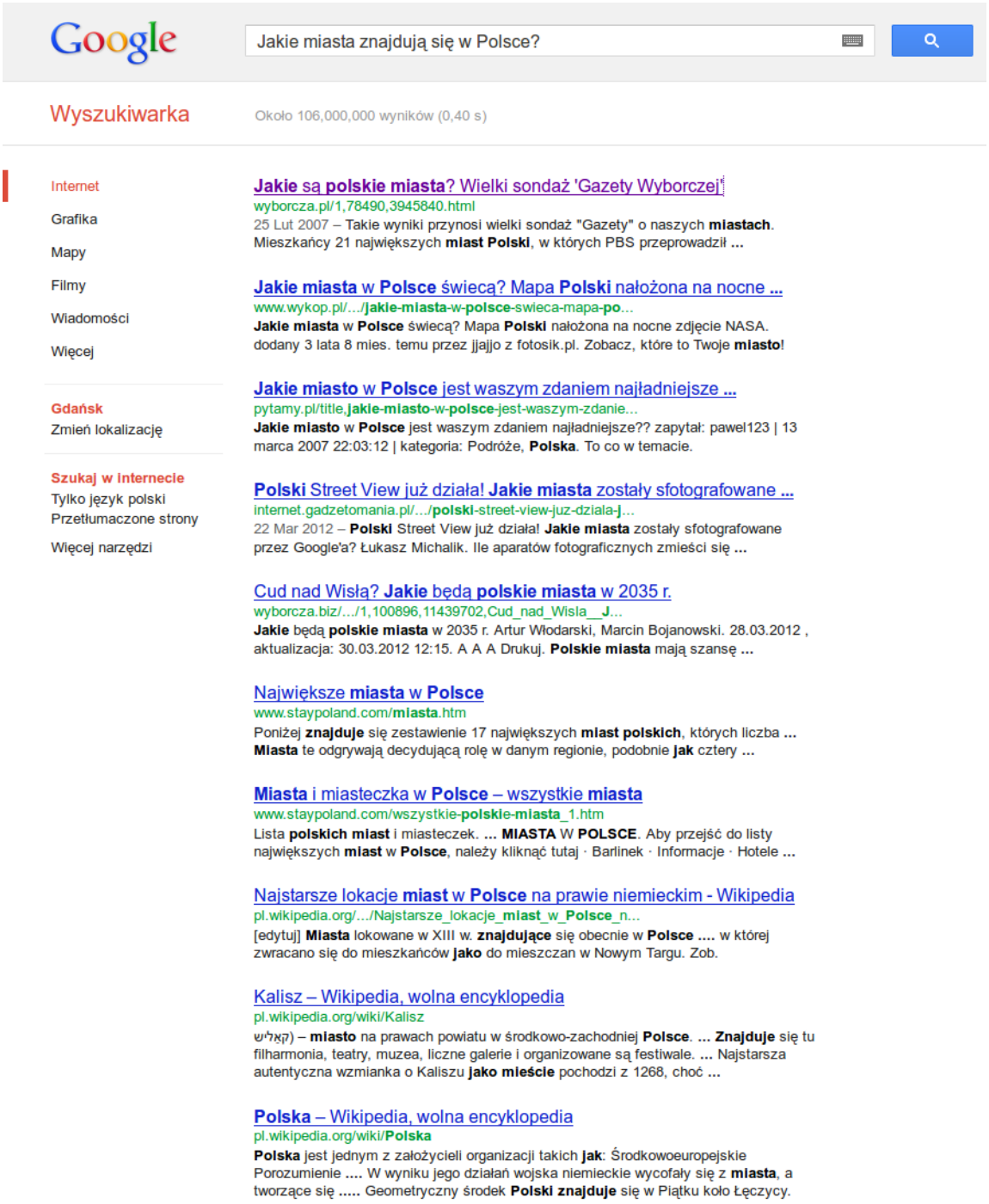
Wyszukiwarki internetowe (Google i Bing) nie są w stanie zwrócić zwięzłej odpowiedzi na pytania o listę elementów. Jest to możliwe jedynie w sytuacji, kiedy odpowiedź na pytanie jawnie występuje w jednym z dokumentów zaindeksowanych przez wyszukiwarkę. W takiej sytuacji odpowiedzią jest cały dokument zawierający odpowiedź w pewnym, nieustandaryzowanym formacie. Przewagą systemu Serel jest generowanie

odpowiedzi na podstawie ciągłego tekstu, przez co nie jest on uzależniony od jawnego wystąpienia kompletnej odpowiedzi w dokumencie.

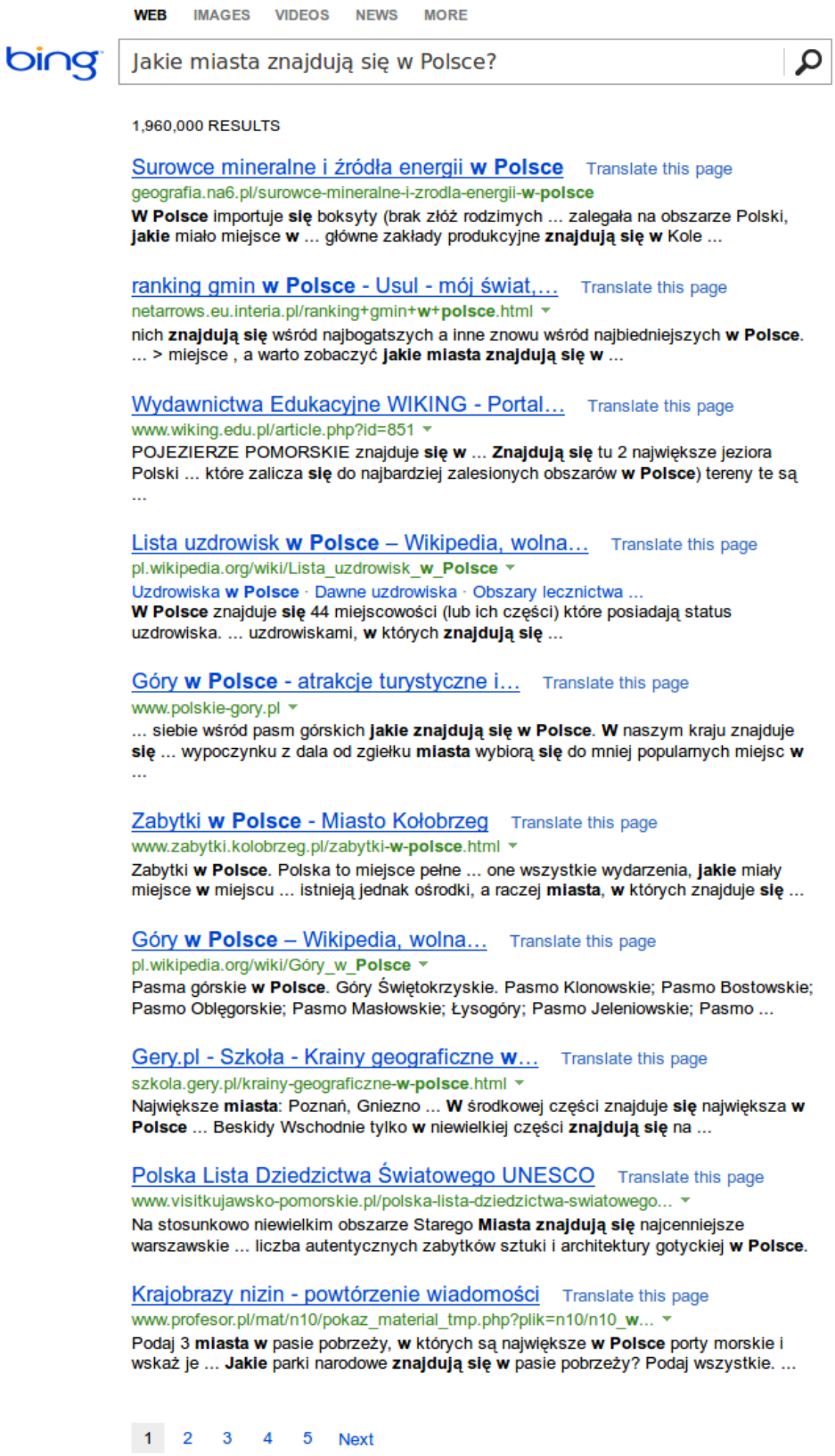
Testowana wyszukiwarka semantyczna KtoCo.pl także nie była w stanie zwrócić precyzyjnej odpowiedzi na pytania o listę elementów. Wynika to z konstrukcji systemu, który zakłada analizę wyników zwracanych przez wyszukiwarkę Google i wskazanie pojedynczej informacji na stronie.

W przypadku pytań o fakty wyszukiwarki internetowe były w stanie zlokalizować fragment tekstu zawierający szukaną odpowiedź. Mimo to zwracaną odpowiedzią było całe zdanie lub paragraf i użytkownik musiał poświęcić dodatkowy czas na przeczytanie przedstawionego fragmentu tekstu i zlokalizowanie odpowiedzi. System Serel był w stanie precyzyjnie wskazać szukaną informację w tekście.

Należy podkreślić, że system Serel w obecnej postaci nie ma tak dużej bazy wiedzy, jaka może być znaleziona za pomocą wyszukiwarek internetowych. Wiąże się to z kilkoma aspektami. Pierwszy, najważniejszy jest taki, że istniejące wyszukiwarki internetowe mają znacznie więcej zaindeksowanych dokumentów niż system Serel. Po drugie, ostateczny wynik działania systemu uzależniony jest od jakości działania poszczególnych jego komponentów, tj. identyfikacji jednostek i relacji. Na poziomie rozpoznawania jednostek nie wszystkie z nich zostają wykryte, przez co nie mogą być zidentyfikowane relacje między nimi. Także na etapie rozpoznawania relacji nie zawsze relacja zostaje wykryta. To ograniczenie może być częściowo zniesione dla informacji, które występują wielokrotnie w różnych źródłach w różnej formie. Jednak to wymaga większego pokrycia dokumentów.



Rys. 6.5. Zrzut ekranu przedstawiający wynik zwrócony przez wyszukiwarkę Google dla pytania *Jakie miasta znajdują się w Polsce?*.



Rys. 6.6. Zrzut ekranu przedstawiający wynik zwrócony przez wyszukiwarkę Bing dla pytania *Jakie miasta znajdują się w Polsce?*.

Rys. 6.7. Zrzut ekranu przedstawiający wynik zwrócony przez system Hipisek dla pytania *Jakie miasta znajdują się w Polsce?*.

Rozdział 7

Podsumowanie

7.1. Realizacja celu rozprawy

Pierwszy cel rozprawy, polegający na opracowaniu nadzorowanej metody do rozpoznawania wybranych kategorii jednostek identyfikacyjnych w tekstach w języku polskim, został osiągnięty poprzez opracowanie hybrydowego modelu rozpoznawania jednostek identyfikacyjnych, łączącego metody uczenia nadzorowanego z metodami regułowymi i słownikowymi. W modelu została użyta metoda warunkowych pól losowych (CRF) wykorzystująca bogaty zbiór cech uwzględniający m.in. informacje ortograficzne, morfologiczne, semantyczne oraz słownikowe. Model został uzupełniony o metodę słownikową rozpoznającą jednoznaczne nazwy własne oraz metodę regułową wykorzystującą kilkanaście reguł, które zostały opracowane ręcznie. Przedstawiona metoda została użyta do stworzenia modelu rozpoznającego 56 kategorii nazw własnych.

Drugi cel rozprawy, polegający na opracowaniu nadzorowanej metody do rozpoznawania relacji semantycznych określonych kategorii między jednostkami identyfikacyjnymi w tekście w języku polskim, został osiągnięty poprzez opracowanie dwufazowego, w pełni nadzorowanego modelu rozpoznawania relacji semantycznych między jednostkami identyfikacyjnymi. W pierwszej fazie został wykorzystany paradygmat indukcyjnego programowania logicznego do automatycznej konstrukcji reguł do rozpoznawania relacji. Cechy były konstruowane dla trzech modeli reprezentacji danych: modelu słów kluczowych, modelu kontekstu jednostek i modelu zależności między tokenami. W drugiej fazie zostały użyte klasyfikatory wektorowe, które wykorzystują reguły wygenerowane w pierwszej fazie jako cechy. Końcowym efektem było opracowanie zestawu ośmiu klasyfikatorów binarnych, po jednym dla każdej kategorii relacji.

Ostatni cel rozprawy, polegający na opracowaniu prototypu systemu odpowiedzi na pytania wykorzystującego bazę wiedzy stworzoną przy użyciu narzędzi do rozpoznawania jednostek identyfikacyjnych i relacji między nimi, został osiągnięty poprzez konstrukcję dwumodułowego systemu ekstrakcji informacji. Pierwszy moduł odpowie-

dzialny jest za przetwarzanie zbioru dokumentów i indeksowanie informacji w relacyjnej bazie danych (indeksowane są jednostki identyfikacyjne i relacje semantyczne). Drugi moduł, który działa niezależnie od pierwszego, odpowiedzialny jest za analizę pytań i transformację ich do postaci zapytania SQL umożliwiającego wyciągnięcie odpowiedzi z relacyjnej bazy danych.

Działanie opracowanego prototypu systemu ekstrakcji informacji o nazwie Serel zostało porównane z istniejącymi wyszukiwarkami internetowymi (Google i Bing) oraz systemami ekstrakcji informacji dla języka polskiego (KtoCo.pl i Hipisek). Zakres obsługiwanych pytań został zawężony do pytań o nazwy własne obiektów będących w określonej relacji względem obiektu o wskazanej nazwie. Na tle istniejących systemów wyszukiwania informacji opracowany system pozwolił na udzielenie precyzyjnej odpowiedzi na zadane pytanie. Tradycyjne wyszukiwarki internetowe także były w stanie wskazać poprawną odpowiedź, ale wymagały od użytkownika dodatkowego wysiłku polegającego na znalezieniu właściwej odpowiedzi w przedstawionym tekście. Odpowiedź systemu Serel zawierała także fragment tekstu, w którym informacja została rozpoznana wraz ze wskazaniem rozpoznanego elementu i elementu będącego argumentem pytania. Dzięki temu użytkownik miał możliwość zweryfikowania danej odpowiedzi.

Realizacja zdefiniowanych celów wykazała, że teza *Wyszukiwanie informacji ograniczone do relacji semantycznych między jednostkami identyfikacyjnymi może być realizowane bardziej efektywnie przy użyciu nadzorowanych metod ekstrakcji informacji niż przy użyciu tradycyjnych wyszukiwarek internetowych.* jest prawdziwa dla zadań, których celem jest pozyskanie listy jednostek identyfikacyjnych spełniających określone warunki (będących w określonej relacji względem podanego obiektu). Dla tego typu zadań przewagą systemu wyszukiwania informacji jest brak uzależnienia od istnienia gotowej listy jednostek wśród zaindeksowanych dokumentów. Takie listy są generowane automatycznie na podstawie informacji rozpoznanych w ciągłym tekście. Natomiast w przypadku pytań, dla których odpowiedzią jest jedna jednostka identyfikacyjna, efektywność wyszukiwania jest porównywalna. Wartością dodaną systemu wyszukiwania informacji jest możliwość precyzyjnego wskazania odpowiedzi w zwróconym fragmencie tekstu, z dokładnością do granicy jednostki identyfikacyjnej.

7.2. Wymierny rezultat pracy

W tej części zostały wymienione efekty realizacji poszczególnych celów w postaci opracowanych narzędzi, zasobów i algorytmów (lista zawiera także elementy, które były opracowane przy udziale współpracowników autora rozprawy).

7.2.1. Rozpoznawanie jednostek identyfikacyjnych

Efektem realizacji pierwszego celu jest:

- korpus raportów giełdowych CSER znakowany jednostkami identyfikacyjnymi,
- korpus wiadomości gospodarczych CEN znakowany jednostkami identyfikacyjnymi,

- warstwa anotacji jednostek identyfikacyjnych w korpusie KPWr,
- słownik nazw własnych zawierający ponad 1,4 miliona nazw własnych należących do ponad 50 kategorii,
- narzędzie Liner2 do rozpoznawania jednostek identyfikacyjnych w tekście,
- model do rozpoznawania 5 kategorii nazw własnych (imion, nazwisk, nazw ulic, miast i państw),
- model do rozpoznawania 56 kategorii nazw własnych.

7.2.2. Rozpoznawanie relacji semantycznych

Efektem realizacji drugiego celu jest:

- warstwa anotacji relacji między jednostkami identyfikacyjnymi w korpusie KPWr,
- 3 modele bazy wiedzy do opisywania relacjami (model słów kluczowych, model kontekstu jednostek i model zależności między tokenami),
- zbiór 8 klasyfikatorów do rozpoznawania 8 kategorii relacji semantycznych między jednostkami.

7.2.3. System ekstrakcji informacji

Efektem realizacji trzeciego celu jest:

- algorytm transformacji pytania w języku naturalnym należącego do określonej klasy do postaci zapytania SQL,
- zbiór reguł transformacji pytań,
- zbiór pytań i ich interpretacji uwzględniających: kategorię relacji, kategorie jednostek identyfikacyjnych, wartość argumentów i kategorię obiektu, którego dotyczy pytanie,
- moduł do indeksowania jednostek identyfikacyjnych i relacji między nimi,
- prototyp systemu ekstrakcji informacji z interfejsem graficznym.

7.3. Unikalny wkład badań

Unikalny wkład badań autora pracy w tematykę ekstrakcji informacji dla języka polskiego stanowi:

- opracowanie i przetestowanie zbioru cech na potrzeby nadzorowanego uczenia modeli do rozpoznawania nazw własnych dla języka polskiego. Został przebadany m.in. wpływ różnych wariantów kodowania cech słownikowych i semantycznych (przy użyciu Słowosieci) na jakość rozpoznawania jednostek w kontekście wykorzystania nadzorowanego uczenia;
- opracowanie 3 modeli reprezentacji danych za pomocą predykatów logiki pierwszego rzędu uwzględniających informacje ortograficzne, morfologiczne, składniowe

i semantyczne na potrzeby rozpoznawania relacji między jednostkami identyfikacyjnymi. Opracowane modele zostały użyte do reprezentacji pary jednostek identyfikacyjnych jako wektora cech na potrzeby klasyfikacji;
- opracowanie zbioru reguł dziedzinowych zawężających przestrzeń przeszukiwania możliwych rozwiązań przez algorytmy indukcyjnego programowania logicznego dla zadania rozpoznawania relacji semantycznych między jednostkami identyfikacyjnymi;
- opracowanie procedury transformacji pytań do postaci kwerend SQL, umożliwiających wyciągnięcie odpowiedzi z relacyjnej bazy danych;
- opracowanie prototypu systemu ekstrakcji informacji udzielającego odpowiedzi na pytania w języku naturalnym o nazwy obiektów będących w określonej relacji z zadanym obiektem dla języka polskiego.

7.4. Kierunek dalszych badań

Opracowany prototyp systemu ekstrakcji informacji powstał w oparciu o kilka ograniczeń, które mogą być wyeliminowane w ramach dalszych badań nad tym zagadnieniem.

Na poziomie rozpoznawania jednostek identyfikacyjnych w tekście możliwe jest zwiększenie pokrycia rozpoznawania istniejących kategorii nazw własnych oraz poprawa kategoryzacji rozpoznawanych jednostek w oparciu o kontekst zdania i dokumentu. W rozważanym podejściu były wykorzystywane wyłącznie cechy występujące w bliskim kontekście nazwy oraz wiedza ogólna w postaci słowników. Wykorzystanie parsera zależnościowego oraz rozwiązywania koreferencji pozwoli na dostarczenie dodatkowych przesłanek umożliwiających prawidłowe rozpoznanie kategorii jednostek. Drugim elementem związanych z rozpoznawaniem jednostek identyfikacyjnych jest rozszerzenie zakresu rozpoznawanych kategorii oraz typów odniesień.

Kierunek dalszych badań na etapie rozpoznawania relacji może obejmować z jednej strony rozszerzenie zakresu rozpoznawanych relacji na nowe kategorie, a z drugiej hierarchiczną strukturalizację istniejących relacji pod kątem bardziej precyzyjnej ekstrakcji informacji, np. rodzaj powiązania osoby z organizacją. Kolejnym kierunkiem jest rozpoznawanie relacji między jednostkami występującymi w różnych zdaniach z wykorzystaniem informacji o koreferencji i anaforze. Także istniejące modele predykatów, za pomocą których reprezentowane są zdania, mogą być rozszerzone o dodatkowe informacje uwzględniające odległość jednostek w zdaniu liczoną po tokenach, frazach składniowych, a także ścieżkach zależności między tokenami. Kolejnym elementem wartym uwagi jest wprowadzenie pewności dla poszczególnych modeli rozpoznających relacje i opracowanie miary pewności pozwalającej na uporządkowanie informacji ze względu na częstość występowania w dokumentach i pewność ich rozpoznania.

Z kolei w ramach zagadnienia związanego z transformacją pytań w języku naturalnym do postaci zapytań SQL możliwym kierunkiem dalszego rozwoju jest zwiększenie kompletności obsługiwanych form pytań. Może to być zrealizowane poprzez

m.in. uwzględnienie odległości semantycznej między słowami w algorytmie częściowego dopasowania pytania do szablonów pytań.

Podsumowując, można zidentyfikować dwa kierunki dalszego rozwoju rozważanego zagadnienia. Pierwszy związany jest z poprawą precyzji i kompletności dla istniejących kategorii relacji. Drugi ukierunkowany jest na rozszerzenie zakresu rozpoznawanych informacji.

Bibliografia

Abramowicz, W., Filipowska, A., Piskorski, J., Krzysztof, W., i Wieloch, K. (2006). Linguistic Suite for Polish Cadastral System. W: *5th International Conference on Language Resources and Evaluation*, str. 2518–2523, Genoa. European Language Resources Association (ELRA), European Language Resources Association (ELRA).

Appelt, D. E. i Israel, D. J. (1999). Introduction to information extraction technology. A tutorial prepared for IJCAI-99, Stockholm, Schweden.

Benajiba, Y., Diab, M., i Rosso, P. (2008). Arabic named entity recognition using optimized feature sets. W: *Proceedings of the Conference on Empirical Methods in Natural Language Processing - EMNLP '08*, number October in EMNLP '08, str. 284–293, Morristown, NJ, USA. Association for Computational Linguistics.

Borkar, V., Deshmukh, K., i Sarawagi, S. (2001). Automatic segmentation of text into structured records. *SIGMOD Rec.*, **30**(2), 175–186.

Broda, B., Marcińczuk, M., Maziarz, M., Radziszewski, A., i Wardyński, A. (2012). KPWr: Towards a free corpus of polish. W: N. Calzolari, K. Choukri, T. Declerck, M. U. Doğan, B. Maegaard, J. Mariani, J. Odijk, i S. Piperidis, red., *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey. European Language Resources Association (ELRA).

Broda, B., Kędzia, P., Marcińczuk, M., Ramocki, R., Radziszewski, A., i Wardyński, A. (2013). Fextor: A feature extraction framework for natural language processing: A case study in word sense disambiguation, relation recognition and anaphora resolution. *Studies in Computational Intelligence*, **458**, 41–62.

Brun, C. i Hagège, C. (2009). Semantically-driven extraction of relations between named entities. *Research in Computing Science*, **41**, 35–46.

Bunescu, R. C. (2007). *Learning for information extraction: from named entity recognition and disambiguation to relation extraction.* Ph.D. thesis, The University of Texas at Austin.

Cafarella, M. J., Re, C., Suciu, D., i Etzioni, O. (2007). Structured querying of web

text data: A technical challenge. W: *CIDR 2007, Third Biennial Conference on Innovative Data Systems Research, Asilomar, CA, USA, January 7-10, 2007, Online Proceedings*, str. 225–234. www.crdrdb.org.

Califf, M. E. (1998). *Relational learning techniques for natural language information extraction*. Ph.D. thesis, The University of Texas at Austin.

Chan, Y. S. i Roth, D. (2010). Exploiting background knowledge for relation extraction. W: C.-R. Huang i D. Jurafsky, red., *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, str. 152–160, Stroudsburg, PA, USA. Association for Computational Linguistics.

Chan, Y. S. i Roth, D. (2011). Exploiting syntactico-semantic structures for relation extraction. W: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, volume 1, str. 551–560, Stroudsburg, PA, USA. Association for Computational Linguistics.

Craven, M. i Kumlien, J. (1999). Constructing biological knowledge bases by extracting information from text sources. W: T. Lengauer, R. Schneider, P. Bork, D. L. Brutlag, J. I. Glasgow, H.-W. Mewes, i R. Zimmer, red., *Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology, August 6-10, 1999, Heidelberg, Germany*, str. 77–86. AAAI.

Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V., Aswani, N., Roberts, I., Gorrell, G., Funk, A., Roberts, A., Damljanovic, D., Heitz, T., Greenwood, M. A., Saggion, H., Petrak, J., Li, Y., i Peters, W. (2011). Text Processing with GATE (Version 6). Technical report.

Dietterich, T. G. (1998). Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms. *Neural computation*, **10**(7), 1895–1923.

Doorenbos, R. B., Etzioni, O., i Weld, D. S. (1997). A scalable comparison-shopping agent for the world-wide web. W: *Proceedings of the first international conference on Autonomous agents*, AGENTS '97, str. 39–48, New York, NY, USA. ACM.

Fellbaum, C., red. (1998). *WordNet: an electronic lexical database*. MIT Press.

Fleischman, M., Hovy, E., i Echihabi, A. (2003). Offline strategies for online question answering: answering questions before they are asked. W: *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, str. 1–7, Morristown, NJ, USA. Association for Computational Linguistics.

Georgiev, G., Nakov, P., Ganchev, K., Osenova, P., i Simov, K. (2009). Feature-rich named entity recognition for bulgarian using conditional random fields. W: G. Angelova, K. Bontcheva, R. Mitkov, N. Nicolov, i N. Nikolov, red., *Proceedings of the International Conference RANLP-2009*, str. 113–117, Borovets, Bulgaria. Association for Computational Linguistics.

Giuliano, C., Lavelli, A., i Romano, L. (2006). Exploiting shallow linguistic information for relation extraction from biomedical literature. W: D. McCarthy i S. Wintner, red., *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2006)*, str. 401–408, Trento, Italy.

Graliński, F., Jassem, K., Marcińczuk, M., i Wawrzyniak, P. (2009). Named Entity

Recognition in Machine Anonymization. W: M. A. Kłopotek, A. Przepiórkowski, A. T. Wierzchoń, i K. Trojanowski, red., *Recent Advances in Intelligent Information Systems.*, str. 247–260. Academic Pub. House Exit.

Grishman, R. i Sundheim, B. (1996). Message understanding conference - 6: A brief history. W: *Proceedings of the 16th International Conference on Computational Linguistics (COLING)*, str. 466–471, Kopenhagen.

Grishman, R., Huttunen, S., i Yangarber, R. (2002). Information extraction for enhanced access to disease outbreak reports. *Journal of Biomedical Informatics*, **35**(4), 236–246.

Grzenia, J. (1998). *Słownik nazw własnych — ortografia, wymowa, słowotwórstwo i odmiana*. Wydawnictwo Naukowe PWN, Warszawa.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., i Witten, I. H. (2009). The WEKA data mining software: an update. *SIGKDD Explor. Newsl.*, **11**(1), 10–18.

Hobbs, J. R. i Riloff, E. (2010). Information extraction. W: N. Indurkhya i F. J. Damerau, red., *Handbook of Natural Language Processing, Second Edition*. CRC Press, Taylor and Francis Group, Boca Raton, FL, 2 edition.

Indurkhya, N. i Damerau, F. J. (2010). *Handbook of Natural Language Processing*. CRC Press, Taylor and Francis Group, Boca Raton, FL, 2 edition.

Kambhatla, N. (2004). Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations. W: *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*, ACLdemo '04, Stroudsburg, PA, USA. Association for Computational Linguistics.

Kopeć, M. i Ogrodniczuk, M. (2012). Creating a coreference resolution system for polish. W: N. C. C. Chair), K. Choukri, T. Declerck, M. U. Doğan, B. Maegaard, J. Mariani, J. Odijk, i S. Piperidis, red., *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey. European Language Resources Association (ELRA).

Kushmeric, N. (1997). *Wrapper Induction for Information Extraction*. Ph.D. thesis, University of Washington.

Lafferty, J. D., McCallum, A., i Pereira, F. C. N. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. W: C. E. Brodley i A. P. Danyluk, red., *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, str. 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Lawrence, S., Giles, C. L., i Bollacker, K. (1999). Digital libraries and autonomous citation indexing. *IEEE Computer*, **32**(6), 67–71.

Linguistic Data Consortium (2008a). ACE (Automatic Content Extraction) English Annotation Guidelines for Entities (Version 6.6). Technical report, Linguistic Data Consortium.

Linguistic Data Consortium (2008b). ACE (Automatic Content Extraction) English Annotation Guidelines for Relations (Version 6.2).

LLC, F. R. (2010). GATE Teamware 1.3 User Guide. Technical report.

Loper, E. (2008). *Encoding structured output values*. Ph.D. thesis, Philadelphia, PA, USA. AAI3346159.

Manning, C. D. i Schütze, H. (1999). *Foundations of statistical natural language processing*. MIT Press, Cambridge, MA, USA.

Marciniak, M., red. (2010). *Anotowany korpus dialogów telefonicznych*. Akademicka Oficyna Wydawnicza EXIT, Warsaw.

Marciniak, M. i Mykowiecka, A. (2007). Automatic processing of diabetic patients' hospital documentation. W: J. Piskorski i H. Tanev, red., *Proceedings of the Workshop on Balto-Slavonic Natural Language Processing*, str. 35–42, Prague, Czech Republic. Association for Computational Linguistics.

Marciniak, M., Mykowiecka, A., Kupść, A., i Piskorski, J. (2005). *Intelligent Content Extraction from Polish Medical Reports*, volume 3490 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin/Heidelberg.

Marcińczuk, M. (2010). Manufakturzysta 2.0 Luna. Dokumentacja techniczna. Technical report.

Marcińczuk, M. i Janicki, M. (2012). Optimizing CRF-based Model for Proper Name Recognition in Polish Texts. W: A. Gelbukh, red., *Computational Linguistics and Intelligent Text Processing — 13th International Conference, CICLing 2012, New Delhi, India, March 11-17, 2012, Proceedings, Part I*, volume 7181 of *Lecture Notes in Computer Science (LNCS)*, str. 258–269. Springer, Heidelberg.

Marcińczuk, M. i Piasecki, M. (2010). Study on Named Entity Recognition for Polish Based on Hidden Markov Models. W: P. Sojka, A. Horák, I. Kopecek, i K. Pala, red., *Proceedings of Text, Speech and Dialogue: 13th International Conference, TSD 2010*, volume 6231 of *Lecture Notes in Computer Science*, str. 142–149. Springer Berlin / Heidelberg.

Marcińczuk, M. i Piasecki, M. (2011). Statistical Proper Name Recognition in Polish Economic Texts. *Control and Cybernetics*, **40**(2), 393–418.

Marcińczuk, M. i Ptak, M. (2012). Preliminary study on automatic induction of rules for recognition of semantic relations between proper names in polish texts. W: P. Sojka, A. Horák, I. Kopecek, i K. Pala, red., *Text, Speech and Dialogue — 15th International Conference, TSD 2012, Brno, Czech Republic, September 3-7, 2012. Proceedings*, volume 7499 of *Lecture Notes in Artificial Intelligence (LNAI)*. Springer-Verlag.

Marcińczuk, M., Stanek, M., Piasecki, M., i Musiał, A. (2011). Rich Set of Features for Proper Name Recognition in Polish Texts. W: P. Bouvry, M. A. Klopotek, F. Leprévost, M. Marciniak, A. Mykowiecka, i H. Rybinski, red., *Security and Intelligent Information Systems - International Joint Conferences, SIIS 2011, Warsaw, Poland, June 13-14, 2011, Revised Selected Papers*, volume 7053 of *Lecture Notes in Computer Science*. Springer.

Marcińczuk, M., Kocoń, J., i Broda, B. (2012). Inforex – a web-based tool for text corpus management and semantic annotation. W: N. C. C. Chair), K. Choukri, T. Declerck, M. U. Doğan, B. Maegaard, J. Mariani, J. Odijk, i S. Piperidis, red.,

Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12), Istanbul, Turkey. European Language Resources Association (ELRA).

McCallum, A. i Li, W. (2003). Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003*, str. 188–191.

Muggleton, S. (1995). Inverse entailment and progol. *New Generation Comput.*, **13**(34), 245–286.

Muggleton, S. H. i Feng, C. (1990). Efficient induction of logic programs. W: *Proceedings of the First Conference on Algorithmic Learning Theory*, str. 368–381, Tokyo, Japan. Ohmsha.

Muslea, I., Minton, S., i Knoblock, C. (1999). A hierarchical approach to wrapper induction. W: O. Etzioni, J. P. Müller, i J. M. Bradshaw, red., *Proceedings of the third annual conference on Autonomous Agents*, AGENTS '99, str. 190–197, New York, NY, USA. ACM.

Mykowiecka, A., Marciniak, M., i Kupść, A. (2009). Rule-based information extraction from patients' clinical data. *J. of Biomedical Informatics*, **42**(5), 923–936.

Niles, I. i Pease, A. (2001). Towards a standard upper ontology. W: *Proceedings of the international conference on Formal Ontology in Information Systems - Volume 2001*, FOIS '01, str. 2–9, New York, NY, USA. ACM.

Nédellec, C. (2005). Learning language in logic — genic interaction extraction challenge. W: *Proceedings of the Learning Language in Logic 2005 Workshop at the International Conference on Machine Learning*. ACM.

Ono, T., Hishigaki, H., Tanigami, A., i Takagi, T. (2001). Automated extraction of information on protein-protein interactions from the biological literature. *Bioinformatics*, **17**(2), 155–161.

Pantel, P. i Pennacchiotti, M. (2006). Espresso: leveraging generic patterns for automatically harvesting semantic relations. W: N. Calzolari, C. Cardie, i P. Isabelle, red., *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, ACL-44, str. 113–120, Stroudsburg, PA, USA. Association for Computational Linguistics.

Patwardhan, S. i Riloff, E. (2006). Learning domain-specific information extraction patterns from the web. W: *Proceedings of the Workshop on Information Extraction Beyond The Document*, IEBeyondDoc '06, str. 66–73, Stroudsburg, PA, USA. Association for Computational Linguistics.

Piasecki, M. (2007). Polish Tagger TaKIPI: Rule Based Construction and Optimisation. *Task Quarterly*, **11**(1–2), 151–167.

Piasecki, M. i Radziszewski, A. (2007). Polish Morphological Guesser Based on a Statistical A Tergo Index. W: *Proceedings of the International Multiconference on Computer Science and Information Technology — 2nd International Symposium Advances in Artificial Intelligence and Applications (AAIA'07)*, str. 247–256.

Piasecki, M., Szpakowicz, S., i Broda, B. (2009). *A Wordnet from the Ground Up*. Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław.

Piskorski, J. (2004a). Extraction of Polish named entities. W: *Proceedings of the Fourth International Conference on Language Resources and Evaluation, LREC 2004 (ELR, 2004)*, str. 313–316, Prague, Czech Republic. ACL.

Piskorski, J. (2004b). Rule-based Named-Entity Recognition for Polish. *Word Journal Of The International Linguistic Association*.

Piskorski, J., Homola, P., Marciniak, M., Mykowiecka, A., Przepiórkowski, A., i Woliński, M. (2004). Information Extraction for Polish Using the SProUT Platform. W: M. A. Kłopotek, S. T. Wierzchoń, i K. Trojanowski, red., *Intelligent Information Processing and Web Mining, Proceedings of the International IIS: IIPWM'04 Conference*, Advances in Soft Computing, Zakopane. Springer-Verlag.

Piskorski, J., Tanev, H., Atkinson, M., van der Goot, E., i Zavarella, V. (2011). Online news event extraction for global crisis surveillance. W: N. T. Nguyen, red., *Transactions on computational collective intelligence*, str. 182–212. Springer-Verlag, Berlin, Heidelberg.

Popowich, F. (2005). Using text mining and natural language processing for health care claims processing. *SIGKDD Explor. Newsl.*, **7**(1), 59–66.

Przepiórkowski, A. (2007). Slavonic information extraction and partial parsing. W: J. Piskorski, B. Pouliquen, R. Steinberger, i H. Tanev, red., *Proceedings of the Workshop on Balto-Slavonic Natural Language Processing: Information Extraction and Enabling Technologies*, ACL '07, str. 1–10, Stroudsburg, PA, USA. Association for Computational Linguistics.

Przepiórkowski, A. (2008). *Powierzchniowe przetwarzanie języka polskiego*. Akademicka Oficyna Wydawnicza EXIT, Warszawa.

Przepiórkowski, A. i Murzynowski, G. (2009). Manual annotation of the National Corpus of Polish with Anotatornia. W: S. Goźdź-Roszkowski, red., *The proceedings of Practical Applications in Language and Computers PALC 2009*, str. 95–104, Frankfurt. Peter Lang.

Przepiórkowski, A., Bańko, M., Górski, R. L., i Lewandowska-Tomaszczyk, B., red. (2012). *Narodowy Korpus Języka Polskiego*. Wydawnictwo Naukowe PWN, Warszawa.

Quinlan, J. R. i Cameron-Jones, R. M. (1993). FOIL: A Midterm Report. W: P. Brazdil, red., *ECML*, volume 667 of *Lecture Notes in Computer Science*, str. 3–20. Springer.

Radziszewski, A. i Śniatowski, T. (2011a). Maca — a configurable tool to integrate Polish morphological data. W: *Proceedings of the Second International Workshop on Free/Open-Source Rule-Based Machine Translation*.

Radziszewski, A. i Śniatowski, T. (2011b). A memory-based tagger for polish. W: *Proceedings of LTC'11*.

Radziszewski, A., Wardyński, A., i Śniatowski, T. (2011). WCCL: A Morpho-syntactic Feature Toolkit. W: I. Habernal i V. Matousek, red., *Proceedings of Text, Speech and Dialogue - 14th International Conference, TSD 2011, Pilsen, Czech Republic*,

volume 6836 of *Lecture Notes in Computer Science*, str. 434—-441, Pilsen. Springer.

Ramakrishnan, G., Joshi, S., Balakrishnan, S., i Srinivasan, A. (2007). Using ilp to construct features for information extraction from semi-structured text. W: *Proceedings of the 17th international conference on Inductive logic programming*, ILP'07, str. 211–224, Berlin, Heidelberg. Springer-Verlag.

Sarawagi, S. (2008). Information extraction. *Foundations and Trends in Databases*, **1**(3), 261–377.

Sarawagi, S. i Bhamidipaty, A. (2002). Interactive deduplication using active learning. W: *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '02, str. 269–278, New York, NY, USA. ACM.

Savary, A. i Piskorski, J. (2011). Language Resources for Named Entity Annotation in the National Corpus of Polish. *Control and Cybernetics*, **40**(2), 361–391.

Sekine, S. (2009). *Named Entities: Recognition, classification and use*. John Benjamins Publishing Company.

Soderland, S. (1999). Learning information extraction rules for semi-structured and free text. *Mach. Learn.*, **34**(1-3), 233–272.

Srinivasan, A. (2006). The aleph manual. Technical report.

Strzalkowski, T., Stein, G. C., Wise, G. B., i Bagga, A. (2000). Towards the next generation information retrieval. W: J.-J. Mariani i D. Harman, red., *RIAO*, str. 1196–1207. CID.

Suchanek, F. M., Ifrim, G., i Weikum, G. (2006). Combining linguistic and statistical analysis to extract relations from web documents. W: T. Eliassi-Rad, L. H. Ungar, M. Craven, i D. Gunopulos, red., *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '06, str. 712–717, New York, NY, USA. ACM.

Tang, L. R., Mooney, R. J., i Melville, P. (2003). Scaling Up ILP to Large Examples: Results on Link Discovery for Counter-Terrorism. W: L. Getoor, T. E. Senator, P. Domingos, i C. Faloutsos, red., *Proceedings of the KDD-2003 Workshop on Multi-Relational Data Mining*, Washingtion DC.

Urbańska, D. i Mykowiecka, A. (2005). Multi-words Named Entity Recognition in Polish texts. W: R. Grabík, red., *SLOVKO 2005 – Third International Seminar Computer Treatment of Slavic and East European Languages, Bratislava, Slovakia*, str. 208–215. VEDA.

Walas, M. (2012). How to answer yes/no spatial questions using qualitative reasoning? W: A. F. Gelbukh, red., *CICLing (2)*, volume 7182 of *Lecture Notes in Computer Science*, str. 330–341. Springer.

Walas, M. i Jassem, K. (2010). Named Entity Recognition in a Polish Question Answering System. W: M. A. Kłopotek, M. Marciniak, A. Mykowiecka, W. Penczek, i S. T. Wierzchoń, red., *Intelligent Information Systems*, str. 181–191, Siedle.

Wróblewska, A. i Woliński, M. (2012). Preliminary experiments in polish dependency parsing. W: P. Bouvry, M. Kłopotek, F. Leprévost, M. Marciniak, A. Mykowiecka, i H. Rybinski, red., *Security and Intelligent Information Systems*, volume 7053 of

Lecture Notes in Computer Science, str. 279–292. Springer Berlin / Heidelberg.

Zhu, G., Bethea, T. J., i Krishna, V. (2007). Extracting relevant named entities for automated expense reimbursement. W: P. Berkhin, R. Caruana, i X. Wu, red., *KDD*, str. 1004–1012. ACM.

Dodatek A

Schemat jednostek identyfikacyjnych

A.1. Antroponimy

Symbol	Opis	Przykłady
`nation_nam`	Nazwa narodowości, nacji	Polacy, Słowianie
`person_add_nam`	Przydomek, przezwisko osoby	Bóg, Bobo
`person_first_nam`	Imię osoby	Michał, Maria
`person_group_nam`	Nazwa identyfikujące nieformalne grupy osób	Laburzyści, kapucyni
`person_last_nam`	Nazwisko osoby, w tym nazwisko panieńskie	Marcińczuk
`person_nam`	Pełna nazwa osobowa składająca się z imienia, nazwiska, przydomku i inicjałów	Michał M. Marcińczuk

A.2. Chrematonimy

Symbol	Opis	Przykłady
`award_nam`	Nazwa nagrody, tytułu lub orderu.	Polsko-Niemiecka Nagroda Dziennikarska
`band_nam`	Nazwa zespołu muzycznego, drużyny, grupy lub innej grupy ludzi.	Deep Purple
`brand_nam`	Nazwa produktów seryjnych (samochodów, produktów spożywczych, chemicznych, itp),	Astra,
`company_nam`	Nazwa firmy, przedsiębiorstwa, działalności gospodarczej	Google, Saab

currency_nam	Nazwa lub symbol waluty	złoty, dolar, PLN, USD, $
document_nam	Nazwa dokumentu urzędowego	Kodeks cywilny
event_nam	Nazwa wydarzenia sportowego, muzycznego, wystawy, pokazu, koncertu lub projektu	Euro 2012, Eurowizja
facility_nam	Nazwa budowli, budynku, lokalu, klubu	Most Średzki
institution_nam	Nazwa instytucja rozumiana jako element organizacji.	Sejm, Rada Ministrów
license_nam	Nazwa licencji lub sposobu dystrybucji oprogramowania.	GPL
media_nam	Nazwa stacji telewizyjnej lub radiowej.	TVN, BBC
organization_nam	Nazwa organizacji społecznej, politycznej, militarnej, ekonomicznej, nazwa fundacji, zrzeszenia itp.	Unia Europejska
organization_sub_nam EU-15	Nazwa wyróżnionego elementu organizacji wynikającego z pewnego podziału tej organizacji	UE-15, EU-25 (podział państw w UE ze względu na datę przystąpienia)
periodic_nam	Nazwa gazety, czasopisma.	Men's Health
political_party_nam	Nazwa partii politycznej.	Platforma Obywatelska, PiS
software_nam	Nazwa programu lub gry komputerowej.	Windows, Warcraft III
system_nam	Nazwa konkretnych instancji fizycznej systemu posiadającego własną infrastrukturę i realizującego określone cele (np. system informatyczny, ostrzegawczy, itp.)	EPI, ECURE
tech_nam	Nazwa technologii, języka programowania.	CD-ROM, ABS, HTTP, C++, Java
title_nam	Nazwa książki, płyty, obrazu, piosenki lub innego utworu artystycznego.	Na Każdy Temat, Wiadomości
treaty_nam	Nazwa porozumienia zawartego między dwoma stronami (np. krajami).	START
vehicle_nam	Nazwa unikalnego pojazdu lub innego obiektu stworzonego przez człowieka zdolnego do przemieszczania się (nazwy statków, samolotów, satelit, itp.). Nie dotyczy nazw marek.	Rudy 102
web_nam	Nazwa portalu lub strony internetowej.	YouTube
www_nam	Nazwa domeny lub adresu WWW strony internetowej	www.youtube.pl

A.3. Hydronimy

Symbol	Opis	Przykłady
`bay_nam`	Nazwa zatoki.	Tampa Bay
`lagoon_nam`	Nazwa laguny	
`lake_nam`	Nazwa jeziora, stawu, zbiornika wodnego	Czarny Staw
`ocean_nam`	Nazwa oceanu	Ocean Atlantycki
`river_nam`	Nazwa rzeki, strumyka, cieku wodnego	Wisła, Odra
`sea_nam`	Nazwa morza	Morze Bałtyckie, Bałtyk

A.4. Kosmonimy

Symbol	Opis	Przykłady
`astronomical_nam`	Nazwa naturalnych obiektów znajdujących się w kosmosie	Ziemia, Księżyc

A.5. Toponimy

Symbol	Opis	Przykłady
`admin1_nam`	Nazwa terenu podziału administracyjnego pierwszego stopnia (województwo, land, stan)	mazowieckie, Bawaria
`admin2_nam`	Nazwa terenu podziału administracyjnego drugiego stopnia (gmina, county, hrabstwo, rejencja)	wągrowiecki
		Loiret
`admin3_nam`	Nazwa terenu podziału administracyjnego trzeciego stopnia (powiat)	Jocelyn
`cape_nam`	Nazwa przylądku.	Krym
`city_nam`	Nazwa miasta, wioski.	Warszawa
`continent_nam`	Nazwa kontynentu.	Europa
`conurbation_nam`	Nazwa aglomeracji miejskiej.	Trójmiasto
`country_nam`	Nazwa kraju.	Polska
`country_region_nam`	Nazwa regionu geograficznego kraju.	Górny Śląsk
`historical_region_nam`	Nazwa regionu historycznego.	Szkocja, Mazowsze
`island_nam`	Nazwa wyspy.	Haiti
`mountain_nam`	Nazwa pasma górskiego, szczytu górskiego.	Beskid Śląski

peninsula_nam	Nazwa półwyspu.	półwysep Helski
region_nam	Nazwa regionu składającego się z kilku państw.	Skandynawia, Europa Środkowa
sandpit_nam	Nazwa pustyni.	
toponym_nam	Pozostałe nazwy geograficzne nieujęte w pozostałych kategoriach	

A.6. Urbanonimy

Symbol	Opis	Przykłady
district_nam	Nazwa dzielnicy miasta.	Śródmieście
park_nam	Nazwa parku.	Park im. Tadeusza Rejtana
road_nam	Nazwa ulicy, drogi, autostrady.	A1, DK44, Trasa Słowackiego
square_nam	Nazwa placu.	Rynek, Plac Centralny
subdivision_nam	Nazwa osiedla.	Na Skarpie

A.7. Zoonimy i Fitonimy

Symbol	Opis	Przykłady
animal_nam	Nazwa zwierzęta lub postaci mitologicznej, fikcyjnej. Organizmy żywe posiadające cechy ludzkie ale nie będące ludźmi.	Ratakan, Koral
plant_nam	Nazwa rośliny, np. pomnik przyrody.	Bartek

Dodatek B

Schemat relacji semantycznych

Przykładowe zdania zawierające relacje między jednostkami identyfikacyjnymi pochodzą z korpusu KPWr, który został opisany w punkcie 3.4.1

B.1. Autorstwo

	Kategorie jednostek	Opis i przykład
1	`band_nam` - `title_nam`	Zespół jest wykonawcą piosenki lub autorem albumu muzycznego. Np. ***Nie zapomnisz nigdy*** *– album zespołu* ***Big Cyc*** *wydany w 1994 roku.*
2	`company_nam` - `brand_nam`	Firma jest twórcą marki. Np. ***Audi 50*** *(nazwa robocza Typ 86) to samochód segmentu B produkowany przez niemiecką firmę* ***Audi*** *w latach.*
3	`company_nam` - `software_nam`	Firma jest twórca oprogramowania. Np. *SimCity 3000 jest trzecią odsłoną* ***SimCity*** *stworzonej przez Willa Wrighta i firmę* ***Maxis****.*
4	`person_nam` - `event_nam`	Osoba jest organizatorem, inicjatorem wydarzenia. Np. *godzina 16.00 – 16:45 / mgr* ***Elżbieta Skonieczna*** */ „****Jak pomóc dziecku z deficytami rozwojowymi****” Prezentacja ćwiczeń usprawniających funkcje percepcyjne i grafomotorykę dla rodziców dzieci 6 letnich i młodszych klas szkolnych.*
5	`person_nam` - `media_nam`	Osoba jest twórcą, współzałożycielem stacji radiowej lub telewizyjnej. Np. ***Aleksandra Zieleniewska*** *- przewodnicząca Rady Fundacji Nowe Media (współtworzyła* ***RMF FM*** *(...)*

6	person_nam - organization_nam	Osoba jest współzałożycielem, twórcą organizacji. Np. ***Zakon św. Jerzego z Karyntii*** *– zakon założony wspólnie przez cesarza* ***Fryderyka III*** *i papieża Pawła II w 1469 roku.*
7	person_nam - title_nam	Osoba jest wykonawcą piosenki lub autorem książki, obrazu. Np. *Do podobnej figury odniósł się też* ***Richard Dawkins*** *w 2006 w swojej książce* ***Bóg urojony****, pisząc tam: Ćzajniczek Russella to oczywiście jeden z nieskończonych szeregów obiektów, których istnienia nie można ani wykluczyć, ani definitywnie odrzucić.*
8	organization_nam - event_nam	Organizacja jest organizatorem wydarzenia. Np. *Pod koniec lat 80. miejscowość dwukrotnie była bazą* ***Ogólnopolskiego Nocnego Rajdu Mazowieckiego*** *- pieszej imprezy turystycznej, organizowanej przez jedno z kół mokotowskiego oddziału* ***PTTK****.*
9	institution_nam - event_nam	Instytucja jest organizatorem wydarzenia. Np. *W 2005 roku po zakończeniu remontu, pałac otrzymał nagrodę w konkursie "****Zabytek zadbany****organizowanym przez* ***Krajowy Ośrodek Badań i Dokumentacji Zabytków****.*

B.2. Kompozycja

	Kategorie jednostek	**Opis** i **przykład**
1	admin1_nam - country_nam	Obszar administracyjny pierwszego poziomu (np. województwo) jest elementem państwa. Np. *Hrabstwo Van Buren – hrabstwo w* ***USA****, w stanie* ***Tennessee****, według spisu z 2000 roku liczba ludności wynosiła 5508.*
2	admin2_nam - admin1_nam	Obszar administracyjny drugiego poziomu (np. powiat) jest elementem obszaru administracyjnego pierwszego poziomu (np. województwo). Np. *Kanie-Stacja – wieś w Polsce położona w województwie* ***lubelskim****, w powiecie* ***chełmskim****, w gminie Rejowiec Fabryczny.*
3	admin3_nam - admin2_nam	Obszar administracyjny trzeciego poziomu (np. gmina) jest elementem obszaru administracyjnego pierwszego poziomu (np. powiat). Np. *Kanie-Stacja – wieś w Polsce położona w województwie* ***lubelskim****, w powiecie chełmskim, w gminie* ***Rejowiec Fabryczny****.*

4	`district_nam` `- city_nam`	Dzielnica jest elementem miasta. Np. *Toronto Dominion Centre - kompleks handlowo-kulturalny w kanadyjskim mieście **Toronto**, w Financial District.*
5	`event_nam` `- event_nam`	Wydarzenie będące integralną częścią innego wydarzenia. Np. ***Niedziela Wielkanocna** (nazywana też Wielką Niedzielą, Niedzielą Zmartwychwstania Pańskiego, I Niedzielą Wielkanocną) – pierwszy dzień **świąt wielkanocnych**.*
6	`institution_nam` `- organization_nam`	Instytucja będąca częścią organizacji. Np. *Za niecałe dwa tygodnie organizujemy w **ICM UW**, w którym pracuję, konferencję "Otwarta nauka w Polsce" .*
7	`institution_nam` `- institution_nam`	Kolejne stopnie podziału administracyjnego w ramach instytucji. Np. ***Centrum Zarządzania Kryzysowego Urzędu Miejskiego Wrocławia** informuje:*

B.3. Narodowość

	Kategorie jednostek	**Opis** i **przykład**
1	`person_nam` `- country_nam`	Osoba posiada obywatelstwo kraju. Np. ***Hristo Zlatanov** (ur. 21 kwietnia 1976 roku w 2Sofii) – siatkarz reprezentacji **Włoch** bułgarskiego pochodzenia.*
2	`person_nam` `- nation_nam`	Osoba należy do grupy narodowej. Np. *Był 2. na mecie wyścigu na 50 km na Zimowych Igrzyskach Olimpijskich 2002 w Salt Lake City, lecz po dyskwalifikacji za stosowanie dopingu zwycięzcy, **Hiszpana Johanna Mühlegga**, otrzymał złoty medal.*

B.4. Pochodzenie

	Kategorie jednostek	**Opis** i **przykład**
1	`band_nam` `- country_nam`	Zespół został założony w kraju, reprezentuje kraj. Np. *Do najwyższej klasy rozgrywkowej trafił w 1978 jako zawodnik najbardziej znanego zespołu z **Gruzji** - **Dinama Tbilisi**.*
2	`band_nam` `- district_nam`	Zespół reprezentuje dzielnicę miasta. Np. ***Barnet Football Club** - angielski klub piłkarski z **Barnet** - północnej dzielnicy Londynu, grający obecnie w Football League Two.*

3	company_nam - city_nam	Firma została założona w mieście, ma swoją siedzibę w mieście. Np. *W roku 1996 hotel wraz z budynkami gospodarczymi i przyległym terenem został zakupiony przez **Fabrykę Urządzeń Mechanicznych "KAMAX"** z **Kańczugi**.*
4	person_nam - admin1_nam	Osoba pochodzi z obszaru województwa (mieszka lub urodziła się na terenie) Np. *W XIX wieku na wyspie mieszkał sułtan **Abdullah** z **Peraku**, zesłany tu przez Brytyjczyków.*
5	person_nam - city_nam	Osoba pochodzi z miasta (mieszka lub urodziła się na terenie) Np. ***Adriaan van der Hoop**, ur. 28 kwietnia 1778 w **Amsterdamie**, zm. 17 marca 1854, tamże – holenderski bankier, polityk i kolekcjoner dzieł sztuki.*
6	person_nam - country_nam	Osoba pochodzi z obszaru państwa (mieszka lub urodziła się na terenie) Np. *Mimo iż **Dickoh** urodził się w **Danii**, jest reprezentantem Ghany, skąd pochodzą jego rodzice.*
7	organization_nam - city_nam	Organizacja została założona w mieście, posiada swoją siedzibę Np. *Dziś **Creative Commons** obchodzi czwarte urodziny - przedsięwzięcie ruszyło dokładnie 16 grudnia 2002 w **San Francisco**.*

B.5. Położenie

	Kategorie jednostek	**Opis** i **przykład**
1	admin2_nam - country_nam	Teren podziału administracyjnego drugiego poziomu znajduje się na terenie państwa. Np. *Powiat **średzki** - powiat w **Polsce** (województwo wielkopolskie), utworzony w 1999 roku w ramach reformy administracyjnej.*
2	admin3_nam - admin1_nam	Teren podziału administracyjnego trzeciego poziomu znajduje się na terenie obszaru pierwszego podziału administracyjnego, Np. ***Rabós** – hiszpańska gmina w Katalonii, w prowincji **Girona**, w comarce Alt Empordà.*
3	admin3_nam - country_nam	Teren podziału administracyjnego trzeciego poziomu znajduje się na terenie państwa. Np. *Pegau - miasto w **Niemczech**, w kraju związkowym Saksonia, w okręgu dyrekcyjnym Lipsk, w powiecie **Lipsk** (do 31 lipca 2008 w powiecie Lipsk Land).*

4	band_nam - city_nam	Zespół znajduje się w mieście (przebywa na stałe lub czasowo na obszarze) Np. *Pierwszy mecz rundy wstępnej Pucharu Polski na szczeblu wojewódzkim zespól Kotana Ozorków rozegra w niedzielę 06.08.2006 godz. 16.30 w* ***Swędowie*** *z tamtejszym* ***Huraganem****.*
5	city_nam - admin1_nam	Miasto znajduje się na terenie pierwszego podziału administracyjnego. Np. ***Smardy Dolne*** *– wieś w Polsce położona w województwie* ***opolskim****, w powiecie kluczborskim, w gminie Kluczbork.*
6	city_nam - admin2_nam	Miasto znajduje się na terenie drugiego podziału administracyjnego. Np. ***Smardy Dolne*** *– wieś w Polsce położona w województwie opolskim, w powiecie* ***kluczborskim****, w gminie Kluczbork.*
7	city_nam - admin3_nam	Miasto znajduje się na terenie trzeciego podziału administracyjnego. Np. ***Smardy Dolne*** *– wieś w Polsce położona w województwie opolskim, w powiecie kluczborskim, w gminie* ***Kluczbork****.*
8	city_nam - country_nam	Miasto znajduje się na terenie państwa. Np. ***Smardy Dolne*** *– wieś w* ***Polsce*** *położona w województwie opolskim, w powiecie kluczborskim, w gminie Kluczbork.*
9	city_nam - mountain_nam	Miasto położone jest w górach. Np. ***Dinner Plain*** *to miejscowość położona w* ***Alpach Australijskich*** *w stanie Wiktoria w Australii przy Great Alpine Road, około 10 km od Mount Hotham i 375 km od Melbourne.*
10	company_nam - city_nam	Firma działa lub posiada swoją siedzibę na terenie miasta. Np. *Tam podjął pracę w laboratorium badawczym firmy* ***Dow Chemical*** *w* ***Sarnii****.*
11	company_nam - country_nam	Firma działa lub posiada swoją siedzibę na terenie państwa. Np. *W latach 1969-1980 model ten powstawał w Uusikaupunki w* ***Finlandii****, w zakładzie firmy* ***Valmet Automotive****, w której firma Saab miała udziały.*
12	event_nam - city_nam	Wydarzenie miało miejsce na terenie miasta. Np. *5. zawodnik* ***Pucharu Europy w Lekkoatletyce*** *(****Brema*** *2001), czym przyczynił się do historycznego triumfu polskiej męskiej reprezentacji.*

13	`event_nam` `- country_nam`	Wydarzenie miało miejsce na terenie państwa. Np. *W 2005 został powołany do kadry na **Młodzieżowe Mistrzostwa Świata** w kategorii U-20, odbywające się w **Holandii**.*
14	`event_nam` `- facility_nam`	Wydarzenie miało miejsce na terenie budynku. Np. *W 1996, podobny koncept - jednorożca, którego nikt nie może zobaczyć - był zaadaptowany jako figura edukacyjna na **Camp Quest**, pierwszym obozie o charakterze świecko-wolnomyślicielskim dla dzieci, założonym w **USA**.*
15	`facility_nam` `- admin1_nam`	Budynek znajduje się na terenie pierwszego podziału administracyjnego. Np. *Najbardziej charakterystyczną cechą **Stratosphere** jest wieża o wysokości 350 m, najwyższa budowla w stanie **Nevada** i druga (po CN Tower w Toronto) najwyższa wieża obserwacyjna na zachodniej półkuli.*
16	`facility_nam` `- city_nam`	Budynek znajduje się na terenie miasta. Np. ***Stadion Punjab** to wielożytkowy stadion znajdujący się w mieście **Lahaur** w Pakistanie, wykorzystywany głównie do rozgrywania meczów piłkarskich.*
17	`facility_nam` `- country_nam`	Budynek znajduje się na terenie państwa. Np. ***Stadion Punjab** to wielożytkowy stadion znajdujący się w mieście Lahaur w **Pakistanie**, wykorzystywany głównie do rozgrywania meczów piłkarskich.*
18	`facility_nam` `- continent_nam`	Budynek znajduje się na terenie kontynentu. Np. *Z myślą o tym ostatnim, w miejscu galeriowca powstaje najwyższy budynek w **Europie** – 750-metrowa **Water Tower**.*
19	`facility_nam` `- facility_nam`	Budynek (obiekt budowlany) znajduje się na terenie innego budynku (kompleksu budynków) Np. *Miejsce : **Sala Senatu**, **Pałac Kazimierzowski**, główny kampus Uniwersytetu Warszawskiego*
20	`institution_nam` `- city_nam`	Instytucja prowadzi działalność lub posiada siedzibę na terenie miasta. Np. *Prezes **Wojewódzkiego Związku Brydża Sportowego** w **Łodzi** mgr inż. Włodzimierz Choinkowski*
21	`institution_nam` `- country_nam`	Instytucja prowadzi działalność lub posiada siedzibę na terenie państwa. Np. *W 1945 rozpoczął pracę dla MSZ, był m.in. kierownikiem Wydziału Konsularnego oraz attaché w **Ambasadzie RP** (PRL) w **Czechosłowacji**.*

22	`institution_nam` `- facility_nam`	Instytucja posiada siedzibę na terenie budynku. Np. *Na środku rynku znajduje się zabytek **Stary Ratusz**, w którym się mieści dzisiaj **Wojewódzka Biblioteka Publiczna**, która przeszła również generalny remont wraz z Rynkiem otaczającym Stary Ratusz.*
23	`island_nam` `- country_nam`	Wyspa znajduje się na terenie państwa. Np. *Yuzawa - miasto w północnej **Japonii**, na wyspie **Honsiu**.*
24	`lake_nam` `- mountain_nam`	Jezioro jest położone w górach w górach. Np. *W polskich **Tatrach Wysokich** znajduje się 12 stawów /jezior polodowcowych/ o powierzchni powyżej 1 ha,najbardziej znane to: **Morskie Oko**, Czarny Staw, Wielki Staw, Czarny Staw Gąsiennicowy.*
25	`mountain_nam` `- mountain_nam`	Szczyt górski jest położony w górach. Np. *Najbardziej znanym szczytem **Pienin** są **Trzy Korony** /982 m n.p.m./, jednak najwyższym szczytem są Wysokie Skałki /1052 m n.p.m./.*
26	`organization_nam` `- admin1_nam`	Organizacja prowadzi działalność lub ma siedzibę na terenie podziału administracyjnego pierwszego rzędu. Np. *Twierdzą oni, że po śmierci Elvisa wyłoniła się specyficzna religia, posiadająca własnych proroków (sobowtóry Króla), święte teksty (muzyka), uczniów (fani Elivsa), relikty (gadżety i pamiątki), pielgrzymki (Tupelo lub Graceland), świątynie (grób Króla) czy kościoły (**Twenty-four-Hour Church of Elvis** w Portland w stanie **Oregon**).*
27	`organization_nam` `- city_nam`	Organizacja prowadzi działalność lub ma siedzibę na terenie miasta. Np. *W latach 1974-1982 studiował na **Papieskim Uniwersytecie Salezjańskim** w **Rzymie** filologię klasyczną oraz starochrześcijańską.*
28	`organization_nam` `- country_nam`	Organizacja prowadzi działalność lub ma siedzibę na terenie państwa. Np. *Zorganizował też pierwsze szkolenia dla księży-dyrektorów diecezjalnych **Caritas** w Wiedniu, Brukseli i **Niemczech**.*
29	`person_nam` `- admin1_nam`	Osoba znajduje się na terenie podziału administracyjnego pierwszego rzędu. Np. *Ponowna lokacja na prawie wołoskim miała miejsce pod koniec XV w. i dokonał jej **Jan Herburt (Arłamowski)** (1470–1508) – poseł województwa **ruskiego** – pierwszy właściciel – wójt samborski.*

30	person_nam - city_nam	Osoba znajduje się na terenie miasta. Np. ***Żeromski*** *dzieciństwo spędził w* ***Ciekotach****, u stóp Gór Świętokrzyskich.*
31	person_nam - country_nam	Osoba znajduje się na terenie państwa. Np. *Dr* ***Wiesław Nowicki*** *długo stał w obronie polskiej przyrody i walczył z różnymi samorządami w* ***Polsce****.*
32	person_nam - facility_nam	Osoba znajduje się na terenie budynku. Np. *W 1926 roku na* ***Kremlu*** *rozegrał ze* ***Stalinem*** *partię szachów, którą przegrał po 37 ruchach, grając czarnymi .*
33	person_nam - title_nam	Osoba występuje w utworze (na obrazie, bohater utworu literackiego). Np. ***Tyzenhauz*** *to też, nazwisko dworzanina królewskiego występującego w dramacie „****Mazepa. Tragedia w pięciu aktach****".*
34	river_nam - city_nam	Rzeka przepływa przez miasto. Np. ***Priesnia*** *(ros.) – niewielka rzeka w centrum w* ***Moskwy****, lewy dopływ rzeki Moskwy.*
35	river_nam - country_nam	Rzeka przepływa przez państwo. Np. *Złoża węgla występują w Zagłębiu Górnośląskim, Dolnośląskim /Wałbrzyskim/ oraz we wschodniej* ***Polsce*** *między* ***Wieprzem*** *a Bugiem /Zagłebie Lubelskie/.*
36	road_nam - admin1_nam	Ulica znajduje się na terenie podziału administracyjnego pierwszego poziomu. Np. ***Droga krajowa nr 44*** *(DK44) - droga krajowa przebiegająca przez województwo* ***śląskie*** *oraz małopolskie.*
37	road_nam - city_nam	Ulica znajduje się na terenie miasta. Np. *Ulica* ***Zacisze*** *w* ***Katowicach*** *(niem. Friedenstraße) jedna z ulic w katowickiej dzielnicy Śródmieście.*
38	road_nam - district_nam	Ulica znajduje się na terenie dzielnicy miasta. Np. *Ulica* ***Zacisze*** *w Katowicach (niem. Friedenstraße) jedna z ulic w katowickiej dzielnicy* ***Śródmieście****.*
39	software_nam - software_nam	Oprogramowanie jest elementem innego oprogramowania. Np. *W nowym* ***Firefoksie*** *jest* ***AwesomeBar****, podobny pasek adresu jest też w Chrome.*
40	title_nam - media_nam	Tytuł artykułu w gazecie, tytuł programu telewizyjnego. Np. *W roku 2001 przeszedł do* ***TV Puls****, gdzie był dziennikarzem i prowadzącym* ***Wydarzenia****.*

41	`title_nam` `- periodic_nam`	Tytuł artykułu, utworu w gazecie, czasopiśmie. Np. *Debiutował wierszem **Mleczarz** w "**Nowym Torze**"w 1953.*

B.6. Przynależność

	Kategorie jednostek	**Opis** i **przykład**
1	`band_nam` `- award_nam`	Zespół otrzymał nagrodę. Np. *W 1996 roku z drużyną **Colorado Avalanche** zdobył **Puchar Stanleya**.*
2	`band_nam` `- event_nam`	Zespół uczestniczył w wydarzeniu. Np. *Karierę w **NBA** rozpoczynał w drużynie **San Diego Rockets**, wybrany z dalekim 85. numerem w drafcie 1968 w wieku 22 lat.*
3	`company_nam` `- company_nam`	Firma należy do koncernu. Np. ***Roewe** – chiński producent samochodów należący do koncernu **Shanghai Automotive Industry Corporation** (SAIC) od listopada 2006 roku produkuje modele na bazie kupionych technologii od upadłego MG Rover.*
4	`country_nam` `- event_nam`	Reprezentaci kraju uczestniczą w wydarzeniu. Np. ***Malediwy** na **Mistrzostwach Świata** w Lekkoatletyce 2009 – reprezentacja Malediwów podczas czempionatu w Berlinie liczyła 2 zawodników.*
5	`country_nam` `- organization_nam`	Kraj należy do organizacji. Np. ***Irlandia** miała szczęście - należała do **EWG**, była biednym krajem, w którym wszyscy mówią w języku zbliżonym do angielskiego, a akurat na fali nowej gospodarki płynęły inwestycje technologiczne (Microsoft - 1985, Intel - 1989, Dell - 1990 i już w innych czasach Google - 2003).*
6	`country_nam` `- system_nam`	Państwo jest członkiem systemu, Np. *Minister właściwy do spraw wewnętrznych określi, w drodze rozporządzenia, sposób wykorzystywania Krajowego Systemu Informatycznego (KSI) jako krajowego interfejsu Wizowego Systemu Informacyjnego, w tym sposób dokonywania wpisów danych VIS, a także wglądu do danych VIS, mając na względzie prawidłowe wykonanie przez **Rzeczpospolitą Polską** zobowiązań wynikających z udziału w **Wizowym Systemie Informacyjnym**.";*

7	`country_nam` `- treaty_nam`	Państwo jest stroną umowy. Np. ***Republika Bułgarii*** *i Rumunia stają się stronami* ***Traktatu ustanawiającego Konstytucję dla Europy*** *oraz Traktatu ustanawiającego Europejską Wspólnotę Energii Atomowej, wraz z ich zmianami i uzupełnieniami.*
8	`facility_nam` `- institution_nam`	Budowla należy, jest zarządzana przez instytucję. Np. *Obecnie mieści się tam* ***Ośrodek Szkoleniowy ZUS-u****.*
9	`institution_nam` `- city_nam`	Instytucja miasta. Np. *Flaga i herb zostały przyjęte uchwałą* ***Rady Miasta Odessy*** *29 czerwca 1999 roku.*
10	`institution_nam` `- country_nam`	Instytucja państwa. Np. *Brzeżański Batalion Obrony Narodowej (Batalion ON "Brzeżany") - pododdział piechoty* ***Wojska Polskiego II RP****.*
11	`institution_nam` `- institution_nam`	Instytucja działa w ramach innej instytucji. Np. *W 1914 szef* ***Sztabu Generalnego Armii Włoskiej****.*

12	`institution_nam`
	`-`
	`political_party_nam`
		Instytucja partii politycznej.
		
		Np. *Kiedy 11 marca 1985 roku Biuro Polityczne* ***KC KPZR*** *powierzyło obowiązki Sekretarza Generalnego KC PZPR Michaiłowi Gorbaczowowi nikt na Kaukazie nie zdawał sobie sprawy, że nadchodzi czas zmian.*

13	`media_nam` `- company_nam`	Stacja telewizyjna lub radiowa należy do firmy. Np. *Jak podaje serwis Wirtualnemedia.pl : 22. grudnia 2006 roku* ***Superstacja sp. z o.o.*** *(właściciel telewizji* ***Superstacja*** *) rozwiązała umowę inwestycyjną z Capital Partners (CP) – zawartą we wrześniu br. – na mocy której CP miała nabyć 30 % udziałów w Superstacji.*
14	`organization_nam` `- event_nam`	Organizacja bierze udział w wydarzeniu. Np. ***Fundacja Nowe Media*** *prowadzi w ramach projektu „****Moje Miasto****”, finansowanego przez Ministerstwo Kultury i Dziedzictwa Narodowego warsztaty dziennikarskie w szkołach Mińska Mazowieckiego.*
15	`organization_nam` `- treaty_nam`	Organizacja jest stroną umowy. Np. *Na mocy niniejszego Aktu nowe Państwa Członkowskie przystępują do* ***Umowy o Partnerstwie*** *między członkami Grupy Państw Afryki, Karaibów i Pacyfiku z jednej strony, a* ***Wspólnotą Europejską*** *i jej Państwami Członkowskimi, z drugiej strony, podpisanej w Cotonou w dniu 23 czerwca 2000 r.*

16	`periodic_nam` `- event_nam`	Gazeta lub czasopismo otrzymało nagrodę w konkursie. Np. *„**Pentagon**” zajął trzecie miejsce w ogólnoukraińskim konkursie **MAM-u**.*
17	`periodic_nam` `- institution_nam`	Gazeta lub czasopismo należy do instytucji. Np. ***Szaniec Kresowy** – oficjalny organ prasowy **Okręgu XIV Lwowskiego** Narodowych Sił Zbrojnych, wychodził od 12 marca 1943 r.*
18	`person_nam` `- award_nam`	Osoba otrzymała nagrodę. Np. *Za swoją działalność **Wacław Milke** otrzymał wiele nagród i prestiżowych odznaczeń, między innymi **Krzyż Komandorski z Gwiazdą Orderu Odrodzenia Polski**, Krzyż Wielki Orderu Odrodzenia Polski, Złoty Krzyż Zasługi, srebrny medal „Za Zasługi dla Obronności Kraju”, odznakę „Zasłużony dla Kultury Polskiej”, medal „Za Zasługi dla Oświaty i Wychowania”, medal Gloria Artis, medal Komisji Edukacji Narodowej.*
19	`person_nam` `- band_nam`	Osoba należy do zespołu. Np. *Mocno zmotywowani gospodarze, grający z nożem na gardle, postawili nam ciężkie warunki - powiedział po meczu szkoleniowiec **GTPS Dziewulski Inkaso Team Jerzy Taczała**.*
20	`person_nam` `- company_nam`	 Np. *W narożniku czerwonym, po stronie popytu, szef **Red Hata Jim Whitehurst** widzi te wszystkie zamówienia na oprogramowanie bez opłat licencyjnych.*
21	`person_nam` `- event_nam`	Osoba bierze udział w wydarzeniu. Np. *Latem 2002 roku **Skoubo** odszedł do niemieckiej Borussii Mönchengladbach, a 24 sierpnia rozegrał swoje pierwsze spotkanie w **Bundeslidze**, wygrane 3:0 z 1. FC Kaiserslautern.*
22	`person_nam` `- institution_nam`	Osoba należy do instytucji. Np. *Biskup koszalińsko-kołobrzeski **Czesław Domin**, który był przewodniczącym **Komisji Charytatywnej Konferencji Episkopatu Polski**, w 1993 poprosił ks. Mariana by w Warszawie stworzył struktury Caritas Polska.*
23	`person_nam` `- organization_nam`	Osoba należy do organizacji. Np. *PZL.13 (PZL-13) – projekt polskiego sześciomiejscowego samolotu pasażerskiego zaprojektowanego w 1931 roku przez inż. **Stanisława Praussa** w **Państwowych Zakładach Lotniczych** w Warszawie*

24	person_nam - political_party_nam	Osoba należy do partii politycznej. Np. *Posłowie **Józef Piotr Klim** i Leszek Cieślik z **Platformy Obywatelskiej** zadadzą pytanie w sprawie harmonogramu prac inwestycyjnych drogi ekspresowej S8 na odcinku Białystok–Warszawa do ministra infrastruktury.*
25	person_nam - periodic_nam	Osoba należy do gazety, czasopisma. Np. ***Janusz Andrzej Rolicki** (ur. 22 października 1938 w Wilnie) – polski dziennikarz, były redaktor naczelny "**Trybuny**"(1996–2001)*
26	person_nam - media_nam	Osoba należy do stacji telewizyjnej lub radiowej. Np. *12.00 – 13.00 - panel dyskusyjny dotyczący polityki redakcyjnej z udziałem: zastępcy redaktora naczelnego Polsatu, Bogusława Chraboty, redaktora naczelnego Newsweeka, **Wojciecha Maziarskiego**, redaktora naczelnego **Gazety Polskiej**, Tomasza Sakiewicza, szefa działu opinii Gościa Niedzielnego, Bogumiła Łozińskiego.*
27	person_nam - person_group_nam	Osoba należy do grupy. Np. ***Katarzyna Załuska** (zm. 1703) – polska zakonnica ze zgromadzenia **bernardynek**, przełożona i fundatorka klasztoru w Przasnyszu.*
28	person_nam - title_nam	Osoba jest bohaterem utworu. Np. ***London Tipton** w serialu **Nie ma to jak hotel** ma suczkę Ivanę która jest szpicem miniaturowym*
29	political_party_nam - country_nam	Partia polityczna działa na terenie państwa. Np. ***Komunistyczna Partia Austrii** – komunistyczna partia w **Austrii**, założona w 1918 roku.*
30	www_nam - institution_nam	Strona www należy do instytucji. Np. *Na stronie internetowej **Urzędu m.st. Warszawy www.um.warszawa.pl** w elektronicznym Wydziale Obsługi Mieszkańców znajduje się wykaz rachunków bankowych dla poszczególnych Dzielnic.*

B.7. Sąsiedztwo

	Kategorie jednostek	**Opis** i **przykład**
1	`city_nam` - `city_nam`	Miasto sąsiaduje z innym miastem. Np. *Łazar Mojsiejewicz Kaganowicz ros. (ur. 22 listopada 1893 w przysiółku **Kabany** (obecnie: Dibrowa) koło **Chabna**, zm. 25 lipca 1991 w Moskwie) - radziecki polityk.*

2	`city_nam`
	-
	`country_region_nam`
		Miasto leży na granicy regionów.
		
		Np. ***Chlebowo*** *(kaszb. Chlebòwò) – wieś w Polsce położona w województwie pomorskim, w powiecie bytowskim, w gminie Miastko na pograniczu pojezierzy **Drawskiego** i Bytowskiego na wzgórzach morenowych dochodzących do 250 m n.p.m. 4 km na wschód od Miastka.*

3	`city_nam` - `lake_nam`	Miasto leży nad/w sąsiedztwie jeziorem. Np. ***Małe*** *(kaszb.Môłé) – mała osada kaszubska w Polsce na Pojezierzu Bytowskim położona w województwie pomorskim, w powiecie bytowskim, w gminie Studzienice nad północnym brzegiem jeziora **Małego**.*
4	`city_nam` - `river_nam`	Miasto leży nad/w sąsiedztwie rzeki. Np. ***Zaopusta*** *osada na terenie Katowic, powstała w XVII wieku obok Podlesia nad brzegiem rzeki **Mleczna**.*
5	`company_nam` - `road_nam`	Siedziba firmy znajduje się przy ulicy. Np. *W Tychach przy **DK44** położone są takie firmy jak: **Kompania Piwowarska** (Tyskie Browary Książęce), fabryka Fiat Auto Poland czy też centrum dystrybucyjne hipermarketów Kaufland.*
6	`country_nam` - `country_nam`	Państwo graniczy z innym państwem. Np. ***Wolne Państwo Irlandzkie*** *(irl. Saorstát Éireann) – istniejące w latach 1922-1937 niepodległe państwo obejmujące 26 z 32 hrabstw Irlandii, oddzielonych od **Zjednoczonego Królestwa Wielkiej Brytanii i Irlandii**.*
7	`district_nam` - `district_nam`	Dzielnica miasta sąsiaduje z inną dzielnicą. Np. ***Bielawy***, *dzielnica Torunia w jego wschodniej części, między **Grębocinem** na północy a osiedlem Na Skarpie i lasami na południu.*

8	`facility_nam` `- road_nam`	Budynek leży przy drodze. Np. *15 grudnia 1907, uruchomiono linię tramwajową linii 1 łączącą pierwotnie **Most św. Jana** (a później Jezioro Długie) przy ulicy **Prostej** z Dworcem Głównym przez ulicę Staromiejską, które przejeżdżały przez wschodnią część Rynku.*
9	`institution_nam` `- road_nam`	Siedziba instytucji znajduje się przy drodze. Np. *Zapraszamy młodych twórców do czytelni **Miejskiej Biblioteki Publicznej** w Ozorkowie przy ul. **Listopadowej** 6b w środę 11 stycznia 2006 r. o godz. 17:00.*
10	`park_nam` `- road_nam`	Park znajduje się przy drodze. Np. ***Park im. Tadeusza Rejtana** (zwany także "Park Nowe Rokicieóraz, nieprawidłowo "Parkiem Skrzywana") - mieści się w Łodzi pomiędzy ulicą **Piękną**, Rejtana, Felsztyńskiego i al. Politechniki.*
11	`road_nam` `- city_nam`	Droga przebiega w sąsiedztwie miasta. Np. ***Droga krajowa nr 44** jest obwodnicą dla miast **Mikołów** i Bieruń Stary.*
12	`road_nam` `- road_nam`	Droga krzyżuje się, łączy z inną drogą. Np. *Budowa Mostu i Trasy Siekierkowskiej po zakończeniu całej inwestycji będzie arterią łączącą węzeł komunikacyjny u zbiegu ulic **Czerniakowskiej** i **Witosa** na Mokotowie ze skrzyżowaniem ulic Ostrobramskiej, Marsa i Płowieckiej w dzielnicy Wawer.*
13	`river_nam` `- river_nam`	Rzeka jest dopływem innej rzeki. Np. ***Priesnia** (ros.) – niewielka rzeka w centrum w Moskwy, lewy dopływ rzeki **Moskwy**.*
14	`river_nam` `- city_nam`	Rzeka przepływa przez miasto. Np. *W **Nowej Wsi** znajduje się leśniczówka, oraz siedziba RSP Łaziska, wieś położona jest nad rzeką **Wełną**, w lasach znajdują się poniemieckie zniszczone budynki, oraz poniemiecki cmentarz.*
15	`square_nam` `- facility_nam`	Plac znajduje się przy budynku. Np. *Na środku rynku znajduje się zabytek Stary Ratusz, w którym się mieści dzisiaj Wojewódzka Biblioteka Publiczna, która przeszła również generalny remont wraz z **Rynkiem** otaczającym **Stary Ratusz**.*

16	`square_nam` - `road_nam`	Plac znajduje się przy drodze. Np. *Pozycja **Rynku** spełniała strategiczną rolę, była pośrodku osi północ-południe, na północ Górne Przedmieście (Oberrohstadt) i ulica **Staromiejska** (dawna ulica Górna – Oberstr.) prowadząca do Bramy Górnej – również znanej jako Wysoka Brama.*
17	`subdivision_nam` - `road_nam`	Osiedle znajduje się przy drodze. Np. ***Osiedle XXV-lecia PRL** - osiedle mieszkaniowe położone we wschodniej części Tarnowa pomiędzy ul. **Lwowską** i ul. Słoneczną.*

B.8. Tożsamość

Relacja tożsamości zachodzi między nazwami własnych obiektów tych samych kategorii, więc liczba podkategorii jest równa liczbie rozpatrywanych nazw własnych. Interpretacja każdej podkategorii jest taka sama i oznacza tożsamość nazw własnych (odnoszenie się do tego samego obiektu).

	Kategorie jednostek	**Przykład**
1	`facility_nam` - `facility_nam`	Np. *Pozycja Rynku spełniała strategiczną rolę, była pośrodku osi północ-południe, na północ Górne Przedmieście (Oberrohstadt) i ulica Staromiejska (dawna ulica Górna – Oberstr.) prowadząca do **Bramy Górnej** – również znanej jako **Wysoka Brama**.*
2	`institution_nam` - `institution_nam`	Np. *Projekt „Otwórz książkę" jest prowadzony przez **Interdyscyplinarne Centrum Modelowania** na Uniwersytecie Warszawskim (**ICM** UW), w ramach projektów Biblioteka Wirtualna Nauki oraz Creative Commons Polska.*
3	`title_nam` - `title_nam`	Np. *"Men's Health"wydaje cieszące się dużą popularnością tematyczne numery specjalne, m.in. "**MH Coach**"(poprzednio "**Twój Osobisty Trener**")*

Dodatek C

Przykładowe wygenerowane reguły

C.1. Autorstwo

Model słów kluczowych

```
relation(A,B,creator) :-
  sentence_has_annotation(C,B),
  sentence_has_base(C,"word_organizować").
```

```
relation(A,B,creator) :-
   sentence_has_annotation(C,B),
   sentence_has_base(C,"word_wydać"),
   sentence_has_hypheronym(C,word_syn_164582_w),
   sentence_has_hypheronym(C,word_syn_20951_cykl_astronomiczny).
```

Model kontekstu jednostek

```
relation(A,B,creator) :-
   annotation_first_token(A,C),
   token_after_token(D,C),
   token_hypheronym(D,word_syn_27592_przedstawienie),
   token_hypheronym(D,"word_syn_2129_całość").
```

Model zależności między tokenami

```
relation(A,B,creator) :-
   annotation_first_token(A,C),
   token_dependency(D,C,adj),
   token_dependency(E,D,adj),
   token_hypheronym(E,word_syn_164582_w),
   token_pos(D,subst).
```

C.2. Kompozycja

Model słów kluczowych

```
relation(A,B,composition) :-
   sentence_has_annotation(C,B),
   sentence_has_base(C,"word_położyć"),
   sentence_has_base(C,word_w),
   sentence_has_base(C,meta_COMMA)
```

```
relation(A,B,composition) :-
   sentence_has_annotation(C,B),
   sentence_has_hypheronym(C,word_syn_53428_instytucja),
   sentence_has_hypheronym(C,word_syn_4833_siedziba).
```

Model kontekstu jednostek

```
relation(A,B,composition) :-
   annotation_first_token(A,C),
   token_hypheronym(C,"word_syn_6434_dział"),
   annotation_last_token(B,D),
   token_after_token(D,E),
   token_pattern(E,"PATTERN_LOWERCASE").
```

Model zależności między tokenami

```
relation(A,B,composition) :-
   annotation_first_token(B,C),
   token_dependency(C,D,adj),
   token_hypheronym(D,word_syn_5592_jednostka_administracyjna).
```

```
relation(A,B,composition) :-
   annotation_first_token(B,C),
   token_dependency(D,C,punct),
   annotation_last_token(A,E),
   token_dependency(E,F,adj),
   token_hypheronym(F,word_syn_7359_struktura).
```

C.3. Narodowość

Model słów kluczowych

```
relation(A,B,nationality) :-
   sentence_has_annotation(C,B),
   sentence_has_base(C,word_urodzony),
   sentence_has_base(C,word_rok).
```

```
relation(A,B,nationality) :-
   sentence_has_annotation(C,B),
   sentence_has_hypheronym(C,word_syn_41068_kraj).
```

Model zależności między tokenami

```
relation(A,B,nationality) :-
   annotation_first_token(B,C),
   token_dependency(C,D,adj),
   token_hypheronym(D,"word_syn_6822_człowiek_ze_względu
                       _na_pełnioną_funkcję"),
   token_hypheronym(C,word_syn_1282_grupa).
```

C.4. Pochodzenie

Model słów kluczowych

```
relation(A,B,origin) :-
   annotation_of_type(B,city_nam),
   annotation_of_type(A,person_nam),
   sentence_has_annotation(C,B),
   sentence_has_base(C,word_urodzony),
   sentence_has_base(C,meta_DASH).
```

```
relation(A,B,origin) :-
   sentence_has_annotation(C,B),
   sentence_has_base(C,"word_się"),
   sentence_has_hypheronym(C,"word_syn_462_część"),
   sentence_has_hypheronym(C,"word_syn_6778_nazwa_człowieka_uwzględniająca
                               _jego_cechy"),
   sentence_has_hypheronym(C,word_syn_164585_z).
```

Model kontekstu jednostek

```
relation(A,B,origin) :-
  annotation_last_token(B,C),
  token_after_token(C,D),
  token_base(D,"word_urodzić").
```

```
relation(A,B,origin) :-
   annotation_first_token(B,C),
   token_after_token(D,C),
   token_after_token(E,D),
   token_orth(D,word_z),
   annotation_first_token(A,F),
   token_after_token(F,E).
```

Model zależności między tokenami

```
relation(A,B,origin) :-
   annotation_first_token(B,C),
   token_dependency(C,D,comp),
   token_dependency(D,E,adj),
   token_dependency(E,F,adj),
   token_hypheronym(F,word_syn_4750_miejsce),
   token_hypheronym(D,word_syn_164585_z).
```

```
relation(A,B,origin) :-
   annotation_last_token(B,C),
   token_dependency(C,D,comp),
   token_dependency(D,E,adj),
   token_dependency(E,F,adj),
   token_dependency(G,F,comp),
   token_hypheronym(D,word_syn_164585_z).
```

C.5. Położenie

Model słów kluczowych

```
relation(A,B,location) :-
   sentence_has_annotation(C,B),
   sentence_has_hypheronym(C,word_syn_3243_stan),
   sentence_has_hypheronym(C,word_syn_4799_pomieszczenie),
   sentence_has_hypheronym(C,"word_syn_25121_miejsce_ze_względu
                              _na_przeznaczenie").
```

```
relation(A,B,location) :-
   annotation_of_type(B,country_region_nam),
   sentence_has_annotation(C,B),
   sentence_has_base(C,"word_być"),
   sentence_has_hypheronym(C,"word_syn_2129_całość").
```

Model kontekstu jednostek

```
relation(A,B,location) :-
  annotation_first_token(B,C),
  token_after_token(D,C),
  token_after_token(E,D),
  token_base(D,word_w),
  annotation_last_token(A,E),
  annotation_of_type(A,facility_nam).
```

```
relation(A,B,location) :-
  annotation_first_token(A,C),
  token_after_token(D,C),
  token_hypheronym(D,"word_syn_4884_miejscowość"),
  token_after_token(E,D),
  token_orth(E,word_w).
```

Model zależności między tokenami

```
relation(A,B,location) :-
   annotation_of_type(A,city_nam),
   annotation_last_token(B,C),
   token_dependency(C,D,comp),
   token_dependency(D,E,adj),
   token_hypheronym(E,"word_syn_4897_przestrzeń"),
   token_hypheronym(D,word_syn_164582_w).
```

```
relation(A,B,location) :-
   annotation_last_token(B,C),
   token_dependency(C,D,adj),
   token_dependency(E,D,comp),
   token_hypheronym(E,word_syn_164582_w),
   annotation_last_token(A,F),
   token_dependency(F,G,app).
```

C.6. Przynależność

Model słów kluczowych

```
relation(A,B,affiliation) :-
  annotation_of_type(B,band_nam),
  sentence_has_annotation(C,B),
  sentence_has_hypheronym(C,"word_syn_6797_człowiek_ze_względu
    _na_swoje_zajęcie").
```

```
relation(A,B,affiliation) :-
   annotation_of_type(A,person_nam),
   sentence_has_annotation(C,B),
   sentence_has_hypheronym(C,"word_syn_29159_członek").
```

Model kontekstu jednostek

```
relation(A,B,affiliation) :-
   annotation_first_token(B,C),
   token_after_token(D,C),
   token_hypheronym(D,word_syn_6178_uczestnik),
   annotation_last_token(A,E),
```

Model zależności między tokenami

```
relation(A,B,affiliation) :-
   annotation_first_token(A,C),
   token_hypheronym(C,word_syn_7650_organizacja),
   annotation_last_token(B,D),
   token_dependency(D,E,adj),
   token_dependency(E,F,comp),
```

C.7. Sąsiedztwo

Model słów kluczowych

```
relation(A,B,neighbourhood) :-
   annotation_of_type(A,facility_nam),
   sentence_has_annotation(C,B),
   sentence_has_base(C,word_przy),
   sentence_has_hypheronym(C,word_syn_289_budynek).
```

Model kontekstu jednostek

```
relation(A,B,neighbourhood) :-
  annotation_first_token(B,C),
  token_after_token(D,C),
  token_base(D,"word_koło").
```

Model zależności między tokenami

```
relation(A,B,neighbourhood) :-
   annotation_first_token(B,C),
   token_dependency(C,D,adj),
   token_dependency(E,D,adj),
   token_pos(E,ign),
   token_hypheronym(D,word_syn_1167_ulica).
```

C.8. Tożsamość

Model kontekstu jednostek

```
relation(A,B,alias) :-
   annotation_first_token(B,C),
   token_after_token(D,C),
   token_after_token(C,E),
   token_after_token(E,F),
   token_base(E,meta_BRACKET_RIGHT),
   annotation_last_token(A,G),
   token_after_token(G,D),
```

Dodatek D

Formalizm języka WCCL do znakowania sekwencji

D.1. Szablon reguły

Reguła WCCL ma następującą postać:

```
rule := apply(
            sec_match,
            (sec_cond,)?
            sec_actions
        )
```

gdzie:

- **sec_match** to sekcja zawierająca listę operatorów określających warunki dopasowania sekwencji tokenów i anotacji,
- **sec_cond** to sekcja zawierająca listę operatorów określających dodatkowe warunki nałożone na dopasowane elementy, np. sprawdzenie, czy token należy do anotacji wskazanej kategorii lub czy zachodzi uzgodnienie między dwoma tokenami (ta sekcja jest opcjonalna),
- **sec_actions** to sekcja zawierająca listę operatorów określających akcje do wykonania, np. dodanie nowej anotacji lub usunięcie istniejącej.

D.2. Oznaczenia pomocnicze

```
token_attr  := (orth|base)[token_index]
token_index := (-)?NUM
group_index := (M)?(:NUM)+
string_list := [STRING(, STRING)]
NUM         := [0-9]+
STRING      := "[a-zA-Z0-9]+"
```

gdzie:

- `token_attr` to forma ortograficzna (`orth`) lub bazowa (`base`) tokenu o pozycji `token_index`,
- `token_index` to indeks tokenu względem bieżącej pozycji; 0 oznacza bieżący token, $+n$ oznacza n-ty token po prawej stronie, a $-n$ oznacza n-ty token po lewej stronie,
- `group_index` to indeks sekwencji dopasowanej przez operator z sekcji *match*.

D.3. Sekcja *match*

Sekcja zawiera listę operatorów określających warunki dopasowania sekwencji tokenów i anotacji. Ma ona następującą strukturę:

```
sec_match       := match(
                     op_match_list
                   )

op_match_list   := op_match
                   (, op_match)*

op_match        := op_match_token | op_match_seq | op_match_ann

op_match_token  := op_equal | op_inter | op_regex | op_isannpart |
                   op_isannhead | op_isannbeg | op_isannend |
                   op_and | op_or | op_not

op_match_seq    := op_text | op_optional | op_repeat | op_longest | op_oneof

op_match_ann    := op_is

op_match_token_list := op_match_token
                       (, op_match_token)*
```

Operatory dopasowania tokenów (`op_match_token`):

- `equal(arg1, arg2)` — dopasowuje bieżący token pod warunkiem, że `arg1`=`arg2`. Operator ma następującą składnię:

  ```
  op_equal := equal(token_attr, STRING)
  ```

- `inter(arg1, arg2)` — dopasowuje bieżący token pod warunkiem, że `arg1` znajduje się w zbiorze `arg2`. Operator ma następującą składnię:

  ```
  op_inter := inter(token_attr, string_list)
  ```

- `regex(arg1, arg2)` — dopasowuje bieżący token pod warunkiem, że `arg1` pasuje do wyrażenia regularnego `arg2`. Operator ma następującą składnię:

  ```
  op_regex := regex(token_attr, STRING)
  ```

- `isannpart(arg1)` — dopasowuje bieżący token pod warunkiem, że jest on częścią anotacji o nazwie `arg1`. Operator ma następującą składnię:

  ```
  op_isannpart := isannpart(STRING)
  ```

- `isannhead(arg1)` — dopasowuje bieżący token pod warunkiem, że jest on głową anotacji o nazwie `arg1`. Operator ma następującą składnię:

  ```
  op_isannhead := isannhead(STRING)
  ```

- `isannbeg(arg1)` — dopasowuje bieżący token pod warunkiem, że jest on początkiem anotacji o nazwie `arg1`. Operator ma następującą składnię:

```
op_isannbeg := isannbeg(STRING)
```

- `isannend(arg1)` — dopasowuje bieżący token pod warunkiem, że jest on końcem anotacji o nazwie `arg1`. Operator ma następującą składnię:

```
op_isannend := isannend(STRING)
```

- `not(arg1)` — dopasowuje token, który nie jest dopasowany przez operator `arg1`. Operator ma następującą składnię:

```
op_not := not(op_match_token)
```

- `or(arg1)` — dopasowuje token, który spełnia jeden z warunków określonych przez listę operatorów `arg1`. Operator ma następującą składnię:

```
op_or := or(op_match_token_list)
```

- `and(arg1)` — dopasowuje token, który spełnia wszystkie warunki określone przez listę operatorów `arg1`. Operator ma następującą składnię:

```
op_and := and(op_match_token_list)
```

Operatory dopasowania sekwencji tokenów (`op_match_seq`):

- `text(arg1)` — dopasowuje sekwencję tokenów, pod warunkiem, że konkatenacja ich form ortograficznych równa się wartości `arg1`. Operator ma następującą składnię:

```
op_text := text(STRING)
```

- `optional(arg1)` — opcjonalne dopasowanie sekwencji tokenów pasującej do listy operatorów `arg1`. Operator ma następującą składnię:

```
op_optional := optional(op_match_token_list)
```

- `repeat(arg1)` — dopasowuje sekwencję tokenów składającą się z jednego lub więcej powtórzeń sekwencji określonej przez `arg1`. Operator ma następującą składnię:

```
op_repeat := repeat(op_match_token_list)
```

- `longest(arg1)` — dopasowuje jedną z podanych sekwencji. Jeżeli kilka sekwencji może być dopasowanych to wybierane jest najdłuższe dopasowanie. Operator ma następującą składnię:

```
op_longest := longest(
                  op_variant
                   (, op_variant)*
              )

op_variant := variant(
                  op_match_token_list
              )
```

Poniżej znajduje się przykład, w którym operator `longest` dopasowuje jedną z sekwencji: *wywodzić się*, *sięgać korzeniami*, *pochodzić* lub *wyrastać*.

```
longest(
  variant(
    inter( base[0], ["wywodzić"] ),
    inter( base[0], ["się"] ) ),
  variant(
    inter( base[0], ["sięgać"] ),
    inter( base[0], ["korzenie", "korzeń"] ) ),
  variant(
    inter( base[0], ["pochodzić"] ) ),
  variant(
    inter( base[0], ["wyrastać"] ) )
)
```

- `oneof(arg1)` — podobnie jak operator `longest` zawiera listę wariantów składających się z sekwencji operatorów dopasowania, z tą różnicą, że zostaje dopasowany pierwszy wariant spełniający określone warunki. W momencie pierwszego dopasowania kolejne warianty nie zostają sprawdzane.

```
op_oneof := oneof(
                op_variant
                 (, op_variant)*
            )
```

Poniżej znajduje się przykład, w którym operator `oneof` dopasowuje jedną z sekwencji: *wywodzić się*, *sięgać korzeniami*, *pochodzić* lub *wyrastać*.

```
oneof(
  variant(
    inter( base[0], ["wywodzić"] ),
    inter( base[0], ["się"] ) ),
  variant(
    inter( base[0], ["sięgać"] ),
    inter( base[0], ["korzenie", "korzeń"] ) ),
  variant(
    inter( base[0], ["pochodzić"] ) ),
  variant(
    inter( base[0], ["wyrastać"] ) )
)
```

Operatory dopasowania anotacji (`op_match_ann`):

- `is(arg1)` — dopasowuje anotację o nazwie `arg1` rozpoczynającą się od bieżącej pozycji. Operator ma następującą składnię:

```
op_is := is(STRING)
```

D.4. Sekcja *cond*

Sekcja zawiera listę operatorów określających dodatkowe warunki nałożone na dopasowane elementy. Ma ona następującą strukturę:

```
sec_cond      := cond(
                     op_cond
                     (, op_cond)*
                 )

op_cond       := op_cond_token | op_cond_group

op_cond_group := op_ann | op_annsub

op_cond_token := op_match_token

op_first      := first(group_index)

op_last       := last(group_index)
```

gdzie:

- `first` zwraca indeks pierwszego tokenu grupy o indeksie `group_index`,
- `last` zwraca indeks ostatniego tokenu grupy o indeksie `group_index`.

Operatory warunków dla grup (`op_cond_group`):

- `ann(arg1, arg2)` — sprawdza, czy zakres tokenów z grupy o indeksie `arg1` pokrywa się w całości z anotacją o nazwie `arg2`.
- `ann(arg1, arg2, arg3)` — sprawdza, czy zakres tokenów rozciągający się od początku grupy o indeksie `arg1` do końca grupy o indeksie `arg2` pokrywa się w całości z anotacją o nazwie `arg3`. Operator ma następującą składnię:

```
op_ann := ann(group_index, STRING)
        | ann(group_index, group_index, STRING)
```

- `annsub(arg1, arg2)` — sprawdza, czy zakres tokenów z grupy o indeksie `arg1` jest fragmentem anotacji o nazwie `arg2`,
- `annsub(arg1, arg2, arg3)` — sprawdza, czy zakres tokenów rozciągający się od początku grupy o indeksie `arg1` do końca grupy o indeksie `arg2` jest fragmentem anotacji o nazwie `arg3`. Operator ma następującą składnię:

```
op_annsub := annsub(group_index, STRING)
           | annsub(group_index, STRING)
```

Operatory warunków dla tokenów (`op_cond_token`) — są to operatory sprawdzające warunki na poziomie tokenów. W sekcji *cond* niedozwolone jest jawne użycie indeksów tokenów (`token_index`). W celu użyciu operatorów odnoszących się do tokenów należy wskazać token w odniesieniu do dopasowanych grup. Do tego służą operatory `first` i `last`, które rzutują indeks grupy na indeks tokenu.

D.5. Sekcja *actions*

Sekcja zawiera listę akcji do wykonania na sekwencji dopasowanej przez operatory z sekcji *match*, np. dodanie nowej anotacji lub usunięcie istniejącej. Ma ona następującą strukturę:

```
sec_actions := actions(
                 op_action
                 (, op_action)*
               )

op_action   := op_mark | op_remark | op_unmark
```

Operatory:

- `mark(:M, arg1)` — znakuje całą sekwencję dopasowaną przez sekcję *match* anotacją o nazwie `arg1`,
- `mark(arg1, arg2)` — znakuje sekwencję dopasowaną jako grupa o indeksie `arg1` anotacją o nazwie `arg2`,
- `mark(arg1, arg2, arg3)` — znakuje sekwencję rozciągająca się od początku grupy o indeksie `arg1` do końca grupy o indeksie `arg2` anotacją o nazwie `arg3`.
  Struktura operatora wygląda następująco:

  ```
  op_mark := mark(:M, STRING)
           | mark(group_index, STRING)
           | mark(group_index, group_index, STRING)
  ```

- `remark(:M, arg1)` — działa analogicznie do operatora `mark` z tą różnicą, że nadpisuje inne anotacje o nazwie `arg1` przypisane do znakowanej sekwencji,
- `remark(arg1, arg2)` — działa analogicznie do operatora `mark` z tą różnicą, że nadpisuje inne anotacje o nazwie `arg2` przypisane do znakowanej sekwencji,
- `remark(arg1, arg2, arg3)` — działa analogicznie do operatora `mark` z tą różnicą, że nadpisuje inne anotacje o nazwie `arg3` przypisane do znakowanej sekwencji.
  Struktura operatora wygląda następująco:

  ```
  op_remark := remark(:M, STRING)
             | remark(group_index, STRING)
             | remark(group_index, group_index, STRING)
  ```

- `unmark(arg1)` — usuwa anotację dopasowaną do grupy o indeksie `arg1`. Struktura operatora wygląda następująco:

  ```
  op_unmark := mark(group_index)
  ```

Dodatek E

Dostęp do narzędzi i zasobów

W załączniku znajduje się lista narzędzi i zasobów opracowanych w ramach rozprawy doktorskiej wraz z informacją o ich dostępności. Część z usług dostępnych on-line jest w fazie testów i mogą wystąpić nieprzewidziane przerwy w ich funkcjonowaniu. W razie wystąpienia problemów z dostępnością proszę o kontakt na mail `marcinczuk@gmail.com`.

E.1. Liner2

Narzędzie do rozpoznawania nazw własnych o nazwie Liner2 jest wynikiem prac nad rozpoznawaniem jednostek identyfikacyjnych (zob. sekcja 4). Narzędzie dostępne na licencji *GNU General Public License* i znajduje się na stronie `http://nlp.pwr.wroc.pl/liner2`. Na tej stronie dostępne są także 3 modele do rozpoznawania jednostek identyfikacyjnych:

- **model-5nam-v1.7z** — model do rozpoznawania 5-ciu kategorii nazw własnych,
- **model-56nam-v1.7z** — model do rozpoznawania 56-ciu kategorii nazw własnych.
- **model-nam-v1.7z** — model do wykrywania granic nazw własnych (bez kategoryzacji). Model został opracowany po zakończeniu prac nad rozprawą w ramach dalszych prac w projekcie SyNaT[1].

E.2. Liner2 on-line

Liner2 jest dostępny w wersji on-line dzięki usłudze sieciowej stworzonej przez Macieja Janickiego w ramach projektu SyNaT. Webowy interfejs graficzny wykorzystujący usługę sieciową do rozpoznawania jednostek identyfikacyjnych został zintegrowany z systemem Inforex.

Uruchomione zostały trzy instancje usługi sieciowej i systemu Inforex. System jest w fazie testów, więc mogą wystąpić nieprzewidziane przerwy w funkcjonowaniu. System dostępny jest pod następującymi adresami:

- `http://nlp.pwr.wroc.pl/inforex/index.php?page=ner` — model do rozpoznawania 56-ciu kategorii nazw własnych. System Inforex jest uruchomiony na serwerze `nlp.pwr.wroc.pl`, usługa sieciowa uruchomiona na serwerze Poznańskiego Centrum Superkomputerowo-Sieciowego.
- `http://156.17.129.140/inforex2/index.php?page=ner` — model do rozpoznawania 56-ciu kategorii nazw własnych. System Inforex i usługa sieciowa uruchomione na roboczym komputerze grupy naukowej G4.19.

1. System Nauki i Techniki; `http://www.synat.pl/`

- `http://188.124.184.105/inforex/index.php?page=ner` — model do rozpoznawania granic nazw własnych. System Inforex i usługa sieciowa uruchomione na prywatnym komputerze autora rozprawy.

E.3. Inforex

System do zarządzania i znakowania korpusów tekstowych o nazwie Inforex jest wynikiem prac przygotowawczych, których celem było opracowanie materiału badawczego na potrzeby rozpoznawania jednostek identyfikacyjnych i relacji semantycznych. W obecnej chwili możliwy jest dostęp do instancji systemu Inforex pod adresem `http://nlp.pwr.wroc.pl/inforex`. System umożliwia publiczny dostęp do korpusów:

- **Korpus wiadomości gospodarczych CEN**
  `http://nlp.pwr.wroc.pl/inforex/?corpus=5&page=browse`
  Korpus CEN oznaczony nazwami własnymi w formacie IOB może być pobrany ze strony `http://nlp.pwr.wroc.pl/en/tools-and-resources/cen`.

- **Korpus raportów giełdowych CSER**
  `http://nlp.pwr.wroc.pl/inforex/index.php?page=browse&corpus=1&subcorpus=55`

W celu wyświetlenia anotacji jednostek identyfikacyjnych należy przejść do perspektywy dokumentu „View Document" i na panelu „View configuration" widocznym z prawej strony kliknąć pozycję „Proper names".

E.4. KPWr

Warstwa anotacji jednostek identyfikacyjnych i relacji semantycznych między jednostkami opracowana i wykorzystana w ramach rozprawy jest integralną częścią Korpusu Politechniki Wrocławskiej (KPWr; zob. 3.4.1). Korpus dostępny jest na licencji *Creative Commons Attribution 3.0 Unported Licence* i do pobrania ze strony `http://nlp.pwr.wroc.pl/kpwr`.

E.5. NELexicon

Słownik nazw własny opracowany na potrzeby rozpoznawania jednostek identyfikacyjnych został udostępniony na stronie `http://nlp.pwr.wroc.pl/nelexicon`.

E.6. Serel

Testowy system odpowiedzi na pytania o nazwie Serel (zob. 6) wykorzystujący metody do ekstrakcji informacji o relacjach semantycznych między jednostkami identyfikacyjnymi (zob. 5) jest dostępny pod adresem `http://188.124.184.105/inforex/?page=serel`. System znajduje się na prywatnym komputerze autora rozprawy i znajduje się w fazie testów, więc mogą wystąpić nieplanowane przerwy w dostępności.

Dodatek F

Słownik

Poniżej znajduje się słownik ważniejszym terminów użytych w rozprawie, skrócone definicje oraz odniesienia do pełniejszej definicji znajdującej się w rozprawie lub w literaturze.

anotacja — sekwencja tokenów z przypisaną interpretacją. Wyróżnia się m.in. anotacje semantyczne (jednostki identyfikacyjne), składniowe (frazy składniowe).

jednostka identyfikacyjna — fragment tekstu będący nazwą pewne obiektu. Jednostka identyfikacyjna może być nazwą własną, deskrypcją określoną lub frazą nominalną.

korpus — zbiór tekstów w postaci elektronicznej często przetworzonych na różnych etapach analizy językowej, np. tokenizacja, segmentacja, analiza morfologiczna, rozpoznanie fraz składniowych, rozpoznanie jednostek identyfikacyjnych.

relacja semantyczna — relacja semantyczna w szerokim znaczeniu rozumiana jest jako pewien określony związek zachodzący między parą elementów. Pojęcie to często jest używane w kontekście wordnetu (relacje między jednostkami leksykalnymi) oraz w kontekście ekstrakcji informacji — wystąpienie pary elementów w tekście, między którymi zachodzi pewna zależność poparta pewnymi przesłankami w tym tekście.

tag morfologiczny — zestaw atrybutów opisujących analizę morfologiczną słowa. W skład analizy morfologicznej zalicza się klasę gramatyczną, przypadek, liczbę, rodzaj, osobę, stopień i aspekt.

token — najmniejszy, niepodzielny fragment tekstu. W fazie segmentacji tekst zostaje podzielony na sekwencje tokenów zgodnie w pewnymi zasadami. W dalszych etapach przetwarzania, po dokonaniu segmentacji tekst reprezentowany jest jako sekwencja tokenów i nie jest już rozpatrywany jako sekwencja znaków. Zasady podziału tekstu na tokeny są kwestia umowną. Najczęściej jest to podział tekstu po znakach interpunkcyjnych i białych znakach. W polskim często też dokonuje się słów [był][bym].

tokenizacja — podział tekstu na słów (w ogólności ciągłe sekwencje znaków będących słowami, liczbami, symbolami itd.). Zasady tokenizacji mogą się różnić w zależności od zastosowania. Najczęściej stosuje się podział po białych znakach i określonym zbiorze znaków interpunkcyjnych. Każda wydzielona sekwencja nazywa się tokenem.

związek binarny — patrz *relacja semantyczna*.