Politechnika Wrocławska

Wydział Informatyki i Zarządzania

Instytut Informatyki

Wrocław, Wybrzeże Wyspiańskiego 27

Rozprawa doktorska

Metody znakowania morfosyntaktycznego i automatycznej płytkiej analizy składniowej języka polskiego

Adam Radziszewski

Promotor: prof. dr hab. inż. Zbigniew Huzar

Wrocław, 2012

Spis treści

Streszczenie

Rozprawa porusza dwa powiązane ze sobą problemy z dziedziny przetwarzania języka naturalnego: znakowanie morfosyntaktyczne oraz płytką analizę składniową. Problemy te rozpatrywane są w kontekście języka polskiego.

Znakowanie morfosyntaktyczne to zadanie klasyfikacji wystąpień słów i innych segmentów występujących w tekście za pomocą *tagów*, tj. znaczników określających własności morfologiczno-składniowe tych wystąpień. Chociaż badania takie prowadzone są dla języka polskiego od ok. 10 lat, istniejące *tagery* (programy, których zadaniem jest znakowanie morfosyntaktyczne) wykazują wciąż dotkliwe braki, a błędy przez nie popełniane przekładają się na widoczne pogorszenie jakości systemów przetwarzania języka polskiego. Znakowanie morfosyntaktyczne języka polskiego, podobnie jak innych języków słowiańskich, jest zadaniem znacznie trudniejszym niż znakowanie języka angielskiego. Dzieje się tak, gdyż języki słowiańskie charakteryzują się z jednej strony swobodnym szykiem wyrazów w zdaniu, z drugiej zaś — mnogością form spowodowaną bogatą odmianą. Ta charakterystyka sprawia, że zadanie to staje się szczególnie ciekawym wyzwaniem z punktu widzenia informatycznego.

Praca dokonuje krytycznej oceny istniejących tagerów języka polskiego. Analiza błędów przez nie popełnianych wskazuje na problem rozpoznawania słów nieznanych, który był dotąd traktowany w sposób niezadowalający. Przedstawiamy również nowe spojrzenie na problem słów nieznanych, które pozwala na uogólnienie do dowolnych rozbieżności między danymi pochodzącymi z zewnętrznego analizatora morfosyntaktycznego a dostępnymi danymi uczącymi. W pracy proponujemy również nową metodę znakowania morfosyntaktycznego, która łączy kilka znanych dotychczas technik (m.in. znakowanie warstwowe i uczenie pamięciowe), a także pozwala zmniejszyć skalę występowania wspomnianych problemów.

Płytka analiza składniowa to ogólne określenie zadań przetwarzania języka naturalnego, których wynikiem jest przypisanie zdaniom częściowej struktury składniowej. Jedną z jej form jest *znakowanie fraz*, gdzie struktura składniowa ograniczona jest do wyróżnienia w tekście płaskich fraz składniowych. Takie frazy znajdują zastosowanie w systemach przetwarzania języka, m.in. wydobywania informacji z tekstu, a także w sytemach przypisujących dokumentom słowa kluczowe. Prawie wszystkie dotychczasowe badania prowadzone dla języków słowiańskich ograniczały się do ręcznego pisania gramatyk rozpoznających frazy. Wadą takiego rozwiązania jest ścisłe uzależnienie opracowanego analizatora składniowego nie tylko od danego języka, lecz także od przyjętych definicji fraz. Stan badań dla języka angielskiego przedstawia się zupełnie inaczej: znaczna część prac zakłada użycie technik maszynowego uczenia. W pracy podjęto próbę dostosowania do języka polskiego trzech metod znakowania fraz korzystających z maszynowego uczenia, które były z powodzeniem stosowane dla języka angielskiego.

W rozprawie opisano szereg przeprowadzonych badań eksperymentalnych nad opracowanymi metodami. Badania uwzględniają także ocenę wpływu błędów tagera na wyniki znakowania fraz, a także wpływu wyboru metody znakowania fraz na wyniki innych zadań przetwarzania języka naturalnego. Prezentowane eksperymenty pokazują skuteczność opracowanych metod.

Podziękowania

Chciałbym podziękować mojemu promotorowi, profesorowi Zbigniewowi Huzarowi, za wsparcie i motywację do pracy, a także przypominanie o zdrowym rozsądku. Podziękowania kieruję również do Macieja Piaseckiego, któremu zawdzięczam zainteresowanie przetwarzaniem języka naturalnego oraz możliwość pracy w grupie naukowej zajmującej się tym zagadnieniem.

Dziękuję również Markowi Maziarzowi i Janowi Wieczorkowi za wielogodzinne dyskusje dotyczące znakowania fraz w języku polskim. Markowi Maziarzowi dziękuję również za uwagi na temat części rozprawy poświęconej definicjom fraz w językach słowiańskich. Podziękowania należą się także Katarzynie Głowińskiej za dyskusje i wyjaśnienia dotyczące fraz znakowanych w Narodowym Korpusie Języka Polskiego. Dziękuję też Milošowi Jakubíčkowi oraz Wojtkowi Kovářowi za wyjaśnienia dotyczące brneńskich wytycznych znakowania fraz.

Dziękuję także Szymonowi Acedańskiemu za dyskusje na temat problemu oceny tagerów, a także owocną współpracę nad opracowywaniem nowej metodyki oceny tegerów. Szczególne podziękowania kieruję do Agnieszki Mykowieckiej za pomoc w przeprowadzeniu eksperymentów z wydobywaniem terminologii z korpusu ekonomicznego. Dziękuję również Bartoszowi Brodzie oraz Dominikowi Piaseckiemu za pomoc w eksperymentach związanych z wydobywaniem miary powiązania znaczeniowego.

Chciałbym też podziękować Adamowi Pawlaczkowi za pomoc w konwersji danych i przeprowadzeniu oceny parsera Spejd.

Spis rysunków

Spis tablic

Rozdział 1

Wprowadzenie

1.1. Przetwarzanie języka naturalnego

Tematyka rozprawy mieści się w dziedzinie *przetwarzania języka naturalnego* (ang. *Natural Language Processing*, *NLP*). Dziedzina ta leży na pograniczu informatyki i językoznawstwa. Podjęto kilka prób jej definicji; w niniejszej pracy powołujemy się na wyjaśnienie Przepiórkowskiego, według którego przetwarzaniem języka naturalnego są *wszelkie prace zmierzające do automatycznego tworzenia lub przetwarzania wypowiedzeń*[1]*, związane ze znaczeniem lub strukturą lingwistyczną tych wypowiedzeń* (Przepiórkowski, 2008, rozdz. 1.).

Powyższe określenie przetwarzania języka naturalnego jest dość szerokie. Obejmuje z jednej strony zarówno analizę istniejących wypowiedzeń, jak i generowanie nowych; z drugiej zaś, przedmiotem przetwarzania może być tekst, ale również i mowa. W niniejszej rozprawie zajmować się będziemy jedynie analizą wypowiedzeń. Co więcej, ograniczamy się tu do analizy *tekstu* w języku naturalnym.

Większość badań prowadzona jest dla języka angielskiego. Badania nad przetwarzaniem języka polskiego też są prowadzone, choć prac takich jest mniej. Niniejsza rozprawa skupia się właśnie na przetwarzaniu tekstu w **języku polskim**.

1.1.1. Cele przetwarzania tekstu

Ostatecznym celem analizy tekstu w języku naturalnego jest na ogół jego *rozumienie*. Marzeniem naukowców i inżynierów jest budowa praktycznych rozwiązań, które w oparciu o analizę tekstu są w stanie przynieść dotychczas nieosiągalne korzyści; przykładowo są to (na podstawie Piasecki, 2008):

— systemy wydobywania informacji (ang. *Information Extraction*), które znajdują w internecie dokładnie te (i tylko te) informacje, których potrzebuje użytkownik systemu;

— systemy automatycznego odpowiadania na pytania zadane w języku naturalnym (ang. *Question Answering*);

[1] *Wypowiedzienie* to zdanie lub równoważnik zdania (Polański, 1999c).

— systemy pozyskujące sformalizowaną wiedzę dzięki „czytaniu" książek i podręczników;
— systemy przełamujące bariery językowe, pozwalające na swobodną komunikację z użytkownikami posługującymi się różnymi językami.

Chociaż osiągnięcie powyższych celów wydaje się wciąż odległe, poczynione dotąd postępy pozwoliły na realizację mniejszych praktycznych przedsięwzięć, przykładowo (na podstawie Przepiórkowski, 2008; van Halteren, 1999):
— wydobywanie informacji ograniczone do danej dziedziny;
— systemy wspomagające tworzenie słowników poprzez udostępnienie syntetycznej informacji o wyrazach lub frazach pozyskanej na podstawie analizy wielkich zbiorów tekstu;
— wspomaganie pracy językoznawcy: wyszukiwanie trafnych przykładów potwierdzających daną hipotezę na temat języka;
— systemy tłumaczenia maszynowego o wystarczającej jakości, by pozwolić na przeglądanie stron internetowych w nieznanym języku.

1.2. Podstawowe pojęcia

W tej części ustalamy rozumienie kluczowych pojęć używanych w tej pracy.

1. **Korpus** jest „dowolnym zbiorem tekstów wykorzystywanych do badań — przede wszystkim językoznawczych" (Rudolf, 2004). Jeśli teksty korpusu wzbogacone są o informacje o charakterze lingwistycznym, powiemy, że korpus jest **oznakowany** (ang. *annotated*; McEnery i Wilson, 2001, s. 32). Jeśli ta informacja została dodana przez lingwistów (w odróżnieniu od tego rodzaju informacji dodanej w sposób automatyczny przez programy komputerowe), powiemy, że korpus zawiera **ręczne** bądź **wzorcowe oznakowanie**.
2. **Segmentem** (tokenem) nazywamy każde wystąpienie w tekście wyrazu, znaku interpunkcyjnego, ciągu cyfr lub symboli[2] (w dobrze uzasadnionych wypadkach za segment można też uznać fragment wyrazu). Przyjmujemy tutaj założenie, że segmenty są ciągłe oraz rozłączne (wzorem Przepiórkowski, 2004, s. 19). Pojęcia segmentu nie da się ściśle zdefiniować, gdyż w zależności od zastosowań, przyjętej tradycji lingwistycznej oraz znakowanego języka, przyjmuje się różne strategie segmentacji.
3. Pojęcie **zdania**[3] będziemy stosować intuicyjnie, gdyż jest ono niezwykle trudne do zdefiniowania. Liczne próby jego definicji można znaleźć w pracy Rudolfa (2004). Ponieważ sam Rudolf podaje w wątpliwość trafność przytaczanych definicji, tutaj pojęcia tego też nie definiujemy. Zakładamy jedynie, że zdanie jest reprezentowane w tekście poprzez ciąg segmentów, oraz, że zdania są ciągłe i rozłączne. Z informatycznego punktu widzenia brak definicji zdania i segmentu nie jest dużą przeszkodą, ponieważ zakładamy, że mamy do dyspozycji korpus oznakowany przez

[2] Odpowiada to temu, co Rudolf (2004) nazywa *napisem*.

[3] Dla uproszczenia, pojęcia zdania używać będziemy zarówno w odniesieniu do faktycznych zdań, jak i równoważników zdań — czyli w odniesieniu do tego, co językoznawcy nazywają ogólnie *wypowiedzeniami*. Słowo *zdanie* wydaje się bardziej intuicyjne, a wprowadzenie dodatkowego rozróżnienia zmniejszyłoby tylko czytelność, skoro żadne z tych pojęć i tak nie jest definiowane.

lingwistów, którzy podjęli właściwe decyzje przy rozróżnianiu konkretnych wystąpień zdań i segmentów.

4. **Tag** (znacznik morfosyntaktyczny, ang. *morpho-syntactic description tag*, *MSD tag*) to symbol, który można przypisać segmentowi, określający jego własności morfologiczno-składniowe. Prawie zawsze symbol taki określa jakieś przybliżenie *części mowy* segmentu; poza tym może określać własności o charakterze fleksyjnym (tj. związane z odmianą, np. przypadek rzeczownika), składniowym (np. z jakim przypadkiem łączy się dany przyimek), a niekiedy także i semantycznym (np. że dana forma jest nazwą własną). W językach o prostej morfologii (np. angielski) tagi są określane często mianem *znaczników części mowy* (ang. *Part-of-Speech tags*, *PoS tags*), gdyż określają one niewiele więcej ponad część mowy segmentu. Tagi takie traktowane są jako symbole niepodzielne. W przypadku języków o bogatej fleksji, takich jak język polski czy inne języki słowiańskie, tagi zwykle traktuje się jako symbole, które składają się z klasy gramatycznej (klasy słowa, czasem nazywanej dla uproszczenia częścią mowy) oraz, w zależności od tej klasy, wartości różnych atrybutów (kategorii gramatycznych[4]). Przykładowo, polskie rzeczowniki odmieniają się przez przypadek, a ponadto konkretne formy rzeczownikowe mają określoną wartość liczby i rodzaju; dlatego też tag określający rzeczownik składa się typowo z symbolu określającego klasę rzeczowników oraz z symboli odpowiadającym wartościom trzech atrybutów: przypadka, liczby i rodzaju.
5. **Tagset** (zestaw znaczników morfosyntaktycznych) to z matematycznego punktu widzenia zbiór możliwych tagów. W praktyce tego pojęcia używa się w szerszym rozumieniu; w takim ujęciu tagset określany jest przez:
   a) zbiór symboli reprezentujących używane klasy gramatyczne, wraz z ich słownymi opisami,
   b) zbiór symboli reprezentujących używane atrybuty, wraz z ich nazwami,
   c) zbiór symboli reprezentujących używane wartości atrybutów, wraz z ich nazwami,
   d) przypisanie klasom gramatycznym zbiorów atrybutów, których wartość musi zostać podana dla danej klasy oraz niekiedy zbiorów atrybutów opcjonalnych,
   e) przypisanie atrybutom zbiorów możliwych wartości,
   f) składnię tekstowej reprezentacji tagów,
   g) kryteria rozróżnienia klas gramatycznych oraz wartości atrybutów w sytuacjach praktycznych,
   h) strategię segmentacji tekstu (tagi przypisywane są do segmentów, a zatem istotne jest ustalenie, czym będą segmenty).
6. **Lemat** (forma podstawowa, forma hasłowa) to forma wyrazowa o określonych wartościach atrybutów wybrana jako reprezentująca cały zbiór form danego leksemu, np. mianownik liczby pojedynczej dla rzeczowników (Piasecki, 2008). Lemat zwykle odpowiada hasłu, pod którym można znaleźć daną formę w słowniku.

[4] W tej pracy używamy głównie neutralnego określenia *atrybuty*. Określenie to przede wszystkim wydaje się bardziej intuicyjne z informatycznego punktu widzenia — można mówić o parach *atrybut–wartość*. Poza tym, nie wszystkie własności wyróżniane w tagsetach odpowiadają rzeczywistym kategoriom gramatycznym. Np. trudno jest nazwać atrybut „wymaganie kropki" stosowany w tagsecie korpusu NKJP (Przepiórkowski i Szałkiewicz, 2012) kategorią *gramatyczną* (atrybut określa, czy forma będąca skrótem wymaga kropki).

7. Dla uproszczenia dalszego opisu, parę (*tag*, *lemat*) nazwiemy **interpretacją morfosyntaktyczną** segmentu.
8. **Problem znakowania ciągu** (ang. *sequence labelling problem*) to zadanie klasyfikacji polegające na przypisanie ciągowi (a_n) o elementach ze zbioru A ciągu (b_n) tej samej długości o elementach ze zbioru B (ogólnie, jest to przekształcenie typu $\mathbf{L} : A^* \to B^*$). Ponieważ przetwarzany ciąg może mieć dowolną długość, w praktyce stosuje się często modele przybliżone, gdzie przyjmuje się, że symbol b_i przypisany symbolowi a_i zależy jedynie od otoczenia $O(a_i) = (a_{i-d}, a_{i-d+1}, \ldots, a_{i+d-1}, a_{i+d})$ przy ustalonej wartości d. W takim ujęciu problem sprowadza się do klasyfikacji otoczeń $O(a_i) \in A^{2d+1}$ w elementy ze zbioru B (Dębowski, 2001).

1.2.1. Poziomy przetwarzania

Ze względu na duży stopień złożoności przedsięwzięć w dziedzinie przetwarzania tekstu, standardową praktyką jest podział prac na *poziomy przetwarzania*. Poziomy te są pochodną poziomów opisu języka przyjętych w lingwistyce. Każdy poziom przetwarzania wiąże się ze swoim poziomem abstrakcji, odpowiadającym w dużym stopniu odrębnemu działowi językoznawstwa. Niezależnie od ostatecznego celu danego zadania przetwarzania języka, początek przetwarzania wygląda zwykle podobnie. Co więcej, pierwsze etapy przetwarzania uległy pewnej standaryzacji, co ułatwia współpracę między naukowcami i inżynierami oraz pozwala na ponowne wykorzystanie dotychczas opracowanych komponentów. Pierwsze etapy przetwarzania tekstu w języku polskim wyglądają zwykle w ten sposób (na podstawie Przepiórkowski, 2008, rozdz. 2):

1. Segmentacja — wstępne przetwarzanie tekstu, które prowadzi do wyodrębnienia w ciągłym tekście segmentów oraz zdań.
2. Znakowanie morfosyntaktyczne, tj. klasyfikacja segmentów, owocująca przypisaniem segmentom znaczników opisujących własności morfologiczno-składniowe. Poziom ten odpowiada przede wszystkim działowi językoznawstwa zwanym *morfologią*, a w pewnym stopniu dotyczy też składni — gdyż dotyczy cech warunkujących zachowanie składniowe segmentów (stąd przymiotnik *morfosyntaktyczny*).
3. Analiza składniowa, tj. przypisanie zdaniom struktur składniowych. W zależności od zastosowania i dostępnych zasobów, może być to analiza głęboka (owocująca pełnym rozbiorem zdania) lub analiza płytka (wyodrębniana jest tylko struktura częściowa; bywa różnie określana).

Często dalszym etapem przetwarzania jest analiza semantyczna (tj. analiza znaczenia słów, zdań bądź całych tekstów), choć istnieją zastosowania, gdzie etap ten nie jest konieczny (np. systemy wspomagające tworzenie słowników często poprzestają na analizie składniowej). Niniejszej praca skupia się na etapach 2 i 3.

1.2.2. Znakowanie morfosyntaktyczne

Znakowanie morfosyntaktyczne (ang. *morpho-syntactic tagging*, *MSD tagging*), czasem określane także *znakowaniem częściami mowy* (ang. *Part-of-Speech tagging*, *PoS tagging*), polega na przypisaniu segmentom w tekście interpretacji morfosyntaktycznych[5]. Znakowanie wykonywane jest kontekstowo, tj. dwa wystąpienia tego samego

[5] Często spotyka się definicje, gdzie segmentom przypisywane są jedynie tagi; praktyka pokazuje jednak, że rzeczywiste implementacje algorytmów znakowania zazwyczaj przypisują też lematy. Taka

segmentu mogą mieć przypisane różne interpretacje w zależności od kontekstu. Przykładowo, w zdaniu 1.1 pierwsze wystąpienie formy kurze oznakowane zostanie jako rzeczownik o lemacie kura, natomiast drugie wystąpienie — jako rzeczownik o lemacie kurz; co więcej, segmentom tym zostaną przypisane tagi określające rzeczownik, lecz o innych wartościach liczby, rodzaju i przypadka.

(1.1) Kazał kurze ścierać kurze.

Z formalnego punktu widzenia, znakowanie morfosyntaktyczne jest problemem znakowania ciągu. To przypisanie każdemu segmentowi w tekście dokładnie jednej interpretacji (definicja wyidealizowana); tj. $\mathbf{Z} : W^* \rightarrow (T \times L)^*$ (W to zbiór segmentów, T to zbiór tagów, a L jest zbiorem lematów). W rzeczywistości czasem dopuszcza się sytuacje, gdy jednemu segmentowi przypisanych jest więcej tagów — niektóre zdania są bowiem inherentnie wieloznaczne i wybór pojedynczego tagu byłby co najwyżej arbitralny (Przepiórkowski, 2004). W takim ujęciu znakowanie morfosyntaktyczne uzyskuje formalizację $\mathbf{Z} : W^* \rightarrow (2^{T \times L})^*$.

Z przyczyn praktycznych często znakowanie morfosyntaktyczne wykonuje się dwuetapowo — pierwszym etapem jest wtedy *analiza morfosyntaktyczna*, drugim — *ujednoznacznianie morfosyntaktyczne*.

Analiza morfosyntaktyczna to problem znakowania ciągu, gdzie każdemu segmentowi w tekście przypisujemy niepusty zbiór interpretacji przy założeniu, że zbiór interpretacji przypisywany segmentowi jest niezależny od kontekstu wystąpienia tego segmentu. Ze względu na tę niezależność od kontekstu, analizę morfosyntaktyczną można sprowadzić do odwzorowania $\mathbf{M} : W \rightarrow 2^{T \times L}$. Tej samej formie wyrazowej (segmentowi) przypisany zostanie zawsze ten sam zbiór interpretacji. Segmentom przypisuje się zbiory, ponieważ mamy do czynienia z *wieloznacznością* — ta sama forma może w różnych kontekstach mieć różne interpretacje. Drugi etap nazywa się wtedy **ujednoznacznianiem morfosyntaktycznym** (ang. *morpho-syntactic disambiguation*). Ujednoznacznianie można sformalizować jako przekształcenie $\mathbf{D} : (2^{T \times L})^* \rightarrow (T \times L)^*$ (przypadek idealny) lub $\mathbf{D} : (2^{T \times L})^* \rightarrow (2^{T \times L})^*$ (gdy dopuścimy wieloznaczność na wyjściu).

Program wykonujący znakowanie morfosyntaktyczne określany jest **tagerem** (ang. *morpho-syntactic tagger*). Pojęcie to nie jest ściśle zdefiniowane; może odnosić się do implementacji różnych wycinków potoku przetwarzania, o ile wyjściem takiego wycinka jest ciąg segmentów z przypisanymi interpretacjami morfosyntaktycznymi (albo samymi tagami). W praktyce pojęcie to najczęściej odnosi się do implementacji całego potoku przetwarzania, który zaczyna się od segmentacji a kończy na znakowaniu morfosyntaktycznym. W takim ujęciu, tager na wejściu przyjmuje czysty tekst.

1.2.3. Płytka analiza składniowa

Jak wspomniano wyżej, analiza składniowa może być realizowana w różnoraki sposób. Celem *analizy głębokiej* jest znalezienie pełnego rozbioru zdania. Chociaż z punktu widzenia wielu zastosowań byłby to efekt pożądany, jego osiągnięcie jest trudne. Prze-

sytuacja ma w szczególności miejsce dla języka polskiego, stąd też w niniejszej pracy przyjmujemy definicję pełną. Mimo to, w centrum naszego zainteresowania pozostaje prawidłowe przypisanie tagów, lematy zaś będziemy traktować jako informację dodatkową.

piórkowski (2008) podaje trzy zasadnicze problemy, których przysparza analiza głęboka:

1. Opracowanie gramatyki potrzebnej do wykonania takiej analizy jest bardzo czasochłonne.
2. Gdy gramatyka taka zostanie już napisana, jej rozmiar i stopień skomplikowania skutecznie utrudnia dalszy jej rozwój i utrzymanie — często sami twórcy nie są w stanie zapanować nad zależnościami między jej regułami.
3. Analizatory wykonane w ten sposób mają tendencję do generowania wielu kandydujących rozbiorów jednego zdania; wybór rozbioru właściwego jest zadaniem trudnym. Problem ten potwierdzają przytaczane przez Przepiórkowskiego doświadczenia z analizą głęboką języka polskiego przy pomocy gramatyki formalnej Marka Świdzińskiego (1992). Zauważono, że implementacja tej gramatyki (parser *Świgra* Marcina Wolińskiego) dla niektórych zdań znajduje ponad tysiąc możliwych rozbiorów, a można znaleźć zdania, gdzie liczba ta przekracza milion (Woliński, 2004).

Problemy te dotyczą przede wszystkim analizatorów opartych o ręcznie pisane gramatyki. Od wielu lat prowadzone są także badania nad uczeniem parserów na podstawie korpusów wzorcowych, np. (Collins, 1999; Charniak, 2000; Nivre, 2003). Korpusy takie składają się z ręcznie przeprowadzonych rozbiorów składniowych zdań (są to tzw. *banki drzew*, ang. *treebanks*). Od niedawna badania takie prowadzone są również dla języka polskiego (Wróblewska i Woliński, 2011). Wadą tego typu rozwiązań jest konieczność posiadania dużego banku drzew[6], podczas gdy znakowanie każdego zdania pełnym rozbiorem jest bardzo pracochłonne.

Alternatywnym podejściem jest **płytka analiza składniowa** (ang. *shallow parsing*; czasem zwana także *analizą częściową* — *partial parsing* — bądź *powierzchniową* — *surface parsing*). Jest to pojęcie ogólne, które odnosić się może do wszelkich form analizy składniowej, gdzie znajdujemy jedynie częściową strukturę składniową. Jedną z form płytkiej analizy składniowej jest **znakowanie fraz**[7] (ang. *chunking*). Takie ujęcie problemu pochodzi od Abneya (1991) i sprowadza się do przypisania zdaniom następującej struktury:

— zdanie zostaje podzielone na fragmenty (będące ciągami segmentów),
— fragmenty te są klasyfikowane: każdemu fragmentowi przypisywana jest albo nazwa frazy (jedna z kilku z góry ustalonych), albo fragment określany jest jako nienależący do żadnej interesującej nas frazy.

Popularną formą znakowania fraz jest *znakowanie fraz rzeczownikowych* (ang. *NP chunking*). Rozpatrujemy tu jedynie frazy rzeczownikowe, a więc fragment może być oznaczony jako fraza rzeczownikowa lub fragment niebędący nią (Ramshaw i Marcus, 1995).

---

[6] W momencie pisania rozprawy budowany jest bank drzew języka polskiego zwany *Składnicą*. Bank zawiera obecnie ok. 8000 drzew (Woliński i inni, 2011).

[7] Maciej Piasecki zaproponował termin *całostka* jako tłumaczenie angielskiego *chunk* (Piasecki, 2008). Późniejsza dyskusja ze Stanisławem Szpakowiczem była jednak powodem do odejścia od tego terminu — określenie *całostka* jest używane w gramatyce Świdzińskiego (1992) w nieco innym znaczeniu — a więc nadawanie mu nowego znaczenia mogłoby prowadzić do niepożądanego zamieszania. W literaturze słowiańskiej można jednak spotkać ciekawe tłumaczenia — przykładowo Kristina Vučković w swoim doktoracie (2009) używa chorwackich terminów *razdjeljivanje* („rozdzielanie"), *razdjelnik* („rozdzielacz", czyli analizator składniowy, które znakuje takie frazy).

Cechą charakterystyczną znakowania fraz jest to, że rozpoznajemy jedynie ich granice oraz typ. Inne ujęcia płytkiej analizy składniowej pozwalają na częściowy opis struktury fraz — w takim wypadku fraza może zawierać frazy innego typu lub, w niektórych ujęciach, nawet inne frazy tego samego typu co fraza nadrzędna (Przepiórkowski, 2008).

Program wykonujący którąś z form analizy składniowej nazywany jest **analizatorem składniowym** albo **parserem** (ang. *parser*, od łac. *pars orationis — część mowy*). Niekiedy używać będziemy także skrótowych określeń **płytki** bądź **głęboki parser** w odniesieniu do programów, których zadaniem jest przeprowadzenie, odpowiednio, płytkiej bądź głębokiej analizy składniowej.

Zdecydowana większość prac poświęcona płytkiej analizie składniowej języków słowiańskich zakłada użycie parserów korzystających z gramatyk napisanych ręcznie. Zaletą takiego podejścia jest możliwość opracowania gramatyki bez dostępu do dużego korpusu oznakowanego ręcznie (chociaż ocenę takiego parsera trudno przeprowadzić bez dostępu do choćby niewielkiego korpusu wzorcowego). Podejście takie ma jednak kilka wad. W szczególności, opracowana gramatyka przywiązana jest nie tylko do danego języka; jest ona również ściśle uzależniona od przyjętych definicji fraz, a być może nawet od dziedziny tekstu. Użycie takiego parsera do analizy składniowej innego, choćby bardzo podobnego, języka, wymagałoby gruntownej przebudowy gramatyki. Zmiana definicji fraz, które z założenia ma rozpoznawać taki system, również pociąga za sobą konieczność przejrzenia całej gramatyki i dokonania korekt. Należy się spodziewać wystąpienia części problemów, o których wspomnieliśmy przy omawianiu analizy głębokiej. W szczególności, korekta jednej reguły może wymagać dokonania rewizji innych reguł.

W przypadku parserów uczących się na podstawie korpusu wzorcowego powyższe problemy nie występują. Ręczne oznakowanie korpusu jest procesem kosztownym, jednak gdy taki korpus już powstanie, może on być podstawą do budowy wielu praktycznych analizatorów składniowych. Dla języka polskiego istnieją od niedawna dwa takie korpusy. Podobna sytuacja ma miejsce dla języka chorwackiego oraz bułgarskiego, podczas gdy trudno jest znaleźć prace poświęcone płytkiej analizie składniowej języków słowiańskich w oparciu o techniki maszynowego uczenia[8]. Dla porównania, znaczna część prac poświęcona płytkiej analizie składniowej języka angielskiego zakłada użycie takich technik. Niniejsza rozprawa wypełnienia tę lukę, gdyż przedstawia metody znakowania fraz w języku polskim korzystające z technik maszynowego uczenia.

1.3. Zakres i cel

W pracy tej poruszane są dwa problemy: **znakowanie morfosyntaktyczne** oraz **płytka analiza składniowa** tekstu polskiego. Zakres analizy składniowej ograniczony został do **znakowania fraz**.

Celami pracy są:

1. udoskonalenie znanych dotychczas metod znakowania morfosyntaktycznego języka polskiego korzystających z technik maszynowego uczenia,

[8] Pomijamy tutaj prace powstałe przy udziale autora tej rozprawy.

2. opracowanie metody automatycznego znakowania fraz w języku polskim, która uczyć się będzie na korpusie oznakowanym ręcznie.

Znakowanie morfosyntaktyczne jest zadaniem kluczowym dla większości praktycznych systemów przetwarzania języka naturalnego, w tym systemów wydobywania informacji z tekstu, odpowiadania na pytania oraz systemów wspomagających pracę leksykografów i lingwistów. Niejednokrotnie zaobserwowano, że błędy popełnione na etapie znakowania morfosyntaktycznego przenoszą się na kolejne warstwy przetwarzania, pogarszając jakość działania końcowych systemów. Opracowanie metody znakowania morfosyntaktycznego języka polskiego, która rzadziej popełnia błędy niż znane obecnie rozwiązania, przyczyni się zatem do poprawy jakości działania praktycznych systemów przetwarzania języka naturalnego.

Znakowanie fraz jest uproszczoną formą analizy składniowej. Znajduje praktyczne zastosowania, m.in. w systemach wydobywania informacji, odpowiadaniu na pytania oraz w systemach przypisujących dokumentom słowa kluczowe. Metody znakowania fraz, które uczą się na korpusach wzorcowych, cechuje duża elastyczność. W szczególności nie są uzależnione od przyjętej definicji fraz. Opracowanie takiej metody dla języka polskiego przyniesie postęp w przyszłych pracach związanych z budową praktycznych systemów przetwarzania języka polskiego.

W ramach realizacji celu wyróżniono następujące zadania badawcze:

1. Przebadanie algorytmów znakowania morfosyntaktycznego stosowanych dla języka polskiego pod kątem rozpoznania ich słabych i mocnych stron.
2. Opracowanie ulepszonego algorytmu znakowania morfosyntaktycznego języka polskiego.
3. Przegląd praktyk i przyjmowanych definicji fraz stosowanych dla zadania znakowania fraz w językach słowiańskich.
4. Opracowanie wytycznych znakowania fraz w języku polskim we współpracy z lingwistami.
5. Dostosowanie znanych metod znakowania fraz opartych na maszynowym uczeniu się do specyfiki języka polskiego.
6. Przeprowadzenie badań eksperymentalnych opracowanych metod znakowania morfosyntaktycznego i znakowania fraz. Analiza wyników.

1.4. Teza

W pracy postawiono następującą tezę:

> Metody znakowania morfosyntaktycznego i płytkiej analizy składniowej oparte na technikach maszynowego uczenia umożliwiają budowę praktycznych systemów przetwarzania języka polskiego.

Tak sformułowanej tezy nie sposób udowodnić na gruncie formalnym. W pracy dokonamy jej uwiarygodnienia poprzez wskazanie dwóch systemów przetwarzania języka polskiego, gdzie wspomniane metody znalazły zastosowanie, mianowicie systemu budującego słownik dziedzinowy na podstawie automatycznej analizy korpusu językowego oraz systemu wydobywającego relacje semantyczne między wyrazami.

Rozwój metod znakowania języka polskiego w oparciu o techniki maszynowego uczenia jest istotnym kierunkiem badań ze względu na dużą elastyczność takich rozwiązań.

Opracowane metody wyuczyć można nie tylko na istniejących obecnie korpusach języka polskiego oznakowanych ręcznie, ale także na korpusach opracowanych w przyszłości. Co więcej, metody te mogą zostać zastosowane do znakowania innych języków słowiańskich. Wykazanie przydatności takich rozwiązań w praktycznych systemach przetwarzania języka polskiego przemawia również za ich przydatnością w podobnych systemach budowanych dla innych języków słowiańskich.

1.5. Struktura rozprawy doktorskiej

W rozdziale 2 opisano problem znakowania morfosyntaktycznego. Dyskusja obejmuje problem oceny tagerów, dostępne zasoby dla języka polskiego, a także znane z literatury metody znakowania morfosyntaktycznego. Rozdział zawiera także propozycję własnej metody łączącej technikę znakowania warstwowego z techniką uczenia na pamięć, a także jej dwa rozszerzenia związane z problemem słów nieznanych i słownikiem analizatora morfosyntaktycznego (zadanie badawcze 2). Rozdział 3 przedstawia wyniki eksperymentalnej oceny metod — zarówno znanych z literatury, jak i zaproponowanych w rozprawie. Podział na rozdziały zatem nie odzwierciedla w pełni kolejności zadań badawczych. Taką strukturę przyjęto dla przejrzystości: najpierw prezentujemy opis metod i algorytmów (zarówno znanych, jak i proponowanych), a następnie ich eksperymentalną ocenę.

Rozdział 4 omawia problematykę płytkiej analizy języka polskiego, a przede wszystkim problem znakowania fraz. Znaczną część rozdziału poświęcono na przegląd praktyk i definicji fraz stosowanych dla zadania znakowania fraz w językach słowiańskich. W dalszym ciągu omówione zostały metody płytkiej analizy składniowej — zarówno te stosowane dla języka angielskiego, jak i języków słowiańskich. Kolejnym punktem jest opis procedury, dzięki której za pomocą płytkiego parsera Spejd można uzyskać strukturę odpowiadającą problemowi znakowania fraz (taka procedura jest konieczna, by porównać Spejd z metodami znakowania fraz). Następnie opisano trzy metody znakowania fraz korzystające ze znanych algorytmów maszynowego uczenia oraz dostosowanego do specyfiki języka polskiego zestawu cech (metody te są wynikiem realizacji zadania badawczego 5). Rozdział 5 przedstawia wyniki eksperymentalnej oceny metod znakowania fraz, a także pilotażowe badania nad ich zastosowaniem w praktycznych systemach przetwarzania języka polskiego.

Rozprawę kończy krótkie podsumowanie najważniejszych osiągnięć pracy.

Rozdział 2

Znakowanie morfosyntaktyczne

Zgodnie z tym, co opisano w rozdziale 1, znakowanie morfosyntaktyczne polega na przypisaniu każdemu segmentowi występującemu w tekście *interpretacji morfosyntaktycznej*, tj. pary składającej się z tagu oraz lematu. Poniżej przedstawiamy przykładowe zdanie wraz z przypisanymi tagami, zgodnie z tzw. tagsetem NKJP (zostanie on dokładniej omówiony w punkcie 2.2, a jego pełną specyfikację przedstawiamy w dodatku B).

(2.1)

Kazał	*kurze*	*ścierać*	*kurze*
kazać	kura	ścierać	kurz
`praet:sg:m1:perf`	`subst:sg:dat:f`	`inf:imperf`	`subst:pl:acc:m3`

Na przykładzie 2.1 widać, że forma kurze może występować w roli rzeczownika (`subst`) rodzaju żeńskiego (`f`) kura, lecz również w roli rzeczownika rodzaju męskiego nieożywionego (`m3`) kurz. Formy występujące w zdaniu różnią się też wartością przypadka i liczby (symbole `dat` i `acc` odpowiadają odpowiednio dopełniaczowi i biernikowi; symbol `sg` oznacza liczbę pojedynczą, a `pl` — mnogą).

Widać tu wyraźnie, że ta sama forma może w zależności od kontekstu otrzymać różne znaczniki, a czasem też różne lematy. Przykładowe zdanie zawiera też dwa czasowniki: formę przeszłą (tzw. *pseudoimiesłów*, `praet`) kazał, w liczbie pojedynczej, rodzaju męskim osobowym oraz w aspekcie dokonanym (`sg:m1:perf`), jak również i bezokolicznik (`inf`) ścierać, którego aspekt został rozpoznany jako niedokonany. Na przykładzie widać również, że znaczniki stosowane w praktyce bywają dość szczegółowe: niosą znacznie więcej informacji niż tylko wskazanie części mowy.

W dalszej części tego rozdziału cytować będziemy wyniki oceny różnych tagerów. Najczęściej stosowaną metodą oceny tagera jest porównanie wyjścia tagera z korpusem wzorcowym oznakowanym przez lingwistę. Standardowa miara, zwana *trafnością*, określa procent segmentów, którym tager przypisał prawidłowe interpretacje. Problem oceny tagerów rozważany jest bardziej szczegółowo w punkcie 2.5. W pierwszej kolejności omówimy jednak zastosowania znakowania morfosyntaktycznego, dostępne korpusy języka polskiego oznakowane morfosyntaktycznie i stosowane w nich tagsety, a także metody znakowania znane z literatury. W dalszej części przedstawiamy problem oceny tagerów; tematyka oceny tagerów wprawdzie nie jest nowa, ale kilka ważnych jej aspek-

tów jest na ogół pomijanych w rozważaniach. Pokażemy, że większość z dotychczas publikowanych testów tagerów języka polskiego przeprowadzono nie w pełni rzetelnie i zaproponujemy alternatywną metodę oceny. Kolejną częścią jest propozycja metody znakowania opartej na uczeniu na pamięć oraz jej modyfikacja pozwalająca na lepsze znakowanie słów niewystępujących w słowniku analizatora morfosyntaktycznego. Ostatnią częścią rozdziału stanowi praktyczna metoda pozwalająca na lepsze użycie dostępnego korpusu uczącego — *ponowna analiza morfosyntaktyczna danych uczących.*

2.1. Zastosowania

Tager jest ważnym elementem typowego potoku przetwarzania tekstu (por. rozdział 1.2.1), w dużej mierze niezależnie od funkcji pełnionej przez cały system. Często podkreśla się ważną rolę znakowania morfosyntaktycznego jako etapu przetwarzania wymaganego przez analizę składniową. Jakość tego oznakowania ma istotny wpływ na wyniki analizy składniowej; w przypadku języków słowiańskich istotne jest nie tylko odgadnięcie części mowy segmentu; ważne są też wartości innych kategorii gramatycznych, np. przypadka (Hajič i inni, 2001; Acedański i Przepiórkowski, 2010). Eksperymenty przeprowadzone w tej rozprawie również potwierdzają istotny wpływ jakości tagera na osiągi analizatora składniowego (rozdział 5.6).

Rozpoznanie klas gramatycznych oraz lematyzacja są kluczowe z punktu widzenia wydobywania informacji z tekstu, ponieważ konieczne jest nie tylko odróżnienie czasowników od rzeczowników (rzeczowniki mogą reprezentować opisane w tekście byty, a czasowniki — relacje między nimi), lecz także rozpoznanie konkretnych jednostek leksykalnych (Feldman i Hana, 2010). Przykładowo, odróżnienie rzeczownika robot od rzeczownika robota może wymagać kontekstowego ujednoznaczniania, jeśli występująca w tekście forma to roboty. Podobne wymagania pojawiają się w przypadku innych zadań przetwarzania języka, np. automatycznego streszczania.

Korpusy oznakowane morfosyntaktycznie są użytecznym materiałem do badań lingwistycznych, a także stanowią nieocenioną pomoc w pracy leksykografa. Współczesne słowniki są w dużej mierze tworzone w oparciu o korpusy językowe, od których często oczekuje się, że będą również oznakowane morfosyntaktycznie (van Halteren, 1999, s. 33). *SketchEngine*, popularny system wspomagający budowę słowników (Kilgarriff i inni, 2004) został niedawno przy udziale autora tej rozprawy uzupełniony o wsparcie dla języka polskiego. Uzupełnienie to polegało na napisaniu reguł rozpoznających proste związki składniowe w korpusie języka polskiego. System SketchEngine używa zgromadzonych związków do pokazania użytkownikowi, w jakim kontekście używane jest dane słowo; np. jakie rzeczowniki są typowymi podmiotami danego czasownika. Reguły napisane dla języka polskiego (podobnie jak i dla innych języków) wymagają tekstu oznakowanego morfosyntaktycznie. Dzieje się tak, gdyż reguły odwołują się bezpośrednio do konkretnych klas gramatycznych, ale również do wartości konkretnych kategorii gramatycznych, np. przypadka. Podczas próby oceny działania systemu na tekstach polskich zaobserwowano, że część nieoczekiwanych wyników systemu wynika bezpośrednio z błędów popełnionych przez tager (Radziszewski i inni, 2011a).

Znakowanie morfosyntaktyczne tekstów polskich odgrywa również ważną rolę w przedsięwzięciu realizowanym na Politechnice Wrocławskiej, mianowicie w budowie

Słowosieci[1] — wielkiej leksykalnej bazy wiedzy, pełniącej m.in. rolę komputerowego słownika wyrazów bliskoznacznych, zawierającego też opis innych relacjach semantycznych między wyrazami (Piasecki i inni, 2009). Budowa Słowosieci w znacznej mierze opiera się na użyciu metod automatycznych, które pozwalają na przyspieszenie pracy lingwistów: system prezentuje prawdopodobne powiązania między wyrazami, a do lingwisty należy ostateczna decyzja, czy daną relację uznać za słuszną, zmodyfikować, bądź całkowicie odrzucić. Podpowiedzi takie były oparte o przesłanki pozyskane na podstawie automatycznej analizy wielkich korpusów tekstów polskich, poddanych znakowaniu morfosyntaktycznemu. Warto też podkreślić, że podczas realizacji przedsięwzięcia zauważono, że błędy popełniane przez tager w znacznym stopniu pogarszają jakość pozyskanej wiedzy (Piasecki i inni, 2009, s. 73). Pokazuje to celowość dalszych działań prowadzących do opracowania lepszych tagerów języka polskiego.

2.2. Korpusy i tagsety języka polskiego

Istnieje kilka korpusów języka polskiego, m.in. (na podstawie Przepiórkowski, 2008 oraz Górski i Łaziński (2012)):

1. korpus utworzony na podstawie *Słownika frekwencyjnego polszczyzny współczesnej* (FREK; pół miliona słów, polszczyzna lat 60-tych),
2. Korpus PWN (ok. 100 mln słów),
3. Korpus PELCRA (ok. 100 mln słów),
4. Korpus IPI PAN (KIPI; ok. 250 mln słów),
5. Narodowy Korpus Języka Polskiego (NKJP; 300 mln segmentów).

Większość tych danych nie jest jednak dostępna publicznie. Jedynie korpusy FREK, KIPI i NKJP są dostępne nieodpłatnie. Co więcej, w celu wyuczenia i przetestowania tagera, potrzebujemy korpusu zawierającego *wzorcowe oznakowanie morfosyntaktyczne*. Na szczęście korpusy KIPI i NKJP zawierają takie oznakowanie. W przypadku korpusu KIPI, ręcznie oznakowana część zawiera ok. 880 000 segmentów, z czego ok. 660 000 segmentów stanowi dane z wspomnianego korpusu FREK, oznakowane zgodnie z przyjętym w KIPI tagsetem. Wadą korpusu FREK jest to, że zawiera on polszczyznę lat 60-tych (Bień i Woliński, 2003). Niestety, jedynie ta właśnie część jest dostępna publicznie na otwartej licencji[2].

Sytuacja wygląda dużo lepiej w przypadku korpusu NKJP: tzw. *podkorpus milionowy NKJP* (Degórski i Przepiórkowski, 2012) w całości jest dostępny na wolnej licencji[3]. Korpus liczy ok. 1,2 mln segmentów, a jego teksty pochodzą ze współczesnych źródeł. Dlatego też korpus ten będzie traktowany jako główny zbiór danych. Warto od razu nadmienić, że podkorpus milionowy NKJP zawiera również oznakowanie na innych poziomach, w tym płytkie oznakowanie składniowe, z czego korzystamy w rozdziale 4.

Jak wspomniano w rozdziale 1.2, tagsety dla języków słowiańskich są na ogół dużo bardziej rozbudowane niż tagsety zdefiniowane dla języka angielskiego. Zwykle tagi składają się z klasy gramatycznej oraz szeregu atrybutów. Sytuacja taka ma miejsce

[1] Projekty finansowane przez Ministerstwo Nauki i Szkolnictwa Wyższego: 3 T11C 018 29 oraz N N516 068637.

[2] Dane te dostępne są na licencji GNU GPL na stronie `http://korpus.pl`.

[3] *Podkorpus milionowy NKJP* dostępny jest na licencji GNU GPL na stronie `http://clip.ipipan.waw.pl/LRT`.

również dla języka polskiego. Na potrzeby korpusu KIPI zdefiniowano nowy tagset, który od tej pory dla uproszczenia nazywać będziemy *tagsetem KIPI*. Głównym założeniem tagsetu jest podział na klasy gramatyczne według możliwie ścisłych kryteriów (Przepiórkowski i Woliński, 2003). W szczególności, autorzy odchodzą od tradycyjnego podziału na części mowy. Decyzja ta umotywowana jest nieprecyzyjną definicją tradycyjnych części mowy i nierozstrzygalnością z niej wynikającą. Przykładowo, tradycyjna kategoria *zaimków* zawiera w sobie zarówno formę nieodmienną *się*, formy odmieniające się jak przymiotniki, ale też formy o ograniczonej fleksji (np. nikt) (Przepiórkowski i Woliński, 2003). Co gorsza, podział ten w dużej mierze odwołuje się do poziomu semantyki, podczas gdy znakowanie morfosyntaktyczne z założenia ma być procesem możliwie niskopoziomowym. Rozwiązaniem przyjętym w tagsecie KIPI jest podział na 32 klasy gramatyczne, które w głównej mierze wyróżniono na podstawie odmiany wyrazowej, a więc kryterium stosunkowo łatwo rozstrzygalnego. Część decyzji odwołuje się do poziomu składniowego (a więc wymaga informacji o tym, z jakimi innymi wyrazami łączy się dana forma), natomiast wpływ poziomu semantycznego ograniczono do minimum. Rozstrzygnięcie klasy gramatycznej formy można w dużej mierze sprowadzić do serii pytań „tak/nie" w stylu: *Czy forma odmienia się przez przypadek? Czy forma ma określoną wartość osoby?* (Przepiórkowski i Woliński, 2003) Każdej z wyróżnionych w ten sposób klas przypisano zbiór atrybutów, których wartości muszą zostać określone. Przykładowo, klasie rzeczowników (`subst`) przypisano trzy atrybuty, mianowicie: liczbę, rodzaj i przypadek, klasie przymiotników (`adj`) przypisano cztery atrybuty: liczbę, rodzaj, przypadek i stopień, zaś klasie przysłówków (`adv`) — jedynie stopień. Występują też klasy bez atrybutów, np. klasa opisująca znaki interpunkcyjne i symbole graficzne (`interp`) (Przepiórkowski, 2004). Poniżej przedstawiamy kilka przykładowych tagów:

(2.2) `subst:pl:inst:n` — rzeczownik, liczba mnoga, narzędnik, rodzaj nijaki; np. drzewami

(2.3) `adj:sg:acc:f:comp` — przymiotnik, liczba pojedyncza, biernik, rodzaj żeński, stopień wyższy; np. dalszej

(2.4) `adv:pos` — przysłówek, stopień równy; np. głośno

Pewnym wyjątkiem od tej ścisłej definicji jest dopuszczenie *atrybutów opcjonalnych*, których wartość można pominąć i, mimo to, uzyskany tag będzie poprawny. Atrybuty opcjonalne są jednak w mniejszości i dotyczą raczej mniej istotnych rozróżnień. Przykładowo, przyimkom przypisano opcjonalny atrybut *wokaliczności*, który określa, czy forma przyimka podlega rozszerzeniu artykulacyjnemu polegającemu na wystąpieniu na końcu samogłoski *-e*. Atrybut ma dwie wartości: `wok` (rozszerzenie zaszło) i `nwok` (nie zaszło). Formie przeze przypisano by tag `prep:acc:wok`, formie przez — tag `prep:acc:nwok`. Wartość tego atrybutu jest określona jedynie dla tych przyimków, gdzie zjawisko to w ogóle zachodzi; w przypadku pozostałych przyimków atrybut ten nie ma przypisanej wartości; np. formie dla należałoby przypisać tag `prep:gen` (`gen` oznacza, że przyimek łączy się z dopełniaczem, `acc` — z biernikiem).

Konsekwencją rozróżnienia klas gramatycznych na podstawie ścisłych testów na odmianę wyrazową jest powstanie nieco sztucznej klasy tworzonej przez formy, które nigdzie indziej nie pasują; klasę tę nazwano *partykuło-przysłówkami* (`qub`), gdyż głównie

te właśnie części mowy tam trafiają. Grupa zawiera również inne nieodmienne formy, w tym formy dźwiękonaśladowcze i wykrzykniki oraz zaimek się.

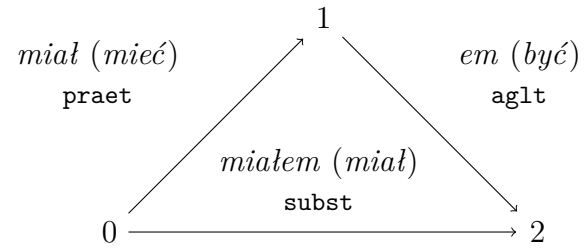
W tagsecie KIPI istnieje ponad 4000 teoretycznie możliwych tagów, choć w korpusie odnotowano ich tylko nieco ponad 1000 (Przepiórkowski, 2005).

Na podstawie tagsetu KIPI opracowano tagset korpusu NKJP (odtąd: *tagset NKJP*). Tagsety te są do siebie bardzo podobne: przyjmują te same założenia teoretyczne, a także zdecydowana większość klas gramatycznych i atrybutów przeniesiona została bezpośrednio z tagsetu KIPI. Powodem wprowadzenia modyfikacji były obserwacje teoretyczne i praktyczne doświadczenia zdobyte w wyniku kilkuletniego stosowania tagsetu KIPI (Przepiórkowski, 2009a). Modyfikacje te sprowadzają się przede wszystkim do wprowadzenia kilku dodatkowych klas gramatycznych i znacznego uporządkowania klasy *partykuło-przysłówków*: klasa ta w tagsecie NKJP nie jest już „zbiorem odrzutów", lecz zdefiniowana jest poprzez wyliczenie. Jedną z nowych w NKJP klas jest klasa skrótów (`brev`), która pozwala na lepszy opis wyrażeń typu prof. czy p. (klasa nie dotyczy akronimów, które opisywane są jako formy rzeczownikowe) (Przepiórkowski, 2009a). Specyfikację tagsetu NKJP zamieszczono w dodatku B.

W praktyce tagset wiąże się z konkretną strategią segmentacji — jest bowiem konieczne ustalenie kryteriów wydzielania jednostek, którym przypisywane będą tagi. Strategia segmentacji w przypadku tagsetów KIPI i NKJP jest identyczna (Przepiórkowski i Szałkiewicz, 2012). Nadrzędnym założeniem jest reguła, że żaden segment nie może zawierać w sobie znaków białych. Powoduje to, że wielowyrazowe nazwy własne (np. Lądek Zdrój) rozbijane są na ciągi segmentów. Podobnie, czasowniki łączące się z zaimkiem się stanowią też ciągi segmentów (np. wydawać się). Bardziej kontrowersyjną decyzją jest podział niektórych form uznawanych tradycyjnie za pojedyncze wyrazy; dzieje się tak w kilku przypadkach, m.in (na podstawie Przepiórkowski, 2004 oraz Przepiórkowski i Szałkiewicz, 2012):

1. tzw. *formy aglutynacyjne* czasownika być traktowane są jako osobne segmenty: zrobił|eś, długo|śmy;
2. jako osobne segmenty traktowane są też partykuły by, -ż(e) i -li, np. przyszedł|by, poszli|by|śmy, potrzebował|że|by|ś;
3. poprzyimkowa nieakcentowana forma zaimka -ń, np. na|ń.

Powyższe decyzje mają dobrą motywację lingwistyczną: wspomniane formy aglutynacyjne i partykuły mogą się przemieszczać bez istotnych zmian w znaczeniu zdania (np. poszedłbyś a byś poszedł). Co więcej, trudno jest przypisać wyrażeniom typu długośmy czy nań sensowną klasę gramatyczną. Warto jednak pamiętać, że taka strategia segmentacji jest mocno nietradycyjna i powoduje czasem praktyczne problemy. W szczególności, pojawia się czasem problem *niejednoznaczności segmentacji*, polegający na tym, że tej samej formie można przypisać kilka alternatywnych ciągów segmentów. Sytuacja taka występuje w praktyce: aby zapewnić pełną obsługę takich niejednoznaczności, analizator morfosyntaktyczny Morfeusz (Woliński, 2006) zwraca nie ciągi segmentów, lecz grafy składające się ze ścieżek reprezentujących alternatywne sposoby podziału wyrazu na segmenty. Przykład grafu zawierającego niejednoznaczność segmentacji przedstawiono na rysunku 2.2. Problem w tym, że praktycznie wszystkie znane z literatury algorytmy ujednoznaczniania morfosyntaktycznego zakładają, że na wejściu jest ciąg segmentów, a nie struktura grafowa. Stosowanym praktycznie rozwiązaniem



Rys. 2.1. Graf zwrócony przez analizator Morfeusz dla formy *miałem*: górna ścieżka odpowiada interpretacji czasownikowej, dolna — rzeczownikowej (*miał*). W nawiasie podano lematy, tagi skrócono do klas gramatycznych dla czytelności.

jest użycie jakiejś heurystyki wyboru ścieżki w grafie[4]. Przykładowo, tager PANTERA pozwala na zdefiniowanie, w przypadku których form wyrazowych należy preferować krótszą ścieżkę, a w przypadku których — dłuższą (Acedański, 2010). Istnieją też inne możliwości rozwiązania tego problemu, np. konwersja tekstu do tagsetu pośredniego, gdzie problem niejednoznaczności segmentacji nie występuje, albo występuje w mniejszym stopniu (Radziszewski i Śniatowski, 2011a).

Jak wspomnieliśmy wyżej, podkorpus milionowy NKJP zawiera wzorcowe oznakowanie morfosyntaktyczne w tagsecie NKJP. Proces ręcznego znakowania wspomagany był przez nową wersję analizatora Morfeusz (Woliński, 2006): znakowanie większości form sprowadzało się do wyboru jednej z interpretacji zwróconych przez analizator jako prawidłowej (Przepiórkowski i Szałkiewicz, 2012). Wyniki działania analizatora również zachowano w podkorpusie milionowym NKJP: każdemu segmentowi przypisano zbiór interpretacji, spośród których dokładnie jedna oznakowana jest jako prawidłowa w tym kontekście. Podczas znakowania korpusu zdarzały się formy, których Morfeusz nie rozpoznał. Zdarzały się również formy, które zostały rozpoznane, lecz analizator nie podał prawidłowej w tym kontekście interpretacji (Przepiórkowski i Szałkiewicz, 2012). W takiej sytuacji zadaniem lingwisty było dodanie brakującej interpretacji ręcznie. Formy nierozpoznane przez analizator zostały opisane w podkorpusie milionowym NKJP w specyficzny sposób; przypisany im zbiór interpretacji składa się z dwóch elementów: prawidłowej interpretacji dodanej ręcznie przez lingwistę oraz sztucznej interpretacji zwróconej przez Morfeusza, składającej się z tagu „słowo nieznane" (klasa `ign`, bez atrybutów) i sztucznego lematu będącego napisem `None`. Zapis w korpusie powzwala więc na łatwą identyfikację form, których używana podczas znakowania wersja analizatora nie rozpoznała.

Oprócz wspomnianych tagsetów KIPI i NKJP istnieje też kilka innych — choć tagsety te są dla nas mniej interesujące, gdyż nie użyto ich do znakowania dostępnych publicznie korpusów wzorcowych. Przykładowo, słownik morfologiczny *Polex/PMDBF* (Obrębski i Stolarski, 2006) rozprowadzany wraz z poznańskim pakietem *UAM Text Tools* definiuje własny tagset. Tagset charakteryzuje się bardzo tradycyjnym podziałem na klasy gramatyczne: odpowiadają one dość dokładnie tradycyjnym częściom mowy.

[4] Zdecydowana większość grafów zwróconych przez Morfeusza wskutek analizy pojedynczych wyrazów i tak składa się z jednej ścieżki.

Podobny charakter ma polski tagset zgodny z wytycznymi projektu *MULTEXT-East* (Kotsyba i inni, 2009) — nadrzędną ideą przedsięwzięcia było bowiem opracowanie możliwie podobnych do siebie tagsetów dla języków wschodnioeuropejskich, które z kolei zbliżone będą do pozostałych tagsetów zdefiniowanych w ramach rodziny MULTEXT.

2.3. Czynniki wpływające na trudność znakowania

Znakowanie języków słowiańskich jest często opisywane jako zadanie trudne, znacznie trudniejsze niż znakowanie języka angielskiego. Dzieje się tak, gdyż języki słowiańskie charakteryzują się z jednej strony swobodnym szykiem wyrazów w zdaniu, z drugiej zaś — mnogością form spowodowaną bogatą odmianą (Hajič i inni, 2001; Przepiórkowski, 2007). Z tego względu proste metody statystyczne, oparte na częstościach konkretnych ciągów wyrazowych, skazane są na niepowodzenie (Sharoff, 2004)[5].

Istotny wpływ na trudność znakowania ma rozmiar i charakter tagsetu. Dla języka angielskiego tagsety zawierają ok. 40–200 różnych tagów (Krenn i Samuelsson, 1997), tagset języka polskiego korpusu IPI PAN dopuszcza zaś 4179 możliwych tagów (Przepiórkowski, 2005). Zazwyczaj bardziej szczegółowy tagset wiąże się z większą niejednoznacznością (tj. większą średnią liczbą tagów przypadającą na segment), co zwiększa trudność ujednoznaczniania (Manning i Schütze, 1999, s. 372). Jest to jeden z częściej przytaczanych powodów wysokiej trudności znakowania morfosyntaktycznego języków fleksyjnych (Vidová-Hladká, 2000; Hajič i inni, 2001; Dębowski, 2001; Piasecki i Godlewski, 2006a).

W przypadku tagerów uczonych na korpusie znakowanym ręcznie, istotne znaczenie ma rozmiar dostępnego korpusu oraz jego pochodzenie. Zazwyczaj publikowane wyniki oceny tagerów dotyczą sytuacji, gdy zarówno materiał uczący, jak i materiał testowy są próbkami tego samego korpusu. Jeśli tager stosowany jest do znakowania tekstu pochodzącego z innego źródła, rzeczywista jakość znakowania może być gorsza (Manning i Schütze, 1999, s. 372).

Odsetek błędów tagera wiąże się także z częstością występowania w tekście analizowanym form nierozpoznanych — **słów nieznanych**[6]. Problem ten pojawia się, gdy analizowany segment nie występuje w słowniku (w przypadku tagera zakładającego wykonanie analizy morfologicznej) lub gdy forma nie pojawiła się w korpusie uczącym (jeśli użyty algorytm znakowania zakłada pozyskiwanie modelu leksykalnego z korpusu uczącego) (Manning i Schütze, 1999; van Halteren, 1999). Niezależnie od rozmiaru dostępnego słownika i korpusu uczącego, należy się liczyć z obecnością słów nieznanych.

[5] To mocne stwierdzenie warto przytoczyć w oryginale:

> Given that the word order in Russian (and other Slavonic languages) is relatively free and a typical word (i.e. lemma) has many forms (typically from 9 for nouns to 50 for verbs), the sequences of exact N-grams are much less frequent than in English, thus rendering purely statistical approaches useless.

[6] Wydaje się, że nazwa *słowa nieznane* utarła się w literaturze, dlatego tutaj też będziemy jej używać. Słowa nieznane są w rzeczywistości segmentami i nie zawsze muszą być tradycyjnie rozumianymi słowami. Poprzez analogię pozostałe segmenty, w tym znaki interpunkcyjne, nazywać będziemy *słowami znanymi*.

Wynika to z charakterystyki języków naturalnych, a w szczególności z tzw. *prawa Zipfa*, które podaje przybliżony rozkład prawdopodobieństwa form wyrazowych w korpusie. Zgodnie z prawem Zipfa, częstotliwość występowania danej formy w korpusie jest odwrotnie proporcjonalna do jej pozycji w rankingu, tj. liście frekwencyjnej (Manning i Schütze, 1999). Z tego rozkładu wynika, że choćby wziąć bardzo duży korpus, znaczny procent segmentów stanowić będą słowa rzadkie. Manning i Schütze (1999) podają następujący przykład: w książce Marka Twaina, zawierającej 71 370 segmentów, 49,8% segmentów występuje dokładnie raz. Konsekwencją takiego rozkładu jest nieunikniony problem znakowania form, które nie zostały zaobserwowane podczas uczenia modelu.

Warto tutaj podkreślić, że nie można bezpośrednio porównać osiągów tagerów dla dwóch języków, nawet podobnych do siebie, gdyż stopień trudności znakowania zależy też od charakteru przyjętego tagsetu. Tagsety zaś różnią się między sobą nie tylko ze względu na charakter opisywanych języków, ale też z powodu różnych tradycji opisu gramatycznego i przewidywanych w momencie projektowania tagsetu zastosowań. Co więcej, porównanie wyników eksperymentalnych jest na ogół niemiarodajne ze względu na istotne różnice w przyjętych metodykach oceny, co wykażemy w rozdziale 2.5.

2.4. Przegląd metod

Badania nad znakowaniem morfosyntaktycznym dla języka angielskiego prowadzone były już w latach sześćdziesiątych XX wieku. Od tego czasu zaproponowano i przebadano wiele różnych metod. Jedynie niewielka część spośród nich została przetestowana pod kątem znakowania języków słowiańskich. Jak wspomniano wyżej, wiele spośród tych metod nie sprawdza się dla języków słowiańskich ze względu na charakter tych języków. W niniejszym opracowaniu skupiamy się przede wszystkim na metodach, które były już testowane dla któregoś z języków słowiańskich. Oprócz tego rozważamy też kilka metod, które ze względu na swoje założenia wydają się stosowne dla języka polskiego.

2.4.1. Reguły pisane ręcznie

Pierwsze tagery powstały na przełomie lat pięćdziesiątych i sześćdziesiątych XX wieku. Były to systemy oparte na niewielkich słownikach pełniących funkcję analizatorów morfosyntaktycznych, heurystykach odgadywania interpretacji słów nieobecnych w słowniku oraz ręcznie pisanych regułach ujednoznaczniania (van Halteren, 1999, s. 10).

Najpopularniejszym formalizmem pozwalającym na zapis reguł ujednoznaczniania jest prawdopodobnie *Constraint Grammar* (dosłownie: *gramatyka ograniczeń*; nazwa często skracana jest do akronimu CG) (Karlsson, 1990; van Halteren, 1999). Formalizm działa na tekście poddanym analizie morfosyntaktycznej. Reguły mają postać „wykonaj operację A w miejscu B w kontekście C". Reguły wykonywane są sekwencyjnie na kontekście każdego segmentu w zdaniu; czasem potrzebne jest wielokrotne przejście przez jedno zdanie, by wszystkie możliwe do wykonania operacje zostały uruchomione (dzieje się tak, gdyż warunki kontekstowe potrzebne do wykonania niektórych operacji mogą zaistnieć dopiero po uruchomieniu innej reguły). Są dwie główne operacje: usunięcie interpretacji spełniających podane warunki (`REMOVE`) oraz działanie odwrotne,

tj. pozostawienie takich interpretacji (`SELECT`). Kontekst C określa warunki konieczne, by uruchomić regułę. Warunki takie mogą odwoływać się do części tagu bądź też do konkretnych lematów. Przykładowo, poniższa reguła (przykład 2.5, za van Halteren, 1999) usuwa interpretację „reczownik w mianowniku liczby pojedynczej" (`N NOM SG`), jeśli istnieje też interpretacja jako czasownik w bezokoliczniku (`INF`) oraz poprzedzający segment ma przypisaną jedynie (`C`) interpretację czasownika modalnego (`AUXMOD`):

(2.5)
```
REMOVE (N NOM SG)
       (-1C (AUXMOD))
       (0 (INF)) ;
```

CG był inspiracją dla formalizmu *JOSKIPI* (*Język Opisu Stanu Korpusu IPI PAN*; Piasecki, 2006) oraz jego następcy zwanego WCCL (*Wrocław Corpus Constraint Language*, Radziszewski i inni, 2011c); oba formalizmy opracowano na Politechnice Wrocławskiej. Formalizm WCCL pozwala na zapis wyrażeń funkcyjnych opisujących cechy dla maszynowego uczenia (to zastosowanie omówimy w punkcie 2.6); formalizm pozwala również na zapis reguł ujednoznaczniania języka polskiego. Każda reguła ujednoznaczniania, podobnie jak w wypadku CG, opisuje operację do wykonania na obecnie przetwarzanym segmencie (stosowana jest identyczna strategia sekwencyjnego uruchamiania reguł oraz wielokrotnego przechodzenia zdania segment po segmencie). Ogólną postać reguły ujednoznaczniania przedstawiono poniżej.

(2.6)
```
<reguła> ::= "rule" "(" <nazwa> "," [<warunek> ","] <akcje>
<warunek> ::= <predykat-WCCL>
<akcje> ::= <akcja> | <akcja> "," <akcje>
<akcja> ::= <akcja-delete> | <akcja-select>
<akcja-delete> ::= "delete" "(" <predykat-WCCL> ")"
<akcja-select> ::= "select" "(" <predykat-WCCL> ")"
```

W formalizmie WCCL dostępne są obie wspomniane operacje: `delete` oraz `select`. Warunki natomiast pozwalają na odwołanie się do bardziej skomplikowanych cech morfosyntaktycznych, takich jak uzgodnienie gramatyczne. Warunki te opisywane są za pomocą *predykatów WCCL*. Predykaty te pozwalają na formalny zapis testów na własności morfosyntaktyczne dotyczące segmentu aktualnie ujednoznacznianego oraz pozostałych segmentów należących do tego samego zdania (predykaty oraz inne wyrażenia funkcyjne omawiamy w punkcie 2.6).

Poniższa reguła pozostawia te interpretacje, które nie mają określonej wartości przypadka (`equal(cas[0], {})`) oraz te, których przypadek zawiera się w zbiorze możliwych przypadków przypisanych poprzedniemu segmentowi (`in(cas[0],cas[-1])`). Reguła określa trzy warunki konieczne: poprzedni segment musi być przyimkiem (`prep`), bieżący segment musi być zaimkiem trzecioosobowym (`ppron3`), natomiast żaden z trzech segmentów następujących po segmencie bieżącym nie ma interpretacji przymiotnikowej, rzeczownikowej ani odsłownikowej (`only(1,3,$V, not(...))`).

(2.7)
```
rule("r1",
       and(
          equal(class[-1],{prep}),
          equal(class[0],{ppron3}),
          only(1,3,$V, not(inter(class[$V],{adj,ger,subst})))
```

```
        )
    ),
    select(or(
          equal(cas[0],{}),
          in(cas[0],cas[-1])
          ))
)
```

Takie reguły znalazły zastosowanie w tagerze języka polskiego TaKIPI (Piasecki i Godlewski, 2006b). Mechanizm jego działania zakłada, że pierwszym etapem jest uruchomienie napisanych ręcznie reguł wykreślających niektóre interpretacje, natomiast pozostałe niejednoznaczności rozwiązywane są za pomocą reguł pozyskanych automatycznie za pomocą techniki indukcji drzew decyzyjnych. Najnowsza wersja tagera zawiera 35 reguł, spośród których znaczna część odwołuje się do konkretnych form wyrazowych (np. istnieje reguła, która narzuca interpretację partykuło-przysłówka formie z w wyrażeniach typu z dala, z bliska, z godzinę).

Podobne metody stosowane były do ujednoznaczniania innych języków, w tym — języka czeskiego (Hajič i inni, 2001), bułgarskiego (Tanev i Mitkov, 2002; Dojchinova i Mihov, 2004).

2.4.2. Metody statystyczne

Metody statystyczne zakładają użycie statystycznego modelu języka, określającego różnie definiowane częstości współwystępowania form wyrazowych i tagów; np. częstość występowania danej formy wyrazowej z danym tagiem, częstość występowania formy wyrazowej w_2 po formie wyrazowej w_1. Model języka budowany jest na podstawie korpusu uczącego. Podczas działania tagera model języka używany jest w celu przypisania ciągowi segmentów ciągu tagów o najwyższym prawdopodobieństwie (Màrquez, 1999, s. 30). Istnieje kilka modeli matematycznych, które znalazły zastosowanie w znakowaniu morfosyntaktycznym; są to m.in. ukryte modele Markowa (Brants, 2000; Dębowski, 2004), warunkowe pola losowe (Lafferty i inni, 2001), model maksymalizacji entropii (Ratnaparkhi, 1996; Mastalerz, 2011), modele Markowa o maksymalnej entropii (McCallum i inni, 2000), a także *tager wykładniczy* (Hajič i Vidová-Hladká, 1998b). W niniejszym rozdziale omówimy dwa pierwsze ze względu na dużą popularność i dobre osiągi w rozwiązywaniu różnych problemów znakowania ciągów.

Ukryty model Markowa n-tego rzędu[7] to proces stochastyczny zdefiniowany na zbiorze *ukrytych stanów* S i zbiorze *obserwacji* W. Proces jest opisany poprzez dwa zbiory prawodopodobieństw:

— *Prawdopodobieństwa przejść* między stanami $P(s_i|s_{i-1} \dots s_{i-n+1})$, zgodnie z założeniem, że jedynie n poprzednich stanów ma wpływ na stan bieżący.

— *Prawdopodobieństwa obserwacji* $P(w_i|s_i)$, zgodnie z założeniem, że jedynie bieżący stan s_i ma wpływ na prawdopodobieństwo pojawienia się obserwacji w_i w chwili i.

W przypadku znakowania morfosyntaktycznego, ukrytymi stanami są tagi, natomiast obserwacje są formami wyrazowymi reprezentowanymi przez segmenty w tekście

[7] Opis ukrytych modeli Markowa oraz ich zastosowania w znakowaniu morfosyntaktycznym opracowano na podstawie van Halteren (1999) oraz Feldman i Hana (2010).

(bądź też pewnymi klasami abstrakcji przypisanymi takim formom by zredukować problem form rzadkich).

Ukryty model Markowa jest modelem generatywnym, tj. modeluje prawdopodobieństwo losowo wygenerowanych ciągów obserwacji (form wyrazowych) na podstawie łącznego rozkładu prawdopodobieństwa dwóch zmiennych: ciągu obserwacji i ciągu stanów. Nazywany jest *ukrytym*, gdyż rozdziela widoczne obserwacje (obecne są bezpośrednio w danych) od stanów będących częścią abstrakcyjnego modelu (dlatego „ukrytych"). W znakowaniu morfosyntaktycznym najczęściej stosuje się modele drugiego rzędu.

Scieżką w modelu Markowa nazywamy ciąg takich przejść między stanami, że stan wyjściowy danego przejścia jest stanem wejściowym przejścia następującego po nim. Przejście przez ścieżkę modelu powoduje wygenerowanie ciągu obserwacji. Jednym z zastosowań modeli Markowa jest ustalenie ciągu stanów, który spowodował wygenerowanie danego ciągu obserwacji. W terminologii znakowania morfosyntaktycznego oznacza to ustalenie ciągu tagów przypisanego do ciągu obserwowanych form wyrazowych. Jako że może istnieć wiele ciągów tagów, które odpowiadają temu samemu ciągowi obserwacji, zadanie polega na ustaleniu ciągu tagów o najwyższym prawdopodobieństwie.

Wspomniane dwa założenia upraszczające sprawiają, że ukryty model Markowa n-tego rzędu można przedstawić za pomocą wzoru (2.8). Należy podkreślić, że założenie o zależności obserwacji jedynie od bieżącego stanu jest dość mocnym uproszczeniem: jedynie dany tag (stan modelu) decyduje o rozkładzie prawdopodobieństwa form wyrazowych, którym tag ten przypisano.

$$P(s_1 \dots s_K, w_1 \dots w_K) = \prod_{i=1}^{K} P(w_i|s_i)P(s_{i+1}|s_i \dots s_{i-n+1}) \tag{2.8}$$

Uczenie tagera opartego na modelu Markowa polega na estymacji prawdopodobieństw obsweracji oraz przejść na podstawie częstości współwystępowania form wyrazowych i tagów w korpusie uczącym.

Zadanie znakowania za pomocą wyuczonego modelu polega na znalezieniu ścieżki o najwyższym prawdopodobieństwie poprzez stany modelu Markowa. Ustalenie takiej ścieżki wymaga analizy prawdopodobieństw przejść i obserwacji. Rozwiązanie dokładne jest niezwykle czasochłonne i prowadzi do eksplozji kombinatorycznej; w praktyce stosuje się rozwiązanie uproszczone, a mianowicie algorytm Viterbiego (Viterbi, 1967). Algorytm opiera się na założeniu, że do każdego stanu prowadzi dokładnie jedna ścieżka o najwyższym prawdopodobieństwie; gdy wiele ścieżek zbiega się w jednym stanie, ścieżki o niskim prawdopodobieństwie można w praktyce pominąć, rozważając w obliczeniach jedynie pozostałe. Pozwala to uniknąć eksplozji kombinatorycznej: w każdym stanie rozważamy jedynie kilka najbardziej prawdopodobnych ścieżek.

Model Markowa był stosowany do znakowania języka czeskiego (Hajič i Vidová-Hladká, 1998a; Vidová-Hladká, 2000; Feldman i Hana, 2010). Przeprowadzono serię eksperymentów, których celem było m.in. znalezienie optymalnego modelu (testowany był model Markowa pierwszego i drugiego rzędu), zbadanie wpływu rozmiaru tagsetu i rozmiaru danych uczących na trafność znakowania. Wnioski z eksperymentów można podsumować w następujący sposób: rozmiar danych uczących ma znaczenie (im więcej danych, tym lepiej), redukcja tagsetu przyniosła prawie dwukrotny spadek błędów tagera[8], natomiast wyniki osiągnięte przez model pierwszego i drugiego rzędu są

[8] Tagset został drastycznie zredukowany: tagset pełny zawiera 1171 różnych tagów, zmniejszony

zbliżone. Kolejnym ciekawym eksperymentem było wprowadzenie przetwarzania dwuetapowego: najpierw stosowana była analiza morfosyntaktyczna, a stosowany po niej model Markowa wybierał jedynie spośród przypisanych tagów. Podejście takie spowodowało spadek błędów o 14%, choć nie jest do końca jasne, czy ocena uwzględniała także błędy analizatora (por. punkt 2.5).

Eksperymenty z modelami Markowa przeprowadzono również dla języka polskiego. Tager Dębowskiego (2004) jest klasycznym tagerem opartym o model Markowa drugiego rzędu. Model Markowa stosowany jest po analizie morfosyntaktycznej, tj. nie ma możliwości wyboru tagu, którego nie zwrócił analizator. Dotychczasowe testy wykazały trafność ok. 90,6% (Dębowski, 2004; Karwańska i Przepiórkowski, 2010). Kuta (2010) przetestował na korpusie FREK tager TnT oparty na modelu Markowa drugiego rzędu (Brants, 2000), opracowany pierwotnie dla języka angielskiego, osiągając trafność ok. 84,3%.

Istnieją też podejścia oparte na ukrytych modelach Markowa, lecz zawierające również nowe elementy. Przykładowo, Schmid (1995) stosuje drzewa decyzyjne jako narzędzie wspomagające estymację parametrów modelu.

Warunkowe pola losowe (ang. *Conditional Random Fields*, CRF)[9] są modelem statystycznym zbliżonym do ukrytych modeli Markowa (Lafferty i inni, 2001). Zasadnicza różnica jest następująca: ukryty model Markowa jest modelem generatywnym, tj. modeluje łączny rozkład prawdopodobieństwa ciągu obserwacji i ciągu stanów (tagów), podczas gdy warunkowe pola losowe są modelem *warunkowym*, gdyż modelują warunkowy rozkład prawdopodobieństwa ciągu tagów zakładając dany ciąg obserwacji, tj. $P(s_1 \dots s_K | w_1 \dots w_K)$. Pozwala to uniknąć niepożądanych założeń, w szczególności zaś prawdopodobieństwo przejścia między tagami nie musi zależeć jedynie od bieżącej obserwacji — możliwe jest uwzględnienie bardzo szerokiego kontekstu, co ma duże znaczenie w językach naturalnych (pomiędzy segmentami powiązanymi składniowo może wystąpić ciąg innych segmentów o nieograniczonej z góry długości). Co więcej, warunkowe pola losowe pozwalają na wprowadzenie do modelu wielu dodatkowych *cech*, tj. funkcji przekształcających dany segment lub jego otoczenie w wartości symboliczne, np. sprawdzenie, czy forma rozpoczyna się wielką literą, pobranie z formy wyrazowej końcówki o ustalonej długości.

Liniowe[10] warunkowe pole losowe można opisać za pomocą wzoru (2.9). Pierwsza suma we wzorze odpowiada za kolejne segmenty w tekście (obserwacje i ich tagi), druga związana jest z cechami f_j i ich wagami λ_j. W modelu przyjmuje się, że cechy są funkcjami charakterystycznymi, tj. przyjmującymi wartość 0 lub 1. Wartość 1 oznacza, że dana cecha jest spełniona na i-tej pozycji oznakowanego ciągu segmentów. Cechy mają postać $\mathbb{N} \times S \times S \times W^* \to \{0, 1\}$ (liczba naturalna wskazuje numer segmentu w wektorze słów o typie W^*; zbiór S oznacza zbiór stanów, czyli tagów; funkcja przyjmuje dwa tagi, gdyż w ogólności rozpatrywane są tagi przypisane do bieżącej pozycji oraz pozycji poprzedniej).

— jedynie 206. Redukcja ta odbyła się kosztem istotnych gramatycznie kategorii, m.in. przypadka i rodzaju (Vidová-Hladká, 2000, s. 27).

[9] Opis warunkowych pól losowych opracowano na podstawie Lafferty i inni (2001); Wallach (2004); Sutton i McCallum (2011).

[10] Oprócz liniowych warunkowych pól losowych (ang. *linear-chain Conditional Random Fields*) istnieją również *ogólne* warunkowe pola losowe (*general CRF*, Sutton i McCallum, 2011). W tej rozprawie skupiamy się na tych pierwszych.

Przykładowo, w ten sposób rozumianą cechą może być funkcja sprawdzająca, czy segment na i-tej pozycji oznakowany jest jako rzeczownik, a poprzedzający segment to forma wyrazowa jest. Uczenie modelu prowadzi do ustalenia wartości wag przypisanych cechom. Wysoka wartość wagi oznacza, że zależności między obserwacjami i tagami opisywane przez daną cechę rzeczywiście zachodzą w danych uczących. W przypadku powyższego przykładu oznaczałoby to, że po większości segmentów o formie wyrazowej jest rzeczywiście wystąpiły rzeczowniki. Z to funkcja normalizująca opisana wzorem (2.10); zapis T^K oznacza zbiór wszystkich możliwych ciągów tagów o długości K przyjmując tagset T.

$$P(s_1 \dots s_K | w_1 \dots w_K) = \frac{1}{Z(w_1 \dots w_K)} \exp \sum_{i=1}^{K} \sum_{j=1}^{J} \lambda_j f_j(i, s_i, s_{i-1}, (w_1 \dots w_K)) \tag{2.9}$$

$$Z(w_1 \dots w_K) = \sum_{(s'_1 \dots s'_K) \in T^K} \exp \sum_{i=1}^{K} \sum_{j=1}^{J} \lambda_j f_j(i, s'_i, s'_{i-1}, (w_1 \dots w_K)) \tag{2.10}$$

Znakowanie za pomocą wyuczonego warunkowego pola losowego polega na znalezieniu ciągu tagów o najwyższym prawdopodobieństwie. Funkcja Z jest niezależna od poszukiwanego ciągu tagów, a zatem problem sprowadza się do maksymalizacji sumy składników $\lambda_j f_j(i, s_i, s_{i-1}, (w_1 \dots w_K))$. Problem ten można rozwiązać za pomocą wspomnianego algorytmu Viterbiego (Viterbi, 1967).

Uczenie tagera opartego na warunkowych polach losowych sprowadza się do znalezienia wartości wag dla których prawdopodobieństwo warunkowe ze wzoru (2.9) jest największe. W tym celu stosuje się metody estymacji parametrycznej, m.in. metodę największej wiarygodności.

Założenie modelu, według którego cechy są funkcjami charakterystycznymi, powoduje pewne utrudnienie. Jeśli cechy mają porównywać tagi oraz formy wyrazowe z pewnymi stałymi, konieczne jest ustalenie z góry zbioru takich stałych. Przykładowo, jeśli cecha ma sprawdzać, czy forma wyrazowa segmentu równa jest napisowi nie, należałoby taką cechę wprost sformułować. W praktyce stosuje się następujące rozwiązanie: tworząc model operuje się na *szablonach cech*, tj. funkcjach parametryzowanych wartościami, które przyrównywane są do tagu i/lub formy wyrazowej. Powszechnie stosuje się dwa typy funkcji: funkcje *unigramowe*, zależne od tagu na bieżącej pozycji i formy wyrazowej znajdującej się na tej samej pozycji — opisane wzorem (2.11a) — oraz funkcje *bigramowe*, gdzie wprowadza się dodatkowo zależność od tagu poprzedzającego bieżącą pozycję — opisane wzorem (2.11b). Podczas uczenia modelu zbierane są wszystkie występujące tagi oraz formy wyrazowe, po czym szablony cech rozwijane są do wszystkich możliwych instancji. W przypadku cech unigramowych instancji tych jest $T_d \cdot W_d$, gdzie T_d to liczba różnych tagów, natomiast W_d to liczba różnych form wyrazowych, które zostały napotkane w danych uczących d. W przypadku cech bigramowych instancji tych jest więcej, a mianowicie $T_d \cdot T_d \cdot W_d$ (Kudo, 2005).

$$f_{t_A, v_A}(i, s_i, s_{i-1}, (w_1 \dots w_K)) = \begin{cases} 1, & s_i = t_A \wedge w_i = v_A \\ 0, & \text{w pp.} \end{cases} \tag{2.11a}$$

$$f_{t_A,t_B,v_A}(i, s_i, s_{i-1}, (w_1 \dots w_K)) = \begin{cases} 1, & s_i = t_A \wedge s_{i-1} = t_B \wedge w_i = v_A \\ 0, & \text{w pp.} \end{cases} \tag{2.11b}$$

Powyższe sformułowanie szablonów cech zakłada, że cechy sprawdzają jedynie formę wyrazową w postaci niezmienionej. W analogiczny sposób można zdefiniować szablony cechy odwołującej się do dowolnego przekształcenia T formy wyrazowej w wartość symboliczną, np. przekształcenia formy wyrazowej w jej końcówkę o ustalonej długości, bądź też sprawdzenia, czy forma wyrazowa zaczyna się wielką literą, czy zawiera cyfry itp. Jeśli przekształcenie T odwzorowuje zbiór form wyrazowych w zbiór mniejszy (np. *prawda* lub *fałsz* w przypadku predykatu sprawdzającego wielkość pierwszej litery), liczba instancji szablonu będzie mniejsza.

Lafferty i inni (2001) stosują warunkowe pola losowe do znakowania morfosyntaktycznego języka angielskiego, osiągając trafność ok. 94,5% dla cech sprawdzających jedynie niezmienione formy wyrazowe, oraz 95,7% dla zestawu cech zawierającego również proste testy na postać graficzną formy wyrazowej. Wyniki te są wyraźnie lepsze od osiągów ukrytych modeli Markowa: ok. 94,3% trafności.

Niestety, użycie warunkowych pól losowych w sposób analogiczny dla języka polskiego jest zadaniem trudnym z technicznego punktu widzenia: tagsety języka polskiego zawierają ponad 1000 możliwych tagów, podczas gdy złożoność czasowa uczenia liniowego warunkowego pola losowego jest proporcjonalna do kwadratu liczby występujących wartości klasy decyzyjnej (tj. liczby różnych tagów pojawiających się w danych). Ściślej rzecz biorąc, złożoność obliczeniowa ma postać $O(T^2 \cdot J \cdot D^2)$, gdzie T to liczba różnych tagów, J to liczba użytych cech, a D to liczba przypadków uczących (Cohn, 2007). Obawy te potwierdzają wstępne eksperymenty przeprowadzone dla języka polskiego przez autora tej rozprawy — próby zastosowania tagera CRF++ (Kudo, 2005) musiały zostać szybko przerwane ze względu na brak pamięci na serwerze (24 GB RAM). Dodatkowe potwierdzenie można znaleźć nie wprost w pracy doktorskiej Kuty (2010, s. 54): przeprowadzono eksperyment ze znakowaniem morfosyntaktycznym przy uproszczeniu tagów do samej klasy gramatycznej (trafność 89,84% na korpusie FREK), natomiast w tabeli przedstawiającej wyniki tagerów na pełnym tagsecie wartość trafności dla tagera CRF++ pominięto (braku wyników w stosownej kolumnie jednak nie wyjaśniono).

2.4.3. Znakowanie poprzez klasyfikację kolejnych segmentów

Opisywane w poprzednim punkcie metody zakładały tworzenie modelu statystycznego opisującego ciągi segmentów i odpowiadające im ciągi tagów. Istnieje również grupa metod, gdzie problem znakowania sprowadza się do problemu klasyfikacji pod nadzorem, tj. użycia klasyfikatorów uczonych na próbie uczącej (Koronacki i Ćwik, 2005, s. 16). Metody takie można scharakteryzować ze względu na dwa czynniki: umiejscowienie klasyfikatora w schemacie działania tagera oraz użyty klasyfikator i algorytm jego uczenia. Zagadnienia te są w dużej mierze niezależne od siebie. Daje to duże możliwości projektowania nowych algorytmów: mamy możliwość wyboru spośród znanych z literatury klasyfikatorów, wyboru strategii podziału segmentów na klasy decyzyjne, a także wprowadzenie dodatkowych źródeł informacji do modelu, np. analizy morfosyntaktycznej. Mimo tego ogromu możliwości, prace zakładające takie podejście do znakowania języków słowiańskich są stosunkowo nieliczne. Taki stan rzeczy był mo-

tywacją do zaproponowania nowej metody znakowania morfosyntaktycznego języków słowiańskich w oparciu o klasyfikację (rozdział 2.7).

W pierwszej kolejności umówimy najprostszy schemat budowy tagera opartego na klasyfikatorze, po czym podamy kilka przykładów jego realizacji znanych z literatury. W dalszej kolejności omówimy modyfikacje tego modelu stosowane do znakowania języków słowiańskich: użycie analizatora morfosyntaktycznego, znakowanie warstwowe oraz klasy niejednoznaczności. Na końcu tego punktu omówimy kilka popularnych klasyfikatorów i algorytmów ich uczenia.

Model podstawowy

Chociaż wejściowy ciąg segmentów może mieć dowolną długość (to samo można też powiedzieć o pojedynczych zdaniach), w praktyce stosuje się model uproszczony: przyjmuje się, że tag T_i przypisany segmentowi W_i zależy jedynie od najbliższego otoczenia tego segmentu, najczęściej rzędu 2–3 segmentów w lewo i w prawo. Pozwala to na sformułowanie zadania znakowania jako problemu klasyfikacji takich otoczeń (van Halteren, 1999, s. 286). Takie ujęcie przedstawiono na rysunku 2.2. Problem znakowania morfosyntaktycznego został sprowadzony do klasyfikacji pięcioelementowych ciągów. Każdy taki ciąg składa się z lewego kontekstu (pozycje „−2” i „−1”), segmentu centralnego podlegającego znakowaniu (pozycja „0”) oraz kontekstu prawego (pozycje „+1” i „+2”). Typowo konteksty przycinane są do granic zdania, co pokazano na rysunku. Wynikiem klasyfikacji jest przypisanie tagu segmentowi położonemu na pozycji centralnej, oznaczonej jako pozycja „0”. Znakowanie odbywa się od lewej do prawej, co powoduje, że kontekst lewy ma przypisane tagi w wyniku poprzednich decyzji klasyfikatora. Tagi te mogą zostać wykorzystane jako dodatkowa przesłanka w procesie decyzyjnym.

−2	−1	**0**	+1	+2
=	=	Kazał	kurze	ścierać
=	=	praet:sg:m1:perf	?	?
=	Kazał	kurze	ścierać	kurze
=	praet:sg:m1:perf	subst:sg:dat:f	?	?
Kazał	kurze	ścierać	kurze	.
praet:sg:m1:perf	subst:sg:dat:f	inf:imperf	?	?
kurze	ścierać	kurze	.	=
subst:sg:dat:f	inf:imperf	subst:pl:acc:m3	?	=
ścierać	kurze	.	=	=
inf:imperf	subst:pl:acc:m3	interp	=	=

Rys. 2.2. Znakowanie morfosyntaktyczne jako problem klasyfikacji.

W powyższym przykładzie kontekst został opisany w najprostszy możliwy sposób: poprzez podanie form wyrazowych należących do otoczenia $(-2, \ldots, +2)$. W terminologii klasyfikacji jest to wektor pięciu cech. Jeśli uwzględnimy również poprzednie decyzje klasyfikatora, to otrzymamy siedmioelementowy wektor: nowymi cechami będą tag przypisany pozycji −2 oraz tag przypisany pozycji −1. W praktyce stosuje się często bardziej zaawansowane cechy, m.in. wstępną klasyfikację „kształtu” formy wyrazowej

(np. czy zawiera tylko małe litery, czy też wielkie litery, znaki interpunkcyjne), przynależność do listy form częstych.

Najprostszy model zakłada użycie jednego klasyfikatora. W fazie uczenia segmenty reprezentowane są jako wektory cech i wraz z prawidłowym tagiem stanowią przypadki uczące dla klasyfikatora. Faza działania tagera (faza znakowania) polega na użyciu wyuczonego modelu do oznakowania kolejnych segmentów, również przedstawionych jako wektory cech. Bird i inni (2009, rozdz. 6) pokazują użycie tego modelu do znakowania języka angielskiego. Stosowany jest klasyfikator bayesowski i kilka prostych zestawów cech.

Często wprowadza się podział segmentów na klasy decyzyjne. Dla każdej z klas uczony jest osobny klasyfikator. Istnieją dwa popularne kryteria podziału: rozróżnienie między słowami znanymi i nieznanymi oraz podział na klasy niejednoznaczności.

Sens podziału na słowa znane i nieznane wynika z obserwacji rozkładów prawdopodobieństwa tagów przypisanych formom częstym i formom rzadkim. Rozkłady te różnią się w istotny sposób: formy rzadkie w praktyce nie należą do zamkniętych klas gramatycznych (np. przyimków, spójników) i najczęściej są nimi rzeczowniki, w szczególności nazwy własne. Praktyka stosowana w budowie tagerów jest następująca: z danych uczących zbierana jest lista frekwencyjna form, a formy o częstości poniżej pewnego progu stosowane są jako przybliżenie *słów nieznanych* (van Halteren, 1999).

Innym kryterium podziału na klasy decyzyjne są tzw. **klasy niejednoznaczności**, tj. zbiory możliwych tagów przypisanych danej formie wyrazowej. Najprostszym sposobem pozyskania takich zbiorów jest analiza danych uczących (Cutting i inni, 1992). Klasy niejednoznaczności są przydatne nie tylko jako klasa decyzyjna *sensu stricto*; klasa niejednoznaczności segmentu może też być użyta jako jedna z cech dla klasyfikatora.

Taki rozszerzony model został zastosowany w tagerze *MBT* (ang. *Memory-Based Tagger*) opartym na uczeniu pamięciowym (Daelemans i inni, 2010b). Podczas uczenia tager ten dzieli segmenty wejściowe na dwie klasy decyzyjne: *słowa znane* i *słowa nieznane*. Dla obu tych klas tworzone są osobne klasyfikatory, co w przypadku uczenia na pamięć sprowadza się do zgromadzenia list przypadków uczących (działanie klasyfikatora pamięciowego omówimy dokładniej na s. 29). MBT pozwala zdefiniować osobne zestawy cech dla słów znanych i nieznanych; standardowo, zbiór cech dla słów znanych uwzględnia m.in. klasę niejednoznaczności uzyskaną na podstawie danych uczących, a dla słów nieznanych stosowane są informacje o postaci graficznej formy (np. czy pisana jest z wielkiej litery, czy zawiera łącznik). Tager MBT testowany był pierwotnie dla języka angielskiego, co dało trafność 96,7% (Daelemans i inni, 2010b). Następnie przeprowadzono testy dla kilku języków europejskich (Zavrel i Daelemans, 1999), m.in. duńskiego (95,7%), hiszpańskiego (97,8%), a także czeskiego (93,6%). Trafność osiągnięta dla języka czeskiego może wydawać się zaskakująco wysoka; tłumaczy ją jednak użyty tagset, zawierający jedynie 42 możliwe tagi. Tager MBT został również przetestowany dla języka słoweńskiego, tym razem na pełnym tagsecie (Džeroski i inni, 1999); osiągnięta trafność jest znacznie niższa: 86,42%. Kuta (2010) przeprowadził również testy tagera MBT na korpusie języka polskiego FREK (por. s. 12) i osiągnął trafność 80,43%.

Znakowanie jako wykreślanie

Jak wspomniano w punkcie 1.2.2, znakowanie języków słowiańskich na ogół wykonuje się dwuetapowo. Istnieje możliwość wykorzystania takiego podejścia również w przypadku modelu opartego na klasyfikacji segmentów: na etapie klasyfikacji mamy dostęp nie tylko do form wyrazowych i przypisanych dotychczas tagów, lecz również do wyników analizy morfosyntaktycznej dla wszystkich segmentów. W szczególności, analiza ta dostępna jest dla formy, której klasyfikator ma przypisać tag. Jeśli uznamy analizę morfosyntaktyczną za wyrocznię, to zawęzimy liczbę sensownych tagów przypisanych segmentowi na pozycji „0" do tych, które zwrócił analizator. Tak więc problem znakowania zostaje tutaj sprowadzony do wykreślania niechcianych tagów spośród pozycji zwróconych przez analizator.

Co więcej, analizy morfosyntaktyczne otaczających segmentów pozwalają na sformułowanie ciekawych cech morfosyntaktycznych, np. możliwych wartości klasy gramatycznej dla danego segmentu, możliwych wartości przypadka itp. — dzięki pozycyjnemu charakterowi tagsetów typowych dla języków słowiańskich możliwe jest naturalne rozbicie tagów na wartości atrybutów. Podobne podejście zostało sformułowane dla języka francuskiego (Tzoukermann i inni, 1997), później — dla języka polskiego (Piasecki i Godlewski, 2006b). Takie ujęcie zilustrowano na rysunku 2.3. Zadaniem klasyfikatora jest przypisanie tagu formie kurze na podstawie cech opisujących możliwe formy wyrazowe, klasy gramatyczne i wartości przypadka otaczających segmentów. Ponieważ znakowanie odbywa się sekwencyjnie od lewej do prawej, segmenty należące do lewego kontekstu zostały już ujednoznacznione (w omawianym przykładzie jest to pojedynczy segment — Kazał). W górnej części rysunku przedstawiono również wyniki analizy morfosyntaktycznej, na podstawie których zostały wygenerowane wartości wspomnianych cech[11].

Pozycja	−1	**0**	+1	+2
Możliwe tagi	praet:sg:m1:perf	adj:sg:nom:n:pos adj:sg:acc:n:pos subst:sg:dat:f subst:sg:loc:f subst:pl:acc:m3 subst:pl:voc:m3	inf:imperf	adj:sg:nom:n:pos adj:sg:acc:n:pos subst:sg:dat:f subst:sg:loc:f subst:pl:acc:m3 subst:pl:voc:m3
Forma	Kazał	kurze	ścierać	kurze
Klasa gram.	{praet}	{adj,subst}	{inf}	{adj,subst}
Przypadek	{}	{nom,dat,acc,loc,voc}	{}	{nom,dat,acc,loc,voc}
Decyzja	praet:sg:m1:perf	subst:sg:dat:f	?	?

Rys. 2.3. Ujednoznacznianie morfosyntaktyczne jako problem klasyfikacji.

Sprowadzenie znakowania do wykreślania niechcianych interpretacji pozwala na łatwiejsze wprowadzenie do modelu klas niejednoznaczności: klasy te pobiera się często

[11] W przykładzie wykorzystano tagset NKJP. Zbiór możliwych tagów przypisanych formie kurze ograniczyliśmy dla czytelności do sześciu pozycji, wybranych arbitralnie. Rzeczywisty zbiór jest znacznie większy. Warto pamiętać, że używany analizator morfosyntaktyczny może również zawierać niepełne bądź błędne dane, więc takie braki mogą też wystąpić w praktyce.

bezpośrednio ze zbiorów możliwych tagów (Tzoukermann i inni, 1997; Màrquez, 1999; Piasecki i Godlewski, 2006a).

Powyższe rozwiązanie zostało zastosowane w tagerze opartym na drzewach decyzyjnych (Màrquez, 1999). Tager stosowany był do znakowania języka angielskiego (udało się osiągnąć trafność 97,27%) oraz hiszpańskiego (97,0%).

Znakowanie warstwowe

W powyższych sformułowaniach pojawia się problem dużego zbioru klas decyzyjnych. Ponieważ tagsety stosowane dla języków słowiańskich zawierają zazwyczaj ponad 1000 możliwych tagów, bezpośrednie użycie ich jako klas decyzyjnych może prowadzić do niskiej trafności klasyfikacji. Zadaniem klasyfikatora jest bowiem przypisanie w jednym przebiegu całego tagu, a zatem jego struktura wewnętrzna nie jest brana pod uwagę. Podejściem alternatywnym jest tzw. **znakowanie warstwowe** (ang. *tiered tagging*), zaproponowane pierwotnie dla języka rumuńskiego (Tufiş, 1999). Wymagało to rzutowania oryginalnego tagsetu (615 tagów) na *tagset uproszczony* (82 tagi); rzutowanie to polegało w głównej mierze na usunięciu mniej istotnych atrybutów. Dzięki temu problem znakowania dało się zrealizować jako proces dwuprzebiegowy: pierwsza **warstwa** znakowania polegała na przypisaniu tagów z tagsetu uproszczonego, natomiast warstwa druga odpowiedzialna była za przypisanie tagów z tagsetu pełnego (oryginalnego).

O ile Tufiş (1999) zainteresowany był przede wszystkim pierwszą warstwą, to jego podejście zostało zastosowane w pełni w tagerze języka polskiego TaKIPI (Piasecki i Godlewski, 2006a): wydzielono trzy warstwy, odpowiadające kolejno za ujednoznacznianie klasy gramatycznej, gramatycznej liczby i rodzaju (razem), a następnie przypadka. Proces znakowania zdania rozpoczyna się od analizy morfosyntaktycznej, po czym w każdej warstwie uruchamiane są klasyfikatory odpowiedzialne za rozstrzyganie wieloznaczności związanych z daną warstwą. Jeśli rozważymy przykład z rysunku 2.3, to ujednoznacznianie formy `kurze` na pozycji „0" sprowadzałoby się do klasyfikacji klasy gramatycznej (w ramach pierwszej warstwy klasyfikator powinien wybrać klasę `subst`, czyli rzeczownik), następnie liczby i rodzaju (klasyfikator drugiej warstwy powinien wybrać symbol `sg:f`, czyli liczbę pojedynczą i rodzaj żeński), a na końcu przypadka (prawidłową decyzją byłby wybór symbolu `dat`, tj. celownika). Można zauważyć, że wybory dokonane na jednej z warstw mogą spowodować redukcję możliwych wyborów na warstwach kolejnych; w przypadku omawianego przykładu, warstwie trzeciej pozostaje wybór spośród zbioru (`dat`, `loc`), gdyż pozostałe wartości przypadka zostały wykluczone przez odrzucenie części tagów w ramach poprzednich warstw. Co więcej, przyjęte w tagerze TaKIPI warstwy nie opisują wszystkich możliwych atrybutów. Powoduje to, że niewielki procent segmentów pozostaje na wyjściu niejednoznaczny. Tager ten został opracowany dla korpusu i tagsetu KIPI i nie był testowany gdzie indziej, a jest prawdopodobne, że w przypadku korpusu NKJP uzyskalibyśmy wyższy procent niejednoznaczności na wyjściu ze względu na większą liczbę klas gramatycznych wyróżnionych w tagsecie[12].

[12] Przetestowanie oryginalnego tagera TaKIPI na korpusie NKJP jest praktycznie niemożliwe ze względu na techniki optymalizacji zastosowane w kodzie. Zmiana tagsetu wymagałaby więc re-implementacji całego algorytmu. Co więcej, tager ten zawiera również zbiór reguł pisanych ręcznie (por. punkt 2.4.1), które wymagałyby też dostosowania do innego tagsetu. Publikowane wyniki oceny

Warto też dodać, że w tagerze TaKIPI zastosowano również mechanizm klas niejednoznaczności, jednak ze względu na jego warstwową budowę, klasy te defiowane są osobno dla każdej warstwy. Przykładowo, na warstwie trzeciej pojawia się klasa `gen, acc` odpowiadająca niejednoznaczności przypadka dopełniacz–biernik (Piasecki i Godlewski, 2006a).

Popularne klasyfikatory

W tym punkcie omówimy kilka popularnych klasyfikatorów. Na potrzeby opisu przyjmujemy, że $X = (x_1, x_2, \ldots, x_N)$ to przypadek do klasyfikacji będący wektorem składającym się z wartości kolejnych N cech, o dziedzinach $F_1, \ldots, F_N$. Symbol $c \in C$ oznaczać będzie klasę (etykietę; wynik klasyfikacji), a każdy przypadek uczący składa się z wektora cech $Y = (y_1, y_2, \ldots, y_N)$ oraz wartości klasy c, co możemy zapisać jednym wektorem jako $V = (y_1, \ldots, y_N, c)$.

Jednym z prostszych klasyfikatorów jest **naiwny klasyfikator bayesowski**[13]. Zakłada on, że dla danej klasy rozkład prawdopodobieństwa wartości poszczególnych cech y_i jest niezależny od rozkładu prawdopodobieństwa wartości pozostałych cech (założenie to jest w ogólności nieprawdziwe; mimo to, klasyfikator daje często zadowalające wyniki). Na mocy twierdzenia Bayesa oraz założenia o niezależności cech, prawdopodobieństwo, że przypadek X należy do klasy c, opisane jest wzorem (2.12). Symbol $P(c)$ oznacza prawdopodobieństwo *a priori* klasy c, zaś $P(x_i|c)$ — prawdopodobieństwo wystąpienia przypadku X w podpróbie przypadków z klasy c_j.

$$P(c|X) = \frac{P(c)P(X|c)}{P(X)} = \frac{P(c)\prod_{i=1}^{N} P(x_i|c)}{P(X)} \tag{2.12}$$

Podczas klasyfikacji dążymy do znalezienia klasy maksymalizującej prawdopodobieństwo warunkowe; wartość wyrażenia z mianownika jest niezależna od wybranej klasy, więc możemy je wyeliminować. Wartość prawdopodobieństw *a priori* klas oraz prawdopodobieństw warunkowych $P(x_i|c)$ można w prosty sposób estymować na podstawie liczności pozyskanych wprost z danych uczących — za pomocą metody największej wiarygodności.

Drzewo decyzyjne[14] to skierowany graf acykliczny i spójny, reprezentujący reguły klasyfikacji. Korzeń drzewa reprezentuje całą próbę uczącą. Węzły potomne odpowiadają *decyzjom*, które dzielą próbę uczącą na rozłączne podzbiory. Do każdego węzła niebędącego liściem przypisane jest kryterium podziału elementów próby uczącej docierających do tego węzła w wyniku poprzednich podziałów. Takie kryterium sprowadza się do testu na wartość jednej z cech. W przypadku cech o wartościach dyskretnych i nominalnych, testy te najczęściej mają postać predykatów $x_i = a$. Każdemu liściowi przypisana jest jedna z klas (może istnieć więcej niż jeden liść przypisujący tę samą klasę). Klasyfikacja przypadku X za pomocą drzewa decyzyjnego polega na przejściu przez ścieżkę od korzenia do liścia, wybierając odpowiednie węzły potomne zgodnie z wartościami zwróconymi przez testy zawarte w węzłach–przodkach.

---

tagerów (Acedański, 2010; Śniatowski i Piasecki, 2011) pokazują, że nawet na korpusie KIPI tager TaKIPI sprawuje się gorzej niż nowsze tagery, dlatego też nie będziemy tego robić.

[13] Omówienie naiwnego klasyfikatora bayesowskiego opracowano na podstawie pracy (Koronacki i Ćwik, 2005, s. 41, 65).

[14] Opis drzew decyzyjnych i algorytmu C4.5 opracowano na podstawie prac (Màrquez, 1999, s. 46–50) oraz (Koronacki i Ćwik, 2005, s. 122–141).

Zadanie budowy drzewa decyzyjnego na podstawie próby uczącej nazywane jest **indukcją drzewa**. Istnieje kilka algorytmów indukcji drzew decyzyjnych. Większość z nich to algorytmy zachłanne, zakładające budowę drzewa od korzenia do liści bez nawrotów. Działanie takich algorytmów rozpoczyna się od analizy całej próby uczącej i wyboru kryterium podziału zbioru, który maksymalizuje pewną funkcję celu. Wybrany podział zostaje przypisany do korzenia, po czym tworzone są węzły potomne odpowiadające dalszym podziałom podzbiorów próby uczącej. W sposób rekurencyjny wybierane są dalsze podziały tych podzbiorów i przypisywane odpowiednim węzłom. Rekurencja kończy się, gdy zachodzi wybrany warunek zatrzymania, np. gdy wszystkie elementy podzbioru trafiającego do węzła należą do jednej klasy, albo gdy liczba elementów w węźle uznana jest za zbyt małą, by miało sens dalsze wnioskowanie statystyczne. Prawdopodobnie najpopularniejszym algorytmem indukcji drzew decyzyjnych jest algorytm C4.5. Algorytm zakłada, że podział wybierany jest w oparciu o statystykę zwaną *Information Gain*. Pozwala ona na oszacowanie, w jakim stopniu każda z cech w izolacji wpływa na wartość klasy; por. wzór (2.13). Statystyka opiera się na entropii warunkowej klasy pod warunkiem wystąpienia danej wartości cechy $H(C|y)$.

$$w_i = H(C) - \sum_{y \in F_i} P(y) H(C|y) \tag{2.13}$$

Algorytm C4.5 zakłada, że po wyuczeniu drzewa w opisany powyżej sposób następuje *przycinanie*, polegające na zastąpieniu wybranych poddrzew przez liście. Celem takiego zabiegu jest zapobieżenie nadmiernemu dopasowaniu drzewa do próby uczącej, zwanego też jego **przeuczeniem**. Przeuczenie stanowi problem, gdyż prowadzi do gorszych wyników klasyfikacji nowych obserwacji (pogarsza zdolność do generalizacji). W algorytmie C4.5 poddrzewa do przycięcia wybierane są na podstawie estymacji błędów klasyfikacji — jeśli prawdopodobieństwo błędu klasyfikacji oszacowane na podstawie podzbioru próby uczącej trafiającej do danego poddrzewa przekracza pewien próg, poddrzewo to jest przycinane. Szczegółowy opis algorytmu C4.5 oraz jego wzorcowej implementacji można znaleźć w pracy (Quinlan, 1993). Algorytm C4.5 wykazuje złożoność czasową $O(D \cdot N)$, gdzie D to liczba przypadków uczących, a N to liczba cech (Su i Zhang, 2006).

Dodatkową zaletą drzew decyzyjnych jest możliwość reprezentacji wyuczonego drzewa jako listy reguł. Taka reprezentacja jest czytelna; pozwala to ekspertowi dziedzinowemu ocenić automatycznie pozyskane reguły, dokonać korekty, a także włączyć je w zbiór reguł napisanych ręcznie (Daelemans i van den Bosch, 2005, s. 5).

Uczenie na pamięć (*uczenie pamięciowe*, ang. *Memory-Based Learning*) jest podejściem alternatywnym w stosunku do popularnych technik opartych na generalizacji. Idea jest prosta: sprowadzamy uczenie klasyfikatora do zapamiętywania wszystkich przypadków uczących, a sam proces klasyfikacji nowego przypadku — do znajdowania w bazie przypadku podobnego (Daelemans i van den Bosch, 2005).

Wyuczony klasyfikator pamięciowy[15] jest zatem zbiorem przypadków uczących (bazą przypadków). Klasyfikacja przypadku X polega na porównaniu jej ze wszystkimi przypadkami uczącymi Y z bazy i wyliczenia dla każdej takiej pary wartości *miary odległości* $\Delta(X, Y)$. Następnym krokiem jest wybór k przypadków uczących o

[15] Opis klasyfikatora pamięciowego opracowano na podstawie prac (Daelemans i van den Bosch, 2005) oraz (Daelemans i inni, 2010b).

najmniejszej odległości od przypadku X i jego klasyfikacja za pomocą etykiety najczęściej pojawiającej się w pozyskanym zbiorze k przypadków. Taki algorytm klasyfikacji jest często nazywany algorytmem k najbliższych sąsiadów (ang. *k-nearest neighbour*, *k-NN*), a wspomniany zbiór k przypadków nazywany jest zbiorem sąsiadów. Miara odległości oraz wartość k są parametrami klasyfikatora.

W przypadku cech o charakterze symbolicznym (głównie z takimi mamy do czynienia w przypadku znakowania morfosyntaktycznego) najprostszą miarą odległości jest *metryka taksówkowa* (ang. *Manhattan distance*, *overlap metric*) opisana poniższym wzorem (δ to symbol Kroneckera):

$$\Delta(X, Y) = \sum_{i=1}^{N} \delta(x_i, y_i) \tag{2.14}$$

Metryka taksówkowa zlicza cechy, których wartości się różnią między porównywanymi przypadkami. Naturalnym rozwinięciem tej metryki jest przypisanie cechom wag.

$$\Delta(X, Y) = \sum_{i=1}^{N} w_i \delta(x_i, y_i) \tag{2.15}$$

Wartości wag można dobrać ręcznie kierując się wiedzą dziedzinową (albo intuicją lingwistyczną w przypadku przetwarzania języka naturalnego). Inną metodą doboru wag jest zastosowanie statystyk oceniający wpływ wartości poszczególnych cech na wartość klasy. Często stosowaną statystyką jest wspomniana wcześniej *Information Gain*, wzór (2.13).

Dotychczas rozważane miary odległości zakładają binarne porównanie wartości danej cechy: albo $x_i = y_i$, albo $x_i \neq y_i$. W przypadku cech o większych dziedzinach typu symbolicznego może być pożądane rozważenie podobieństwa między różnymi wartościami tej samej cechy. Przykładowo, intuicja wskazuje, że klasa gramatyczna *przymiotnik* jest bardziej podobna do klasy gramatycznej *imiesłów przymiotnikowy czynny* niż do klasy *czasownik w bezokoliczniku*. Tego typu informację o podobieństwie wartości cech uwzględnia miara Scotta–Salzberga — *Modified Value Difference*. Miara realizowana jest przez modyfikację schematu z wzoru (2.14) — symbol δ otrzymuje nową interpretację, opisaną wzorem (2.16). Odległość między wartościami x_i i y_i cechy F_i ustalana jest na podstawie różnicy między warunkowymi rozkładami prawdopodobieństwa klas decyzyjnych pod warunkiem konkretnej wartości cechy.

$$\delta(x_i, y_i) = \sum_{j=1}^{|C|} |P(c_j|x_i) - P(c_j|y_i)| \tag{2.16}$$

Standardowo decyzja klasyfikatora wybierana jest ze zbioru sąsiadów na drodze głosowania większościowego, tj. jako ostateczna decyzja wybierana jest ta etykieta, która przypisana jest największej liczbie sąsiadów. Istnieje też możliwość wprowadzenia głosowania z wagami, gdzie wagi nadawane są sąsiadom w zależności od ich odległości od klasyfikowanego przypadku X. Popularnym schematem przypisywania wag jest schemat Dudaniego — *Inverse Linear*, zakładający przypisanie sąsiadom wag malejących liniowo wraz ze wzrostem ich odległości od klasyfikowanego przykładu. Wzór (2.17) określa wagę w_j decydującą, jaki wpływ ma j-ty najbliższy sąsiąd przypadku

X na wynik głosowania. d_1 to odległość przypadku X od najbliższego sąsiada, d_k to odległość przypadku X od najdalszego sąsiada.

$$w_j = \begin{cases} \frac{d_k - d_j}{d_k - d_1}, & d_k \neq d_1 \\ 1, & \text{w pp.} \end{cases} \tag{2.17}$$

Klasyfikator pamięciowy działa w oparciu o funkcję podobieństwa. Dzięki temu w naturalny sposób wspiera wnioskowanie zarówno na podstawie mocnych przesłanek (tj. reprezentowanych przez dużą liczbę przypadków uczących), jak i słabych przesłanek, związanych nawet z pojedynczymi przypadkami uczącymi. Jest to szczególnie istotne w przypadku danych pochodzących z języka naturalnego — zjawiskiem bardzo powszechnym w języku są bowiem wyjątki, które cechuje pewna regularność (ang. *subregularities* — odstępstwa od ogólnej reguły, które rządzą się jednak pewnymi regułami). Uczenie oparte na generalizacji prowadzi do pominięcia rzadszych wyjątków, co często jest niekorzystne (Daelemans i van den Bosch, 2005, s. 124). Drugą ważną zaletą modelu jest możliwość użycia opisanych wyżej statystyk do estymacji wartości wag przypisanym cechom. Pozwala to na definiowanie dużych zbiorów cech w oparciu o różnorakie intuicje lingwistyczne; jeśli nawet intuicja okaże się błędna, użyta technika estymacji wag ma szansę zminimalizować negatywny wpływ niepotrzebnych cech na wynik klasyfikacji (Daelemans i inni, 2010b).

Wadą klasyfikatora pamięciowego jest złożoność obliczeniowa klasyfikacji pojedynczego przypadka: $O(N \cdot D)$, gdzie N to liczba cech, a D to liczba przypadków uczących. Istnieją techniki, które pozwalają na zmniejszenie praktycznej złożoności przy niewielkim spadku trafności klasyfikacji, np. usuwanie mało istotnych przypadków z bazy (Daelemans i van den Bosch, 2005). Zaletą uczenia pamięciowego jest za to bardzo szybkie „uczenie", które sprowadza się jedynie do zapamiętania przypadków uczących.

2.4.4. Automatyczne pozyskiwanie reguł

Istnieje też grupa metod, które sprowadzają się do automatycznego pozyskiwania reguł na podstawie analizy korpusu uczącego. Część tych metod można również uznać za przynależne do wcześniej omawianej grupy, tj. metod znakowania przez klasyfikację poszczególnych segmentów. W niniejszym opracowaniu postanowiliśmy jednak metody te omówić osobno, gdyż cechą wspólną jest tutaj zapis pozyskanej wiedzy w postaci reguł, które są zrozumiałe dla eksperta dziedzinowego. Co więcej, nie zawsze reguły takie operują na poziomie pojedynczych segmentów, więc nie wszystkie metody da się wyrazić jako problem klasyfikacji segmentów i ich kontekstów.

Prawdopodobnie pierwsze podejście do automatycznego pozyskiwania reguł znakowania morfosyntaktycznego zaproponował Hindle (1989). Założeniem jest poprawa jakości istniejącego tagera opartego na liście reguł ujednoznaczniania napisanych ręcznie. Stosowana lista reguł jest listą decyzyjną, tj. wybierana jest zawsze pierwsza reguła spośród tych, które da się zaaplikować. Algorytm pozyskiwania reguł składa się z dwóch etapów: generowania reguł oraz usuwania reguł. Generowanie reguł polega na odnalezieniu miejsc, gdzie zastosowanie listy decyzyjnej prowadzi do błędu oraz dodaniu nowych reguł korygujących te błędy. Etap usuwania reguł polega na ocenie reguł pod kątem trafności; jeśli odsetek wprowadzanego błędu jest zbyt duży, reguła jest usuwana. Wadą opisanego podejścia jest generowanie znacznej liczby nadmiarowych reguł

— reguły oceniane są jedynie pod kątem wprowadzanego błędu, reguły nadmiarowe nie są natomiast rozpoznawane.

Ciekawe podejście do indukcji reguł zaproponował (Brill, 1992). Jedną z głównych zalet algorytmu Brilla jest pozyskiwanie stosunkowo niedużych zbiorów prostych reguł. Idea jest prosta: w pierwszej kolejności segmentom przypisywane są tagi za pomocą prostej heurystyki, po czym w kolejnych iteracjach odkrywane są reguły poprawiające błędne oznakowanie. Algorytm zakłada, że z korpusu uczącego wydzielona jest niewielka część, która stanowi tzw. *korpus poprawkowy* (ang. *patch corpus*). Pozyskiwane reguły są stosunkowo proste. Akcją reguły jest zamiana tagu t_a na tag t_b, o ile zachodzą określone warunki kontekstowe. Warunki kontekstowe można wyrazić jako realizację jednego z poniższych szablonów (na podstawie Brill, 1992 oraz Acedański, 2010):

1. Segmentowi na pozycji $p \in \{-2, -1, 0, 1, 2\}$ przypisano tag t_z (p i t_z są parametrami szablonu).
2. Istnieje segment na którejś z pozycji należących do podzbioru $P \in \{\{-2, -1\}, \{1, 2\}, \{-3, -2, -1\}, \{1, 2, 3\}\}$, któremu przypisano tag t_z.
3. Segmentowi na pozycji -1 przypisano tag t_z, a segmentowi na pozycji 1 przypisano tag t_w.
4. Segmentowi na pozycji -1 przypisano tag t_z, a segmentowi na pozycji -2 przypisano tag t_w.
5. Segmentowi na pozycji 1 przypisano tag t_z, a segmentowi na pozycji 2 przypisano tag t_w.
6. Segment bieżący (na pozycji 0) (nie) zaczyna się wielką literą.
7. Segment na pozycji -1 (nie) zaczyna się wielką literą.

Wspomniana heurystyka początkowego przypisania tagów zakłada, że formom przypisywane są tagi, które najczęściej były im przypisane w korpusie uczącym. W przypadku form, które w korpusie uczącym nie pojawiły się w ogóle, przypisywane są tagi najczęstsze dla form, które kończą się tymi samymi trzema znakami. W każdej iteracji algorytmu tworzone są instancje powyższych szablonów, po czym instancje te są oceniane. Wynikiem iteracji jest wybranie reguły, która przynosi największy zysk, liczony jako różnica między liczbą segmentów z korpusu poprawkowego, które dzięki zyskały poprawne oznakowanie a liczbą segmentów, które utraciły prawidłowe oznakowanie. Algorytm zatrzymuje się, gdy uzyskany w danej iteracji zysk nie przekracza podanego progu (Brill, 1992).

Brill (1992) testował swój algorytm dla języka angielskiego, osiągając trafność 95%–96,5% w zależności od użytych korpusów. Tager Brilla został później przetestowany na korpusie języka słoweńskiego, udało się osiągnąć trafność znakowania 85,95% (Džeroski i inni, 1999). Kuta (2010) przetestował oryginalny algorytm Brilla na korpusie FREK, osiągając 84,66% trafności.

Jak zauważają Acedański i Gołuchowski (2009), oryginalny algorytm Brilla narażony jest na problem rzadkich tagów, jeśli zastosujemy go do znakowania języka polskiego. W celu uniknięcia tego problemu, autorzy ci proponują wprowadzić do algorytmu kilka modyfikacji. Proponowane modyfikacje do tagera Brilla zostały w pełni rozwinięte w pracy (Acedański, 2010) i na tej wersji algorytmu się skupimy. Główną modyfikacją jest wprowadzenie znakowania warstwowego: pierwsza warstwa odpowiedzialna jest za klasę gramatyczną, ale też za dwa atrybuty: przypadek i osobę. Druga

warstwa odpowiada pozostałym atrybutom tagsetu (rozważane są tagsety KIPI i NKJP, gdzie zbiór atrybutów jest prawie identyczny).

Drugą ważną modyfikacją opisaną przez Acedańskiego (2010) jest uogólnienie szablonów. Po pierwsze, akcja reguły ma postać „zamień wartość atrybutu a (lub klasy gramatycznej) z wartości v na wartość w". Po drugie, warunki kontekstowe poddane zostały podobnej modyfikacji: wszystkie testy na wartości tagu zamieniono w sposób analogiczny na testy na wartość klasy gramatycznej lub konkretnego atrybutu, np. „jeśli segmentowi na pozycji p przypisano tag o wartości v atrybutu a...". Zestaw szablonów dodatkowo rozszerzono o testy na 2- i 3-literowe prefiksy i sufiksy form wyrazowych.

Pozostałe modyfikacje algorytmu wiążą się z poprawą jego wydajności. Algorytm został zaimplementowany w postacie tagera o nazwie PANTERA, którego kody udostępniono na otwartej licencji GNU GPL. Acedański (2010) przeprowadził ocenę tagera na ręcznie ujednoznacznionej części korpusu KIPI, osiągając słabą poprawność 92,44%, oraz na fragmencie korpusu NKJP zawierającym 648 000 segmentów, osiągając trafność 92,68% lub 92,82% (w zależności od przyjętego progu zysku). Jako że wyniki te są bardzo dobre, tager PANTERA będzie głównym punktem odniesienia dla metod proponowanych w tej rozprawie.

Indukcyjne programowanie w logice (ang. *Inductive Logic Programming*, *ILP*) to rodzina technik indukcji reguł łącząca metody maszynowego uczenia z programowaniem w logice. Zadanie sformułowane jest jako poszukiwanie hipotezy $\mathcal{H}$, takiej że wraz z dostarczoną wiedzą dziedzinową $\mathcal{B}$ (ang. *background knowledge*), dają podstawy do udowodnienia zbioru przypadków uczących $\mathcal{E}$, tj. $\mathcal{H} \cap \mathcal{B} \models \mathcal{E}$. Zarówno wiedza dziedzinowa, jak i wydobyte reguły zapisywane są w postaci programów logicznych opartych o język rachunku predykatów, na ogół wyrażonych w języku programowania Prolog. Stosuje się dwa ogólne podejścia do indukcji hipotez: *uogólnianie* przypadków uczących oraz *uszczegóławianie* najbardziej ogólnej hipotezy. Technika uogólniania polega na przeszukiwaniu przestrzeni hipotez począwszy od najbardziej szczegółowych formuł, które implikują przypadki uczące. Hipotezy takie są iteracyjnie uogólniane. W przypadku uszczegóławiania, przeszukiwanie zaczyna się od najbardziej ogólnej hipotezy (Lavrač i Džeroski, 1994). Indukcyjne programowanie w logice jest atrakcyjne z punktu widzenia przetwarzania języka naturalnego, gdyż pozwala na włączenie wiedzy dziedzinowej w naturalny sposób do modelu, a siła ekspresji logiki predykatów jest wystarczająca z punktu widzenia wielu problemów przetwarzania języka (Cussens i inni, 1997).

Zastosowanie techniki indukcyjnego programowania w logice do problemu znakowania morfosyntaktycznego języka angielskiego zaproponował Cussens (1997). Przyjęte rozwiązanie zakładało dwuetapowe podejście, gdzie analizator morfosyntaktyczny przypisywał zbiory możliwych tagów, natomiast zadaniem wydobytych reguł było kontekstowe ujednoznacznianie morfosyntaktyczne. Wydobywane reguły miały postać *„usuń tag T z tagów przypisanych segmentowi na pozycji centralnej, jeśli z lewej strony jest ciąg tagów T_L, a z prawej — T_P*. Co ciekawe, wiedza dziedzinowa zawierała predykaty definiujące proste frazy, np. frazę rzeczownikową złożoną z przymiotnika i rzeczownika. Gramatyka ta miała charakter płytki (gramatyki pisane na cele płytkiej analizy składniowej omawiamy w rozdziale 4.4.1). Autor sam podkreśla, że zdaje sobie sprawę z niedoskonałości gramatyki, jednak jej celem nie jest analiza składniowa, lecz wspo-

maganie indukcji reguł ujednoznaczniania. Eksperymenty pokazują, że gramatyka ta faktycznie poprawia skuteczność modułu.

Indukcyjne programowanie w logice znalazło również zastosowanie w ujednoznacznianiu morfosyntaktycznym języków słowiańskich. Cussens i inni (1999) opisują indukcję reguł wykreślających tagi w tekście słoweńskim. Wiedza dziedzinowa zawierała predykaty pozwalające na sprawdzenie wartości poszczególnych atrybutów, np. liczby, rodzaju, przypadka, ale także predykaty sprawdzające czy dwa segmenty są zgodne co do wartości liczby, przypadka i rodzaju (z osobna, tj. zdefiniowano 3 oddzielne predykaty). Trafność systemu oceniona została na 87,5%.

Technika ILP znalazła również zastosowanie w ujednoznacznianiu morfosyntaktycznym języka czeskiego (Nepil i inni, 2001). Założenia pierwszych eksperymentów były jednak inne: technika ta była jedynie narzędziem wspomagającym działanie innyc metod, m.in. gramatyki pisanej ręcznie. Późniejsze prace zakładały większą rolę odgrywaną przez system ILP (Nepil, 2003): inne techniki nie były stosowane, choć system pozostawiał na wyjściu spory procent niejednoznacznych segmentów. Eksperymenty jednak pokazały, że odsetek błędów wprowadzonych przez system był minimalny (ok. 0,1%), natomiast redukcja niejednoznaczności wejściowej była dwukrotna. Podobnie jak w przypadku wcześniej omawianego tagera języka słoweńskiego, i tu wiedza dziedzinowa zawierała predykaty sprawdzające wartości poszczególnych atrybutów a także pewnych uzgodnień.

W pracy (Šmerk, 2004) opisano ciekawą modyfikację powyższej metody, która pozwala na indukcję reguł ujednoznaczniania na podstawie korpusu nieoznakowanego. Metoda opiera się na wykorzystaniu dostępnego analizatora morfosyntaktycznego: wielki zbiór tekstu zostaje poddany analizie morfosyntaktycznej, przez co możliwe jest znalezienie segmentów jednoznacznych, które tworzą interesujące wzorce. Wzorce te sprowadzane są do postaci reguł, które w przybliżeniu opisują również ciągi prawidłowych tagów, które należy wybrać dla słów wieloznacznych.

2.4.5. Metody hybrydowe i łączenie tagerów

Popularną metodą budowy tagerów jest połączenie reguł pisanych ręcznie, na ogół niosących informację w miarę pewną pewną, z metodą statystyczną bądź opartą na klasyfikacji, która to podejmuje decyzje, które trudniej było sformułować w postaci reguł. Omawiany wcześniej tager TaKIPI stosuje tę właśnie technikę: w pierwszej kolejności uruchamiane są reguły usuwające niektóre interpretacje, po czym drzewa decyzyjne wykonują dalszy ciąg ujednoznaczniania. Podejście to było stosowane wcześniej, m.in. w tagerze języka czeskiego (Hajič i inni, 2001), gdzie po ujednoznacznianiu za pomocą reguł stosowana była metoda statystyczna oraz w tagerze języka bułgarskiego, gdzie reguły pisane ręcznie połączono z zastosowaniem sieci neuronowych (Simov i Osenova, 2001).

Innym popularnym podejściem pozwalającym na połączenie kilku różnych technik jest głosowanie. Idea jest prosta: mając do dyspozycji kilka gotowych tagerów, które mylą się w różnych miejscach, przeprowadzamy głosowanie na poziomie tagów zwróconych przez pojedyncze tagery i wybieramy tag najczęstszy (w przypadku głosowania większościowego) lub tag, który otrzymał największą liczbę punktów (gdy przypiszemy różne wagi poszczególnym tagerom). Zaobserwowano, że najlepsze wyniki daje połączenie metod, które różnią się w istotny sposób, oraz że zbiór powinien

uwzględniać przynajmniej jeden tager, który samodzielnie daje dobre wyniki (Sjöbergh, 2003). Badania przeprowadzone dla języka angielskiego pozwoliły na redukcje błędu o 19% (Van Halteren i inni, 2001), dla języka szwedzkiego — o 19%. Eksperymenty z głosowaniem większościowym dla języka polskiego przeprowadził Śniatowski (2011) i odnotował spadek liczby błędów o 15% na korpusie KIPI.

Technika *bootstrappingu*, tj. przyrostowego uczenia jest opisana w pracy (Clark i inni, 2003). Autorzy biorą dwa tagery i wynikiem jednego uczą drugi, potem na odwrót. Taki zabieg określany jest jako *co-training* (*współuczenie*). Autorzy twierdzą, że przynosi on dobre wyniki w różnych problemach przetwarzania języka. Użyto tagera opartego na modelu Markowa oraz tagera implementującego metodę maksymalizacji entropii. Pokazano, że można osiągnąć zaskakująco ciekawe wyniki stosując niewielką ilość ręcznie oznakowanego tekstu i znacznie większy korpus nieoznakowany. Omówiona jest też metoda wybierania danych do ujednoznacznienia podczas iteracji: wybierana jest taka część danych, która maksymalizuje zgodność obu tagerów.

Chociaż powyższe metody hybrydowe mają duży potencjał, przebadanie ich leży poza zakresem tej rozprawy. Wierzymy jednak, że zaproponowane przez nas metody znakowania morfosyntaktycznego mogą poprawić wyniki tagera głosującego, gdyż metody te samodzielnie dają dobre wyniki, a także różnią się w sposób istotny od metod do tej pory stosowanych w ramach tagerów głosujących języka polskiego (Śniatowski, 2011).

2.5. Problem oceny tagerów

Jak wspomniano w punkcie 2.1, jakość użytego tagera ma istotny wpływ na wyniki osiągnięte w kolejnych etapach przetwarzania tekstu. Istotnym problemem praktycznym jest zatem wybór najlepszego spośród dostępnych tagerów pod kątem danego zastosowania. Idealnym rozwiązaniem wydaje się ocena tagera przez ocenę wyników osiągniętych przez cały system, którego komponentem jest dany tager; w ten sposób można poznać rzeczywisty wpływ wybranego tagera na wyniki całego systemu.

Taka procedura oceny jest jednak mało praktyczna, gdyż musi być przeprowadzana każdorazowo pod kątem konkretnego zastosowania — a proces ten może być bardzo pracochłonny. Dlatego też koniecznością jest przeprowadzenie możliwe rzetelnej oceny jakości samego znakowania morfosyntaktycznego. Ocena taka, choć nie zawsze w pełni odzwierciedla użyteczność tagera jako narzędzia wspomagającego rozwiązywanie danego problemu, jest praktyczną „heurystyką" dającą podstawy do wyboru tagera bez konieczności przeprowadzania czasochłonnych badań. Miara taka jest również kluczowa, by ocenić wyniki prac nad nowymi tagerami (bądź ulepszaniem istniejących) — pozwala na porównanie algorytmów w oparciu o pewne abstrakcyjne kryterium zgodności wyjścia tagera z oznakowaniem wzorcowym.

2.5.1. Popularne miary oceny tagerów

Najprostszym sposobem oceny tagera jest porównanie wyjścia tagera z korpusem wzorcowym oznakowanym przez lingwistę. Standardowa miara, zwana *trafnością* (ang. *accuracy*) określa procent segmentów, którym tager przypisał prawidłowe interpretacje — tj. interpretacje identyczne z interpretacjami pochodzącymi z korpusu wzorcowego

(Manning i Schütze, 1999). Miara ta na ogół nie jest definiowana bardzo precyzyjnie, co pozostawia pewną dowolność w jej interpretacji. Po pierwsze, na ogół oceniane jest *de facto* jedynie przypisanie tagów, a trafność przypisanych lematów nie podlega ocenie. Przypomnijmy, że przez interpretację morfosyntaktyczną rozumiemy parę (*tag*, *lemat*). W większości rzeczywistych przypadków przypisanie prawidłowego tagu pozwala na jednoznaczny wybór prawidłowego lematu (dzięki słownikowi analizatora morfosyntaktycznego). Są jednak sytuacje, gdy tag nie determinuje lematu dla danej formy wyrazowej. Przykładowo, formie kręgu możemy przypisać lemat krąg lub krąg, nawet jeśli wiemy, że dane jej wystąpienie ma tag `subst:sg:gen:m3` (rzeczownik, liczba poj., dopełniacz, rodzaj męski nieożywiony). W niniejszej pracy skupiamy się na przypisywaniu tagów i, o ile nie zostanie powiedziane inaczej, podane miary dotyczyć będą jedynie prawidłowego przypisania tagów. W ten sposób rozumiana trafność uzyskuje definicję 2.18 (N to liczba segmentów w korpusie, *tag* opisuje przyporządkowanie numerom segmentów tagów wykonane przez tager, natomiast *ref* to przyporządkowanie tagów z korpusu wzorcowego).

$$Acc = \frac{|\{i : tag(i) = ref(i), 0 < i \leqslant N\}|}{N} \tag{2.18}$$

Naturalnym rozwinięciem oceny jest podanie osobnego wyniku dla trafności przypisywania lematów. Istnieje też inna możliwość: oceniać trafność przypisywania całych interpretacji (jako trafne liczymy jedynie te interpretacje, gdzie przypisano zarówno prawidłowy tag, jak i prawidłowy lemat).

Drugim problem jest założenie, że wyjście z tagera oraz korpus wzorcowy zawierają identyczny podział na segmenty. Problem ten dotyczy wszystkich powszechnie używanych miar i zostanie szczegółowo omówiony w następnym punkcie.

Istnieją sytuacje, gdy miary trafności nie da się zastosować. Sytuacja taka ma miejsce, jeśli:

1. wzorcowy korpus przypisuje więcej niż jeden tag niektórym segmentom lub
2. tager przypisuje więcej niż jeden tag niektórym segmentom.

Obecność wieloznaczności w korpusie wzorcowym może być wytłumaczona na gruncie lingwistycznym. Przepiórkowski (2004) podaje m.in. przykładowe zdanie 2.19, gdzie nie jest możliwe określenie, czy formę go należy interpretować jako formę biernika, czy też dopełniacza. Wieloznaczność taką zachowano w oznakowaniu korpusu IPI PAN (Przepiórkowski, 2004).

(2.19) Pożądała go.

Druga sytuacja może zajść, gdy twórcy algorytmu ujednoznaczniania świadomie zrezygnują z rozstrzygania niejednoznaczności pewnego typu. Decyzję taką podjęto w przypadku tagera TaKIPI. Przykładowo, rozróżnienie między zwykłymi rzeczownikami a odsłownikami (gerundiami) uznano za zbyt trudne i tager takich decyzji nie podejmuje (np. słowo zdanie może być interpretowane jako zdanie egzaminu, albo zdanie proste/złożone; por. Piasecki i Godlewski, 2006a). Jako że tager ten był uczony i testowany na korpusie IPI PAN, występowały tam oba typy wieloznaczności.

W pracy Karwańska i Przepiórkowski (2010) opisano szczegółowo tę problematykę. Zaproponowano kilka miar, które pozwalają ocenić jakość oznakowania w sytuacji wystąpienia niejednoznaczności tego typu. Warto przytoczyć dwie spośród tych miar:

słabą poprawność (ang. *weak correctness*, WC) oraz *silną poprawność* (ang. *strong correctness*, SC). Obie miary zdefiniowano analogicznie do trafności, tj. przez procent segmentów poprawnie rozpoznanych. W przypadku silnej poprawności (wzór 2.21) segment uznajemy za poprawnie rozpoznany jedynie jeśli otrzymał on dokładnie ten sam zbiór tagów, jaki przypisano mu w korpusie wzorcowym; w przypadku słabej poprawności (2.20) warunkiem wystarczającym jest niepuste przecięcie obu zbiorów — tj. musi istnieć przynajmniej jeden tag przypisany przez tager, który przypisano również w korpusie wzorcowym danemu segmentowi.

$$WC = \frac{|\{i : tag(i) \cap ref(i) \neq \emptyset, 0 < i \leqslant N\}|}{N} \tag{2.20}$$

$$SC = \frac{|\{i : tag(i) = ref(i), 0 < i \leqslant N\}|}{N} \tag{2.21}$$

Warto tutaj jednak zaznaczyć, że wspomniane rodzaje niejednoznaczności wiążą się ze stosunkowo niewielką liczbą korpusów i tagerów, a na ogół przyjmuje się, że zarówno korpus wzorcowy, jak i tager przypisują zawsze dokładnie jeden tag per segment. Sytuacja taka ma miejsce również w Narodowym Korpusie Języka Polskiego (NKJP). Wytyczne znakowania NKJP nakazują podjąć niezbędny wysiłek, by wybrać najtrafniejszą interpretację morfosyntaktyczną, choćby oznaczało to konieczność analizy szerszego kontekstu niż jedno zdanie. Gdy niejednoznaczności nie można rozstrzygnąć, instrukcja nakazuje wybór interpretacji, która wydaje się najbardziej typowa (Przepiórkowski i inni, 2012).

2.5.2. Czy popularne metody oceny są rzetelne?

Ocena tagera na podstawie korpusu wzorcowego jest ważna ze względów praktycznych. Ocena taka powinna być zatem możliwie rzetelna, by ułatwić wybór najlepszego z dostępnych rozwiązań. Co więcej, stosowanie nierzetelnej procedury oceny może powodować długoterminowe konsekwencje: jeśli sposób oceny zaniedbuje pewien podzbiór rzeczywistych błędów popełnianych przez tagery, błędy te pozostają niezauważone przez środowisko naukowe. Zmniejsza to szanse, że błędy te zostaną kiedykolwiek naprawione.

W dalszej części tego punktu wykazujemy, że na ogół stosowana procedura oceny tagerów języka polskiego pozostawia wiele do życzenia. Proponujemy również alternatywną metodę oceny tagerów, która pozwala uniknąć przytoczonych problemów. Proponowana metoda nie jest zupełnie nowa, tj. podobna metoda została już zastosowana do oceny polskich tagerów (Karwańska i Przepiórkowski, 2010); przeprowadzonej ocenie nie towarzyszyła jednak żadna dyskusja na temat użytej metody. Problem ten wydaje się ważny, zarówno z praktycznego, jak i teoretycznego punktu widzenia, dlatego też celowe jest jego uwidocznienie i dyskusja. Elementy te stanowią wkład tej rozprawy w dziedzinę znakowania morfosyntaktycznego. Zaproponowana ostatecznie metoda oceny tagerów uwzględniająca zmiany w segmentacji jest natomiast wynikiem współpracy autora tej rozprawy z Szymonem Acedańskim (Radziszewski i Acedański, 2012).

Jak wspomniano w punkcie 1.2.2, z przyczyn praktycznych znakowanie morfosyntaktyczne języków słowiańskich często wykonuje się dwuetapowo — najpierw dokonuje

się analizy morfosyntaktycznej (pobrania możliwych interpretacji ze słownika), po czym zbiory interpretacji poddane zostają ujednoznacznieniu. W przypadku języków słowiańskich praktyka ta stała się tak powszechna[16], że *tagerem* często nazywany jest sam moduł ujednoznaczniania morfosyntaktycznego. Obswerwacja ta wiąże się bezpośrednio z oceną tagerów: w zależności od części całego systemu, którą zamkniemy w czarną skrzynkę opisaną jako „tager", przeprowadzimy ocenę *de facto* różnych układów i uzyskamy różne wyniki. Chociaż spostrzeżenie to może wydawać się oczywiste, praktyka pokazuje, że tak nie jest: nie dość że kwestia precyzyjnej definicji tagera nie pojawia się w literaturze na temat znakowania języków słowiańskich, to różne publikacje przyjmują milcząco różne założenia. Konsekwencją tego jest fakt, że opublikowane wyniki eksperymentów przeprowadzonych nawet na tych samych danych nie są ze sobą porównywalne. Istnieją przynajmniej trzy możliwe definicje tagera jako czarnej skrzynki; każda z nich jest przynajmniej niekiedy przyjmowana w literaturze:

1. Tager dostaje na wejściu czysty tekst, tj. ciąg znaków; definicja zakładana w: Karwańska i Przepiórkowski (2010).
2. Tager przetwarza ciąg nieoznakowanych segmentów (zgrupowanych w zdania), definicja zakładana w: Džeroski i inni (1999); Schmid i Laws (2008); Acedański i Przepiórkowski (2010); Daelemans i inni (2010b).
3. Tager wykonuje jedynie ujednoznacznianie morfosyntaktyczne tekstu; zakładane w: Piasecki (2007); Hajič i Vidová-Hladká (1998b); Acedański (2010); Śniatowski i Piasecki (2011); Radziszewski i Śniatowski (2011b).

Drugie podejście zakłada, że segmentacja tekstu wykonana przez tager jest bezbłędna (oceny dokonuje się na wzorcowym podziale na segmenty, tj. pochodzącym z korpusu wzorcowego). Podejście trzecie jest najbardziej kontrowersyjne, gdyż zaniedbuje zarówno błędy w segmentacji, jak i błędy popełnione na etapie analizy morfologicznej. Mimo to, podejście przyjmowane jest w literaturze zdecydowanie najczęściej, przynajmniej gdy oceniane są tagery języka polskiego.

Następujące punkty wykazują, że jedynym rzetelnym podejściem do oceny tagerów jest ocena zakładająca znakowanie czystego tekstu; dwa pozostałe zaś podejścia obciążone są błędem systematycznym i powinno się ich unikać.

1. W normalnych okolicznościach użytkownik ma dostęp do czystego tekstu[17]. Uruchamiając tager, dąży do uzyskania sensownej segmentacji i możliwie trafnego oznakowania otrzymanych segmentów interpretacjami morfosyntaktycznymi.
2. Jednym z symptomów nierzetelnej procedury oceny jest brak oddzielnych wyników dla słów znanych i nieznanych w publikacjach na temat polskich tagerów; praktyka ta jest powszechna w przypadku prac wykonanych dla języka angielskiego. Wyniki takie nie są podawane, gdyż podczas testów samego modułu ujednoznaczniania problem słów nieznanych w ogóle nie występuje. Testowana jest bowiem zdolność tagera do wykreślania interpretacji z korpusu wzorcowego — a w korpusie wzorcowym każdy segment ma przypisana przynajmniej jedną interpretację. Prowadzi to

[16] Hajič i Vidová-Hladká (1998b) twierdzą, że „biorąc pod uwagę charakter języków fleksyjnych (...), konieczne jest zastosowanie analizy morfologicznej przed właściwym znakowaniem morfosyntaktycznym".

[17] Tekst może zawierać częściowe formatowanie, np. podział na akapity. W tym rozdziale przez *czysty tekst* rozumiemy tekst niepoddany nawet najprostszej formie analizy językowej. Oznacza to m.in., że dostępny tekst nie został podzielony na zdania ani segmenty.

do absurdalnej sytuacji, gdzie właściwe oznakowanie słów nieznanych (nieobecnych w słowniku analizatora morfosyntaktycznego) jest zadaniem trywialnym, łatwiejszym niż prawidłowe oznakowanie słów znanych[18].

3. Ocena ograniczona do zdolności ujednoznaczniania pomija wpływ konkretnego analizatora morfosyntaktycznego na całościowy wynik. Problem ma naturę praktyczną, gdyż istnieje kilka analizatorów dla języka polskiego (Radziszewski i Maziarz, 2011; Radziszewski i Śniatowski, 2011a; Woliński, 2006; Hajnicz i Kupść, 2001), podczas gdy wspomniana dwuetapowa realizacja znakowania morfosyntaktycznego pozwala na łatwą podmianę użytego analizatora.
4. Jedynie rzetelna ocena tagerów pokazuje różnice wynikające z obecności różnych strategii rozpoznawania słów nieznanych; stymuluje to dalsze prace nad takimi strategiami.

Należy tutaj podkreślić, że rzetelna ocena tagerów języka polskiego została już przeprowadzona: praca Karwańska i Przepiórkowski (2010) opisuje porównanie dwóch tagerów języka polskiego — statystycznego tagera Dębowskiego (2004) z tagerem TaKIPI (Piasecki, 2007); testy przeprowadzone zostały na czystym tekście, a uzyskane wartości miar są wyraźnie niższe niż opublikowane wcześniej. Sam fakt, że ocena została przeprowadzona w oparciu o znakowanie czystego tekstu nie jest jednak nawet wzmiankowany w artykule[19], nie podjęto też próby wyjaśnienia dużej różnicy w stosunku do wcześniej publikowanych wyników. Warto też dodać, że następujące po tym artykule publikacje wiążące się z oceną tagerów języka polskiego stosują „stare" podejście, tj. ocenę jedynie ujednoznaczniania: Acedański (2010); Śniatowski i Piasecki (2011); Radziszewski i Śniatowski (2011b). Podobna sytuacja miała miejsce dla języka czeskiego: Hajič przyznaje w przypisie na s. 3 artykułu (2000), że „w przeszłości problem słów nieznanych był po prostu ignorowany".

2.5.3. Proponowana metoda oceny tagerów

Jak wykazaliśmy w poprzednim punkcie, jedynie testy tagera na czystym tekście pozwalają przetestować wszystkie składniki systemu i uzyskać w ten sposób wiarygodne przybliżenie rzeczywistego odsetka błędów, które mogą się pojawić na wyjściu. Przeprowadzenie takich testów wymaga jednak przyjęcia kilku dodatkowych założeń. Należy się bowiem spodziewać, że ponowne przetworzenie tekstu przez moduł segmentacji prowadzić będzie do różnic w segmentacji tekstu; problem może dotyczyć podziału na segmenty, jak i podziału na zdania.

Jest kwestią sporną, czy błędy w podziale na zdania należy uznać za błędy tagera. Z jednej strony, błędy takie mogą utrudnić dalsze przetwarzanie, np. analizę składniową zdań. Z drugiej zaś strony, nie każde zastosowanie wymaga podziału na zdania, a jakość podziału stosunowo łatwo ocenić z osobna. Co więcej, jest prawdopodobne, że błędny podział na zdania odbije się negatywnie na automatycznym oznakowaniu morfosyntaktycznym (większość algorytmów zakłada, że każde zdanie rozpatrywane

[18] Sytuacja taka ma miejsce podczas oceny przeprowadzanej w następujących publikacjach: Acedański (2010); Śniatowski i Piasecki (2011); Radziszewski i Śniatowski (2011b). Korpus użyty do oceny (NKJP) przypisuje dokładnie dwie możliwe interpretacje słowom nieznanym: prawidłową interpretację przypisaną przez lingwistów oraz specjalny tag oznaczający słowo nieznane. Strategią zapewniającą stuprocentową trafność w przypadku słów nieznanych jest niewybieranie tagu „słowo nieznane".

[19] Fakt ten został potwierdzony przez Danutę Karwańską (komunikacja e-mailowa, 6.10.2011).

jest z osobna) — przez co ocena na poziomie segmentów po części uwzględnia błędy w podziale na zdania. Głównym problemem jest jednak trudność opisania jedną miarą błędów umiejscowionych na poziomie segmentów i tagów oraz błędów na poziomie całych zdań. Dlatego też zalecamy, by miara oceny tagera nie uwzględniała wprost błędów w podziale na zdania.

Różnice w podziale na segmenty powodują, że nie zawsze możliwe jest bezpośrednie porównanie tagów przypisanych danemu segmentowi w wariancie wzorcowym i wariancie wyprodukowanym przez tager. Dalsze rozważania opieramy o założenie, że korpus wzorcowy i korpus wynikający z oznakowania czystego tekstu zawierają ten sam tekst (pomimo możliwych różnic w podziale na segmenty). Należy się spodziewać, że znaczna część (większość) segmentów z korpusu wzorcowego obecna będzie w korpusie wynikowym w postaci niezmienionej. W przypadku pozostałych segmentów z korpusu wzorcowego powiemy, że *podlegają zmianie segmentacji*.

Poniżej przedstawiamy kilka przykładów takich zmian segmentacji[20]. W przykładach użyto strzałek obustronnych, gdyż teoretycznie możliwe są oba kierunki zamiany (tj. zarówno lewa, jak i prawa strona może teoretycznie przedstawiać segmentację wzorcową).

(2.22) [...] ↔ [.][.][.]

(2.23) [m.in.] ↔ [m][.][in][.]

(2.24) [człowiek–demolka] ↔ [człowiek][–][demolka]

(2.25) [dałżebyś] ↔ [dał][że][byś]

(2.26) [void*][ptr] ↔ [void][*ptr]

(2.27) [Lądek Zdrój] ↔ [Lądek] [Zdrój]

Najprostszy typ różnic segmentacji dotyczy różnego traktowania zbitek znaków interpunkcyjnych (2.22). Problem pojawia się, gdy w jednym z porównywanych wariantów ciąg znaków interpunkcyjnych potraktowany jest jako jeden segment, podczas gdy w drugim wariancie każdy znak stanowi osobny segment. Sytuacja taka typowo dotyczy kropek, a także ciągów łączników pełniących rolę pauzy lub półpauzy (np. `---`), choć teoretycznie może wystąpić w przypadku ciągów dowolnych znaków interpunkcyjnych lub symboli graficznych.

Drugi typ różnic, który często można spotkać, dotyczy ciągów zawierających naprzemian znaki interpunkcyjne i ciągi liter. Jego typowym przykładem jest inne traktowanie skrótów wieloczłonowych (2.23), a także słów połączonych półpauzą lub łącznikiem (2.24).

Część problemów może wynikać z nietradycyjnej strategii segmentacji przyjętej w tagsetach NKJP i KIPI (por. punkt 2.2). W przypadku nierozpoznania konkretnej formy przez moduł analizy morfosyntaktycznej, prawdopodobnie zostanie ona zwrócona jako jeden segment, wbrew temu, co zostało oznakowane w korpusie wzorcowym (2.25).

Dotychczasowe przykłady uwzględniały jedynie sytuacje, gdzie jeden z wariantów zawierał dokładnie jeden segment, a drugi — więcej. Teoretycznie istnieje możliwość

[20] Przytoczone w tym punkcie typy różnic segmentacji zostały w dużej mierze zainspirowane dyskusjami z Szymonem Acedańskim.

pojawienia się bardziej skomplikowanej różnicy, gdzie segmenty będą się na siebie częściowo nakładać (2.26).

Możliwe jest również wystąpienie różnic w segmentacji związanych z obecnością znaków białych (2.27). Warto zauważyć, że sytuacja taka nie może wystąpić w tagsecie NKJP ani KIPI — bezwzględnym nakazem jest tam podział na każdym znaku białym (por. 2.2).

Nie jest jasne, jak należy traktować takie różnice w segmentacji. Intuicja wskazuje na to, że zmiany w segmentacji dotyczące jedynie znaków interpunkcyjnych powinny być karane mniej surowo niż nieprawidłowa segmentacja wyrażeń zawierających np. rzeczowniki. Trudno jest jednak podać jakiekolwiek liczbowe współczynniki kar. Należy przy tym pamiętać, że każdy z wyodrębnionych segmentów ma przypisane pewne tagi, a niezwykle trudno jest sformułować jednoznaczne i uniwersalne kryteria porównania tagów przypisanych segmentom różnie wydzielonym w ramach tego samego wyrażenia.

Problem ten można obejść poprzez określenie przedziału, w jakim bez wątpienia znajduje się trafność osiągana przez tager, niezależnie od tego, które z tych zmian segmentacji uznamy za istotne. Proponowane w tej pracy miary zakładają uprzednie dopasowanie korpusu wzorcowego z korpusem wynikowym (wyjściem tagera) na poziomie segmentów. Dopasowanie takie polega na podziale tekstu na najkrótsze fragmenty, których granice zgodne są z granicami segmentów w obu korpusach. Każdy taki fragment można przedstawić jako parę (*ciąg segmentów z korpusu wzorcowego*, *ciąg segmentów z korpusu wynikowego*), gdzie oba elementy pary dają ten sam tekst. Wszystkie segmenty z korpusu wzorcowego niepodlegające zmianom segmentacji tworzą samodzielne fragmenty. Pozostałe segmenty, tj. te podlegające zmianie segmentacji, tworzą fragmenty, gdzie przynajmniej jeden z elementów pary zawiera więcej niż jeden segment.

Proponujemy użycie dwóch miar, stanowiących widełki, w których mieści się rzeczywista trafność tagera: **dolnego ograniczenia trafności** oraz **górnego ograniczenia trafności**. Obie miary zakładają, że segmenty z korpusu wzorcowego podlegające zmianie segmentacji nie są analizowane (nie sprawdzamy ich tagów). Dolne ogranicznie trafności zakłada, że wszystkie takie segmenty traktowane są jako nietrafione; ograniczenie górne traktuje wszystkie takie segmenty jako trafione, niezależnie od przypisanych im tagów.

Przyjmijmy, że odwzorowanie $match : \mathbb{N} \to \mathbb{N}$ przypisuje numerom segmentów niepodlegającym zmianom segmentacji z korpusu wzorcowego numery segmentów w korpusie wynikowym. Jako że odwzorowanie to jest określone jedynie dla segmentów niepodlegających zmianie segmentacji, zapis $i \in match$ oznacza, że segment i-ty należy do korpusu wzorcowego oraz nie podlega zmianie segmentacji. W takim ujęciu, dolne i górne ograniczenie trafności uzyskuje definicję, odpowiednio, 2.28 i 2.29. Jak w poprzednich wzorach, N określa liczbę segmentów w korpusie wzorcowym.

$$Acc_{lower} = \frac{|\{i : tag(i) = ref(match(i)), i \in match\}|}{N} \tag{2.28}$$

$$Acc_{upper} = \frac{|\{i : tag(i) = ref(match(i)), i \in match\}| + |\{i : 0 < i \leqslant N \wedge i \notin match\}|}{N} \tag{2.29}$$

Dopasowanie korpusu wynikowego do korpusu wzorcowego zilustrowano poniżej (2.30). Korpus wzorcowy (górny wiersz) składa się z sześciu segmentów, natomiast wynikowy (wiersz dolny) — z siedmiu. Spośród sześciu segmentów korpusu wzorcowego jedynie segmenty Dawno, w oraz żyli nie podlegają zmianom segmentacji. Pozostałe segmenty tworzą następujące fragmenty: (PRL-u, PRL - u) oraz (. ., ..).

(2.30)

Dawno	*w*	*PRL-u*		*żyli*	. .
Dawno	w	PRL	- u	żyli	..

Obliczenie dolnego ograniczenia trafności polegałoby zatem na ustaleniu równości tagów przypisanym tym trzem segmentom. Jeśli założymy, że wszystkim tym segmentom tager przypisał tagi prawidłowe, wartość dolnego ograniczenia trafności wyniesie $\frac{3}{6} = 50\%$. Wartość górnego ograniczenia trafności wyniesie natomiast $\frac{3+3}{6} = 100\%$ (przyjmujemy, że tager przypisał prawidłowe tagi tym trzem segmentom, natomiast pozostałe trzy segmenty podlegają zmianie segmentacji — a więc, w przypadku górnego ograniczenia, uznajemy je arbitralnie za trafione).

Która zatem wartość powinna być użyta jako ostateczna miara oceny tagerów? Zalecamy, aby zawsze, gdy przeprowadzana jest eksperymentalna ocena tagerów, publikować wartości obu miar. Wartości te pozwalają ocenić skalę problemu, jakim są zmiany segmentacji. Jako miarę o charakterze decyzyjnym (determinującą, który tager uznamy za lepszy) rekomendujemy[21] **dolne ograniczenie trafności**. Miara ta karze surowo każdą zmianę w segmentacji. Chociaż kara taka może być czasem nieadekwatna do sytuacji, jest to jedyny sposób, aby promować wysiłek włożony w dostosowywanie modułu segmentacji tagera do towarzyszących tagsetowi wytycznych znakowania. Ocena przy użyciu *górnego ograniczenia trafności* w skrajnym wypadku może zachęcać do sztucznego wprowadzenia zmian segmentacji, gdyż zawyżają one osiągnięte przez tager wyniki.

2.6. Generowanie cech w formalizmie WCCL

Formalizm WCCL (*Wrocław Corpus Constraint Language*) został opracowany jako język zapisu cech morfosyntaktycznych na potrzeby różnych zadań przetwarzania języka polskiego (Radziszewski i inni, 2011c), w szczególności znakowania morfosyntaktycznego w oparciu o maszynowe uczenie. Formalizm pozwala na zapis zarówno bardzo prostych, jak i złożonych cech. W niniejszym punkcie omówimy w skrócie wybrane elementy języka WCCL, do których odwoływać się będziemy w następnych punktach. Szerszy opis samego formalizmu, jak i jego implementacji można znaleźć w artykule Radziszewski i inni (2011c), natomiast pełna specyfikacja formalizmu dostępna jest on-line (Radziszewski i inni, 2011b).

Podstawowym wyrażeniem języka WCCL jest **cecha**, czyli wyrażenie funkcyjne przekształcające kontekst występowania wskazanego segmentu w wartości symboliczne. Przyjęto założenie, że kontekst zawsze przycinany jest do granic zdania, do którego należy wskazany segment. Zdanie rozumiane jest jako ciąg segmentów, którym przypisano zbiory interpretacji morfosyntaktycznych. Dzięki temu możliwe jest wartościowanie

[21] Zalecenie to, wraz z jego motywacją, pochodzi bezpośrednio od Szymona Acedańskiego.

cech zarówno na zdaniach w pełni ujednoznacznionych, jak i na zdaniach zawierających segmenty niejednoznaczne (tj. takie, którym przypisano więcej niż jedną interpretację morfosyntaktyczną). Takie ujęcie cech morfosyntaktycznych zbieżne jest z założeniami modelu *znakowania poprzez wykreślanie* opisanego na s. 26.

Język WCCL jest silnie typizowany. Każde wyrażenie funkcyjne oraz każda stała języka ma jednoznacznie określony zwracany typ (typ każdego wyrażenia da się wywieść na podstawie składni). Zdefiniowano 4 podstawowe typy danych:

1. Typ *zbiór symboli z tagsetu* (typ T). Jego definicja zależna jest od przyjętego w danym momencie tagsetu. Wartościami tego typu są wszystkie możliwe zbiory składające się z klas gramatycznych i wartości atrybutów zdefiniowanych w tagsecie. Jeśli C_t to zbiór klas gramatycznych zdefiniowanych w tagsecie t, V_t to zbiór wszystkich wartości atrybutów zdefiniowanych w tagsecie t, to $T_t = 2^{(C_t \cup V_t)}$. Przyjęto założenie, że dla każdego zestawu cech opisanych w formalizmie WCCL tagset t jest ustalony, tj. nie przewidziano możliwości mieszania wyrażeń opierających się o różne tagsety. Jeśli przyjmiemy tagset NKJP, to następujące symbole są prawidłowymi stałymi typu T: `{subst,adj}`, `{nom}`, `{}`.
2. Typ *zbiór napisów* (S). Podobnie jak w przypadku powyższego typu danych, wartościami prostymi typu S są zbiory, a nie pojedyncze wartości (w tym przypadku napisy). Takie rozwiązanie wynika z założenia, że przetwarzane zdania mogą zawierać niejednoznaczne segmenty. Próba pobrania wartości danej kategorii gramatycznej może więc powodować konieczność zwrócenia kilku elementów; podobna sytuacja ma miejsce w przypadku pobrania lematu segmentu, który również może być wieloznaczny. Przykładowe wartości typu S: `["patrzeć","patrzyć"]`, `[]` (nawiasy kwadratowe wprowadzono dla odróżnienia składniowego pustego zbioru symboli od pustego zbioru napisów).
3. *Wartości logiczne* (B): `True` i `False`. Funkcje zwracające wartości tego typu to predykaty.
4. *Pozycje* (typ P): liczby całkowite wskazujące segment względem segmentu wskazanego jako centralny (np. -1, 0; por. punkt 2.4.3), bądź specjalne symbole określające pierwszy segment w zdaniu (`begin`) lub segment ostatni (`end`).

Do prostych wyrażeń funkcyjnych należy m.in:

1. pobranie formy wyrazowej segmentu znajdującego się na podanej pozycji (forma zwracana jest jako zbiór jednoelementowy lub zbiór pusty, jeśli podana pozycja wskazuje poza granice zdania), np. `orth[0]` zwraca formę wyrazową segmentu centralnego;
2. pobranie lematu segmentu (zbiór jednoelementowy w przypadku jednoznacznego lematu, wieloelementowy w przypadku wieloznaczności albo pusty w przypadku przekroczenia granicy zdania); np. `base[-1]` zwraca zbiór możliwych lematów przypisanych segmentowi poprzedzającemu segment centralny;
3. pobranie wartości klasy gramatycznej (zwracany jest zbiór symboli z tagsetu, zbiór może być pusty, jedno- lub wieloelementowy); np. `class[0]`;
4. pobranie wartości danego atrybutu (j.w.) — formalizm automatycznie traktuje symbole atrybutów zdefiniowane w tagsecie jako nazwy funkcji zwracających ich wartości, np. nazwa `cas` używana jest jako funkcja zwracająca wartość przypadka segmentu na podanej pozycji; przykładowo, wyrażenie `cas[0]` pobiera zbiór możliwych wartości przypadka przypisanych do segmentu centralnego.

Formalizm pozwala także na odwołanie się do zewnętrznego słownika (listy form wyrazowych). Odwołanie takie realizowane jest za pomocą wyrażenia `lex`, które pozwala na przefiltrowanie danego zbioru napisów, tj. pozostawienie ze zbioru jedynie tych elementów, które znajdują się w zewnętrznym słowniku.

Zdefiniowano również szereg bardziej skomplikowanych wyrażeń funkcyjnych. Szczególnie istotnymi z punktu widzenia tego rozdziału są predykaty sprawdzające *uzgodnienie gramatyczne*. Uzgodnienie gramatyczne rozumiane tutaj jest w sposób techniczny[22]: oznacza, że dane segmenty mają przypisane te same wartości określonym z góry atrybutom. Najprostszym predykatem sprawdzającym uzgodnienie jest predykat `agrpp(P1,P2,T1)`. Predykat ten spełniony jest wtedy i tylko wtedy, gdy segmentom znajdującym się na podanych pozycjach (`P1` i `P2`) przypisano te same wartości atrybutów, których wartości znajdują się w zbiorze `T1`. Przykładowo, wyrażenie `agrpp(-1,1,{nmb,gnd,cas})` spełnione będzie, jeśli segmenty znajdujące się na pozycji -1 i $+1$ mają określoną wartość liczby, rodzaju i przypadka (`{nmb,gnd,cas}` w tagsecie NKJP) oraz zachodzi między nimi uzgodnienie, tj. segmenty te mają identyczną wartość podanych atrybutów[23].

Naturalnym rozszerzeniem predykatu `agrpp` na przedział pozycji jest predykat `agr(P1,P2,T1)`. Predykat ten wymaga, aby wszystkie segmenty należące do przedziału (`P1`, `P2`) były uzgodnione, tj. wszystkie miały określoną wartość atrybutów należących do zbioru `T1`, a wartości tych atrybutów dla wszystkich segmentów były identyczne.

Pewną modyfikacją predykatu jest predykat `wagr(P1,P2,T1)`, sprawdzający *słabe uzgodnienie* przedziału segmentów (`P1`, `P2`). Zwykłe (silne) uzgodnienie na przedziale wymaga, aby wszystkie segmenty należące do tego przedziału miały określone wartości wszystkich podanych atrybutów. W przypadku słabego uzgodnienia warunek ten złagodzono: silne uzgodnienie musi zachodzić jedynie pomiędzy segmentami stanowiącymi krańce przedziału (tj. znajdującymi się na pozycjach `P1` i `P2`), natomiast od pozostałych segmentów wymaga się, by tagi do nich przypisane miały nieokreśloną wartość danego atrybutu **lub** wartość ta była określona i zgodna z wartością przypisaną segmentom leżącym na krańcach przedziału (dotyczy to wszystkich podanych atrybutów).

Przykłady zastosowania testów na uzgodnienie w budowie tagera TaKIPI oraz dalsze rozważania na ich temat można znaleźć w pracy (Piasecki i Radziszewski, 2009).

Implementację formalizmu omawiamy w dodatku A.

2.7. Algorytm: ujednoznacznianie morfosyntaktyczne w oparciu o uczenie na pamięć

W tym punkcie przedstawiamy nowy algorytm ujednoznaczniania morfosyntaktycznego. Chociaż algorytm opiera się o znane z literatury metody, nowością jest połączenie

[22] Z lingwistycznego punktu widzenia część rozpatrywanych przez nas „uzgodnień" to w rzeczywistości związki rządu.

[23] W formalizmie WCCL można użyć symbolu atrybutu w ramach stałej typu T; symbol atrybutu rozbijany jest wtedy na zbiór jego możliwych wartości, zgodnie z definicją tagsetu. Tak więc zapis `{nmb,gnd,cas}` jest skrótem zapisu `{sg,pl,f,m1,m2,m3,n,nom,gen,dat,acc,loc,inst,voc}`. Predykatowi sprawdzającemu uzgodnienie można również przekazać pojedyncze wartości pewnych atrybutów — w takiej sytuacji wymagane jest, aby atrybut reprezentowany przez podane wartości miał dokładnie te wartości, które podano. Z możliwości tej nie będziemy jednak korzystać w tej pracy.

kilku technik, szczególnie obiecujących z punktu widzenia przetwarzania języków słowiańskich. Techniki te wymieniamy poniżej:

1. uczenie pamięciowe ze względu na zdolność do wnioskowania na podstawie słabych przesłanek i bogatego zbioru cech (por. s. 28),
2. znakowanie warstwowe ze względu na duży rozmiar i pozycyjny charakter tagsetów typowych dla języków słowiańskich,
3. znakowanie jako analiza morfosyntaktyczna i ujednoznacznianie (działanie dwuetapowe) ze względu na mnogość form w językach fleksyjnych.

Problem wnioskowania na podstawie słabych przesłanek wydaje się szczególnie istotny w przypadku dużych tagsetów: należy się liczyć z obecnością nie tylko rzadkich form wyrazowych, ale także klas niejednoznaczności reprezentowanych przez niewielką liczbę przykładów uczących. Metoda znakowania warstwowego oraz podejścia dwuetapowego sprawdziły się niejednokrotnie w znakowaniu języków słowiańskich (por. rozdz. 2.4). Stosowane wcześniej metody cechował jednak duży stopień komplikacji. Zasadniczą zaletą proponowanego przez nas algorytmu jest duża prostota. Co więcej, jest on praktycznie niezależny od języka: jedyną wymaganą informacją związaną z językiem jest definicja tagsetu oraz definicja zbioru cech. Poza tym algorytm nie wymaga żadnych reguł pisanych ręcznie ani nawet definicji klas niejednoznaczności. Algorytm wprowadza również modyfikację stosowanych wcześniej modeli znakowania warstwowego (Tufiş, 1999; Piasecki i Godlewski, 2006b; Acedański, 2010): zamiast wymagać definicji atrybutów tagsetu stanowiących osobne warstwy, klasa gramatyczna, a także każdy atrybut zdefiniowany w tagsecie traktowany jest jako samodzielna warstwa. Innymi słowy, stosowany jest model warstwowy, gdzie wprowadzamy $A + 1$ warstw dla tagsetu zawierającego A atrybutów.

W algorytmie świadomie zrezygnowaliśmy z jawnego podziału danych na klasy niejednoznaczności. Dzięki temu proponowany model jest bardzo prosty. Klasy niejednoznaczności pojawiają się niejako nie wprost: stosowany przez nas zbiór cech zawiera cechy odpowiadające możliwym wartościom klasy gramatycznej oraz wartościom wszystkich atrybutów w tagsecie — co efektywnie stanowi klasy niejednoznaczności zdefiniowane z osobna dla każdej warstwy.

Opisywany algorytm oraz jego implementacja wykonane zostały na potrzeby projektu NEKST[24]. Powstały w ten sposób tager nazwany został *Wrocław Memory-Based Tagger*, *WMBT*[25]. Od tej pory w ten sposób będziemy nazywać zarówno zaproponowany w tym punkcie algorytm, jak i jego implementację w postaci działającego tagera.

Algorytm WMBT zakłada, że korpus uczący zawiera przyporządkowanie każdemu segmentowi zbioru możliwych interpretacji morfosyntaktycznych (wynik analizy morfosyntaktycznej), a jedna spośród tych interpretacji oznaczona jest jako właściwa (wzorcowa). Znakowanie za pomocą wyuczonego modelu wymaga przeprowadzenia segmentacji i wykonania analizy morfosyntaktycznej — poniższy opis algorytmu działania tagera (punkt 2.7.2) zakłada, że proces ten został już wykonany, a więc segmentom z korpusu wejściowego przypisano już wskutek analizy morfosyntaktycznej zbiory możli-

[24] Projekt finansowany z funduszy europejskich w ramach Programu Operacyjnego Innowacyjna Gospodarka, numer umowy POIG.01.01.02-14-013/09. Bezpośrednim beneficjentem grantu jest Instytut Podstaw Informatyki Polskiej Akademii Nauk, zaś Politechnika Wrocławska odgrywa rolę partnera, odpowiedzialnego za opracowanie technologii językowych związanych m.in. z przetwarzaniem morfosyntaktycznym.

[25] Opracowane oprogramowanie opisujemy w dodatku A.

wych interpretacji morfosyntaktycznych. Opisywana w tym punkcie podstawowa wersja algorytmu WMBT zakłada, że tager zawsze wybiera jedną z dostępnych interpretacji, tj. nie może dodać interpretacji, której pośród przypisanego zbioru nie było.

Algorytm parametryzowany jest zbiorem **cech**. Cecha rozumiana jest jako funkcja przekształcająca kontekst danego segmentu (niewykraczający poza granice zdania) w wartości symboliczne (por. s. 26). Przykładowo, prostymi cechami mogą być funkcje zwracające formę wyrazową segmentu analizowanego, formę wyrazową segmentu poprzedzającego segment analizowany itp. Podobnie można zdefiniować cechy pobierające wartości klasy gramatycznej czy wartości podanych atrybutów. Warto tutaj podkreślić, że tego rodzaju cechy mogą również zwracać wieloelementowe zbiory symboli, jeśli na danym etapie ujednoznaczniania segmentowi przypisanych jest kilka możliwych tagów. W opisywanym tutaj algorytmie wartości zwrócone przez cechy traktujemy jak symbole niepodzielne, tj. pytamy jedynie o równość dwóch wartości cech, nie rozpatrujemy zaś relacji typowych dla zbiorów, takich jak podzbiór czy niepuste przecięcie. Dzięki temu możliwe jest wykorzystanie cech będących funkcjami o różnych przeciwdziedzinach bez konieczności uprzedniego definiowania tych przeciwdziedzin. Rozpatrywać będziemy też cechy będące bardziej złożonymi funkcjami, np. predykaty sprawdzające, czy segmenty sąsiadujące z segmentem analizowanym mają tę samą wartość przypadka, liczby i rodzaju. Proponowany przez nas zestaw cech omawiamy w punkcie 2.7.3.

Parametrami algorytmu są również parametry samego klasyfikatora pamięciowego, takie jak liczba sąsiadów i funkcja podobieństwa, oraz pomocnicza wartość F określająca długość listy najczęstszych form wyrazowych pozyskiwanej z korpusu uczącego (lista służy jako dodatkowa informacja, do której mogą odwoływać się cechy).

2.7.1. Uczenie

Uczenie odbywa się warstwowo, tj. najpierw generowana jest baza przypadków uczących dla klasy gramatycznej, następnie dokonywane jest ujednoznacznienie klasy gramatycznej (wykreślenie tych tagów, które mają inną jej wartość niż tag uznany za wzorcowy), po czym algorytm przechodzi do generowania bazy przypadków uczących dla pierwszego atrybutu tagsetu — itd.

Wynikiem uczenia jest lista najczęstszych form wyrazowych zawierająca F pozycji oraz A+1 baz przypadków uczących, tj. jedna dla klasy gramatycznej i pozostałe dla kolejnych atrybutów tagsetu (innymi słowy, baza przypadków generowana jest dla każdej warstwy). Procedurę uczenia opisujemy poniżej jako algorytm 1. Algorytm rozpoczyna się od zebrania listy F najczęstszych form wyrazowych z korpusu uczącego. Główna pętla algorytmu polega na generowaniu przypadków uczących dla danego atrybutu i danego segmentu. Każdy przypadek uczący to para (*wektor cech*, *decyzja*), gdzie wektor cech to ciąg wartości otrzymanych wskutek aplikacji kolejnych cech do otoczenia danego segmentu, a decyzja to wartość danego atrybutu pobrana z wzorcowego tagu przypisanego segmentowi.

Użyte w opisie algorytmu 1 sformułowanie *segment jest niejednoznaczny ze względu na dany atrybut* oznacza, że dany segment ma na obecnym etapie przetwarzania przypisane tagi, które zawierają łącznie co najmniej dwie różne wartości danego atrybutu. Przykładowo, jeśli na etapie przetwarzania odpowiadającemu atrybutowi przypadka gramatycznego segmentowi przypisane będą trzy tagi rzeczownikowe, lecz wszystkie z nich będą miały określoną tę samą wartość przypadka (np. biernik), powiemy, że seg-

Algorytm 1 Uczenie algorytmu WMBT

Dane: korpus uczący *korp* oznakowany morfosyntaktycznie
zbiór cech przypisany każdemu z atrybutów $cechy(a)$
Wyniki: bazy przypadków uczących B_a dla $a \in [klasa, atr_1, \ldots, atr_k]$
zbierz F najczęstszych form z korpusu uczącego *korp*
for $zdanie \in korp$ **do**
for $a \in [klasa, atr_1, \ldots, atr_k]$ **do**
for $seg \in zdanie$ **do**
if *seg* jest niejednoznaczny ze względu na a **then**
$wek\_cech \leftarrow [f(seg, zdanie)$ **for** $f \in cechy(a)]$
$decyzja \leftarrow$ prawidłowa wartość atrybutu a dla segmentu *seg*
dodaj do bazy a przykład uczący $(wek\_cech, decyzja)$
usuń z segmentu *seg* tagi z nieprawidłową wartością a
end if
end for
end for
end for

ment taki **nie** jest niejednoznaczny ze względu na przypadek gramatyczny — otrzymujemy bowiem jednoelementowy zbiór możliwych wartości przypadka. Co więcej, gdyby segmentowi przypisane były np. jedynie interpretacje przysłówkowe, które z natury nie mają określonej wartości przypadka, otrzymalibyśmy pusty zbiór możliwych wartości przypadka, a zatem również nie mielibyśmy do czynienia z niejednoznacznością w tej warstwie.

2.7.2. Znakowanie

Znakowanie za pomocą wyuczonego modelu zostało przedstawione jako algorytm 2. Algorytm zakłada, że na podstawie każdej bazy przypadków uczących tworzony jest osobny klasyfikator pamięciowy. Procedura **klasyfikuj** sprowadza się do użycia klasyfikatora związanego z warstwą a do klasyfikacji podanego wektora cech. Algorytm zakłada, że decyzja zwrócona przez klasyfikator porównywana jest ze zbiorem możliwych wartości danego atrybutu (wartości te pobierane są z tagów, które pozostały przypisane do ujednoznacznianego w danej chwili segmentu). Jeśli otrzymana decyzja należy do tego zbioru, to przeprowadzamy częściowe ujednoznacznienie polegające na usunięciu tagów, dla których wartość tego atrybutu jest inna niż otrzymana decyzja. W przeciwnym wypadku (w praktyce sytuacja ta rzadko występuje) nie podejmowana jest na tym etapie żadna decyzja, co powoduje tymczasowe pozostawienie wieloznaczności związanej z danym atrybutem. Warto zauważyć, że ta sama wieloznaczność może zostać rozwiązana później przy okazji ujednoznaczniania innego atrybutu (często wieloznaczności związane z różnymi atrybutami są od siebie zależne). Jeśli pomimo przejścia przez wszystkie warstwy wieloznaczność pozostanie, na samym końcu podejmowana jest arbitralna decyzja celem wymuszenia jednego tagu na segment.

Algorytm 2 Ujednoznacznianie pojedynczego zdania przez WMBT

Dane: *zdanie* poddane analizie morfosyntaktycznej
zbiór cech przypisany każdemu z atrybutów $cechy(a)$
wyuczony klasyfikator K_a dla każdego z atrybutów
Wyniki: ujednoznacznione *zdanie*
for $a \in [klasa, atr_1, \ldots, atr_k]$ **do**
for $seg \in zdanie$ **do**
if *seg* jest niejednoznaczny ze względu na a **then**
$wek\_cech \leftarrow [f(seg, zdanie)$ **for** $f \in cechy(a)]$
$decyzja \leftarrow$ klasyfikuj (K_a, wek_cech)
if *decyzja* $\in$ możliwe wartości a pobrane z tagów przypisanych segmentowi *seg* **then**
usuń z segmentu *seg* tagi, dla których wartość$(a) \neq$ *decyzja*
end if
end if
end for
end for
for $seg \in zdanie$ **do**
wybierz arbitralnie „pierwszy wg tagsetu" tag jeśli pozostało ich kilka
end for

Opisany algorytm zakłada ujednoznacznianie tagów przypisanych do segmentów. Algorytm działa na poziomie tagów a nie całych interpretacji, a zatem możliwe jest pozostawienie na wyjściu kilku interpretacji o tym samym tagu, lecz różnych lematach.

2.7.3. Parametry i cechy

Zaproponowany algorytm parametryzowalny jest zarówno parametrami samego klasyfikatora pamięciowego, jak i zestawem cech. Dobór parametrów i cech został przeprowadzony w oparciu o wstępne eksperymenty przeprowadzone na ręcznie oznakowanej części korpusu KIPI. Proponowane przez nas parametry klasyfikatora pamięciowego są następujące:

— liczba najbliższych sąsiadów $k = 11$,
— miara podobieństwa Scotta–Salzberga — *Modified Value Difference*,
— schemat głosowania Dudaniego — *Inverse Linear*.

Podczas testowania tagera używamy implementacji klasyfikatora pamięciowego z popularnego pakietu TiMBL (Daelemans i inni, 2010a). Powyższe parametry można włączyć w tymże pakiecie za pomocą opcji `-mM -k11 -dIL`.

Proponowany przez nas zestaw cech jest w dużej mierze inspirowany cechami użytymi w ramach tagera języka polskiego TaKIPI (Piasecki, 2007), a po części też standardowym zestawem cech tagera pamięciowego MBT (Daelemans i inni, 2010b). Zestaw zawiera następujące grupy cech:

1. Możliwe wartości klasy gramatycznej dla każdego segmentu z okna $(-3, -2, -1, 0, +1, +2)$ oraz wartości trzech atrybutów: przypadka, rodzaju

i liczby dla segmentów z tego samego okna. Dokładnie taki sam zestaw cech używany był dla każdej klasy niejednoznaczności w TaKIPI (oprócz tego niektóre z klas używały kilku dodatkowych cech). Ze względu na możliwą niejednoznaczność, wartości tych cech są z natury zbiorami. Jeśli dany atrybut (lub klasa gramatyczna) będzie na danym etapie przetwarzania już ujednoznaczniony, wartością będzie zbiór jednoelementowy.

2. Cechy leksykalne: formy wyrazowe wszystkich segmentów z okna $(-3, \ldots, 2)$ sprowadzone do małych liter. Formy wyrazowe filtrowane są do F najczęstszych, pobranych z listy tworzonej podczas uczenia; formy niewystępujące na liście zamieniane są na symbol reprezentujący formę rzadką (podobna praktyka stosowana jest w tagerze MBT, por. Daelemans i inni, 2010b). Przyjmujemy wartość $F = 500$.
3. Cechy o wartościach prawda/fałsz sprawdzające uzgodnienie wartości liczby, rodzaju i przypadka. Uzgodnienie takie charakteryzuje proste frazy rzeczownikowe w języku polskim (por. rozdział 4.2). Testy te mają za zadanie ułatwić ujednoznacznianie uzgodnionych ciągów segmentów. Użyto następujących cech:
   a) test na uzgodnienie między pozycjami -1 i 0: `agrpp(-1,0,{nmb,gnd,cas})`,
   b) test na uzgodnienie między pozycjami 0 i $+1$,
   c) test na *słabe uzgodnienie przedziałowe* (`wagr`; por. punkt 2.6) między segmentami z okna $(-2, -1, 0)$,
   d) j.w., okno $(-1, 0, +1)$,
   e) j.w., okno $(0, +1, +2)$.
4. Sufiksy formy wyrazowej na pozycji 0. Pobierane są trzy sufiksy: trzyliterowy, dwuliterowy oraz jednoliterowy.
5. Dwa testy na graficzną postać formy wyrazowej na pozycji 0: czy zaczyna się małą literą oraz czy zaczyna się wielką literą.

Wszystkie te cechy zostały zapisane w postaci wyrażeń funkcyjnych w formalizmie WCCL. Wyrażenia te zamieszczono w punkcie C.1 dodatku C.

2.7.4. WMBT a MBT

Proponowany algorytm jest w dużej mierze inspirowany algorytmem znakowania morfosyntaktycznego zaimplementowanym w tagerze MBT. W szczególności wykazuje następujące podobieństwa:

1. Używany jest ten sam klasyfikator pamięciowy (podczas oceny eksperymentalnej używamy też tej samej implementacji klasyfikatora pamięciowego — pakietu TiMBL).
2. Klasy niejednoznaczności obecne są w modelu nie wprost, tj. dostarczane są jako cechy.
3. Stosowane są cechy leksykalne (formy wyrazowe odpowiadające segmentom z najbliższego otoczenia) z filtrowaniem form rzadkich.

W celu dostosowania algorytmu do znakowania języków słowiańskich wprowadzono **kilka istotnych modyfikacji**:

1. Tager MBT nie używa zewnętrznego analizatora morfosyntaktycznego, wszystkie informacje leksykalne pozyskiwane są z danych uczących. Użycie analizatora w tagerze WMBT pozwala na uzyskanie opisu morfosyntaktycznego części form, które nie wystąpiły w korpusie uczącym.
2. Tager WMBT wprowadza do znakowania model warstwowy.

3. Cechy stosowane w MBT zakładały użycie cech typu zbiorowego jedynie dla klas niejednoznaczności (dotyczy segmentów na pozycjach 0, 1, 2). W tagerze WMBT używamy znacznie więcej takich cech, dzięki czemu możliwy jest prosty opis kontekstu wystąpienia danego segmentu, niezależnie od obecnego etapu ujednoznaczniania (bieżącej warstwy).
4. Wersja algorytmu WMBT omawiana w tym punkcie nie wprowadza podziału na słowa znane i nieznane. Upraszcza to model: całe znakowanie sprowadza się do ujednoznaczniania interpretacji przypisanych segmentow w wyniku analizy morfosyntaktycznej.

Tager MBT pozwala zdefiniować proste zestawy cech, jeden dla słów znanych, drugi dla słów nieznanych. Domyślny zestaw cech tagera MBT dla słów znanych wygląda następująco:

1. formy wyrazowe przypisane segmentom z okna $(-3, -2, \ldots, +2)$, gdzie formy rzadkie zastępowane są symbolem specjalnym;
2. tagi dotychczas przypisane segmentom z lewego kontekstu, tj. segmentom na pozycjach $(-3, -2, -1)$,
3. zbiory tagów (klasy niejednoznaczności) przypisane segmentowi centralnemu (0) oraz prawemu kontekstowi $(+1, +2)$.

Dla słów nieznanych standardowy zestaw cech zawiera wszystkie powyższe cechy oraz następujące pozycje:

1. pierwszy znak formy wyrazowej na pozycji centralnej,
2. dwa ostatnie znaki formy wyrazowej na pozycji centralnej,
3. predykat sprawdzający, czy forma wyrazowa na pozycji centralnej zawiera wielkie litery,
4. j.w., czy zawiera łącznik,
5. j.w., czy zawiera cyfrę.

Cechy zdefiniowane w powyższy sposób odtąd będziemy nazywać **zestawem cech 0**.

MBT pozwala na drobne modyfikacje tego zestawu cech: część cech można wyłączyć, poza tym istnieje możliwość zmiany długości lewego i prawego kontekstu.

Istnieje też możliwość działania w oparciu o własny zestaw cech[26] — wtedy cechy użytkownika dodawane są do wspomnianego zestawu, a ściślej rzecz biorąc — dodawane zarówno do zestawu zdefiniowanego dla słów znanych, jak i nieznanych. Rozpatrywać będziemy dwa zestawy zawierające cechy dodatkowe.

Zestaw cech 1 odpowiada zestawowi cech 0 wzbogaconego o możliwe wartości klasy gramatycznej dla każdego segmentu z okna $(-3, -2, -1, 0, +1, +2)$ oraz wartości trzech atrybutów: przypadka, rodzaju i liczby dla segmentów z tego samego okna. Zestaw ten jest pierwszym krokiem w kierunku przybliżenia cech używanych przez tager MBT do zestawu zaproponowanego na potrzeby tagera WMBT.

Zestaw cech 2 dodaje do zestawu cech 1 następujące pozycje:

1. testy na uzgodnienie zdefiniowane w punkcie 3 na stronie 49,

[26] Z technicznego punktu widzenia funkcjonalność ta nie jest wspierana przez sam tager: wymaga to samodzielnego przygotowania pliku uczącego i pliku do oznakowania rozszerzonego o wartości cech wyliczone dla każdego segmentu. Gdy będzie mowa o testowaniu tagera MBT przy użyciu dodatkowych cech, dodatkowe cechy dostarczamy właśnie w ten sposób, korzystając z formalizmu WCCL oraz narzędzia `wccl-run` dostępnego w ramach implementacji formalizmu.

2. pobranie sufiksów formy wyrazowej, zdefiniowane w punkcie 4 tej samej strony oraz
3. testy na postać graficzną formy zdefiniowane w punkcie 5.

Uzyskany w ten sposób zestaw cech 2 jest prawie identyczny z zestawem cech zaproponowanym na potrzeby algorytmu WMBT (punkt 2.7.3).

2.8. Modyfikacja algorytmu: rozpoznawanie słów nieznanych

Powyższa propozycja algorytmu WMBT zakłada, że wszystkie segmenty traktowane są jednakowo (brak klas decyzyjnych), a zadaniem algorytmu jest wybór spośród możliwych tagów. W tym punkcie proponujemy modyfikację algorytmu, która opiera się na wprowadzeniu podziału segmentów na dwie klasy: *słowa znane* i *słowa nieznane*, podobnie jak w przypadku tagera MBT. Co więcej, zmodyfikowany algorytm pozwala na przypisanie segmentowi zupełnie nowego tagu, nieobecnego w zbiorze przypisanym podczas analizy morfosyntaktycznej. Od tej pory opisywany w tym punkcie algorytm nazywać będziemy **algorytmem WMBT z modułem odgadującym**.

Ogólny zarys algorytmu jest następujący. Segmenty rozpoznane jako słowa znane traktowane są w dotychczasowy sposób, tj. ich znakowanie dokonywane jest poprzez wybór właściwej interpretacji spośród interpretacji przypisanych przez analizator morfosyntaktyczny (jak opisano w punkcie 2.7.2). Segmentom pozostałym, tj. słowom nieznanym, przypisywane są najpierw tagi typowe dla form rzadkich, po czym w sposób analogiczny wybierane są spośród tych zbiorów pojedyncze tagi. Proponowany tutaj algorytm łączy zatem technikę znakowania poprzez wykreślanie z heurystyką odgadywania słów nieznanych. Modyfikacje w stosunku do podstawowego algorytmu WMBT wprowadzono w sposób konserwatywny — w przypadku słów nieznanych główne zadanie również sprowadza się do wykreślania tagów — a efekt odgadywania osiągnięto dzięki uprzedniemu rozszerzeniu zbiorów tagów przypisanych słowom nieznanym o tagi typowe dla takich słów. Modyfikacja zakłada również podwojenie liczby baz przypadków uczących: otrzymujemy $A + 1$ baz przypadków uczących dla słów znanych oraz $A + 1$ baz przypadków uczących dla słów nieznanych (A to liczba atrybutów zdefiniowanych w tagsecie; zwiększamy ją o jeden, gdyż pierwsza baza tworzona jest dla klasy gramatycznej).

Rozróżnienie na słowa znane i nieznane dokonywane jest z osobna podczas uczenia i podczas działania tagera. Podczas znakowania za słowa nieznane uznawane są segmenty nierozpoznane przez analizator morfosyntaktyczny, tj. te, którym przypisano tag „słowo nieznane” (w tagsetach NKJP i KIPI jest to tag `ign`). Odnalezienie słów nieznanych w korpusie uczącym jest zadaniem mniej oczywistym. My zakładamy, że segmenty takie zostały uprzednio oznakowane poprzez obecność tagu „słowo nieznane” pośród zbioru możliwych tagów przypisanych segmentowi. Przypomnijmy, że sytuacja taka ma miejsce w korpusie NKJP: formom nierozpoznanym przez analizator morfosyntaktyczny użyty jako narzędzie wspomagające wzorcowe znakowanie przypisano zbiór interpretacji składający się z dwóch elementów: prawidłowej interpretacji dodanej ręcznie przez lingwistę oraz sztucznej interpretacji o tagu „słowo nieznane” (por. punkt 2.2). Wymóg obecności tej informacji w korpusie wzorcowym może wydawać się istotnym ograniczeniem proponowanego tutaj algorytmu. Na szczęście istnieje prosta

metoda pozwalająca na przypisanie tej informacji automatycznie w oparciu o analizator morfosyntaktyczny; opiszemy ją w następnym punkcie.

Zmodyfikowaną wersję algorytmu uczenia przedstawiamy jako algorytm 3. Lista frekwencyjna to zbiór par $(tag, c(tag))$, gdzie $c(tag)$ to liczba wystąpień w korpusie uczącym słów nieznanych oznakowanych tagiem *tag*. Algorytm przewiduje możliwość filtrowania tagów rzadkich, choć my tej możliwości nie testowaliśmy (przyjmujemy, że $U = 1$). W toku działania algorytmu przypadki uczące trafiają do baz; zapis B_a^K oznacza bazę związaną z atrybutem a oraz słowami znanymi; zapis B_a^U oznacza bazę związaną z atrybutem a i słowami nieznanymi.

Algorytm 3 Uczenie algorytmu WMBT z modułem odgadującym

Dane: korpus uczący *korp* oznakowany morfosyntaktycznie
zbiór cech przypisany każdemu z atrybutów $cechy(a)$
Wyniki: bazy przypadków uczących B_a^K i B_a^U dla $a \in [klasa, atr_1, \ldots, atr_k]$
zbierz F najczęstszych form z korpusu uczącego *korp*
zbierz listę frekwencyjną tagów przypisanych słowom nieznanym z *korp*
usuń z listy tagi pojawiające się rzadziej niż U razy
for *zdanie* $\in$ *korp* **do**
  for *seg* $\in$ *zdanie* **do**
    if *seg* to słowo nieznane **then**
      dodaj tagi z listy frekwencyjnej do zbioru tagów segmentu *seg*
    end if
  end for
  for $a \in [klasa, atr_1, \ldots, atr_k]$ **do**
    for *seg* $\in$ *zdanie* **do**
      if *seg* jest niejednoznaczny ze względu na a **then**
        $wek\_cech \leftarrow [f(seg, zdanie)$ **for** $f \in cechy(a)]$
        decyzja $\leftarrow$ prawidłowa wartość atrybutu a dla segmentu *seg*
        if *seg* to słowo znane **then**
          dodaj do bazy B_a^K przykład uczący $(wek\_cech, decyzja)$
        else
          dodaj do bazy B_a^U przykład uczący $(wek\_cech, decyzja)$
        end if
        usuń z segmentu *seg* tagi z nieprawidłową wartością a
      end if
    end for
  end for
end for

Zapis „dodaj tagi z listy frekwencyjnej do zbioru tagów segmentu *seg*" oznacza, że do zbioru interpretacji morfosyntaktycznych przypisanych danemu segmentowi dodawanych jest tyle sztucznych interpretacji, ile pozycji znajduje się na liście frekwencyjnej tagów słów nieznanych. Każda taka sztuczna interpretacja składa się z tagu *tag* pochodzącego ze wspomnianej listy oraz lematowi o wartości równej formie wyrazowej (jest to minimalna próba odgadnięcia właściwego lematu — nierzadko lemat jest równy formie wyrazowej, zwłaszcza w przypadku nazw własnych, które często podczas znakowania są właśnie słowami nieznanymi). Opisany zabieg dodawania nowych elementów do zbio-

rów interpretacji wykonywany jest w osobnej iteracji po segmentach, poprzedzającej właściwe wyliczanie wartości cech. Ta osobna iteracja jest konieczna, by wyliczone wartości cech uwzględniały rozszerzone w ten sposób zbiory interpretacji przypisane segmentom znajdującym się na różnych pozycjach zdania.

Algorytm znakowania poddany został analogicznym modyfikacjom; wersję zmodyfikowaną przedstawiono jako algorytm 4.

Algorytm 4 Ujednoznacznianie pojedynczego zdania przez algorytm WMBT z modułem odgadującym

Dane: *zdanie* poddane analizie morfosyntaktycznej
zbiór cech przypisany każdemu z atrybutów $cechy(a)$
wyuczony klasyfikator K_a^K i K_a^U dla każdego z atrybutów
Wyniki: ujednoznacznione *zdanie*
for $a \in [klasa, atr_1, \ldots, atr_k]$ **do**
for $seg \in zdanie$ **do**
if *seg* to słowo nieznane **then**
dodaj tagi z listy frekwencyjnej do zbioru tagów segmentu *seg*
end if
end for
for $seg \in zdanie$ **do**
if *seg* jest niejednoznaczny ze względu na a **then**
$wek\_cech \leftarrow [f(seg, zdanie)$ **for** $f \in cechy(a)]$
if *seg* to słowo znane **then**
$decyzja \leftarrow$ klasyfikuj (K_a^K, wek_cech)
else
$decyzja \leftarrow$ klasyfikuj (K_a^U, wek_cech)
end if
if $decyzja \in$ możliwe wartości a pobrane z tagów przypisanych segmentowi *seg* **then**
usuń z segmentu *seg* tagi, dla których wartość$(a) \neq decyzja$
end if
end if
end for
end for
for $seg \in zdanie$ **do**
wybierz arbitralnie „pierwszy wg tagsetu” tag, jeśli pozostało ich kilka
end for

2.9. Ponowna analiza morfosyntaktyczna danych uczących

Opisany w poprzednich punktach algorytm zakłada, że proces decyzyjny opiera się na zbiorach interpretacji morfosyntaktycznych przypisanych podczas analizy morfosyntaktycznej. Podczas uczenia tagera zakłada się natomiast, że dane uczące już zostały poddane analizie morfosyntaktycznej. Podobne założenia zresztą przyjmuje się m.in. w przypadku tagera PANTERA.

Należy się zatem spodziewać najlepszych osiągów tagera, jeśli do przygotowania danych uczących zostanie użyty ten sam analizator morfologiczny, z którego będziemy korzystać podczas właściwego znakowania. Jeśli użyjemy innych analizatorów, prawdopodobna jest sytuacja, że dokładnie ten sam segment zostanie rozpoznany jako inna klasa decyzyjna, co może owocować błędną decyzją. Dla przykładu załóżmy, że podczas uczenia i testowania trafiliśmy na dokładnie to samo zdanie, zawierające słowo piec. Załóżmy, że w korpusie uczącym słowu temu przypisano jedynie dwa tagi rzeczownikowe (np. w mianowniku i w bierniku), spośród których jeden oznaczony jest jako poprawny, natomiast analizator morfosyntaktyczny użyty podczas testowania tagera przypisał temu słowu jeszcze trzeci, czasownikowy, tag. Cecha pobierająca wszystkie możliwe wartości danego atrybutu dla segmentu przyjmie różne wartości podczas działania i podczas uczenia, a zatem otrzymamy różne wektory cech opisujące te przypadki. Może to spowodować błędną klasyfikację.

Warto tutaj dodać, że problem ten nie był dotąd rozważany w literaturze dotyczącej znakowania morfosyntaktycznego języka polskiego. Najprawdopodobniej wynika to z powszechnie stosowanych w tych kręgach testów ograniczonych jedynie do zdolności wykreślania interpretacji z korpusu wzorcowego (por. rozdział 2.5) — takie testy nie są w stanie wykazać różnicy, gdyż analizator morfosyntaktyczny podczas znakowania nie jest używany.

W niniejszym rozdziale proponujemy prostą metodę, która pozwala na lepsze wykorzystanie istniejących zasobów: ponowną analizę morfosyntaktyczną danych uczących. Celem tego zabiegu jest zmniejszenie rozbieżności między zbiorami interpretacji morfosyntaktycznych dostępnych w danych uczących, a zbiorami, które podczas działania otrzymamy z analizatora morfosyntaktycznego. Metoda ta została opracowana pod kątem zastosowania w proponowanym algorytmie opartym na uczeniu pamięciowym, aczkolwiek jej zastosowanie jest szersze (co wykażemy eksperymentalnie).

Procedura prowadzi do przetworzenia pierwotnego korpusu uczącego w wynikowy korpus uczący, którego informacja morfologiczna będzie w miarę możliwości odpowiadać interpretacjom pochodzącym z używanego analizatora. Przypomnijmy, że w podkorpusie milionowym NKJP każdemu segmentowi przypisany jest zbiór możliwych interpretacji, spośród których dokładnie jedna oznaczona jest jako wzorcowa/właściwa w kontekście jej wystąpienia. Nie jest możliwe bezpośrednie zastąpienie wszystkich interpretacji z korpusu wzorcowego interpretacjami z analizatora, gdyż część interpretacji uznanych za właściwych w korpusie wzorcowym może w analizatorze się w ogóle nie pojawić. Proponowana procedura opisuje sposób postępowania w obu przypadkach:

1. Zamieniamy *pierwotny korpus uczący* na czysty tekst (pozbywamy się więc informacji o segmentacji, analizie morfosyntaktycznej i prawidłowym oznakowaniu).[27]
2. Czysty tekst przetwarzamy przez moduł segmentacji i analizator morfosyntaktyczny. Otrzymany w ten sposób korpus nazwijmy *korpusem pośrednim*.

[27] Krok ten zakłada, że korpus wzorcowy zawiera przynajmniej szczątkową informację o znakach białych występujących pomiędzy segmentami. Informacja ta jest potrzebna, by prawidłowo złączyć segmenty w czysty tekst. Informacja o tym, czy znak interpunkcyjny poprzedziły znaki białe może być kluczowa do prawidłowego podziału na zdania. Na szczęście, informacja taka jest dostępna w korpusach NKJP i KIPI w postaci znacznika `<ns/>` oznaczającego brak znaków białych między dwoma segmentami (*no space*). Znacznik taki można spotkać np. pomiędzy segmentem reprezentującym ostatni wyraz zdania a segmentem–kropką.

3. Scalamy pierwotny korpus uczący z korpusem pośrednim w następujący sposób (otrzymany korpus nazwijmy *wynikowym*):
   a) Podział na segmenty i zdania bierzemy z korpusu wzorcowego.
   b) Segmenty podlegające zmianom segmentacji bierzemy niezmienione z korpusu wzorcowego (dla uproszczenia przypadki te pomijamy jako występujące rzadko). Zatem segmentom takim w korpusie wynikowym pozostaną przypisane zbiory interpretacji wraz z interpretacją oznakowaną jako wzorcowa pobrane bezpośrednio z korpusu wzorcowego.
   c) Pozostałe segmenty (zdecydowana większość w praktyce) możemy porównać w stosunku 1:1. Jeśli interpretacja oznakowana jako wzorcowa w korpusie wzorcowym również pojawia się pośród interpretacji z korpusu pośredniego, do korpusu wynikowego bierzemy zbiór możliwych interpretacji z korpusu pośredniego i oznaczamy pożądaną interpretację jako wzorcową. Jeśli interpretacja wzorcowa z korpusu wzorcowego nie wystąpi w korpusie pośrednim, to tager nie byłby w stanie takiego segmentu prawidłowo oznakować — a zatem uznajemy taki segment za *słowo nieznane*. Stosujemy zabieg omówiony w punkcie 2.2, by jawnie oznakować takie segmenty: w korpusie wynikowym segmentowi przypisujemy dokładnie dwie interpretacje: tę wzorcową (oznakowaną wprost jako wzorcowa) oraz sztuczną nie-wzorcową interpretację, na którą składa się tag „słowo nieznane" (zgodnie z zaleceniami tagsetu) i lemat równy formie napotkanej słowa sprowadzonej do małych liter.

Opisana procedura jest uproszczona, tj. problem zmian segmentacji został w niej świadomie pominięty. Motywacja jest praktyczna: eksperymenty wskazują, że zmiany segmentacji dotykają mimo wszystko niewielkiego odsetka segmentów, podczas gdy opracowanie ogólnej strategii radzenia sobie z takimi przypadkami wydaje się zadaniem niełatwym.

2.10. Podsumowanie

W tym rozdziale omówiliśmy problem znakowania morfosyntaktycznego w kontekście języka polskiego. Przedstawiliśmy dostępne dla języka polskiego korpusy, używane tagsety, a także dokonaliśmy przeglądu metod znakowania morfosyntaktycznego. Rozdział przedstawia także elementy oryginalne, stanowiące wkład tej rozprawy w dziedzinę znakowania morfosyntaktycznego języka polskiego:

1. rozważenie na nowo problemu oceny tagerów i propozycja nowej metodyki oceny;
2. nową metodę znakowania morfosyntaktycznego języka polskiego łącząca znane techniki: znakowanie warstwowe, uczenie pamięciowe oraz analizę morfosyntaktyczną;
3. wariant powyższej metody, który pozwala na ropoznawanie słów nieznanych;
4. technikę ponownej analizy morfosyntaktycznej danych uczących, której zadaniem jest poprawa osiągów tagera.

Rozdział 3

Eksperymentalna ocena algorytmów znakowania morfosyntaktycznego

3.1. Cel

Zasadniczym celem eksperymentów przedstawionych w tym rozdziale jest porównanie osiągów kilku metod znakowania morfosyntaktycznego. Ocena przeprowadzana jest na podstawie miar opartych na zgodności wyników automatycznego znakowania z oznakowaniem wzorcowym. Porównanie to pozwala na wskazanie metody o najlepszych osiągach i analizę różnic pomiędzy metodami; w szczególności zaś daje odpowiedź na pytanie, czy zmiana metody przynosi istotną poprawę. Oceniono zarówno różne algorytmy *sensu stricto*, jak i kilka praktycznych konfiguracji, w jakich algorytmów tych można użyć: pokazano m.in. wpływ zestawu cech na osiągi tagera pamięciowego MBT, a także zysk, jaki można osiągnąć dzięki zastosowaniu ponownej analizy morfosyntaktycznej danych uczących (technika zaproponowana w rozdziale 2.9).

Drugim celem eksperymentów jest pokazanie różnic pomiędzy wynikami testów samego modułu ujednoznaczniania, a wynikami pełnego testu tagera przeprowadzonego na czystym tekście (zgodnie z zaleceniami z rozdziału 2.5). Porównanie to wskazuje skalę problemu: rzeczywisty odsetek błędów popełnianych przez tagery jest niemal dwukrotnie wyższy niż przedstawiany w większości publikacji.

3.2. Kryterium oceny i stosowany zbiór danych

Z powodów omówionych w punkcie 2.5.3, za ogólną miarę oceny tagerów uznajemy **dolne ograniczenie trafności**. Rozważania na temat różnic w osiągach tagerów, w tym również i ocena istotności statystycznej różnic, odnosić się będą do wartości tej właśnie miary.

Oprócz tego podajemy wartości kilku miar pomocniczych: górnego ograniczenia trafności, dolnego ograniczenia trafności liczonego z osobna dla słów znanych i nieznanych, a także trafności samego modułu ujednoznaczniania. Wartości te mają na celu pokazanie skali problemów pojawiających się przy ocenie tagerów omawianych w

poprzednim rozdziale, w szczególności: zmian segmentacji wprowadzonych przez tagery, znakowania słów nieznanych oraz błędów analizatora morfosyntaktycznego.

Wszystkie eksperymenty przeprowadzono na danych pochodzących z *podkorpusu milionowego NKJP*, tj. ręcznie oznakowanej części Narodowego Korpusu Języka Polskiego w wersji 1.0 (patrz rozdział 2.2). Korpus ten został wybrany jako materiał testowy z trzech powodów:

1. jest to największy publicznie dostępny ręcznie oznakowany korpus języka polskiego, prawie dwa razy większy od korpusu *FREK*;
2. ręczne oznakowanie przeprowadzono zgodnie z rygorystycznymi normami jakości: dwóch lingwistów znakowało równolegle te same fragmenty, po czym rozbieżności były rozstrzygane przez trzeciego (Przepiórkowski i Szałkiewicz, 2012); niestety, nie można tego powiedzieć o korpusie FREK[1];
3. korpus zawiera polszczyznę współczesną (Degórski i Przepiórkowski, 2012).

Jesteśmy przekonani, że użycie publicznie dostępnego zbioru danych jest ważne, gdyż ułatwia odtworzenie wyników opisanych tutaj eksperymentów — a jest to kluczowe w przypadku nauk empirycznych (Pedersen, 2008).

Wszystkie eksperymenty przeprowadzono na danych pochodzących z *podkorpusu milionowego NKJP* (patrz 2.2). Statystyki korpusu przedstawiono w tabeli 3.1.

Segmentów	1 215 513
Zdań	85 663
Akapitów	18 484

Tabela 3.1. Statystyki podkorpusu milionowego NKJP 1.0.

Tam, gdzie testy przeprowadzane były na czystym tekście, użyto analizatora morfosyntaktycznego Morfeusz SGJP[2], będącego udoskonaleniem starszego analizatora Morfeusz SIaT (Woliński, 2006). Morfeusz zwraca analizy w postaci struktury grafowej, podczas gdy wszystkie rozpatrywane algorytmy znakowania morfosyntaktycznego operują na ciągach segmentów. Dlatego też konieczne było niekiedy zastosowanie heurystyk, które wybiorą ścieżki w grafie przedstawiającym niejednoznaczności segmentacji (problem ten został omówiony na s. 14). Tager PANTERA zawiera kilka reguł heurystycznych napisanych ręcznie, które decydują o wyborze ściezki. Gdy testowana jest PANTERA, reguły te są wykorzystywane. W przypadku pozostałych tagerów, gdy wystąpi niejednoznaczność segmentacji, przyjmujemy heurystykę wyboru najkrótszej ścieżki w grafie. Heurystyka ta została zaimplementowana w pakiecie Maca wspomagającym analizę morfosyntaktyczną (por. dodatek A oraz Radziszewski i Śniatowski, 2011a) i z tej właśnie implementacji korzystamy.

[1] Przykładowo, fragmenty zdań ewidentnie wycięto, a w korpusie pojawiają się niejednokrotnie sztuczne segmenty w stylu `DELETED_TOKEN` czy `DELETED_END_OF_SENTENCE`; segmenty takie występują na równi z pozostałymi segmentami w korpusie, a prawidłowe ich traktowanie wymagałoby na dobrą sprawę wstępnego przetworzenia tekstu, być może przy udziale lingwisty.

[2] Morfeusz SGJP w wersji 64-bitowej pobrany ze strony `http://sgjp.pl/morfeusz/dopobrania.html`; wersja kodu 0.82 (2010.02.22), dane lingwistyczne z 2011.04.15. Autorem kodu jest Marcin Woliński, natomiast dane lingwistyczne pochodzą od Zygmunta Saloniego, Włodzimierza Gruszczyńskiego, Marcin Wolińskiego oraz Roberta Wołosza. Morfeusz wraz z danymi lingwistycznymi dostępny jest publicznie na bardzo swobodnej licencji BSD.

3.3. Metodyka analizy wyników

Wszystkie eksperymenty opisane w tym rozdziale opierają się na porównaniu automatycznego oznakowania dokonanego przez dany tager z oznakowaniem wzorcowym, przypisanym przez lingwistów w korpusie wzorcowym. Niezależnie od użytej miary, testy tagerów uczonych na wzorcowo oznakowanym korpusie wymagają wydzielenia części testowej i uczącej z korpusu wzorcowego. Najprostszą metodą oceny jest jednokrotny eksperyment polegający na wyuczeniu tagera na części uczącej, oznakowaniu za pomocą wyuczonego modelu części testowej (uprzednio oczyszczonej z oznakowania wzorcowego) i porównanie wynikowego oznakowania z wzorcowym oznakowaniem części testowej. Porównanie takie polega na wyliczeniu wartości danej miary, np. trafności.

Podejście takie ma jednak zasadniczą wadę: wybrany podział na część uczącą i testową może mieć duży wpływ na uzyskane wyniki, zwłaszcza jeśli nie mamy do dyspozycji dużego korpusu wzorcowego.

Powszechnie stosowaną praktyką jest *K-krotny sprawdzian krzyżowy* (kroswalidacja, ang. *cross-validation*) (Koronacki i Ćwik, 2005, s. 90). Celem takiego działania jest zmniejszenie wariancji obserwowanych wyników, a zatem uwiarygodnienie wyników eksperymentów przeprowadzonych na dostępnym zbiorze danych. Praktyka ta zakłada podział korpusu wzorcowego na K możliwie równych części (najczęściej przyjmuje się $K = 10$). Następnie generowanych jest K par (*część ucząca*, *część testowa*), gdzie część testowa i-tej pary stanowi i-tą część z pierwotnego podziału korpusu, natomiast część ucząca uzyskiwana jest poprzez scalenie wszystkich pozostałych części z pierwotnego podziału korpusu. Każda para zawiera więc łącznie materiał z całego korpusu wzorcowego, gdzie część testowa zajmuje $\frac{1}{K}$ całości, natomiast część ucząca stanowi $\frac{K-1}{K}$, przy czym części testowe są wzajemnie rozłączne. Ocena tagera polega na K-krotnym powtórzeniu uczenia i testowania na odpowiednich częściach. Jako wynik końcowy podaje się średnią wartość danej miary oceny obliczoną na podstawie wszystkich K powtórzeń eksperymentu.

Wszystkie wartości miar oceny tagerów podane w tym rozdziale zostały obliczone na podstawie dziesięciokrotnego sprawdzianu krzyżowego. Korpus wzorcowy został podzielony na dziesięć równych części z dokładnością do akapitów (żadnego akapitu nie przedzielono).

Zaletą K-krotnego sprawdzianu krzyżowego jest możliwość oceny istotności statystycznej różnicy między wynikami osiągniętymi przez dwa różne tagery. Stosujemy w tym celu test t-Studenta dla prób zależnych (Dietterich, 1998). Test zakłada, że używamy tego samego korpusu wraz z tym samym podziałem na K części uczących i testowych. Na w ten sposób przygotowanych danych przeprowadzamy K prób dla tagera A i tyle samo prób dla tagera B. i-ta próba polega na wyuczeniu tagera na i-tej części uczącej i przetestowanie go na i-tej części testowej. Niech $Acc_A^{(i)}$ oznacza procent segmentów poprawnie oznakowanych[3] przez tager A podczas i-tej próby. Niech $Acc^{(i)} = Acc_A^{(i)} - Acc_B^{(i)}$ oraz $\overline{Acc} = \frac{1}{K}\sum_{i=1}^{K} Acc^{(i)}$. Wtedy statystyka t przyjmuje postać opisaną wzorem (3.1).

[3] Symbolu *Acc* używamy tutaj dla uogólnienia wszelkich miar oceny tagera, które da się wyrazić jako odsetek segmentów z korpusu wzorcowego, które uznano za prawidłowo oznakowane. Zgodnie z zaleceniami z punktu 2.5.3, jako miarę oceny tagerów będziemy przyjmować *dolne ograniczenie trafności*.

$$t = \frac{\overline{Acc} \cdot \sqrt{K}}{\sqrt{\frac{\sum_{i=1}^{K}\left(Acc^{(i)} - \overline{Acc}\right)^2}{K-1}}} \tag{3.1}$$

Przyjmijmy hipotezę H_0: oba tagery popełniają tyle samo błędów, oraz hipotezę H_1: tager A popełnia więcej błędów niż tager B. Zgodnie z testem t-Studenta dla prób zależnych, hipotezę H_0 możemy odrzucić, jesli zachodzi nierówność (3.2). Parametr α jest poziomem istotności, natomiast $K-1$ to liczba stopni swobody. Wartość $t_{K-1;1-\alpha}$ odczytujemy z tablic statystycznych.

$$t > t_{K-1;1-\alpha} \tag{3.2}$$

Wszystkie testy istotności statystycznej opisane w tym i kolejnych rozdziałach przeprowadzono dla poziomu istotności $\alpha = 0,05$. Tam, gdzie stosowany był K-krotny sprawdzian krzyżowy, przyjęto $K = 10$.

3.4. Wyniki oceny tagerów

Pierwszy z eksperymentów polegał na ocenie działania trzech tagerów: MBT (przetestowano trzy zestawy cech, które omówione zostały w punkcie 2.7.4), PANTERA oraz proponowanego w tej pracy tagera WMBT (jak opisano w punkcie 2.7, tj. bez modułu rozpoznawania słów nieznanych i bez ponownej analizy morfosyntaktycznej danych uczących).

W tabeli 3.2 przedstawiono wyniki testów tagerów na podkorpusie milionowym NKJP 1.0. Użyte miary to:

1. trafność samego ujednoznaczniania (Acc_{dis}; użyto danych morfologicznych i segmentacji wprost z korpusu wzorcowego, zgodnie z definicją 2.18 ze s. 36),
2. dolne ograniczenie trafności (Acc_{lower}, definicja 2.28 ze s. 41),
3. górne ograniczenie trafności (Acc_{upper}, definicja 2.29 ze s. 41),
4. dolne ograniczenie trafności liczone jedynie dla słów znanych (Acc^{K}_{lower}),
5. dolne ograniczenie trafności liczone jedynie dla słów nieznanych (Acc^{U}_{lower}).

Tager	Acc_{dis}	Acc_{lower}	Acc_{upper}	Acc^{K}_{lower}	Acc^{U}_{lower}
MBT: cechy 0	79,31%	79,11%	79,44%	80,30%	40,49%
MBT: cechy 1	88,03%	84,14%	84,46%	85,79%	30,74%
MBT: cechy 2	87,12%	83,39%	83,72%	85,00%	31,36%
PANTERA	92,95%	88,79%	89,09%	91,08%	14,70%
WMBT	93,00%	87,50%	87,82%	89,78%	13,57%

Tabela 3.2. Porównanie tagerów na podkorpusie milionowym NKJP 1.0.

Pierwszym wnioskiem z eksperymentu są najlepsze osiągi tagera PANTERA. Wyniki osiągnięte przez tager WMBT są nieco gorsze (różnica ta jest istotna statystycznie). Tager MBT daje zdecydowanie gorsze rezultaty; wyniki te zależne są od przyjętego zestawu cech, najlepszą wartość dolnego ograniczenia trafności udało się osiągnąć dla uproszczonego zestawu cech (*cechy 1*). Porównanie osiągów tagera MBT i WMBT wskazuje na duży zysk osiągnięty dzięki wprowadzeniu modelu warstwowego.

Widoczna jest różnica między wartościami dolnego i górnego ograniczenia trafności. Mimo to, między wartościami obu miar dla wszystkich testowanych tagerów zachodzą te same nierówności.

Bardzo ciekawe wnioski można wyciągnąć z porównania odnotowanej trafności samego ujednoznaczniania (*Acc*) z wartościami dolnego ograniczenia trafności. W przypadku tagerów zakładających użycie zewnętrznego analizatora morfosyntaktycznego, tj. tagerów PANTERA i WMBT, mamy do czynienia z rażącą rozbieżnością między wartościami obu miar — wartości te dla WMBT to, odpowiednio, 87,5% i 93,0%. Rozbieżność ta wynika z prostego faktu, że ocena trafności samego ujednoznaczniania zaniedbuje błędy popełnione na etapie analizy morfosyntaktycznej i segmentacji. Jak widać — błędy te stanowią prawie połowę całkowitego odsetka błędów popełnianych przez tager. Zgodnie z przewidywaniami, obie miary dają zbliżone wyniki dla tagera MBT, który z założenia nie ma dostępu do zewnętrznego analizatora morfosyntaktycznego — a więc i nie jest w stanie skorzystać ze wzorcowego oznakowania morfosyntaktycznego dostępnego w korpusie testowym. Jest to kolejnym dowodem na to, że ocena działania tagera na tekście poddanym wzorcowej analizie morfosyntaktycznej jest nierzetelna: tagery, które zakładają działanie dwuetapowe, mają możliwość „podejrzenia" wzorcowego oznakowania morfosyntaktycznego i zyskują niezasłużone punkty za trafienie, natomiast pozostałe tagery testowane są prawie rzetelnie (tj. z dokładnością do błędów segmentacji).

Przeprowadzenie testów tagera zgodnie z proponowaną metodyką pozwoliło również poznać przybliżenie trafności znakowania słów nieznanych. Problem ten nie był dotąd poruszany w polskich publikacjach, a wyniki z tabeli 3.2 sugerują, że warto się tym problemem zająć. Tager MBT, którego dolne ograniczenie trafności wypada najgorzej na tle pozostałych, daje najlepsze oznakowanie słów nieznanych (najlepszy wynik to 40,5% dla naiwnego zestawu cech). Pozostałe tagery wykazują bardzo niską trafność rozpoznawania słów nieznanych, poniżej 15%. Różnica ta wynika prawdopodobnie stąd, że tager MBT jako jedyny z testowanych zawiera moduł odgadywania tagów słów nieznanych.

3.5. Testy modułu odgadującego nieznane słowa

Tager	Acc_{lower}	Acc_{upper}	Acc^{K}_{lower}	Acc^{U}_{lower}
PANTERA	88,79%	89,09%	91,08%	14,70%
WMBT bez	87,50%	87,82%	89,78%	13,57%
WMBT z	88,44%	88,76%	89,89%	**41,43**%

Tabela 3.3. Wpływ modułu odgadującego nieznane słowa na wyniki tagera pamięciowego. Wyniki tagera wraz z tymże modułem umieszczono w wierszu *„WMBT z"*.

W tabeli 3.3 przedstawiono porównanie wyników osiągniętych przez tager PANTERA, tager pamięciowy oraz tager pamięciowy rozszerzony o algorytm odgadujący słowa nieznane (rozdział 2.8). Poprawa osiągnięta przez moduł odgadujący jest istotna statystycznie. Wyniki tagera PANTERA są jednak wciąż lepsze (różnica ta również jest

istotna). Warto zwrócić uwagę na wartości dolnego ograniczenia trafności liczonego jedynie dla słów nieznanych: dla tych słów odsetek błędów jest o ponad 30% niższy niż w przypadku tagera PANTERA.

3.6. Ponowna analiza morfosyntaktyczna danych uczących

Tager	**Re-analiza**	Acc_{lower}	Acc_{upper}	Acc^{K}_{lower}	Acc^{U}_{lower}
PANTERA	nie	88,79%	89,09%	91,08%	14,70%
	tak	88,99%	89,28%	91,27%	14,74%
WMBT bez	nie	87,50%	87,82%	89,78%	13,57%
	tak	88,75%	89,08%	91,07%	13,62%
WMBT z	nie	88,44%	88,76%	89,89%	41,43%
	tak	**89,71%**	90,04%	91,20%	41,45%

Tabela 3.4. Wpływ ponownej analizy morfosyntaktycznej danych uczących na wyniki tagerów.

W tabeli 3.4 przedstawiono wyniki osiągnięte dzięki zastosowaniu zaproponowanej w punkcie 2.9 techniki *ponownej analizy morfosyntaktycznej danych uczących*. Wyniki potwierdzają skuteczność tej techniki: pozwoliła ona osiągnąć istotny statystycznie wzrost trafności znakowania przez tager WMBT (obu wariantów: bez modułu odgadującego i z nim), ale również i przez tager PANTERA. Co więcej, osiągi tagera WMBT wyposażonego w moduł odgadujący po zastosowaniu tej techniki przewyższają najlepsze osiągi tagera PANTERA (różnica również jest istotna statystycznie). Tager WMBT jest w stanie lepiej wykorzystać informację pochodzącą z ponownej analizy morfosyntaktycznej niż PANTERA dzięki modułowi odgadującemu: dokonuje on jawnego podziału segmentów na słowa znane i nieznane, podczas gdy informacja ta zakodowana jest wprost w wynikach ponownej analizy morfosyntaktycznej.

3.7. Podsumowanie

W rozdziale zostały opisane badania kilku metod znakowania morfosyntaktycznego języka polskiego:

1. za pomocą (niepolskiego) tagera pamięciowego MBT wyposażonego w swój standardowy zestaw cech,
2. za pomocą tagera MBT wyposażonego w dwa zestawy cech zaproponowane w tej rozprawie dla języka polskiego,
3. za pomocą tagera języka polskiego PANTERA,
4. używając metody zaproponowanej w tej pracy (WMBT) przetestowanej w kilku wariantach.

Eksperymenty pokazały, że dzięki pełnej metodzie WMBT możliwe jest osiągnięcie wyników lepszych niż za pośrednictwem tagera PANTERA. Co więcej, eksperymenty potwierdziły, że osiągnięcie tych wyników możliwe było dzięki wprowadzeniu do metody WMBT dwóch technik:

1. metody odgadywania słów nieznanych dzięki tworzeniu dla nich osobnych baz przypadków uczących;
2. techniki ponownej analizy morfosyntaktycznej danych uczących.

Co więcej, przeprowadzone badania pokazały skalę problemu związanego ze stosowaniem nierzetelnych metod oceny tagerów. Obawy przedstawione w rozdziale 2.5 zostały potwierdzone empirycznie: popularne metody oceny zaniedbują błędy popełniane przez tagery na etapie segmentacji i analizy morfologicznej, co powoduje, że raportowany w publikacjach odsetek błędu jest w skrajnych przypadkach dwukrotnie zaniżony. Ponadto, metody oceny zaniedbujące tego rodzaju błędy w sposób nieuzasadniony faworyzują tagery zakładające użycie analizatora morfosyntaktycznego nad tagerami, które w ten sposób nie działają.

Warto też dodać, że przeprowadzenie testów tagerów zgodnie z metodyką proponowaną w punkcie 2.5.3 pozwoliło zauważyć dotychczas nieomawiany w literaturze problem niskiej trafności znakowania słów nieznanych przez tagery języka polskiego. Dostrzeżenie tego problemu dało podstawę do opracowania metody odgadywania słów nieznanych.

Rozdział 4

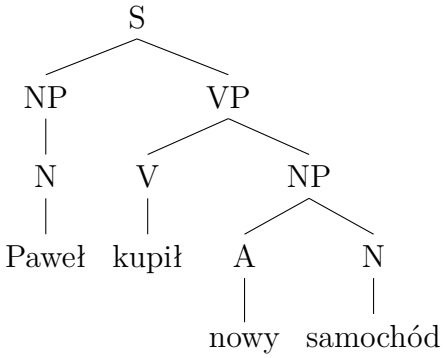
Znakowanie fraz

Zgodnie z tym, co napisano w rozdziale 1, znakowanie fraz to jedna z form płytkiej analizy składniowej i polega na na przypisaniu zdaniom następującej struktury:

— zdanie zostaje podzielone na fragmenty będące rozłącznymi ciągami segmentów,

— fragmenty te są klasyfikowane: każdemu fragmentowi przypisywana jest albo nazwa frazy (jedna z kilku z góry ustalonych), albo fragment określany jest jako nienależący do żadnej interesującej nas klasy fraz.

W takim ujęciu frazy są „płaskie", tj. samym frazom nie jest przypisana struktura. Sposób podziału zdania na takie płaskie frazy na ogół wiąże się z drzewem rozbioru składniowego, czyli popularną reprezentacją wyniku pełnej analizy składniowej. Przykład (4.1) przedstawia proste drzewo rozbioru składniowego (na podstawie Polański, 1999b). Jeśli założymy, że zbiór interesujących nas fraz ogranicza się do fraz rzeczownikowych (NP), rozbiorowi temu odpowiadałoby oznakowanie frazami przedstawione jako przykład (4.2). Jeśli rozszerzymy zbiór o frazy czasownikowe (VP), jedną z możliwych realizacji oznakowania frazami jest przykład (4.3).



(4.1)

(4.2) [$_{NP}$ Paweł] kupił [$_{NP}$ nowy samochód]

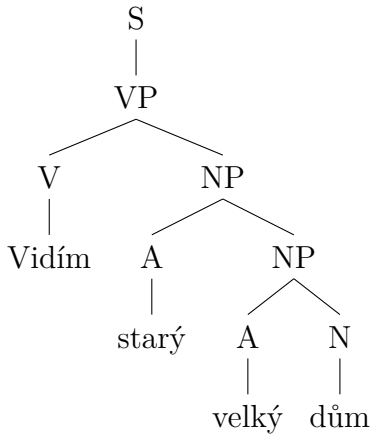
(4.3) [$_{NP}$ Paweł] [$_{VP}$ kupił] [$_{NP}$ nowy samochód]

Często frazy (płaskie) odpowiadają bezpośrednio poddrzewom drzewa rozbioru składniowego. Nie zawsze tak jednak jest; Abney (1991) w definicjach fraz podaje wyjątki od tej reguły dla języka angielskiego. Pomijając nawet te wyjątki, decyzja, które poddrzewo należy wybrać jako frazę nie zawsze jest łatwa i zależy nie tylko od

charakteru języka, ale także od założeń teoretycznych i przewidywanych zastosowań modułu znakowania fraz. Łatwo tutaj dostrzec analogię do problemu definicji tagsetu (por. punkt 2.3 na stronie 16). Zagadnienie wyboru poddrzewa omawiają między innymi Jakubíček i inni (2009), podając przykładowe proste zdanie czeskie (4.4) oraz dwie kandydujące (płaskie) frazy rzeczownikowe, które można ze zdania wydobyć: starý velký dům oraz velký dům. Obie te frazy odpowiadają poddrzewom drzewa rozbioru składniowego przedstawionego jako przykład (4.5). Problem definicji fraz dla języków słowiańskich, a w szczególności języka polskiego omówimy szerzej w punkcie 4.2.

(4.4) *Vidím starý velký dům*
Widzę stary wielki dom

(4.5)



Wraz z definicją zadania znakowania fraz, Abney (1991, 1996b) zaproponował także ogólną „filozofię" płytkiej analizy składniowej, która opiera się na poniższych zasadach:

— **łatwe decyzje** są podejmowane w pierwszej kolejności; jeśli decyzja wydaje się zbyt trudna, nie podejmujemy jej wcale;
— **niejednoznaczności** rozwiązywane są stopniowo; nie wszystkie z nich muszą zostać rozwiązane, lecz wynikowa struktura powinna pozostawiać jawną informację o pozostałych niejednoznacznościach;
— powstrzymujemy się od podjęcia zbyt trudnych decyzji, więc możemy liczyć na **wysoką dokładność** (niewielki odsetek błędnie podjętych decyzji).

Do wspomnianych niejednoznaczności i niepodjętych decyzji można wrócić w razie potrzeby w ramach następnych etapów przetwarzania tekstu. Abney (1991) proponuje następny etap przetwarzania w postaci modułu **podpinającego** frazy pod inne frazy (ang. *attacher*). Podobne rozwiązania rozważane są w późniejszych pracach, np. Federici i inni (1997); Daelemans i inni (1999).

W dalszej części tego rozdziału omówimy zastosowania znakowania fraz, przedstawimy dostępne korpusy oznakowane frazami i porównamy definicje fraz stosowane dla języków słowiańskich. Następnie przedstawimy stosowane metody oceny modułów znakowania fraz, a także dokonamy przeglądu metod znakowania fraz w tekście. W ostatniej części tego rozdziału przedstawimy kilka metod, które można zastosować do znakowania fraz w języku polskim. Pierwszą z tych metod jest konwersja wyjścia płytkiego parsera Spejd na płaskie frazy, pozostałe metody opierają się o techniki maszynowego uczenia. Chociaż te ostatnie metody nie są nowe i były już stosowane do znakowania fraz w języku angielskim, nowością jest zaproponowany zestaw cech.

Nowością jest także zastosowanie technik maszynowego uczenia do znakowania fraz w językach słowiańskich (trudno jest dotrzeć do publikacji na ten temat).

4.1. Zastosowania

Znakowanie fraz zostało zaproponowane przez Abneya (1996b) jako etap pośredni prowadzący do głębokiej analizy składniowej. Taką rolę pełniły również moduły znakowania fraz opracowane później dla innych języków, m.in. włoskiego (Federici i inni, 1997) czy bułgarskiego (Osenova, 2002).

Znakowanie fraz i, bardziej ogólnie, płytka analiza składniowa, znalazły inne praktyczne zastosowania. Zaobserwowano, że pełny rozbiór składniowy dostarcza informacji dużo bardziej szczegółowej niż wymaga tego wiele praktycznych zastosowań. Naukowcy zajmujący się budową systemów wydobywania informacji z tekstu, automatycznego streszczania, czy automatycznej odpowiedzi na pytania w języku naturalnym są zainteresowani przede wszystkim rozpoznaniem fraz czasownikowych i rzeczownikowych oraz niekiedy relacji składniowo-semantycznych między nimi, natomiast pełny rozbiór składniowy zdań na ogół nie jest im potrzebny (Shen, 2004).

Systemy automatycznego odpowiadania na pytania zadane w języku naturalnym (ang. *Question Answering*) budowane są w oparciu o standardowe moduły przetwarzania języka naturalnego, takie jak tager czy właśnie moduł znakowania fraz. Znakowanie fraz znajduje tu dwa zastosowania: może wspomagać zarówno analizę pytań, jak i ułatwiać znalezienie w odpowiedzi w dokumentach tekstowych. Przykładowo, Tufiş (2011) opisuje algorytm, który pozwala na analizę pytań w języku naturalnym i ich transformację na zapytania do systemu wyszukującego terminy w zbiorach dokumentów. Algorytm opiera się na rozpoznanych w pytaniu płaskich frazach rzeczownikowych — dla każdej takiej frazy tworzone jest osobne zapytanie do bazy danych. Jinguji i inni (2006) opisują inny system automatycznego odpowiadania na pytania, gdzie moduł znakowania fraz rzeczownikowych znalazł zastosowanie jako narzędzie wspomagające znajdowanie odpowiedzi w analizowanych zbiorach dokumentów. System zakłada, że odpowiedzią na pytanie jest zawsze pojedyncza fraza rzeczownikowa lub zbiór takich fraz w przypadku pytania typu „lista odpowiedzi".

Hobbs i Riloff (2010) opisują budowę typowego systemu wydobywania informacji z tekstu. Schemat składa się z pięciu etapów przetwarzania:

1. rozpoznawanie jednostek wielowyrazowych i nazw własnych;
2. znakowanie prostych fraz rzeczownikowych, czasownikowych i partykuł;
3. znakowanie złożonych fraz rzeczownikowych i czasownikowych;
4. rozpoznawanie wyrażeń opisujących zdarzenia i byty związane z dziedziną za pomocą dopasowania wzorców; wzorce takie składają się z ciągów fraz rozpoznanych w poprzednich etapach przetwarzania;
5. złączenie wyrażeń rozpoznanych w ramach poprzedniego etapu rozproszonych po różnych fragmentach dokumentu w informację całościową.

Jak widać, moduł znakowania fraz rzeczownikowych i czasownikowych pełni kluczową rolę w takim systemie, gdyż w oparciu o te frazy formułowane są wzorce pozwalające na rozpoznanie w tekście odwołań do bytów, o których jest mowa, i relacji między nimi. To zastosowanie modułu znakowania fraz potwierdzają również prace przedsta-

wiające konkretne systemy wydobywania informacji, np. FASTUS (Hobbs, 1992; Appelt i inni, 1993) czy TEXTRUNNER (Etzioni i inni, 2008).

Znakowanie fraz znajduje również zastosowanie w systemach przypisujących dokumentom tekstowym zbiory słów kluczowych (często „słowa kluczowe" są w rzeczywistości całymi frazami). Hulth (2003) przeprowadziła eksperyment z zastosowaniem trzech technik rozpoznania granic potencjalnych słów kluczowych; jedną z owych technik było właśnie znakowanie fraz rzeczownikowych. Wyniki eksperymentu pokazują, że znakowanie fraz pozwala na osiągnięcie najwyższej dokładności działania systemu, choć inne techniki mają przewagę, jeśli dążymy do maksymalizacji kompletności.

4.2. Korpusy oznakowane frazami i problem definicji fraz

Istnieją dwa dostępne publicznie korpusy języka polskiego, zawierające ręczne oznakowanie na poziomie płytkiej analizy składniowej:

1. Korpus Języka Polskiego Politechniki Wrocławskiej (w skrócie *Korpus Politechniki Wrocławskiej* albo *KPWr*) zawiera wzorcowe oznakowanie frazami wykonane przez lingwistów.
2. Podkorpus milionowy Narodowego Korpusu Języka Polskiego (NKJP; por. strona 12) został również oznakowany na poziomie składniowym. Oznakowanie to ma charakter płytki, lecz wykracza poza ramy *znakowania fraz* zgodnie z naszą definicją. Mimo to, przyjmując kilka dodatkowych założeń, można oznakowanie to sprowadzić do oznakowania frazami, a co za tym idzie, wykorzystać materiał z NKJP do testów algorytmów znakowania fraz. Założenia te omówimy w dalszych punktach tego rozdziału.

Korpus KPWr (Broda i inni, 2012) jest obecnie tworzony w Instytucie Informatyki Politechniki Wrocławskiej. Teksty korpusu są znakowane przez lingwistów na kilku poziomach. Jednym z poziomów jest znakowanie fraz. Głównym zastosowaniem korpusu jest przeprowadzenie badań nad zastosowaniem algorytmów maszynowego uczenia do rozpoznawania struktur lingwistycznych w tekście. Badania te prowadzone są w ramach projektu „SyNaT" (system nauki i techniki)[1]. Opisane w tym rozdziale prace stanowią wkład w te właśnie badania.

Oznakowanie składniowe korpusów NKJP i KPWr wykonywane było w dużej mierze niezależnie. Gdy rozpoczął się projekt „SyNaT" (rok 2010), korpus NKJP był w trakcie budowy. Nieznana była jego przyszłość, a w szczególności nie było jasne, czy jakakolwiek część korpusu zostanie udostępniona w postaci umożliwiającej przeprowadzenie planowanych badań. Co więcej, nieznane były również wytyczne znakowania składniowego stosowane w ramach przedsięwzięcia. By zapewnić dostęp do niezbędnych danych, podjęto decyzję o budowie korpusu KPWr. Opracowano również własne wytyczne znakowania fraz, których zadaniem jest wskazywanie właściwych decyzji lingwistom znakującym korpus. Zostały one opracowane przez trzyosobową grupę, składającą się z autora tej rozprawy oraz dwóch lingwistów: Marka Maziarza i Jana Wieczorka. Same wytyczne są ponadsześćdziesięciostronicowym dokumentem, który w pierwszej części omawia główne zasady znakowania, a w drugiej — konkretne rozstrzygnięcia w

[1] Projekt finansowany przez Narodowe Centrum Badań i Rozwoju, numer umowy SP/I/1/77065/10. Politechnika Wrocławska odgrywa rolę jednego z wykonawców, odpowiedzialnego za opracowanie technologii językowych związanych z wydobywaniem informacji z tekstu.

sytuacjach praktycznych (część druga jest znacznie dłuższa). Główne zasady zawarte w wytycznych, a także przykłady ich zastosowania i dyskusję zawarto również w artykule opublikowanym w czasopiśmie *Cognitive Studies* (Radziszewski i inni, 2012). Opracowanie wytycznych poprzedził przegląd podejść do znakowania frazami korpusów innych języków słowiańskich. W dalszej części tego punktu przedstawimy w skrócie ten przegląd oraz decyzje podjęte w ramach wytycznych KPWr. Przedstawimy także główne zasady znakowania składniowego przyjęte w NKJP.

Obecność dwóch korpusów, z których można pozyskać frazy oznakowane według różnych zasad stwarza ciekawe możliwości badań eksperymentalnych. Badania takie opisujemy w rozdziale 5. Warto też dodać, że cechą wyróżniającą korpus Politechniki Wrocławskiej jest jego bardzo swobodna licencja, tj. Creative Commons ShareAlike. Podkorpus milionowy NKJP dostępny jest również na wolnej licencji (GNU GPL), choć licencja ta nakłada większe ograniczenia na możliwe sposoby wykorzystania korpusu.

W dalszej części tego punktu przedstawiono problem definicji fraz w kontekście języków słowiańskich, a w szczególności języka polskiego. Punkt ten zawiera w większości rozważania o naturze lingwistycznej i jako taki został w całości sprawdzony i zaakceptowany przez lingwistę — Marka Maziarza.

4.2.1. Znakowanie fraz a języki słowiańskie

Definicje fraz dla języka angielskiego są w dużej mierze ustalone oraz poparte pracami przedstawiającymi wytyczne i dyskusje na temat ich kształtu, np. Abney (1996a, 1995). Co więcej, istnieją standardowe zbiory danych, na których testowane są różne metody znakowania fraz, np. korpus Wall Street Journal oznakowany frazami na potrzeby konferencji–konkursu CoNLL-2000 (Tjong Kim Sang i Buchholz, 2000).

W przypadku języków słowiańskich sytuacja jest odmienna. Budowie płytkich parserów języków słowiańskich na ogół nie towarzyszą szersze dyskusje na temat kształtu struktury, którą owe parsery mają rozpoznawać, a przynajmniej niełatwo jest znaleźć takie dyskusje w literaturze.

Duża część prac nie czyni wyraźnego rozróżnienia między opisem struktury, która ma zostać oznakowana (np. w postaci wytycznych) a procedurą, dzięki której można uzyskać oznakowanie tekstu zgodnego z oczekiwaniami. Przykładem takiego podejścia jest „formalny model fraz rzeczownikowych w języku serbsko-chorwackim" (Nenadić i Vitas, 1998a). Na model ten składa się zarówno formalizm zapisu gramatyk pozwalających na rozpoznawanie fraz rzeczownikowych, jak i konkretna gramatyka napisana na potrzeby języka serbsko-chorwackiego. Gramatyka ta została potem rozszerzona o inne definicje fraz rzeczownikowych (Nenadić i Vitas, 1998b; Nenadić, 2000). Prace te nie podejmują jednak próby oceny gramatyk jako narzędzia znakowania frazami tekstu. Co więcej, jeśli podane gramatyki rzeczywiście traktować jako „model frazy rzeczownikowej", zmuszeni jesteśmy uznać je za bezbłędne (skoro ta sama gramatyka pełni funkcję działającego parsera, jak i definicji wzorcowych fraz). Zbliżone podejście zastosowano w przypadku znakowania fraz chorwackich: opis konkretnych typów fraz przeplata się z regułami gramatycznymi, które te typy fraz mają rozpoznawać (Vučković, 2009; Vučković i inni, 2010). Mimo to, prace Chorwatów zakładały ocenę na podstawie korpusu, który został uprzednio oznakowany frazami przez lingwistów — a zatem prawdopodobnie istniał choćby bardzo uproszczony opis koncepcyjny struktury, którą owi lingwiście mieli oznakować w tekście.

Znakowanie składniowe NKJP zostało poparte rozważaniami teoretycznymi i definicjami fraz (Głowińska, 2012). Mimo to, na kształt tych definicji wywarło wyraźny wpływ założone z góry użycie parsera regułowego Spejd — a zatem przyjęte definicje fraz dostosowane były do możliwości i ograniczeń konkretnego narzędzia. Niewątpliwą zaletą takiego podejścia jest większa szansa na osiągnięcie dużej dokładności opisu i, zgodnie z duchem płytkiej analizy składniowej, niepodejmowanie zbyt trudnych decyzji. Wadą takiego rozwiązania jest natomiast ograniczenie rozpatrywanych struktur składniowych wymuszone przez sposób działania założonego z góry narzędzia.

W przypadku KPWr, głównym założeniem było wyraźne rozdzielenie tych dwóch aspektów: opracowano wytyczne znakowania fraz o charakterze deklaratywnym, nie narzucając z góry żadnej konkretnej procedury, dzięki której można osiągnąć oznakowanie nowych tekstów. Decyzję, czy dane rozstrzygnięcie wymaga podjęcia zbyt trudnych decyzji, oparto na ocenie, czy rozstrzygnięcia te sprawiają trudności człowiekowi (ocena ta jest oczywiście subiektywna i należy się liczyć z tym, że nie wszystkie decyzje okażą się słuszne podczas stosowania schematu znakowania). Głównym źródłem inspiracji do przyjęcia takiego podejścia była praktyka stosowana w znakowaniu morfosyntaktycznym: zestaw znaczników morfosyntaktycznych oraz wytyczne znakowania korpusu za ich pomocą opracowywane są na postawie rozważań na temat struktur obecnych w języku, przewidywanych zastosowań tekstu oznakowanego oraz zdolności człowieka do dokonania niezbędnych rozróżnień w praktyce (Przepiórkowski i Woliński, 2003; Babarczy i inni, 2005; van Halteren, 1999), natomiast trudno jest spotkać decyzje podyktowane możliwościami i ograniczeniami konkretnego tagera.

Prawie wszystkie prace dotyczące znakowania fraz w językach słowiańskich uwzględniają **frazy rzeczownikowe** (grupy nominalne, ang. *noun phrases*, *NP*). Warto przytoczyć encyklopedyczną definicję frazy rzeczownikowej (Karolak, 1999a):

> Struktura składniowa o dowolnej liczbie składników, której ośrodkiem jest rzeczownik lub zaimek rzeczowny. Minimalna prosta grupa nominalna jest tożsama z rzeczownikiem. W złożonej grupie nominalnej wyróżnia się grupę nominalną *nuklearną*, czyli tę jej część, która pozostaje po wyodrębnieniu wykładników kwantyfikatorów referencyjnych i liczbowych, oraz grupę determinantów obejmującą te właśnie wykładniki. Zakres elementów zaliczanych do kategorii determinantów, ich podział na podklasy syntaktyczne oparty na regułach kookurencji (współwystępowania) i substytucji oraz terminy odnoszące się do nich nie są w tradycji gramatycznej w pełni ustabilizowane. (. . .)

W dalszych rozważaniach istotne będzie też pojęcia **nadrzędnika** oraz **podrzędnika** (Derwojedowa i inni, 2005, s. 441):

> **Składnik nadrzędny**, zwany **nadrzędnikiem** konstrukcji składniowej tworzącej wypowiedzenie pojedyncze, to ten jej człon syntaktyczny, który pełni funkcję członu określanego, a w wypadku redukcji tekstu reprezentuje całą konstrukcję z zachowaniem ogólnego sensu (jest jej reprezentantem).
>
> **Składnik podrzędny**, zwany **podrzędnikiem** konstrukcji składniowej tworzącej wypowiedzenie pojedyncze, to ten jej człon syntaktyczny, który pełni funkcję członu określającego (określenia), a w wypadku redukcji tekstu może być reprezentowany przez człon nadrzędny bez szkody dla ogólnego sensu konstrukcji.

Poprzez **składnik** należy rozumieć najmniejszy element wypowiedzenia, który pełni w nim samodzielną funkcję (Derwojedowa i inni, 2005, s. 436). Składnik może być realizowany poprzez pojedynczy segment lub ciąg kilku segmentów.

Przykładowo, w zdaniu (4.6) występuje fraza rzeczonikowa bardzo stary dom, której nadrzędnikiem jest rzeczownik dom. Fraza ta może być bowiem zredukowana do samego składnika dom z zachowaniem ogólnego sensu. Przymiotnik stary jest określeniem (modyfikatorem) rzeczownika dom, natomiast przysłówek bardzo określa składnik stary.

(4.6) Widzę bardzo stary dom.

W literaturze poświęconej znakowaniu fraz słowiańskich stosuje się różne definicje fraz rzeczownikowych. Cechą wspólną jest założenie, że nadrzędnikiem frazy jest rzeczownik bądź inny składnik pełniący jego funkcję. Drugim powszechnie stosowanym założeniem jest nieprzekraczanie granic zdań cząstkowych, tj. jeśli fraza rzeczownikowa zawiera określenie będące zdaniem podrzędnym, to określenie to rozpatrywane jest z osobna. Przykład (4.7) przedstawia konstrukcję, którą tradycyjnie można by scharakteryzować jako jedną frazę rzeczownikową. Nawiasami zaznaczono granice fraz rzeczownikowych (NP), które zostałyby wyróżnione przyjmując założenie o nieprzekraczaniu zdań cząstkowych (w zależności od dalszych założeń, składnik który można uznać za frazę rzeczownikową lub nie). Jak widać, zdanie podrzędne zostało poddane osobnej analizie i wyodrębniono w nim frazę samolotem, chociaż z tradycyjnego punktu widzenia cały przykład (4.7) stanowi frazę rzeczownikową.

(4.7) $[_{NP}$ Człowiek], który leciał $[_{NP}$ samolotem]

Istotne różnice w definicjach występują natomiast odnośnie rodzajów podrzędników, które włączane są w granice fraz. Definicja zaproponowana w pracy (Nenadić i Vitas, 1998b) ograniczona jest do fraz uzgodnionych co do liczby, rodzaju i przypadka. Oznacza to, że frazy mają nadrzędnik rzeczownikowy, a poza nim dopuszczone są określenia rzeczownika z nim uzgodnione, tj. przymiotniki i wyrazy pełniące ich funkcję. Oprócz tego dopuszczone są nieodmienne określenia tych przymiotników (przysłówki i partykuły).

Analogiczna definicja stosowana jest w pracy Vučković i inni (2008) dla języka chorwackiego. Wyjątkiem od wymogu uzgodnienia jest tutaj dopuszczenie **apozycji** (dopowiedzeń), tj. konstrukcji składających się z dwóch lub więcej fraz rzeczownikowych pod rząd (apozycją jest np. fraza książka „Potop" i pan profesor Jan Miodek).

Ograniczenie fraz do fragmentów wykazujących uzgodnienie prowadzi do rozbicia wielu nazw własnych i utartych konstrukcji na mniejsze frazy, por. (4.8) i (4.9). Przykładowo, składnik Umarłych jest określeniem dopełniaczowym składnika Tybetańska Księga, a zatem włączenie go we frazę naruszałoby uzgodnienie. Z praktycznego punktu widzenia może to być wadą, jeśli od fraz rzeczownikowych oczekujemy, że będą odpowiadać nazwom bytów opisywanych przez tekst. Z drugiej strony, podejście takie można uznać za bliskie oryginalnym wytycznym Abneya (1991) — wytyczne te nakazują podział fraz na każdym przyimku; konstrukcje anglojęzyczne realizowane przez wyrażenia przyimkowe są po polsku często realizowane za pomocą przypadka zależnego (tj. innego niż mianownik). Gdyby przetłumaczyć wyrażenie (4.8) na język angielski, otrzymalibyśmy wyrażenie, które, wraz z oznakowaniem zgodnie z wytycznymi Abneya, przyjęłoby realizację (4.10); zapis PP oznacza *frazę przyimkową* (ang. *prepositional*

phrase). Jak widać, mamy do czynienia z rozbiciem nazwy własnej na dwie frazy w sposób analogiczny.

(4.8) [$_{NP}$ Tybetańska Księga][$_{NP}$ Umarłych]
(4.9) [$_{NP}$ Ministerstwo][$_{NP}$ Finansów]
(4.10) [$_{NP}$ Tibetan Book] [$_{PP}$ of the Dead],

Powyższe sformułowanie serbsko-chorwackiej frazy rzeczownikowej (Nenadić i Vitas, 1998b) zostało później rozszerzone o możliwość włączenia podrzędników kilku rodzajów (Nenadić i Vitas, 1998a). Po pierwsze, dopuszczono określenia dopełniaczowe — a zatem możliwe byłoby włącznie składnika Umarłych z przykładu (4.8). Dopuszczono również możliwość włączenia całych fraz przyimkowych. Nenadić i Vitas (1998a) podają przykłady (4.11) i (4.12).

(4.11) *datoteka* *na* *disku*
plik na dysku
(4.12) *upis* *na* *disk*
zapis na dysk

W dalszej kolejności (Nenadić, 2000) rozszerzono powyższe ujęcie o **szeregowe frazy rzeczownikowe** (ang. *coordination of noun phrases*, CNP). Frazy szeregowe zdefiniowano jako ciąg fraz rzeczownikowych połączonych spójnikami bądź przecinkiem.

Chorwackie frazy rzeczownikowe z definicji Vučković i inni (2008) także poddane zostały późniejszej rozbudowie[2] (Vučković, 2009; Vučković i inni, 2010). Dotychczasową definicję frazy uzgodnionej zachowano pod hasłem NP, natomiast wprowadzono nową frazę, nazwaną *AT* (frazę z określeniem dopełniaczowym; chorw. *atributska sintagma*, ang. *attribute phrase*). Frazy AT składają się z dwóch członów będących (uzgodnionymi) frazami rzeczownikowymi, gdzie drugi człon jest frazą w dopełniaczu. Definicja taka pozwoliłaby na opisanie przykładów (4.8) i (4.9) jako całych fraz AT.

Dotąd przytaczane prace zakładały definicję frazy „od dołu", tj. zaczynając od składnika będącego nadrzędnikiem, po czym wyliczając typy określeń, które dopuszczamy. Praca Czechów (Grác i inni, 2010) określa ramy przyjętych fraz rzeczownikowych „od góry", tj. podane jest kilka kryteriów, kiedy frazę należy rozbić, natomiast punktem wyjścia są największe frazy (tj. największe poddrzewa odpowiadające frazom rzeczownikowym w drzewie rozbioru składniowego). Artykuł przytacza dwa takie kryteria: wspomnianą wyżej zasadę analizy każdego zdania cząstkowego w izolacji oraz zasadę, że przyimki zawsze dzielą frazy. Jak wspomnieliśmy, ta ostatnia zasada wywodzi się bezpośrednio od wytycznych Abneya (1991). Umotywowana jest obserwacją, że rozstrzygnięcie, czy fraza przyimkowa należy do frazy rzeczownikowej, czy też do całej konstrukcji czasownikowej, jest niełatwe i często wymaga wiedzy o charakterze semantycznym (tj. nie wystarczy tu informacja czysto składniowa). Rozstrzygnięcie to uznano za zbyt trudne, w związku z czym każde wystąpienie przyimka rozpoczyna nową frazę. Problem ten ilustrują to przykłady (4.13) i (4.14), gdzie zaznaczono granice całych fraz NP i PP, zakładając że nie rezygnujemy z tego rozstrzygnięcia. Jeśli z niego zrezygnujemy, zdanie (4.13) uzyska strukturę (4.15), tj. taką samą, jak zdanie (4.14).

[2] Sądząc po braku wzajemnych cytowań, można wnioskować, że przytoczone prace nad znakowaniem fraz były prowadzone niezależnie dla języka serbsko-chorwackiego i języka chorwackiego.

Identyczną decyzję podjęto również w przypadku fraz chorwackich (Vučković, 2009), bułgarskich (Osenova, 2002), a także polskich w ujęciu NKJP (Głowińska, 2012).

(4.13) Jem [$_{NP}$ pizzę z pieczarkami]
(4.14) Jem [$_{NP}$ pizzę][$_{PP}$ z przyjaciółmi]
(4.15) Jem [$_{NP}$ pizzę][$_{PP}$ z pieczarkami]

Warto dodać, że takie rozwiązanie ma tę praktyczną wadę, że nazwy własne i inne utarte konstrukcje mogą ulec rozbiciu na dwie lub więcej fraz, por. (4.16) i (4.17).

(4.16) [$_{NP}$ Frankfurt][$_{PP}$ nad Menem]
(4.17) [$_{NP}$ wyciskanie sztangi][$_{PP}$ w leżeniu][$_{PP}$ na ławce poziomej]

Prace o podobnym charakterze przeprowadzono również dla języka bułgarskiego (Osenova, 2002; Osenova i Simov, 2003). Należy jednak podkreślić, że bułgarski nie jest typowym językiem słowiańskim, w szczególności bułgarskie rzeczowniki i przymiotniki nie odmieniają się przez przypadki (Sussex i Cubberley, 2006, s. 249), a frazom rzeczownikowym można przypisać kategorię określoności, nieobecną w języku polskim czy czeskim (Sussex i Cubberley, 2006, s. 258).

Marciniak (2010) opisuje problem rozpoznawania polskich fraz (rzeczownikowych, czasownikowych, przymiotnikowych i liczebnikowych) pod kątem konkretnego zastosowania — znakowania korpusu składającego się z transkrypcji rozmów klientów z infolinią Zarządu Transportu Miejskiego w Warszawie. Użyte definicje fraz uwzględniają specyfikę języka mówionego oraz dziedziny, w tym częstą obecność nazw ulic i numerów tramwajów, liczne wtrącenia, powtórzenia, urwane wypowiedzi, a także naruszenia uzgodnień gramatycznych. Frazy rzeczownikowe zostały podzielone na trzy kategorie: zaimki osobowe, nazwy własne oraz frazy składające się z przymiotników i rzeczowników. W ramach ostatniej kategorii dopuszczono zarówno podrzędniki przymiotnikowe, ale także podrzędniki realizowane przez rzeczowniki w dopełniaczu. Taki podrzędnik w dopełniaczu może również mieć swoje określenia przymiotnikowe, por. przykład (4.18).

(4.18) [$_{NP}$ pojazdach komunikacji miejskiej]

Z przytaczanych przez Marciniak (2010) oraz (Mykowiecka i inni, 2007) przykładów można wnioskować, że frazy rzeczownikowe i tu rozbijane są na przyimkach (choć niewykluczone, że w przypadku nazw własnych obowiązują inne zasady).

Frazy, których nadrzędnikiem jest liczebnik główny mogą być traktowane jako samodzielna **fraza liczebnikowa**, np. chorwackie frazy *M* (Vučković, 2009), frazy liczebnikowe z NKJP (Głowińska, 2012). Można je też dla uproszczenia traktować jako jeden z przypadków fraz rzeczownikowych, por. Osenova (2002). Frazy liczebnikowe mogą wykazywać dużą zależność od dziedziny i rodzaju znakowanego tekstu. Marciniak (2010) opisuje kilka problemów związanych z rozpoznawaniem fraz liczebnikowych we wspomnianym korpusie dialogów; jednym z problemów jest tu wielość możliwych sposobów wyrażenia fraz liczebnikowych określających numery autobusów.

Oprócz fraz rzeczownikowych, często znakowane są wspomniane już kilkakrotnie **frazy przyimkowe**. Zakładając, że nakreślono już kształt fraz rzeczownikowych, definicja fraz przyimkowych nie przysparza większych problemów: definiowane są one jako następujące po sobie przyimek i fraza rzeczownikowa (np. na ławce poziomej, przez

odpowiedzialne za to osoby). W pracy Grác i inni (2010) przyjęto inne ciekawe rozwiązanie: frazy rzeczownikowe oraz przyimkowe dla uproszczenia znakowane są zbiorowo jako „frazy *N/P*". Rozwiązanie takie ma sens ze względu na prostą definicję frazy przyimkowej: w prosty sposób można automatycznie oddzielić jedne od drugich, a w razie potrzeby również odciąć przyimki rozpoczynające frazy przyimkowe i uzyskać z nich frazy rzeczownikowe.

Duża część prac dotycząca znakowania fraz w językach słowiańskich uwzględnia **frazy czasownikowe** (grupy werbalne, ang. *verb phrases*, *VP*). Karolak (1999b) definiuje frazę czasownikową jako strukturę składniową, „której ośrodkiem jest czasownik w formie finitywnej lub niefinitywnej (bezokolicznik, imiesłów)". Według niektórych tradycji gramatycznych do frazy czasownikowej zalicza się podmiot czasownika, według innych — nie; włączanie do frazy dopełnień nie budzi zaś większych kontrowersji (Karolak, 1999b).

Standardową praktyką przy znakowaniu (płaskich) fraz czasownikowych wydaje się znakowanie jedynie elementów należących do samego czasownika (prostego lub złożonego), nie włączając w granice frazy ani podmiotu, ani dopełnień czasownika (Abney, 1996a). Praktyka ta jest o tyle naturalna, że podmiot i dopełnienia czasownika są na ogół już oznaczone jako frazy rzeczownikowe bądź przyimkowe — a zatem jako (płaskie) frazy czasownikowe znakowane są pozostałe elementy całej frazy czasownikowej rozumianej jako największe poddrzewo z etykietą VP; por. drzewo składniowe (4.1) ze strony 63 i odpowiadające mu oznakowanie frazami (4.3).

Powyższe założenie przyjęto w pracy Vučković (2009) oraz Vučković i inni (2010) odnośnie chorwackich fraz czasownikowych. Nadrzędnikiem frazy czasownikowej jest czasownik. Oprócz niego, do frazy można włączyć następujące składniki:

1. jeden lub dwa czasowniki posiłkowe (w zależności od czasu),
2. zaimek zwrotny se (pol. się),
3. negację ne (nie),
4. bezokolicznik w roli podrzędnika.

Składniki te mogą występować w różnych kombinacjach, tworząc różne czasy, tryby, a także stronę czynną lub bierną.

Podobne ujęcie dla języka serbsko-chorwackiego przedstawiają Nenadić i inni (1999). Niestety, praca nie podaje choćby przybliżonych definicji rozpoznawanych fraz. Pada stwierdzenie, że frazy są albo proste (tj. złożone jedynie z formy czasownikowej), albo złożone — składające się z kilku jednostek, „zazwyczaj czasownika posiłkowego i czasownika głównego". Sądząc po przytoczonych przykładach, wyróżniane są podobne konstrukcje jak w przypadku prac Vučković (2009); Vučković i inni (2010). Autorzy zwracają za to uwagę na istotny szczegół: pewna ilość fraz czasownikowych jest nieciągła. Oznacza to, że w pełni poprawny ich opis wykracza poza możliwości zadania znakowania frazami (rozumianego jako znakowania ciągłych i rozłącznych ciągów segmentów). W pracy podano m.in. przykład (4.19). Fraza je dolazila (przyszła) jest formą jednego z serbsko-chorwackich czasów przeszłych (dosł. jest przyszła).

(4.19) *bog* *cyega* $[_{VP}$ *je* **1**$]$ *gospodja* $[_{VP}$ *dolazila* **1**$]$
dla czego $[_{VP}$ (jest) **1**$]$ pani $[_{VP}$ przyszła **1**$]$
'dlaczego pani przyszła'

W pracy (Mráková i Sedláček, 2003) opisano problem rozpoznawania fraz czasow-

nikowych w języku czeskim. Nie podano wprawdzie definicji takich fraz, ale można znaleźć informację, że do fraz czasownikowych można włączyć różne grupy czasownikowe, a także formy zaimka zwrotnego se oraz si. Autorzy również zauważają, że zdarzają się nieciągłe frazy czasownikowe.

Problem nieciągłych fraz występuje również w języku polskim. Przypadki takie nie są wprawdzie częste, choć nie można ich ograniczyć do fraz czasownikowych. Przykład (4.20) pokazuje zdanie zawierające nieciągłą frazę czasownikową, przykład (4.21) — rzeczownikową (fragmenty tej samej nieciągłej frazy oznakowano cyferką 1). Warto też dodać, że konieczność wprowadzenia nieciągłych fraz może wystąpić jako artefakt płytkiej analizy składniowej. Problem ten występuje zarówno w NKJP, jak i KPWr (choć nie zawsze w tych samych sytuacjach). Głowińska (2012) podaje przykładowe zdanie (4.22). Można zauważyć, że wycięcie z frazy na najwyższą od lat kumulację fragmentu od lat jest wymuszone przez przyjętą w NKJP zasadę, że przyimki stanowią granice rozpoznawanych fraz. Gdyby nie ta zasada, fragment na najwyższą od lat kumulację można by oznakować jako ciągłą frazę przyimkową.

(4.20) $[_{VP}$ Tańczyć **1**] to $[_{NP}$ my] $[_{VP}$ lubimy **1**]!
(4.21) $[_{NP}$ Wino **1**] $[_{VP}$ piliśmy] $[_{NP}$ czerwone **1**].
(4.22) $[_{VP}$ Skusili się] $[_{PP}$ na najwyższą **1**] $[_{PP}$ od lat **2**] $[_{PP}$ kumulację **1**].

We wspomnianych transkrypcjach rozmów telefonicznych znakowane są także frazy czasownikowe (Marciniak, 2010). Frazy te są pojedynczymi czasownikami bądź też formami analitycznymi (np. będę potrzebował, będzie trzeba). Co ciekawe, przykład z pracy (Mykowiecka i inni, 2007) sugeruje, że zaimek się nie jest włączany do fraz czasownikowych — przesiąść się rozbity jest na frazę czasownikową przesiąść i „frazę partykułową" się.

Niekiedy znakuje się także **frazy przymiotnikowe** (ang. *adjective phrases*, AdjP bądź AP). Są to frazy, których nadrzędnikami są przymiotniki lub wyrazy pełniące ich funkcje, np. godny uwagi (Grzegorczykowa, 2006, s. 22). Jako że frazy przymiotnikowe stanowią najczęściej podrzędniki większej frazy rzeczownikowej — por. przykłady (4.1) i (4.5) — typowym podejściem jest znakowanie jedynie tych fraz przymiotnikowych, które nie należą do żadnej frazy rzeczownikowej (Osenova i Simov, 2003). Podejście to ilustrują przykłady (4.23) i (4.24).

(4.23) $[_{NP}$ Pies] $[_{VP}$ jest] $[_{AdjP}$ głodny].
(4.24) $[_{NP}$ Głodny pies] $[_{VP}$ szczeka].

4.2.2. Frazy w KPWr

Frazy w korpusie KPWr zdefiniowano kierując się następującymi postulatami:

1. przydatność praktyczna, w szczególności w zadaniu wydobywania informacji,
2. zgodność z tradycyjnym opisem składniowym, a także z praktyką stosowaną przy płytkiej analizie składniowej języków słowiańskich,
3. łatwość rozpoznawania fraz przez człowieka i podejmowania niezbędnych rozstrzygnięć.

Postulaty te są w pewnym stopniu sprzeczne, dlatego opracowane wytyczne są wynikiem szeregu kompromisów (z podobnym problemem zetknęli się Grác i inni, 2010 próbujący ustalić priorytety przy znakowaniu frazami czeskiego korpusu).

Dodatkowy wymóg związany był z następnym etapem znakowania składniowego, który zaplanowano w ramach projektu SyNaT. Etapem tym jest znakowanie wybranych relacji składniowych pomiędzy frazami, w szczególności relacji *podmiot* i *dopełnienie* (są to tzw. relacje predykatowo-argumentowe). Relacje te wiążą frazy czasownikowe z frazami rzeczownikowymi, przyimkowymi, a w niektórych sytuacjach — z frazami przymiotnikowymi. Semantyką tych relacji jest wskazanie frazy, która jest podmiotem danego czasownika lub jego dopełnieniem. Przykład (4.25) przedstawia oznakowanie takimi relacjami prostego zdania. Więcej szczegółów na ten temat, a także wytyczne znakowania relacjami można znaleźć w pracy Radziszewski i inni (2012). Sam problem znakowania relacji między frazami leży poza zakresem tej rozprawy i nie będzie tu dalej rozważany. Istotne tutaj jest jednak to, że decyzja o znakowaniu tych relacji miała wpływ na przyjęty kształt fraz. Przede wszystkim, istotne było to, aby frazy stanowiące podmiot bądź dopełnienie czasownika były w miarę możliwości oznakowane jako całość. Było to głównym powodem podjęcia decyzji, że w odróżnieniu od większości przytaczanych w poprzednim punkcie prac, znakowane jako całość są także frazy rzeczownikowe zawierające w sobie frazy przyimkowe (innymi słowy, wytyczne KPWr nie nakazują „ucinania" fraz na każdym wystąpieniu przyimka, nakazują zaś podjęcie decyzji, czy dana fraza przyimkowa należy do innej większej frazy, czy nie).

(4.25) PODMIOT DOPEŁNIENIE
Kwiat wiśni **symbolizuje** **ludność** regionu .

Jedną z pierwszych decyzji było wprowadzenie do modelu dwóch zbiorów fraz, odpowiadających dwóm alternatywnym spojrzeniom na płytką analizę składniową zdania:

1. Frazy zdefiniowane „oddolnie" na podstawie lokalnych związków składniowych. Do tego zbioru należą jedynie **frazy uzgodnione**.
2. Frazy zdefiniowane „odgórnie", na podstawie związków predykatowo-argumentowych panujących w zdaniu. Definicje opierają się na wyliczeniu sytuacji, gdzie elementu nie można włączyć do frazy — w pozostałych wypadkach należy włączyć największe poddrzewo rozbioru składniowego odpowiadające frazie danego typu (wytyczne wskazują też pewne wyjątki, a także wskazówki uściślające rozstrzygnięcia w praktycznych sytuacjach). Do tej grupy należą **frazy rzeczownikowe**, **frazy przymiotnikowe** oraz **frazy czasownikowe**.

Frazy uzgodnione (AgP) zostały bezpośrednio zainspirowane uzgodnionymi frazami rzeczownikowymi rozpatrywanych przez Nenadić i Vitas (1998b) oraz Vučković i inni (2008). Frazy AgP muszą być zatem uzgodnione co do liczby, rodzaju i przypadka. Podobnie jak w cytowanych pracach, dopuszczamy także elementy nieodmienne, które określają rzeczowniki, przymiotniki, bądź inne nieodmienne określenia, por. przykład (4.26). Do frazy AgP można także włączyć apozycje, jeśli nie naruszają one uzgodnienia, por. (4.27).

(4.26) [$_{AgP}$ wcześniej omawiany projekt]
(4.27) [$_{AgP}$ pan wicemarszałek Jerzy Szmajdziński]

W odróżnieniu od cytowanych prac, definicja AgP przyjęta w KPWr obejmuje zarówno uzgodnione frazy rzeczownikowe, jak i uzgodnione frazy przymiotnikowe (o ile nie stanowią one części większego AgP), por. (4.28).

(4.28) [$_{AgP}$ godny][$_{AgP}$ uwagi]

Ze względu na wymóg uzgodnienia, szeregowe frazy rzeczownikowe rozbijane są na osobne frazy AgP (szeregi rzeczownikowe nie gwarantują uzgodnienia). Jako że szeregi przymiotników określają i tak jedną frazę rzeczownikową, całe takie szeregi nie naruszają uzgodnienia. Dlatego też szeregi przymiotników włączane są do AgP, por. (4.29) i (4.30).

(4.29) [$_{AgP}$ ciekawa i trafna uwaga]
(4.30) [$_{AgP}$ ciekawa uwaga] i [$_{AgP}$ riposta]

Dla uproszczenia, rząd przyimka traktowany tutaj jest na równi z uzgodnieniem, a zatem przyimki włączane są także do fraz AgP. Powoduje to, że proste frazy przyimkowe też stanowią AgP. Ilustrują to przykłady (4.31) i (4.32). Zabieg ten jest podobny do złączenia fraz NP i PP przyjętego w pracy Grác i inni (2010).

(4.31) [$_{AgP}$ przez rzekę] pływa [$_{AgP}$ prom]
(4.32) [$_{AgP}$ miasto położone] [$_{AgP}$ w województwie dolnośląskim] [$_{AgP}$ nad Odrą]

Wyjątkiem od wymogu uzgodnienia jest włączenie złożonych liczebników porządkowych w większe frazy, nawet jeśli podrzędniki tych liczebników naruszają uzgodnienie. Nadrzędnik takiego liczebnika porządkowego nie może jednak naruszać uzgodnienia, por. przykłady (4.33) i (4.34).

(4.33) [$_{AgP}$ rok tysiąc dziewięćset dziewięćdziesiąty piąty]
(4.34) [$_{AgP}$ dwudziesty piąty] [$_{AgP}$ stycznia]

W korpusie znakowane są także nieciągłe frazy AgP. Przedstawiają to poniższe przykłady (cyferka 1 oznacza, że fragmenty nią oznakowane należą do tej samej frazy).

(4.35) [$_{AgP}$ zapomniane **1**] [$_{AgP}$ przez nas] [$_{AgP}$ książki **1**]
(4.36) [$_{AgP}$ konsekwencje szybko **1**] [$_{AgP}$ przez sąd] [$_{AgP}$ uznane **1**]

Frazy rzeczownikowe (NP) to frazy, które w strukturze predykatowo-argumentowej mogą pełnić rolę argumentu (podmiotu bądź dopełnienia), bądź też okolicznika. Są to maksymalne frazy rzeczownikowe lub liczebnikowe bez zdań podrzędnych (zdania takie analizowane są z osobna w poszukiwaniu fraz). Podobnie jak w przypadku fraz AgP, nie dokonujemy rozróżnienia między prawdziwymi frazami rzeczownikowymi, a frazami przyimkowymi. Oba typy fraz są znakowane dla uproszczenia jako NP. Ilustrują to przykłady (4.37), (4.38). Przykład (4.39) pokazuje NP będącą okolicznikiem (przez okno).

(4.37) [$_{NP}$ przez rzekę] [$_{VP}$ pływa] [$_{NP}$ prom]
(4.38) [$_{NP}$ w mieście położonym w województwie dolnośląskim nad Odrą]
(4.39) [$_{VP}$ wyrzucił][$_{NP}$ spaloną jajecznicę][$_{NP}$ przez okno]

Nadrzędnikiem frazy rzeczownikowej może być rzeczownik, liczebnik, zaimek pełniący funkcję rzeczownika, odsłownik (gerundium), a także przymiotnik pełniący wyraźną funkcję rzeczownika, por. (4.40) i (4.41).

(4.40) [$_{NP}$ niepalący] [$_{VP}$ stanowią] [$_{NP}$ większość społeczeństwa]
(4.41) [$_{NP}$ palenie] [$_{VP}$ szkodzi] [$_{NP}$ zdrowiu]

Fraza rzeczownikowa w KPWr powinna mieć jeden nadrzędnik. Jeśli mamy do czynienia z frazą szeregową na poziomie nadrzędnym, szeregi takie są rozbijane. Motywacją do podjęcia tej decyzji było przybliżenie ram fraz rzeczownikowych do nazw własnych i utartych konstrukcji występujących w tekście (konstrukcje takie mają znaczenie praktyczne m.in. w wydobywaniu informacji z tekstu). Przykład (4.42) przedstawia szereg na poziomie nadrzędnym, zaś przykład (4.43) — na poziomie podrzędnym (elementy szeregu mają wspólny nadrzędnik Ministerstwo). Nadrzędniki fraz wyróżniono tłustym drukiem.

(4.42) [$_{NP}$ **Minister** Finansów] i [$_{NP}$ jego **podwładni**]
(4.43) [$_{NP}$ **Ministerstwo** Nauki i Szkolnictwa Wyższego]

Jak wcześniej wspomnieliśmy, zasady znakowania nakazują podjęcia niezbędnego wysiłku, aby ustalić, czy fraza przyimkowa należy do większej frazy, czy nie. Ilustrują to przykłady (4.44) i (4.45).

(4.44) [$_{VP}$ wróciła] [$_{NP}$ do domu] [$_{NP}$ z podbitym okiem]
(4.45) [$_{VP}$ wróciła] [$_{NP}$ do domu z odrapanym dachem]

Należy się liczyć z pojawieniem się sytuacji, gdzie obie decyzje będą wydawać się jednakowo sensowne. W takich sytuacjach wytyczne nakazują podjąć arbitralną decyzję o rozdzieleniu fraz.

Wytyczne dopuszczają także możliwość oznakowania nieciągłych fraz rzeczownikowych. Sytuacje te są jednak stosunkowo rzadkie.

Frazy przymiotnikowe (AdjP) rozumiane są jako maksymalne frazy przymiotnikowe, zdefiniowane w sposób analogiczny do NP. W tym wypadku nadrzędnikiem jest przymiotnik lub imiesłów przymiotny. Frazy takie znakujemy jedynie w wypadku, gdy nie są one częścią większej frazy NP. Frazy te są najczęściej fragmentem konstrukcji predykatywnych, np. (4.46). Jak wspomniano wyżej, przymiotniki o użyciu wyraźnie rzeczownikowym tworzą frazy rzeczownikowe, a nie przymiotnikowe, np. (4.47). Zdarzają się frazy nieciągłe, np. (4.48).

(4.46) [$_{NP}$ Książki te] [$_{VP}$ są] [$_{AdjP}$ nam wszystkim znane].
(4.47) [$_{NP}$ Młodzi] [$_{VP}$ witali] [$_{NP}$ gości].
(4.48) [$_{NP}$ Dwadzieścia pięć kwitnących okazów bluszczu] [$_{AdjP}$ uznanych **1**] [$_{VP}$ zostało] [$_{AdjP}$ za pomnik przyrody **1**].

Frazy czasownikowe (VP) to frazy zawierające czasownik główny w roli nadrzędnika. Ściślej rzecz biorąc, nadrzędnik ten może być:

1. formą osobową czasownika,
2. formą bezosobową zakończoną na -no lub -to,
3. imiesłowem czasownikowym zakończonym na -wszy, -łszy lub -ąc,
4. bezokolicznikiem, o ile nie jest on podrzędnikiem większej frazy czasownikowej,
5. predykatywem (np. widać).

Gerundia czy imiesłowy przymiotnikowe nie są uznawane za ośrodki fraz czasownikowych (mogą być ośrodkami fraz AgP, AdjP i NP).

Do fraz czasownikowych zaliczane są podrzędniki będące czasownikami. Jako całość traktujemy więc frazy typu (4.49), (4.50). Nie są zaś zaliczane doń podmioty, dopełnienia ani okoliczniki wyrażane frazami NP czy AdjP — por. (4.51). W przypadku

orzeczenia imiennego jako frazę czasownikową znakujemy tylko łącznik — orzecznik będzie już oznakowany jako NP lub AdjP; por. (4.52).

(4.49) [$_{VP}$ chcę śpiewać]
(4.50) [$_{VP}$ boję się iść]
(4.51) [$_{NP}$ Jan][$_{VP}$ przyniósł] [$_{NP}$ we wtorek] [$_{NP}$ do domu] [$_{NP}$ karabin].
(4.52) [$_{NP}$ Ona] [$_{VP}$ jest] [$_{AdjP}$ piękna].

Do fraz czasownikowych można włączyć także określenia o charakterze przysłówkowym, jeśli w wyraźny sposób należą one do frazy. Pokazuje to przykład (4.53). Nie włączamy do fraz przysłówków, które są wtrąceniami albo pełnią funkcję łączników międzyzdaniowych, np. (4.54). Znakowane też są frazy nieciągłe, np. (4.55).

(4.53) [$_{VP}$ chcę głośno śpiewać]
(4.54) Prawdopodobnie [$_{VP}$ uciekł] [$_{NP}$ do lasu].
(4.55) [$_{VP}$ Muszę szybko **1**] [$_{NP}$ ją] [$_{VP}$ odnaleźć **1**]!

W wytycznych KPWr obowiązuje nadrzędna zasada, że zdania cząstkowe znakowane są z osobna. Jeśli zdanie podrzędne zawiera zaimek względny pełniący w nim funkcję podmiotu, zaimek ten znakowany jest jako fraza rzeczownikowa, choćby był to zaimek przymiotnikowy (np. który, jaki). Zaimek względny może się także łączyć z przyimkiem, wtedy w całości tworzy frazę, np. (4.58).

(4.56) [$_{VP}$ Wiedziałem], [$_{NP}$ co] [$_{VP}$ zamierzali zrobić].
(4.57) [$_{VP}$ Szanuję] [$_{NP}$ ludzi], [$_{NP}$ którzy] [$_{VP}$ mają] [$_{NP}$ cel] [$_{NP}$ w życiu].
(4.58) [$_{NP}$ Dom], [$_{NP}$ do którego] [$_{VP}$ chce się wracać].

Chociaż wytyczne KPWr nakazują znakować nieciągłe frazy wszystkich rozpatrywanych typów, nieciągłości pojawiają się stosunkowo rzadko. Zakres tej rozprawy ograniczamy do rozpoznawania ciągłych fraz, zgodnie z definicją znakowania fraz przywołaną na początku tego rozdziału (strona 63). Będziemy w tym celu stosować następujące uproszczenie: każdy ciągły fragment nieciągłej frazy traktować będziemy jako osobną frazę tego samego typu. Uproszczenie to ma pewne wady (np. prowadzi do oznakowania niektórych przysłówków jako samodzielnych fraz czasownikowych), jednak z praktycznego punktu widzenia jest bardzo wygodne. Co więcej, jeśli opracowana metoda rozpoznawania fraz prawidłowo oznakuje takie ciągłe fragmenty nieciągłych fraz, istnieje możliwość opracowania kolejnego modułu odpowiedzialnego za rozpoznawanie, które fragmenty stanowią w rzeczywistości części większych fraz.

4.2.3. Frazy w NKJP

Znakowanie składniowe w NKJP zostało podzielone na dwa poziomy: poziom **słów składniowych** oraz poziom **grup składniowych** (Głowińska, 2012).

Słowa składniowe to poziom pośredni między drobnym podziałem tekstu na segmenty (strategia segmentacji przyjęta w NKJP została omówiona w punkcie 2.2), a poziomem właściwych grup składniowych. Objawia się to m.in. tym, że każdy segment występujący w korpusie należy do jakiegoś słowa składniowego (większość słów składniowych jest jednosegmentowa, choć występują też słowa nawet siedmiosegmentowe). Słowa składniowe pozwalają na złączenie niektórych ciągów segmentów w jednostki, dzięki czemu możliwe jest opisanie m.in. tradycyjnie rozumianych form czasownikowych

(np. ciąg segmentów robiła | by | ś czy będę | robiła), nieciągłych spójników, jednostek wielowyrazowych (np. po | ludzku, Bachleda | - | Curuś). Zdarzają się nieciągłe słowa składniowe, por. przykład (4.59) (Głowińska, 2012).

(4.59) [$_{Verbfin}$ niech **1**] tutaj [$_{Verbfin}$ przyjdzie **1**]

Znakowanie słów składniowych wiąże się nie tylko z wyznaczeniem ich granic, lecz również z przypisaniem każdemu słowu tagu i lematu. Tagi przypisywane słowom składniowym pochodzą ze specjalnie zaprojektowanego dla tego poziomu tagsetu. Dzięki temu, że opisywane są jednostki większe niż pojedyncze segmenty, w tagsecie można było uwzględnić kategorie gramatyczne właściwe takim jednostkom, np. czas i tryb czasownika. Obecne w tym tagsecie klasy gramatyczne są bliższe tradycyjnie rozumianym częściom mowy, a atrybuty bardziej przypominają tradycyjnie przypisywane tym częściom kategorie gramatyczne (Głowińska, 2012).

Granice niektórych słów składniowych można uznać ze ciekawe przypadki (płaskich) fraz. W szczególności można to powiedzieć o słowach, którym przypisano tagi należące do typowo czasownikowych klas gramatycznych. Za takie możemy uznać klasy `Verbfin` (formy osobowe), `Winien` (leksemy winien, powinien), `Imps` (formy bezosobowe), `Inf` (bezokoliczniki) oraz `Pred` (predykatywy). Jeśli słowa składniowe wszystkich tych klas złączymy w jedną grupę, uzyskujemy definicję frazy czasownikowej zbliżoną do stosowanej dla języka chorwackiego (por. punkt poprzedni). Tak rozumiana fraza czasownikowa wydaje się prostsza od ujęcia stosowanego w KPWr, gdyż nie pozwala na włączenie przysłówków do VP.

Zdarza się, że jeden segment należy jednocześnie do większej liczby słów składniowych. Jeśli słowa składniowe potraktować jako frazy, powoduje to, że mamy do czynienia z problemem nakładających się na siebie fraz (co, podobnie jak problem fraz nieciągłych, stoi w sprzeczności z przyjętą przez definicją znakowania fraz). Ilustruje to przykład (4.60).

(4.60) [*bał* *się*] *odezwać*
bał [*się* *odezwać*]

Rozróżnienie między słowami a grupami składniowymi nie zawsze jest oczywiste. Nadrzędnym kryterium rozróżnienia jest oparcie definicji słów na konstrukcjach zleksykalizowanych, podczas gdy grupy zdefiniowane są głównie w oparciu o klasy gramatyczne. Znakowanie grup składniowych polega na oznaczeniu granic frazy, ale także oznakowaniu dwóch *centrów*: nadrzędnika (składniowego) oraz centrum semantycznego (tj. składnika, który znaczeniowo reprezentuje grupę). W pracy podana jest przykładowa fraza nad morzem — nad jest nadrzędnikiem składniowym, podczas gdy morzem jest centrum semantycznym frazy.

W NKJP wyróżniono następujące rodzaje grup składniowych (na podstawie Głowińska, 2012):

1. grupy nominalne (NG), np. sala posiedzeń senatu, bieżących wydarzeń politycznych;
2. grupy liczebnikowe (NumG), np. dwie dziewczyny, ostatnie pięć minut;
3. grupy przymiotnikowe (AdjG), np. wyjątkowo piękny, przyzwyczajony pracować;
4. grupy przyimkowo-nominalne (PrepNG), np. nad głównym wejściem;
5. grupy przyimkowo-przymiotnikowe (PrepAdjG), np. na zmęczonego;
6. grupy przyimkowo-liczebnikowe (PrepNumG), np. z dwiema osobami;

7. grupy przysłówkowe (AdvG), np. gdzieś daleko, niemal natychmiast;
8. *dyskurs* — „elementy zdania, które nie sa składniowo zwiazane", np. a nuż, m.in., moim zdaniem;
9. zdanie podrzędne z że, żeby, iż, aby, by (CG);
10. zdanie podrzędne pytajne (KG).

Powyższy zestaw fraz wydaje się niezwykle duży w porównaniu do przytaczanych wcześniej prac dla języków słowiańskich (a nawet w stosunku do fraz wyróżnianych w języku angielskim, por. Abney, 1996a). Uwzględniono w nim kilka szczegółowych rozróżnień — np. frazy rzeczownikowe wyraźnie oddzielono od liczebnikowych, zaś frazy przyimkowe podzielono na trzy grupy, w zależności od centrum semantycznego takiej grupy. Kolejną decyzją nietypową dla pracy z dziedziny płytkiej analizy składniowej jest uwzględnienie dwóch grup odpowiadającym całym zdaniom składowym (CG i KG). Obecność tego typu struktur jest cenna, gdyż zwiększa możliwości użycia fraz do badań lingwistycznych a także na potrzeby systemów przetwarzania języka naturalnego. W sytuacji, gdy rozróżnienie danego typu nie jest istotne, można zawsze złączyć kilka grup w jedną (z możliwości takiej będziemy korzystać podczas oceny algorytmów znakowania fraz).

W przypadku grup składniowych ustalono, że jedno słowo składniowe może być elementem co najwyżej jednej grupy składniowej (Głowińska, 2012). Nie wyklucza to jednak możliwości pokrywania się dwóch grup o tej samej nazwie. Przykładowo, w korpusie pojawia się równoważnik zdania (4.61). Oznakowano tam dwie grupy NG: siłowanie się oraz szarpanie się. Obie grupy tworzone są przez pojedyncze słowa składniowe opisujące formy rzeczownika. Te słowa składniowe łączy użycie tego samego wystąpienia zaimka się.

(4.61) [*Siłowanie się*] , *szarpanie*
Siłowanie [*się* **1**] , [*szarpanie* **1**]

Przypadki takie są jednak bardzo rzadkie, dzięki czemu można zastosować prymitywną heurystykę, która umożliwia „spłaszczenie" takiej struktury do rozłącznych fraz: jeśli dwie lub więcej fraz posiada segmenty wspólne, segmenty te przyporządkowujemy arbitralnie do frazy, której początek znajduje się najbliżej początku zdania.

Zdarza się również, że między grupami o innych nazwach zachodzi relacja zawierania. W szczególności to dotyczy grup CG i KG odpowiadającym całym zdaniom podrzędnym — w środku takich zdań oznaczono również inne typy grup, np. rzeczownikowe.

Występuje również zjawisko nieciągłych fraz omawiane już wcześniej. Podobnie jak w KPWr, zjawisko to występuje tu stosunkowo rzadko, dlatego też stosować będziemy to samo rozwiązanie, które omawialiśmy w punkcie 4.2.2: za frazy uznamy ciągłe fragmenty nieciągłych grup lub słów składniowych.

Nadrzędnikiem **grupy nominalnej** (czyli frazy rzeczownikowej) jest słowo składniowe pełniące funkcję rzeczownika, zaimka osobowego lub zaimka siebie. Podrzędnikiem grupy może być:

1. rzeczownik — w mianowniku (co tworzy apozycję, np. terroryści samobójcy), w dopełniaczu (brat ojca), a czasem w innym przypadku (spacer ulicami Wrocławia);
2. liczebnik (kurtki trojga dzieci, spacer trzema ulicami Wrocławia);
3. przymiotnik (bieżących wydarzeń politycznych, coś dobrego),

4. partykuło-przysłówek (prawie geniusz).

Podobne zasady dotyczą **grup przyimkowych** oraz **grup przymiotnikowych**. Jak już wspomnieliśmy, nadrzędną zasadą w NKJP jest „ucinanie" fraz rzeczownikowych, przymiotnikowych oraz przyimkowych na każdym wystąpieniu przyimka. W pracy Głowińska (2012) podano m.in. przykłady (4.62)–(4.64). Można zauważyć, że w przypadku znakowania KPWr, przykłady te stanowiłyby całe frazy.

(4.62) [$_{AdjG}$ odporny] [$_{PrepNG}$ na zabrudzenia]
(4.63) [$_{NG}$ ochota] [$_{PrepNG}$ na kawę]
(4.64) [$_{NG}$ spektakl] [$_{PrepNG}$ pt. „Dziady"]

Wyjątkiem od tej reguły są tzw. konstrukcje elektywne, np. jeden z wielu, gdyż „dało się je precyzyjnie opisać za pomocą reguł" (Głowińska, 2012). Kolejny „punkt cięcia fraz" związany jest z obecnością imiesłowów przymiotnikowych określających rzeczowniki. Jeśli imiesłów taki występuje przed rzeczownikiem, zostaje włączony do frazy. Jeśli następuje natomiast po niej, nie jest już on włączany. Praca (Głowińska, 2011) podaje przykłady (4.65) i (4.66) oraz następującą motywację: imiesłowy poprzedzające rzeczowniki „zachowują się bardziej jak przymiotniki", zaś te następujące po rzeczownikach — „często jako ekwiwalent zdania względnego". W KPWr imiesłowy oraz ich podrzędniki włączane są do NP niezależnie od ich umiejscowienia.

(4.65) [$_{NG}$ nadchodzące zmiany]
(4.66) [$_{NG}$ zapaleńcy] [$_{AdjG}$ prowadzący] [$_{NG}$ swoje wojenki]

Wytyczne NKJP dopuszczają także oznakowanie szeregowych fraz rzeczownikowych jako NG. Niestety prace nie podają bliższego określenia takich fraz; Głowińska (2012) podaje jedynie dwa przykłady: (4.67) i (4.68). W szczególności nie jest jasne, jak rozbudowane mogą być elementy takich szeregów — czy mogą być nimi dowolne elementy, które uznane byłyby samodzielnie za NG, czy też jakiś ich podzbiór. Głowińska (2011) zaznacza, że szeregi uwzględniają jedynie elementy połączone zaimkami szeregowymi, natomiast przecinki i inne znaki interpunkcyjne zawsze dzielą frazy. Można stąd wnioskować, że szeregu (4.69) nie można by oznakować w NKJP jako jednego NG (podane tu oznakowanie frazami NG jest jedynie domysłem autora rozprawy).

(4.67) [$_{NG}$ Jan albo Maria]
(4.68) [$_{NG}$ rządu i parlamentu]
(4.69) [$_{NG}$ Adam], [$_{NG}$ Jan albo Maria]

Nadrzędnikiem **grupy przymiotnikowej** jest przymiotnik. Podrzędnikiem może być czasownik (np. gotowy zostać, przyzwyczajony pracować) oraz przysłówek lub partykuła (wyjątkowo piękny, dość głupi). Nie są za to włączane podrzędniki rzeczownikowe. Praca (Głowińska, 2012) podaje przykłady (4.70) i (4.71) oraz wyjaśnienie, że włączenie tych podrzędników wymagałoby informacji pochodzącej ze słownika walencyjnego (tj. określającego wymagania konkretnych czasowników). Według wytycznych KPWr przykłady te należałoby oznakować jako całe frazy przymiotnikowe.

(4.70) [$_{AdjG}$ miły] [$_{NG}$ sercu]
(4.71) [$_{AdjG}$ winny] [$_{NG}$ braku nadzoru]

4.3. Ocena płytkich parserów

Porównanie parserów jest zadaniem trudnym. Aby móc porównać osiągi dwóch różnych parserów, muszą zostać one przetestowane w tych samych warunkach. Oznacza to po pierwsze konieczność testowania parsera na tym samym korpusie wzorcowym. Co więcej, aby porównanie było uczciwe, parsery te muszą przyjmować takie same założenia odnośnie rozpoznawanej struktury składniowej. Jest to szczególnie kłopotliwe w przypadku parserów opartych na regułach pisanych ręcznie: są one przywiązane do danych definicji fraz i nie ma możliwości przetestowania ich na korpusie zakładającym inne definicje. Problem ten jest szczególnie widoczny w przypadku języków słowiańskich: prace różnią się w istotny sposób przyjętymi definicjami fraz, nawet jeśli języki są do siebie podobne (por. punkt 4.2.1). W dalszych częściach tego rozdziału przytaczamy wartości liczbowe miar oceniających osiągi parserów odnotowane dla różnych języków. Ze względu na wspomniane różnice w założeniach i definicjach fraz, wartości te należy traktować jedynie poglądowo.

Podobnie jak w przypadku tagerów, płytkie parsery ocenia się najczęściej pod kątem zgodności ich wyjścia z oznakowaniem pochodzącym z korpusu wzorcowego. Do oceny modułów znakowania fraz stosuje się powszechnie trzy standardowe miary (na podstawie Tjong Kim Sang i Buchholz, 2000; miary te są na ogół liczone z osobna dla poszczególnych typów fraz):

Dokładność (ang. *precision*, P) — odsetek fraz zwróconych przez parser, który występuje również w korpusie wzorcowym. Aby fraza została uznana za prawidłowo rozpoznaną, jej granice muszą leżeć dokładnie w tym samym miejscu, gdzie oznakowano je w korpusie wzorcowym. Dokładność można zapisać wzorem (4.72); A to zbiór fraz zwróconych w wyniku automatycznej analizy składniowej (zbiór fraz zwróconych przez parser), a K to zbiór fraz w korpusie wzorcowym.

Kompletność (ang. *recall*, R) — odsetek fraz oznakowanych w korpusie wzorcowym, które zostały prawidłowo rozpoznane przez parser (mają identyczne granice). Komplentość można opisać wzorem (4.73).

Miara F (ang. *F-measure*, F) — średnia harmoniczna dokładności i kompletności, tj. (4.74). Miara ta jest na ogół używana jako ogólna ocena działania parsera.

$$P = \frac{|A \cap K|}{|A|} \tag{4.72}$$

$$R = \frac{|A \cap K|}{|K|} \tag{4.73}$$

$$F = \frac{2PR}{P + R} \tag{4.74}$$

Analiza odnotowanych wartości dokładności i kompletności ułatwia wnioskowanie na temat typowych błędów popełnianych przez parser. Jeśli osiąga on wysoką kompletność przy niskiej dokładności, oznacza to, że parser ma tendencję do rozpoznawania zbyt dużej liczby fraz, lecz pośród zwróconego zbioru znajduje się większość fraz, które należało rozpoznać. Sytuacja odwrotna oznacza, że parser zwrócił zbyt małą liczbę fraz, natomiast frazy zwrócone przez parser można z dużą pewnością uznać za rozpoznane prawidłowo.

Zastosowanie parsera decyduje o tym, czy ważniejsza jest kompletność, czy też dokładność. Jeśli ocena ma charakter ogólny, za miarę o charakterze decyzyjnym uznawana jest miara F.

Większość algorytmów znakowania fraz wymaga, by tekst wejściowy był oznakowany morfosyntaktycznie. Zauważono, że jakość tego oznakowania ma wyraźny wpływ na jakość uzyskanego przy użyciu parsera oznakowania frazami (Hajič i inni, 2001). Ważne jest zatem, aby ocena eksperymentalna działania analizatora składniowego uwzględniała wpływ błędów tagera na wyniki końcowe. Innymi słowy, rzetelna ocena parsera powinna być wykonywana na danych oznakowanych morfosyntaktycznie przy użyciu automatycznego tagera, bez użycia wzorcowego oznakowania pochodzącego od lingwistów (Tjong Kim Sang i Buchholz, 2000). Zalecenie to ma podobny charakter do proponowanego w tej rozprawie zalecenia, by ocena tagera uwzględniała błędy popełnione na poprzednich etapach przetwarzania, tj. segmentacji i analizy morfosyntaktycznej (punkt 2.5). Eksperymenty omówione w punkcie 5.6 potwierdzają praktyczne znaczenie tego zalecenia: obserwowane osiągi parserów są dużo wyższe, jeśli oceny dokonujemy na korpusie z wzorcowym oznakowaniem morfosyntaktycznym — jednak te zawyżone wyniki mają niewielką wartość praktyczną, gdyż w normalnych okolicznościach użytkownik nie ma dostępu do wzorcowego oznakowania morfosyntaktycznego.

4.4. Przegląd metod płytkiej analizy składniowej

Jak wspomnieliśmy w punkcie 1.2.3, płytka analiza składniowa bywa rozumiana różnie. Zgodnie z zakresem tej rozprawy, skupimy się tutaj na metodach **znakowania fraz**. Wyjątkiem od tego założenia będzie omówienie płytkiej analizy składniowej za pomocą reguł pisanych ręcznie — reguły takie na ogół rozpoznają strukturę zawierającą pewien stopień zagnieżdżenia.

W dalszej części tego punktu omówimy kolejno techniki regułowe oraz techniki zakładające uczenie się (tj. statystyczne i oparte na klasyfikacji). Na samym końcu przedstawimy podsumowanie metod i systemów opracowanych dla języków słowiańskich.

4.4.1. Reguły pisane ręcznie

Popularną metodą płytkiej analizy składniowej jest ręczne pisanie gramatyk opisujących rozkład fraz na czynniki bezpośrednie. Są to formalizmy odpowiadające gramatykom regularnym lub gramatykom bezkontekstowym, niekiedy wzbogacone o dodatkowe mechanizmy, np. testy na równość wartości atrybutów. Parsery takich gramatyk na ogół implementowane są za pomocą technik znanych z teorii kompilacji, mianowicie transduktorów czy też ich kaskad (tj. wyjście jednego transduktora jest wejściem dla kolejnego; por. Müller, 2005) albo parserów LR(k) (Aho i Ullman, 1972).

Systemów takich powstało wiele, zwłaszcza dla języka angielskiego i niemieckiego, np. Hobbs (1992); Grefenstette (1996); Cunningham i inni (2000); Müller (2005). Podejście to przedstawimy na przykładzie modułu gramatyk bezkontekstowych Abneya (1991). Przegląd systemów regułowych dla języków germańskich wykracza poza zakres niniejszej rozprawy; przegląd taki można znaleźć w przytoczonej pracy Müller (2005), a

także w pracy Przepiórkowskiego (2008). Do systemów regułowych wrócimy w punkcie 4.4.3 przy okazji omawiania technik przetwarzania języków słowiańskich.

Abney, twórca koncepcji znakowania fraz, opisuje prostą technikę znakowania fraz za pomocą gramatyki bezkontekstowej (Abney, 1991). Reguły uruchamiane są na tekście oznakowanym morfo-syntaktycznie, dzięki czemu mogą odwoływać się do klas gramatycznych przypisanych segmentom. Gramatyka taka składa się z reguł przepisywania. Każda reguła opisuje możliwy schemat rozbioru frazy na składniki, będące frazami — symbolami nieterminalnymi w terminologii gramatyk formalnych (Aho i Ullman, 1972) — bądź też pojedynczymi segmentami o określonych klasach gramatycznych (symbolami terminalnymi). Poniżej przedstawiono fragment gramatyki Abneya:

$$\texttt{PP} \rightarrow \texttt{P DP}$$
$$\texttt{DP} \rightarrow \texttt{Predet? D? NP}$$
$$\texttt{DP} \rightarrow \texttt{QP}_{\texttt{Pron}}$$

Przykładowo, pierwsza reguła mówi, że fraza przyimkowa (PP) składa się z przyimka (P), po którym następuje fraza określnikowa (DP). Ta ostatnia opisana jest dwoma alternatywnymi regułami: albo składa się z opcjonalnego określnika poprzedzającego (Predet?), opcjonalnego określnika centralnego (D?) oraz frazy rzeczownikowej (NP), albo też frazy kwantyfikatorowej wyrażonej zaimkiem ($\texttt{QP}_{\texttt{Pron}}$).

Warto zauważyć, że gramatyki takie mogą być użyte do opisu dowolnego poziomu zagnieżdżenia; w szczególności opis ten nie wyklucza rekurencji (np. fraza NP może być zdefiniowana poprzez odwołanie do frazy NP niższego rzędu). Można zatem powiedzieć, że jest to już głęboka analiza składniowa. Mimo to, często podejścia takie nazywane są wciąż analizą płytką ze względu na charakter gramatyki. W przypadku analizy płytkiej, twórcy gramatyk ograniczają się do opisu wybranego zestawu fraz, gdzie frazy zazwyczaj okrojone są do fragmentów tradycyjnie rozumianych fraz tak, aby uniknąć podejmowania decyzji zbyt trudnych.

4.4.2. Metody statystyczne i uczenie maszynowe

Jeden z pierwszych modułów znakowania fraz został zaproponowany w pracy Church (1988). Moduł implementuje bardzo prosty algorytm statystyczny, który estymuje prawdopodobieństwo, że między danymi dwoma tagami znajduje się początek lub koniec frazy.

Większość późniejszych metod opiera się na spostrzeżeniu, że znakowanie fraz można w prosty sposób sprowadzić do problemu znakowania ciągu. Obserwacji tej dokonali Ramshaw i Marcus (1995) i zaproponowali sposób reprezentacji fraz za pomocą znaczników przypisywanych kolejnym segmentom. Zaproponowany zbiór składa się z trzech znaczników: O (*outside*), I (*inside*) oraz B (*begin*). Ten „tagset" wraz z zaproponowaną interpretacją został później nazwany **reprezentacją IOB1** (Sang i Veenstra, 1999):

— O oznacza segment nienależący do frazy,
— B oznacza segment rozpoczynający frazę, która następuje bezpośrednio po poprzedniej frazie,
— I oznacza „zwykły" segment należący do frazy (tj. niespełniający powyższego warunku).

Ramshaw i Marcus (1995) podają następujący przykład:

(4.75) In [$_{NP}$ early trading] in [$_{NP}$ Hong Kong] [$_{NP}$ Monday], [$_{NP}$ gold] was quoted at [$_{NP}$ $ 336.50] [$_{NP}$ an ounce] .

Fragment ten uzyskuje reprezentację IOB1 (4.76).

(4.76)

In	*early*	*trading*	*in*	*Hong*	*Kong*	*Monday*	*,*	*gold*	*was*	*quoted*	*at*	*$*
O	I	I	O	I	I	B	O	I	O	O	O	I

336.50	*an*	*ounce*	*.*
I	B	I	O

Sang i Veenstra (1999) proponują sześć innych reprezentacji i stwierdzają, że wybór reprezentacji ma stosunkowo niewielki wpływ na działanie modułu znakowania fraz. Warto przytoczyć jedną z nich — **reprezentację IOB2** — gdyż stała się ona bardzo popularna (prawdopodobnie za sprawą konferencji–konkursu CoNLL-2000, Tjong Kim Sang i Buchholz, 2000). Reprezentacja korzysta z tych samych znaczników, lecz zmienono semantykę znaczników `B` oraz `I`:

— `B` oznacza każdy segment rozpoczynający frazę,
— `I` oznacza segment należący do frazy, lecz jej nie rozpoczynający,
— `O` oznacza, jak w IOB1, segment nienależący do frazy.

W takim ujęciu, zdanie (4.75) uzyskuje oznakowanie (4.77).

(4.77)

In	*early*	*trading*	*in*	*Hong*	*Kong*	*Monday*	*,*	*gold*	*was*	*quoted*	*at*	*$*
O	B	I	O	B	I	B	O	B	O	O	O	B

336.50	*an*	*ounce*	*.*
I	B	I	O

Reprezentacja IOB2 wydaje się koncepcyjnie prostsza i bardziej naturalna. Z tego powodu oraz ze względu na jej dużą popularność będzie ona stosowana również w tej rozprawie.

Jeśli rozpatrywany zbiór fraz zawiera więcej niż jedną frazę, istnieją dwie możliwości:

1. rozpatrywać znakowanie każdego typu frazy jako osobny problem znakowania ciągu,
2. rozszerzyć zbiór znaczników.

Pierwsze podejście ma tę zaletę, że jest koncepcyjnie prostsze. Podejście to ma tę własność, że frazy znakowane są niezależnie, a co za tym idzie, reprezentacja nie wymusza, by były wzajemnie rozłączne. W niektórych sytuacjach może to być wadą. Przykładowo, może być celowe mocne założenie, że frazy rzeczownikowe nie mogą mieć wspólnych segmentów z frazami czasownikowymi.

Ramshaw i Marcus (1995) realizują drugie podejście, wprowadzając warianty znaczników `B` oraz `I` dla każdej z fraz. Przykładowo, jeśli przyjmiemy reprezentację IOB2 oraz dwa typy fraz: NP i VP, otrzymujemy pięć możliwych znaczników:

— `B-NP` oznacza segment rozpozczynający frazę NP,
— `I-NP` oznacza segment należący do NP, lecz jej nie rozpoczynający,
— `B-VP` oznacza segment rozpozczynający frazę VP,
— `I-VP` oznacza segment należący do VP, lecz jej nie rozpoczynający,
— `O` oznacza segment nienależący do żadnej z rozpatrywanych fraz.

Sformułowanie problemu znakowania fraz jako problemu znakowania ciągu jest bardzo wygodne, gdyż pozwala stosować praktycznie wszystkie znane dotąd metody roz-

wiązywania takich problemów, w tym metody znakowania morfosyntaktycznego opisane w punkcie 2.4. Od strony obliczeniowej, znakowanie fraz jest poniekąd prostsze w realizacji ze względu na niewielki rozmiar zbioru klas. Podczas gdy tagsety mają od kilkudziesięciu do ponad tysiąca możliwych tagów, w przypadku znakowania pojedynczej frazy mamy jedynie trzy możliwe klasy (`I`, `O`, `B`), a w przypadku n rozłącznych fraz — $2n + 1$ klas.

Warto w tym miejscu podkreślić, że ta prostota realizacji nie pociąga za sobą lepszych wyników znakowania fraz niż znakowania morfosyntaktycznego. Miałoby to miejsce gdyby oceniać osiągi modułu znakowania fraz za pomocą *trafności* rozumianej jako procent segmentów z poprawnie przypisanymi tagami IOB2. Miary tej się jednak nie używa, a stosowana powszechnie miara F implikuje bardziej surowe kary. Przykładowo, jeśli mamy do czynienia z bardzo długą frazą, a moduł znakowania zwróci poprawnie wszystkie znaczniki IOB2 z wyjątkiem jednego, to cała fraza zostanie potraktowana jako nietrafiona. Trafność przypisywania znaczników nie jest dobrym sposobem oceny oznakowania frazami, gdyż stosunkowo łatwo osiągnąć wysokie wartości, jeśli prawidłowo rozpoznamy tagi `O` oznaczające, że przez segmenty nie przebiega żadna fraza. W szczególności jeśli wartość trafności podalibyśmy dla problemu znakowania pojedynczej frazy, która występuje w tekście nieczęsto, „parser", który każdemu segmentowi przypisuje znacznik `O`, uzyskałby wysoką ocenę.

Ramshaw i Marcus (1995) pokazali nie tylko sposób reprezentacji fraz za pomocą znaczników IOB1, ale także sposób, w jaki można użyć algorytmu Brilla znanego ze znakowania morfosyntaktycznego do znakowania fraz. Przypomnijmy, że algorytm Brilla pozwala na indukcję reguł, które dokonują kolejnych poprawek istniejącego już oznakowania ciągu segmentów (por. str. 32). Algorytm wymaga zastosowania heurystyki, która pozwala na początkowe przypisanie znaczników — przypisanie, które korygować będą reguły w kolejnych iteracjach. W przypadku algorytmu z pracy (Ramshaw i Marcus, 1995) heurystyka ta korzystała z oznakowania morfosyntaktycznego: danemu segmentowi w przypisujemy znacznik IOB1, który najczęściej przypisano w danych uczących segmentom oznakowanym tagiem morfosyntaktycznym, który tager przypisał także segmentowi w. Na danych pochodzących z korpusu Wall Street Journal (WSJ) udało się osiągnąć wartość miary F 92,05%.

Praca (Ramshaw i Marcus, 1995) jest istotna także ze względu na stosowany zbiór danych: wspomniane dane pozyskane z korpusu WSJ wraz z ich podziałem na część uczącą i testową stały się standardowym materiałem testowym, na którym później przetestowano szereg algorytmów znakowania fraz rzeczownikowych[3]. Drugi standardowy zbiór danych został opracowany na potrzeby konferencji–konkursu CoNLL-2000 (Tjong Kim Sang i Buchholz, 2000). Dane te także pochodzą z tego samego fragmentu korpusu WSJ, lecz tym razem tekst oznakowano aż jedenastoma typami fraz. Zbiór ten uwzględnia frazy rzeczownikowe (NP), przyimkowe (PP), przymiotnikowe (AdjP), czasownikowe (VP), przysłówkowe (AdvP), a także kilka mniej typowych fraz, które odpowiadają partykułom, spójnikom itp. (te nietypowe frazy występują w tekście stosunkowo rzadko). W przypadku obu zbiorów danych, część ucząca składa się z 211 727

[3] Oprócz fraz rzeczownikowych, oznaczonych tam jako N, autorzy wprowadzają frazy pomocnicze, zwane V. Frazy V nie są jednak prawdziwymi frazami czasownikowymi, a jedynie pomocniczym zgrupowaniem kilku rodzajów fraz nierzeczownikowych i nie są one używane w większości późniejszych prac.

segmentów, natomiast część testowa zawiera ich 47 377. Dla uproszczenia, pierwszy zbiór danych będziemy nazywać odtąd **korpusem WSJ-NP**, natomiast zbiór drugi — **korpusem CoNLL 2000**.

Podsuwanie wybranych prac prezentujących ocenę algorytmów znakowania fraz na tych korpusach przedstawimy tabelarycznie. Kilka z tych prac omówimy bardziej szczegółowo poniżej. Tabela 4.1 podsumowuje wyniki prac, gdzie eksperymenty przeprowadzono na korpusie WSJ-NP. Tabela 4.2 dotyczy wyników eksperymentów przeprowadzonych na korpusie CoNLL-2000. Użyte w tabeli skróty wyjaśniamy poniżej:

CRF — warunkowe pola losowe,
HMM — ukryty model Markowa,
MBL — uczenie na pamięć,
MaxEnt — maksymalizacja entropii,
SVM (ang. *Support Vector Machine*) — maszyna wektorów wspierających, tj. klasyfikator, którego uczenie polega na znajdowaniu płaszczyzny rozdzielającej przykłady należące do różnych klas (Cortes i Vapnik, 1995),
WPDV (ang. *weighted probability distribution voting*) — głosowanie między rozkładami prawdopodobieństwa, algorytm uczenia maszynowego zaproponowany w pracy (von Halteren, 2000)

Publikacja	**Użyta technika**	**Miara F dla NP**
Shen i Sarkar (2005)	HMM i głosowanie	95,23%
Sha i Pereira (2003)	SVM	94,38%
Sun i inni (2008)	CRF	94,34%
McDonald i inni (2005)	CRF	94,29%
Kudoh i Matsumoto (2001)	SVM	94,22%
Hollingshead i inni (2005)	Głęboki parser	94,20%
Kudoh i Matsumoto (2000)	SVM	93,79%
Sang i Veenstra (1999)	MBL	92,37%
Veenstra (1998)	MBL	91,57%

Tabela 4.1. Wybrane prace, gdzie testy przeprowadzono na korpusie WSJ-NP

Prace (Sang i Veenstra, 1999; Veenstra i van den Bosch, 2000) przedstawiają proste podejście do znakowania fraz przy użyciu klasyfikatora pamięciowego. Model ten jest prawie identyczny z modelem znakowania morfosyntaktycznego poprzez klasyfikację kolejnych segmentów omówionym w punkcie 2.4.3 na stronie 24: przyjmuje się, że znacznik IOB2 przypisany segmentowi w_i zależy jedynie od najbliższego otoczenia tego segmentu. Problem znakowania fraz można więc sprowadzić do klasyfikacji otoczeń kolejnych segmentów w zdaniu. W przypadku przypisywania znaczników IOB2 mamy do dyspozycji nie tylko formy wyrazowe, lecz także przypisane im tagi morfosyntaktyczne (wskutek użycia tagera), a także przypisane dotąd znaczniki IOB2 (zakłada się klasyfikację kolejnych segmentów w zdaniu, a więc jedynie segmenty stanowiące lewy kontekst segmentu klasyfikowanego mają już przypisane znaczniki IOB2). Veenstra i van den Bosch (2000) stosują następujące cechy:

1. formy wyrazowe w oknie $(-5, -4, \ldots, +2, +3)$,
2. tagi morfosyntaktyczne w tym samym oknie.

Publikacja	**Użyta technika**	**NP**	**VP**	**F**
Kudoh i Matsumoto (2000)	SVM	93,87%	93,8%	93,48%
van Halteren (2000)	WPDV	93,84%	93,65%	93,32%
Tjong Kim Sang (2000)	MBL + głosowanie	93,23%	92,64%	92,5%
Zhou i inni (2000)	hybrydowa	92,39%	92,81%	92,12%
Déjean (2000)	indukcja reguł	92,54%	92,70%	92,09%
Koeling (2000)	MaxEnt	93,01%	92,88%	91,97%
Osborne (2000)	MaxEnt	92,19%	92,7%	91,95%
Veenstra i van den Bosch (2000)	MBL	91,76%	92,3%	91,54%
Pla i inni (2000)	HMM	89,84%	91,55%	90,14%
Johansson (2000)	MBL	89,3%	89,75%	87,23%

Tabela 4.2. Wybrane prace, gdzie testy przeprowadzono na korpusie CoNLL-2000. Kolumny **NP** i **VP** podają wartości miary F dla tych fraz, kolumna **F** podaje wartości miary F liczone łącznie dla wszystkich fraz

Sang i Veenstra (1999) stosują praktycznie te same cechy, lecz przy mniejszym rozmiarze okna: $(-2, \ldots, +1)$. Praca opisuje także drugi eksperyment polegający na wprowadzeniu drugiego przebiegu znakowania, którego zadaniem było ponowne przypisanie znaczników IOB2 przy użyciu dodatkowych cech, tj. znaczników IOB2 z najbliższego otoczenia przypisanych w poprzednim przebiegu.

Praca (Sha i Pereira, 2003) prezentuje sposób zastosowania warunkowych pól losowych do problemu znakowania fraz. W tabeli 4.3 przedstawiamy zastosowane szablony cech (ideę szablonów cech dla warunkowych pól losowych omówiliśmy na stronie 23). Zapis s_i oznacza znacznik IOB2 przypisany segmentowi na pozycji centralnej (0), w_i oznacza formę wyrazową segmentu na pozycji centralnej, a t_i oznacza tag morfosyntaktyczny segmentu na pozycji centralnej. Greckie litery oznaczają stałe, do których przyrównywane są podane wartości. Przykładowo, zapis $t_{i-1} = \tau_1$ oznacza, że tag przypisany segmentowi na pozycji -1 przyrównywany jest do stałej τ_1.

Kilka prac stosuje także modele matematyczne, które są w zasadzie rozszerzeniem warunkowych pól losowych. Sun i inni (2008) stosują warunkowe pola losowe z ukrytą dynamiką (ang. *Latent-Dynamic Discriminative Conditional Random Fields*, *LDCRF*). Idea polega na wprowadzeniu do modelu dodatkowego poziomu stanów ukrytych, pozwalających na przypisanie form wyrazowych do pewnych klas abstrakcji. McDonald i inni (2005) wprowadzają klasyfikację wieloklasową, której zadaniem jest określenie, do której frazy dany segment należy.

Dwie z cytowanych prac stosują zabieg *głosowania między reprezentacjami* (Shen i Sarkar, 2005; Tjong Kim Sang, 2000). Zabieg ten polega na konwersji danych uczących do kilku różnych reprezentacji (w tym wspomnianych IOB1 i IOB2), po czym na tak przetworzonych danych uczonych jest kilka instancji danego algorytmu. Podczas działania parsera, wyuczone modele są stosowane do oznakowania tekstu, wyniki są konwertowane na wspólną reprezentację, a na końcu dokonywane jest głosowanie większościowe.

1. $s_i = \sigma$
2. $s_i = \sigma_1 \wedge s_{i-1} = \sigma_2$
3. $s_i = \sigma_1 \wedge s_{i-1} = \sigma_2 \wedge s_{i-2} = \sigma_3$
4. $s_i = \sigma \wedge w_{i+N} = \phi$ dla $N \in \{-2, -1, 0, 1, 2\}$
5. $s_i = \sigma \wedge t_{i+N} = \tau$ dla $N \in \{-2, -1, 0, 1, 2\}$
6. $s_i = \sigma \wedge w_{i-1} = \phi_1 \wedge w_i = \phi_2$
7. $s_i = \sigma \wedge w_i = \phi_1 \wedge w_{i+1} = \phi_2$
8. $s_i = \sigma \wedge t_{i-1} = \tau_1 \wedge t_i = \tau_2$
9. $s_i = \sigma \wedge t_i = \tau_1 \wedge t_{i+1} = \tau_2$
10. $s_i = \sigma \wedge t_{i-2} = \tau_1 \wedge t_{i-1} = \tau_2 \wedge t_i = \tau_3$
11. $s_i = \sigma \wedge t_{i-1} = \tau_1 \wedge t_i = \tau_2 \wedge t_{i+1} = \tau_3$
12. $s_i = \sigma \wedge t_i = \tau_1 \wedge t_{i+1} = \tau_2 \wedge t_{i+2} = \tau_3$

Tabela 4.3. Szablony cech zaproponowane przez Sha i Pereira (2003) na potrzeby znakowania fraz

4.4.3. Płytka analiza składniowa języków słowiańskich

Zdecydowana większość znanych z literatury metod płytkiej analizy składniowej języków słowiańskich opiera się na gramatykach pisanych ręcznie. Istnieje kilka formalizmów, które pozwalają na zapis takich gramatyk oraz implementacji, które pozwalają na uzyskanie działającego parsera na ich podstawie.

Jeden z takich formalizmów związany jest ze wspomnianym w punkcie 4.2.1 „formalnym modelem fraz rzeczownikowych w języku serbsko-chorwackim" (Nenadić i Vitas, 1998a). Formalizm ten oparty jest o pojęcie *morfosyntaktycznych wyrażeń regularnych* (ang. *regular morphosyntactic expressions*). Wyrażenia te przypominają język regularny nad alfabetem, którego symbolami są *rozszerzone opisy morfosyntaktyczne*. Rozszerzony opis morfosyntaktyczny rozumiany jest jako specyfikacja ograniczeń, które musi spełniać segment, a konkretnie jego tag, lemat lub forma wyrazowa. Przykładowym morfosyntaktycznym wyrażeniem regularnym jest zapis `N? Ng Ng`. Wyrażenie to dopasuje się do ciągu rzeczowników (`N`), z których pierwszy jest opcjonalny (`N?`), a dwa kolejne muszą występować w dopełniaczu (`g`). Reguły formalizmu są uruchamiane na tekście poddanym analizie morfosyntaktycznej. Aplikacja reguł powoduje nie tylko oznakowanie fraz, lecz także częściowe ujednoznacznienie morfosyntaktyczne. Formalizm ten został potem rozszerzony o mechanizm unifikacji, który pozwolił na opis uzgodnień gramatycznych (Nenadić i Vitas, 1998b). Do gramatyki wprowadzono później reguły opisujące szeregowe frazy rzeczownikowe (Nenadić, 2000), a także frazy czasownikowe (Nenadić i inni, 1999). Niestety, autorzy tych prac nie podejmują jednak próby oceny jakości opracowanych gramatyk.

Podobne gramatyki napisano dla języka chorwackiego przy pomocy systemu NooJ (Silberztein, 2003). Prace (Vučković i inni, 2008; Vučković, 2009; Vučković i inni, 2010) prezentują kolejne wersje gramatyki, która pozwala na rozpoznanie m.in. fraz rzeczownikowych, przyimkowych oraz czasownikowych. W zależności od użytego korpusu i sposobu oceny, udało się osiągnąć wartości miary F dla fraz rzeczownikowych od 88,4% do

96,6%, a dla fraz czasownikowych — ok. 94%–97% (przyczyną dużych różnic w wynikach prawdopodobnie jest fakt, że w części eksperymentów skorzystano ze wzorcowego oznakowania morfosyntaktycznego, a w innych użyto tagera).

Podobna gramatyka została napisana również dla języka bułgarskiego. Większość reguł opisuje frazy rzeczownikowe uzgodnione pod względem liczby i rodzaju (rzeczowniki bułgarskie nie odmieniają się przez przypadki). Opisane eksperymenty pozwoliły osiągnąć dokładność 90,1% i kompletność 87,7% (Osenova, 2002).

Dla języka polskiego opracowano również podobny formalizm — formalizm Spejd[4] (Przepiórkowski, 2008). Formalizm pozwala na zwięzły zapis reguł znakowania słów i grup składniowych (por. punkt 4.2.3), a także reguł ujednoznaczniania morfosyntaktycznego. Podobnie jak w przytaczanym wyżej języku morfosyntaktycznych wyrażeń regularnych, jedna reguła może odpowiadać za rozpoznawanie grupy składniowej i jednoczesne częściowe ujednoznacznienie morfosyntaktyczne jej składników.

Reguła w formalizmie Spejd składa się z pięciu części (niektóre z nich można pominąć; Przepiórkowski, 2008, s. 121):

1. `Rule` (nazwa reguły),
2. `Left` (lewy kontekst),
3. `Match` (obszar dopasowania reguły),
4. `Right` (prawy kontekst),
5. `Eval` (warunki dodatkowe i operacje do wykonania).

Przepiórkowski (2008) podają następujący przykład prostej reguły:

```
Rule   "Reguła przykładowa"
Match: [pos~~"prep"]     # na pewno przyimek
       [base~"co|kto"];  # forma CO lub KTO
Eval:  unify(case,1,2);  # uzgodnij przypadek
       group(PG,1,2);    # twórz grupę przyimkową
```

Powyższa reguła dopasowuje ciąg składający się z dwóch segmentów: przyimek (`prep`) oraz następująca po nim forma o lemacie `co` lub `kto`. Jeśli dopasowanie to powiedzie się, następuje unifikacja (`unify`) wartości przypadka obu segmentów — tj. usuwane są te interpretacje, które naruszają uzgodnienie co do wartości przypadka. Druga operacja (`group`) prowadzi do oznaczenia grupy PG składającej się z tych dwóch segmentów. Pierwszy segment (przyimek) zostanie oznaczony jako nadrzędnik (składniowy) grupy, drugi segment (`co` lub `kto`) — jako centrum semantyczne grupy.

Oprócz obszaru dopasowania głównego (`Match`) istnieje możliwość podania dodatkowych dopasowań — lewego kontekstu (`Left`) oraz prawego kontekstu (`Right`). Wszystkie te dopasowania opisywane są językiem przypominającym wspomniany serbsko-chorwacki język morfosyntaktycznych wyrażeń regularnych. Formalizm Spejd ma jednak wyraźnie większą siłę ekspresji. Umożliwia on m.in. odwołanie się do oznaczonych już grup składniowych, których nadrzędniki (lub centra semantyczne) spełniają podane warunki. Co więcej, sposób dopasowania poszczególnych segmentów jest bardzo elastyczny i umożliwia użycie wyrażeń regularnych na poziomie form wyrazowych lub lematów. Przykładowo, zapis `[orth~~"o.*ć"]` opisuje dopasowanie do dowolnego segmentu odpowiadającego formie wyrazowej rozpoczynającej się literą `o` i kończącej

[4] Przepiórkowski (2008) używa także nazwy skróconej do symbolu ♠.

się literą `ć`. Język dopasowań wspiera także operacje koniunkcji, alternatywy i negacji (ta ostatnia operacja ograniczona jest do warunków atomowych).

Spejd doczekał się dwóch implementacji[5]: prototypowej implementacji w języku Java (już nie jest wspierana) oraz nowej implementacji w języku C++, w której zastosowano zaawansowane techniki skończenie-stanowe, by osiągnąć wysoką wydajność przetwarzania (Zabłocki, 2010). Formalizm poddany drobnym modyfikacjom został również zaimplementowany w parserze Puddle (Manicki, 2009).

Dwie gramatyki napisane w formalizmie Spejd są dostępne publicznie: *przykładowa gramatyka powierzchniowa* opisana w rozdziale 8 pracy (Przepiórkowski, 2008) oraz gramatyka napisana na potrzeby NKJP (Waszczuk i inni, 2010). Pierwsza z nich jest opisana jako „gramatyka robocza", której głównym celem jest wydobycie z dużych zbiorów tekstu informacji o możliwych argumentach czasownika. Spośród 468 reguł, znaczna część odpowiada za korektę błędów analizatora morfosyntaktycznego oraz rozpoznanie skrótów, liczb i symboli. Gramatyka rozpoznaje także grupy składniowe, których opis sugeruje, że są to grupy podobne do tych uwzględnionych w NKJP (opisaliśmy je w punkcie 4.2.3). Praca (Przepiórkowski, 2008) nie podejmuje próby oceny jakości gramatyki. Jak podaje Przepiórkowski (2009b), gramatyka ta nie jest kompletna i pozwala na „kompletną analizę składniową"[6] jedynie ok. 42% zdań z korpusu KIPI. Mimo to, użycie tej gramatyki w połączeniu z technikami statystycznymi pozwoliło na pozyskanie wysokiej jakości słownika przypisującego czasownikom informację na temat możliwych argumentów (Przepiórkowski, 2009b).

Drugą z tych gramatyk jest składająca się z 1181 reguł gramatyka napisana na potrzeby korpusu NKJP[7] (Głowińska, 2012; Waszczuk i inni, 2010). Będziemy ją odtąd nazywać w skrócie **gramatyką NKJP**. Rozpoznaje ona wszystkie typy słów i grup składniowych opisane w punkcie 4.2.3. Proces pisania gramatyki przeplatał się ze znakowaniem składniowym NKJP: początkowa wersja gramatyki posłużyła do oznakowania korpusu, po czym zespół lingwistów identyfikował błędy popełniane przez parser. Uwagi te były wykorzystywane w celu dodania reguł korygujących do gramatyki. Proces ten był powtarzany, co prowadziło do stopniowej poprawy zarówno oznakowania korpusu, jak i gramatyki. Zgodnie ze stanem wiedzy autora rozprawy, nie podjęto dotąd próby oceny jakości oznakowania, które można uzyskać za pomocą parsera Spejd wyposażonego w tę gramatykę. Pewną formę takiej oceny prezentujemy w rozdziale 5.

W pracy (Mykowiecka i inni, 2007) przedstawiono metody przetwarzania korpusu transkrypcji rozmów klientów z infolinią Zarządu Transportu Miejskiego w Warszawie. Jednym z etapów przetwarzania jest znakowanie fraz (rodzaje znakowanych tam fraz podsumowaliśmy na stronie 71). Autorzy opracowali w tym celu własny program, który pozwala na aplikację prostych reguł do ropoznawania takich fraz. W pracy nie podjęto jednak próby oceny jakości uzyskiwanego oznakowania.

---

[5] Obie implementacje można znaleźć na stronie `http://zil.ipipan.waw.pl/Spejd/`.

[6] Stwierdzenie to nie jest w pełni jednoznaczne, dlatego przytoczymy je w oryginale:

> Because of the partial nature of the grammar and the parser, not all sentences were fully parsed; after syntactic processing, some sentences contained sequences of lexical segments not assigned to any syntactic constituents. Only fully parsed sentences underwent the statistical processing. There were 1 137 014 (41.74%) such sentences and they contained 8 516 676 segments.

[7] Gramatyka jest dostępna pod adresem `http://nkjp.pl/index.php?page=14`.

Niewiele jest prac poświęconych znakowaniu fraz w językach słowiańskich za pomocą technik statystycznych bądź maszynowego uczenia. Mráková i Sedláček (2003) opisują system przetwarzania morfologiczno-składniowego języka czeskiego, w którego skład wchodzi m.in. regułowy parser Dis. Większość reguł została napisana ręcznie na podstawie istniejących gramatyk języka czeskiego. Wyjątkiem stanowią reguły rozpoznawania fraz czasownikowych, które pozyskiwane są automatycznie. Algorytm opiera się na założeniu, że frazy czasownikowe składają się z form czasownikowych i czasem zaimka se (się), a elementy te mogą być przedzielone innymi frazami (w przypadku nieciągłych fraz czasownikowych). Pierwszym etapem jest znalezienie tych elementów i zapamiętanie wzorców ich występowania. Wzorce te mogą się składać z lematów i tagów form należących do frazy oraz specjalnego znacznika, który oznacza, że w frazie wystąpiła przerwa spowodowana nieciągłością. Następnym etapem jest uogólnianie takich wzorców. Algorytm tego uogólniania nie został dokładnie opisany; wydaje się, że oparte jest ono na pewnych regułach pomocniczych napisanych ręcznie. Ostatnim krokiem jest zapis pozyskanej reguły w języku programowania Prolog.

Jakubíček i inni (2009) opisują reguły, które pozwalają na pozyskanie (płaskich) fraz rzeczownikowych z wyjścia głębokiego parsera języka czeskiego synt. Praca (Grác i inni, 2010) przedstawia podobne przedsięwzięcie, choć metodyka w niej przyjęta zakłada użycie trzech głębokich parserów oraz ręczną weryfikację pozyskanych fraz.

Tylko część cytowanych powyżej prac prezentuje testy opracowanych płytkich analizatorów składniowych. Prace takie podsumowujemy w tabeli 4.4. Tabela przedstawia zastosowaną technikę, osiągniętą wartość miary F dla problemu znakowania fraz rzeczownikowych oraz rozmiar użytego korpusu testowego. Dwie prace pojawiają się w tabeli wielokrotnie, gdyż prezentują wyniki eksperymentów przeprowadzonych na tych samych danych, lecz przy różnych założeniach.

Język	**Publikacja**	**Technika**	**F dla NP**	**Rozmiar korpusu**
Bułgarski	Osenova (2002)	Reguły	$88,9\%^{t}$	3500 słów
	Osenova (2002)	Reguły	$94,8\%^{m}$	3500 słów
Chorwacki	Vučković (2009)	Reguły	$96,56\%^{u}$	135 zdań
	Vučković i inni (2008)	Reguły	$92,31\%^{a}$	137 zdań
	Vučković i inni (2010)	Reguły	$86,4\%^{a}$	10 131 seg. (459 zd.)
	Vučković i inni (2010)	Reguły	$88,4\%^{t}$	10 131 seg. (459 zd.)
	Vučković i inni (2010)	Reguły	$91,3\%^{m}$	10 131 seg. (459 zd.)
Czeski	Grác i inni (2010)	Głęboki parser	$93,1\%^{m}$	300 zdań

Tabela 4.4. Wyniki testów płytkich parserów języków słowiańskich. Kolumna *F dla NP* cytuje wartości miary F dla rozpoznawania fraz rzeczownikowych. Wyniki oznakowane literą t uwzględniają błędy tagera; a oznacza, że tager dokonywał jedynie częściowego ujednoznacznienia, przez co parser dostawał segmenty wieloznaczne; m oznacza, że eksperymenty przeprowadzono na wzorcowym oznakowaniu morfosyntaktycznym (błędy tagera zostały zaniedbane), natomiast u oznacza, że autorowi rozprawy nie udało się dotrzeć do dokładniejszych informacji.

Tabela 4.4 pozwala zaobserwować duże różnice w wynikach między pracami. Część z tych różnic wynika zapewne z istotnych rozbieżności między definicjami fraz rzeczow-

nikowych, a co za tym idzie, z różnym stopniem trudności zadania znakowania fraz. Drugim istotnym źródłem różnic jest metoda oceny: w eksperymentach, gdzie uwzględniono błędy tagera (t), osiągnięto wyraźnie niższe wartości miary F niż w eksperymentach, gdzie błędy te zaniedbano (m). Widać to szczególnie wyraźnie w wynikach z prac (Vučković i inni, 2010) oraz (Osenova, 2002), gdzie eksperymenty przeprowadzono z osobna w wariancie uwzględniającym błędy tagera i w wariancie je zaniedbującym.

4.5. NKJP i Spejd a znakowanie fraz

W tym punkcie opiszemy, w jaki sposób można użyć parsera Spejd do oznakowania zdań (płaskimi) frazami, a także — w jaki sposób będziemy rzutować wzorcowe oznakowanie składniowe NKJP na takie frazy. Celem tych zabiegów jest uzyskanie danych do testów algorytmów znakowania fraz, jak również i przetestowanie Spejda wyposażonego w gramatykę NKJP jako narzędzia znakowania fraz.

Analizator składniowy Spejd pozwala na znakowanie grup składniowych i słów składniowych na podstawie gramatyki napisanej ręcznie. Struktura generowana przez ten analizator jest bardzo zbliżona do oryginalnej struktury składniowej zawartej w NKJP (por. 4.2). Istnieją dwie istotne różnice między tymi strukturami:

1. Oznakowanie składniowe NKJP zawiera czasami frazy nieciągłe (por. 4.2), podczas gdy Spejd nie ma możliwości wygenerowania struktur nieciągłych, gdyż zakłada użycie formalizmu zbliżonego do gramatyk regularnych.
2. W NKJP, z wyjątkiem grup składniowych odpowiadających całym zdaniom podrzędnym, pozostałe grupy składniowe nie zawierają w sobie innych grup. Oznakowanie wykonane przez Spejd zawiera takie zagnieżdżenia, odpowiadające kolejno aplikowanym regułom. Przykładowo, jeśli reguła gramatyki nakazuje oznakowanie grupy przyimkowej, jeśli wystąpi przyimek poprzedzający oznakowaną już grupę rzeczownikową, to wynikowa grupa rzeczownikowa zawierać będzie w sobie wspomnianą grupę rzeczownikową.

Pierwsza różnica nie stanowi dla nas istotnego problemu, gdyż, jak wspomnieliśmy, frazy nieciągłe w korpusie wzorcowym występują rzadko i dla uproszczenia i tak będziemy rozpatrywać każdy ciągły fragment nieciągłej frazy jako osobną frazę. Najprostszym rozwiązaniem drugiego problemu byłoby wybieranie zawsze frazy największej (frazy, która nie zawiera się w innej frazie). Rozwiązanie to nie jest jednak zadowalające ze względu na obecność fraz odpowiadającym całym zdaniom podrzędnym (CG i KG). Pojawienie się takiego zdania podrzędnego oznaczałoby, że żadna z fraz rzeczownikowych, czasownikowych i przymiotnikowych, które należą do takiego zdania, nie zostałaby przez nas zauważona. Rozwiązanie, które ostatecznie przyjęto, wiąże się z wyborem podzbioru fraz (ściślej rzecz biorąc, grup i słów składniowych), które będziemy w ogóle rozpatrywać w dalszych rozważaniach i eksperymentach. Zakładamy, że słowa składniowe i grupy składniowe rozpatrywane są z osobna. W obu przypadkach, pomijamy najpierw typy słów/grup, które nie należą do interesującego nas podzbioru, po czym wybieramy słowo/grupę największą. Rozwiązanie to stosujemy zarówno w stosunku do wyjścia Spejda (które zawiera liczne zagnieżdżenia), jak i wzorcowego oznakowania składniowego NKJP (pozwala to na uniknięcie wspomnianego problemu zdań podrzędnych — zdania te nie należą do rozpatrywanego przez nas podzbioru fraz).

Ze względu na duży zbiór rodzajów grup i słów składniowych, którymi oznakowano korpus NKJP, a także na „drobnoziarnisty" ich podział (por. punkt 4.2.3), zdecydowaliśmy się skupić jedynie na podzbiorze dostępnych grup i słów składniowych. Co więcej, dokonaliśmy także pewnych uogólnień polegających na scaleniu kilku typów grup i fraz w „zgrubne" typy, które nazywamy tutaj frazami rzeczownikowymi, przymiotnikowymi i czasownikowymi. Uproszczenia te mają przede wszystkim na celu ułatwienie przeprowadzenia eksperymentów, a także interpretacji ich wyników. Dodatkowym kryterium dokonywanych wyborów było pewne upodobnienie otrzymanych fraz do tych zdefiniowanych w korpusie KPWr.

Jak wspomnieliśmy w punkcie 4.2.3, słowa składniowe należące do klas czasownikowych możemy uznać za **frazy czasownikowe** (VP). W ten właśnie sposób będziemy je traktować: do VP zaliczamy wszystkie słowa składniowe, którym przypisano tagi należące do klas gramatycznych `Verbfin`, `Winien`, `Imps`, `Inf` lub `Pred`.

Jako **frazy rzeczownikowe** (NP) traktować będziemy zarówno grupy rzeczownikowe zdefiniowane w NKJP (NG[8]), grupy liczebnikowe (NumG), grupy przyimkowo-rzeczownikowe (PrepNG) oraz grupy przyimkowo-liczebnikowe (PrepNumG). Podobnie jak w KPWr (a także Grác i inni, 2010) zdecydowaliśmy się na złączenie rzeczywistych fraz rzeczownikowych z frazami przyimkowymi. Choć decyzja ta może wydawać się nieco kontrowersyjna, chcieliśmy po pierwsze uzyskać definicje bliższe tym z KPWr, po drugie zaś rozdzielenie rzeczywistych fraz rzeczownikowych od fraz przyimkowych za pomocą prostej reguły wydaje się zadaniem trywialnym.

Analogicznie, jako **frazy przymiotnikowe** (AdjP) traktować będziemy grupy przymiotnikowe (AdjG i AdjGk) oraz grupy przyimkowo-przymiotnikowe (PrepAdjG).

Opisany powyżej sposób rzutowania grup i słów składniowych we frazy stosujemy zarówno w stosunku do danych wzorcowych z NKJP, jak również i do wyjścia Spejda.

4.6. Algorytm: znakowanie fraz w oparciu o uczenie na pamięć

W tym punkcie przedstawiamy pierwszy z algorytmów znakowania fraz, który testować będziemy na danych z korpusów języka polskiego. Sam w sobie algorytm nie jest nowy — został on bezpośrednio zainspirowany pracą (Veenstra i van den Bosch, 2000), natomiast elementem nowym jest użycie bogatego zbioru cech, uwzględniającego specyfikę języków słowiańskich oraz pozycyjny charakter tagsetu NKJP.

Algorytm sprowadza się do znakowania kolejnych segmentów za pomocą znaczników IOB2 (por. str. 84) w oparciu o decyzje klasyfikatora pamięciowego. Zadaniem rozpatrywanym przez nas jest znakowanie fraz kilku typów. Jak wspomnieliśmy na stronie 84, istnieją dwa podejścia: znakowanie każdej frazy w izolacji lub znakowanie wszystkich fraz jednocześnie przy użyciu rozszerzonego zestawu znaczników. Prezentowany tutaj algorytm zakłada rozwiązanie pierwsze. Podejście takie wydaje się prostsze. Co więcej, w korpusie KPWr mamy do czynienia z nakładającymi się na siebie frazami (np. frazy AgP mogą pokrywać się z frazami NP), a zatem i tak konieczne byłoby wydzielenie przynajmniej dwóch podzbiorów fraz, które rozpatrywane byłyby jako osobne problemy

[8] W rzeczywistych danych pojawiają się jeszcze bardziej szczegółowe rozróżnienia w obrębie NG, PrepNG, NumG itp.. Przykładowo, grupy rzeczownikowe opisujące adresy nazywają się `NGadres`. Dla czytelności, wszystkie takie podtypy nazywamy tutaj w skrócie NG (PrepNG, NumG, ...), choć w rzeczywistości chodzi o grupy, których nazwa zaczyna się od NG (PrepNG, NumG, ...).

znakowania fraz. Dalszy opis algorytmu zakładać będzie więc, że znakowana jest fraza jednego typu. Tam, gdzie będzie mowa o wynikach znakowania kilku typów fraz, oznaczać to będzie, że dla każdej frazy przeprowadzono z osobno uczenie i znakowanie wyuczonym modelem.

Algorytm przedstawiany w tym punkcie jest podobny do algorytmu ujednoznaczniania morfosyntaktycznego w oparciu o uczenie na pamięć (punkt 2.7). Różnice są następujące:

1. algorytm ujednoznaczniania dostosowany jest do tagsetów pozycyjnych, podczas gdy przedstawiany tutaj algorytm znakowania fraz opiera się na reprezentacji IOB2, składającej się jedynie z trzech atomowych znaczników (`I`, `O` i `B`),
2. nie stosujemy tutaj żadnego odpowiednika analizy morfosyntaktycznej, a zadaniem klasyfikatora nie jest wykreślanie znaczników, lecz ostateczny ich wybór (wybieramy zawsze spośród trzech możliwości),
3. z oczywistych przyczyn nie stosujemy tutaj znakowania warstwowego.

Ze względu na użyty klasyfikator pamięciowy, algorytm tutaj opisywany będziemy odtąd skrótowo nazywać **algorytmem znakowania fraz MBL**.

4.6.1. Uczenie

Algorytm uczony jest na tekście oznakowanym morfosyntaktycznie, gdzie segmentom przypisano dodatkowo znaczniki IOB2 określające wzorcowe oznakowanie frazą rozpatrywanego typu. Wynikiem uczenia jest lista najczęstszych form wyrazowych zawierająca F pozycji oraz baza przypadków uczących. Procedurę uczenia opisujemy poniżej jako algorytm 5. Algorytm rozpoczyna się od zebrania listy F najczęstszych form wyrazowych z korpusu uczącego.

Główna pętla algorytmu polega na generowaniu przypadków uczących na podstawie kolejnych segmentów. Każdy przypadek uczący to para (*wektor cech*, *decyzja*), gdzie wektor cech to ciąg wartości otrzymanych wskutek aplikacji kolejnych cech do otoczenia danego segmentu, a decyzja to znacznik IOB2 pobrany z wzorcowego oznakowania składniowego.

Algorytm 5 Uczenie algorytmu znakowania fraz MBL

Dane: korpus uczący *korp* oznakowany morfosyntaktycznie i znacznikami IOB2
zbiór cech *cechy*
Wyniki: baza przypadków uczących B
zbierz F najczęstszych form z korpusu uczącego *korp*
for *zdanie* $\in$ *korp* **do**
for *seg* $\in$ *zdanie* **do**
wek_cech $\leftarrow$ $[f(seg, zdanie)$ **for** $f \in cechy]$
decyzja $\leftarrow$ prawidłowy znacznik IOB2 przypisany segmentowi *seg*
dodaj do bazy B przykład uczący (*wek_cech*, *decyzja*)
end for
end for

4.6.2. Znakowanie

Znakowanie za pomocą wyuczonego modelu zostało przedstawione jako algorytm 6. Wejściem algorytmu jest tekst oznakowany morfosyntaktycznie. Wynikiem działania jest przypisanie każdemu segmentowi znacznika IOB2 odpowiadającemu problemowi znakowania fraz danego typu. Procedura **klasyfikuj** sprowadza się do aplikacji klasyfikatora pamięciowego do podanego wektora cech.

Algorytm 6 Znakowanie pojedynczego zdania za pomocą algorytmu MBL

Dane: *zdanie* oznakowane morfosyntaktycznie
zbiór cech *cechy*
baza przypadków uczących B
Wyniki: *zdanie* oznakowane morfosyntaktycznie z przypisanymi znacznikami IOB2
for $seg \in zdanie$ **do**
$wek\_cech \leftarrow [f(seg, zdanie)$ **for** $f \in cechy(a)]$
decyzja $\leftarrow$ klasyfikuj (B, wek_cech)
przypisz znacznik IOB2 *decyzja* segmentowi *seg*
end for

4.6.3. Parametry i cechy

Algorytm MBL parametryzowalny jest zarówno parametrami samego klasyfikatora pamięciowego, jak i zestawem cech. Wartości parametrów oraz zestaw cech są praktycznie identyczne z tymi zaproponowanymi na potrzeby ujednoznaczniania morfosyntaktycznego w oparciu o uczenie na pamięć (algorytm ze strony 44). Używane przez nas parametry klasyfikatora pamięciowego są następujące:

— liczba najbliższych sąsiadów $k = 9$,
— miara podobieństwa Scotta–Salzberga — *Modified Value Difference*,
— schemat głosowania Dudaniego — *Inverse Linear*.

Podobnie jak w przypadku tagera pamięciowego, używamy implementacji klasyfikatora pamięciowego z pakietu TiMBL (Daelemans i inni, 2010a). Powyższe parametry można włączyć w tymże pakiecie za pomocą opcji `-mM -k9 -dIL`.

Zestaw cech zawiera następujące pozycje:

1. Wartość klasy gramatycznej dla każdego segmentu z okna $(-3, -2, -1, 0, +1, +2)$ oraz wartości trzech atrybutów: przypadka, rodzaju i liczby dla segmentów z tego samego okna. Jako że tekst wejściowy jest ujednoznaczniony morfosyntaktycznie, wartościami tych cech będą zawsze pojedyncze symbole (w przypadku algorytmu WMBT pojawiały się zbiory kilkuelementowe).
2. Cechy leksykalne: formy wyrazowe wszystkich segmentów z okna $(-3, \ldots, 2)$ sprowadzone do małych liter. Formy wyrazowe filtrowane są do F najczęstszych, pobranych z listy tworzonej podczas uczenia; formy niewystępujące na liście zamieniane są na symbol reprezentujący formę rzadką. Przyjmujemy wartość $F = 500$.
3. Cechy o wartościach prawda/fałsz sprawdzające uzgodnienie wartości liczby, rodzaju i przypadka. Testy te mają za zadanie ułatwić rozpoznanie fraz uzgodnionych. Użyto następujących cech:
   a) test na uzgodnienie między pozycjami -1 i 0: `agrpp(-1,0,{nmb,gnd,cas})`,
   b) test na uzgodnienie między pozycjami 0 i $+1$,

c) test na *słabe uzgodnienie przedziałowe* (`wagr`; por. punkt 2.6) między segmentami z okna $(-2, -1, 0)$,
d) j.w., okno $(-1, 0, +1)$,
e) j.w., okno $(0, +1, +2)$.

4. Dwa testy na graficzną postać formy wyrazowej na pozycji 0: czy zaczyna się małą literą oraz czy zaczyna się wielką literą.

Wszystkie te cechy zostały zapisane w postaci wyrażeń funkcyjnych w formalizmie WCCL. Wyrażenia te zamieszczono w punkcie C.2 dodatku C. Dla wszystkich typów fraz używamy dokładnie tego samego zestawu cech i wartości parametrów.

4.7. Algorytm: znakowanie fraz w oparciu o drzewa decyzyjne

Algorytm ten jest prostą modyfikacją algorytmu MBL opisanego w poprzednim punkcie. Modyfikacja polega na zastąpieniu klasyfikatora pamięciowego techniką indukcji drzew decyzyjnych (por. str. 28). Modyfikację tę będziemy odtąd nazywać **algorytmem znakowania fraz DT**.

Faza uczenia algorytmu wygląda identycznie jak w poprzednim punkcie z jednym rozszerzeniem: po zebraniu wszystkich przypadków uczących uruchamiany jest algorytm indukcji drzew decyzyjnych C4.5. Wyuczony model algorytmu DT jest zatem pojedynczym drzewem decyzyjnym, którego zadaniem jest zwrócenie znacznika IOB2 dla podanego wektora cech. Modyfikacja algorytmu znakowania sprowadza się do użycia tego właśnie drzewa decyzyjnego zamiast używanego w algorytmie MBL klasyfikatora pamięciowego.

Stosowany zbiór cech jest identyczny jak w przypadku algorytmu MBL. Nasza implementacja algorytmu znakowania fraz DT korzysta z implementacji drzew decyzyjnych zawartej w pakiecie NLTK (Bird i inni, 2009).

4.8. Algorytm: znakowanie fraz w oparciu o warunkowe pola losowe

Algorytmy MBL i DT zakładały, że każdy segment klasyfikowany jest niezależnie od pozostałych. Użycie warunkowych pól losowych pozwala na klasyfikację całych ciągów segmentów odpowiadających poszczególnym zdaniom, wskutek czego otrzymujemy ciągi znaczników IOB2. Opisywany tutaj algorytm zbliżony jest do metody zaproponowanej w pracy (Sha i Pereira, 2003). Podobnie jak w poprzednich punktach, elementem nowym jest zbiór cech dostosowany do specyfiki języków słowiańskich.

Algorytm opisywany w tym punkcie nazywać będziemy odtąd **algorytmem CRF**.

4.8.1. Uczenie

Procedura opisana jako algorytm 7 polega na zgromadzeniu danych uczących i wyuczeniu na nich modelu CRF. W stosunku do algorytmu 5 (uczenia MBL) pojawiają się dwie różnice: zapamiętujemy pomocniczy znacznik końca zdania oraz rezygnujemy z generowania listy form najczęstszych. Pierwsza modyfikacja wynika z chęci klasyfikacji całych zdań na raz, druga zaś jest inspirowana pracą (Sha i Pereira, 2003) oraz domyślnym zestawem cech pakietu CRF++ (Kudo, 2005), gdzie formy nie były filtrowane.

Algorytm 7 Uczenie algorytmu znakowania fraz CRF

Dane: korpus uczący *korp* oznakowany morfosyntaktycznie i znacznikami IOB2
zbiór cech *cechy*
Wyniki: wyuczony model CRF
for *zdanie* ∈ *korp* **do**
for *seg* ∈ *zdanie* **do**
wek_cech ← [f(*seg*, *zdanie*) **for** $f \in$ *cechy*]
decyzja ← prawidłowy znacznik IOB2 przypisany segmentowi *seg*
zapamiętaj przykład uczący (*wek_cech*, *decyzja*)
end for
zapamiętaj znacznik końca zdania
end for
ucz CRF na zapamiętanych przykładach uczących

4.8.2. Znakowanie

Znakowanie za pomocą wyuczonego modelu CRF zostało przedstawione jako algorytm 8. Różnicą w stosunku do algorytmu 6 jest poprzedzające klasyfikację zgromadzenie ciągu wektorów cech wygenerowanych dla kolejnych segmentów z jednego zdania. Ten ciąg wektorów oznaczono jako *repr_zdania*. W opisie algorytmu użyto funkcji **zip** odpowiadającej standardowej operacji znanej z programowania funkcyjnego polegającej na zamianie pary list w listę par. Procedura **klasyfikuj** użyta tutaj polega na klasyfikacji całego ciągu wektorów cech (reprezentacji zdania), wskutek czego otrzymujemy ciąg znaczników IOB2. Zadaniem drugiej pętli algorytmu jest przypisanie tych znaczników kolejnym segmentom zdania.

Algorytm 8 Znakowanie pojedynczego zdania za pomocą algorytmu CRF

Dane: *zdanie* oznakowane morfosyntaktycznie
zbiór cech *cechy*
wyuczony model CRF
Wyniki: *zdanie* oznakowane morfosyntaktycznie z przypisanymi znacznikami IOB2
repr_zdania ← []
for *seg* ∈ *zdanie* **do**
wek_cech ← [f(*seg*, *zdanie*) **for** $f \in$ *cechy*(a)]
dodaj *wek_cech* na końcu listy *repr_zdania*
end for
for (*seg*, *decyzja*) ∈ **zip**(*zdanie*, klasyfikuj(*repr_zdania*)) **do**
przypisz znacznik IOB2 *decyzja* segmentowi *seg*
end for

4.8.3. Parametry i cechy

Proponowany zastaw cech jest prawie identyczny z cechami zaproponowanymi na potrzeby algorytmów MBL i DT. Różnicą jest wspomniana rezygnacja z filtrowania

form wyrazowych do najczęstszych. Taki zestaw możemy sformalizować za pomocą szablonów cech z tabeli 4.5 (ideę szablonów cech dla warunkowych pól losowych omówiliśmy na stronie 23).

1. $s_i = \sigma_1 \wedge s_{i-1} = \sigma_2 \wedge w_i = \phi$
2. $s_i = \sigma \wedge w_{i+N} = \phi$ dla $N \in \{-2, -1, 0, 1, 2\}$
3. $s_i = \sigma \wedge w_{i+N-1} = \phi_1 \wedge w_{i+N} = \phi_2 \wedge w_{i+N+1} = \phi_3$ dla $N \in \{-1, 0, 1\}$
4. $s_i = \sigma \wedge \texttt{class}_{i+N} = \tau$ dla $N \in \{-2, -1, 0, 1, 2\}$
5. $s_i = \sigma \wedge \texttt{class}_{i+N} = \tau_1 \wedge \texttt{class}_{i+N+1} = \tau_2$ dla $N \in \{-2, -1, 0, 1\}$
6. $s_i = \sigma \wedge \texttt{cas}_{i+N} = \tau$ dla $N \in \{-2, -1, 0, 1, 2\}$
7. $s_i = \sigma \wedge \texttt{gnd}_{i+N} = \tau$ dla $N \in \{-2, -1, 0, 1, 2\}$
8. $s_i = \sigma \wedge \texttt{nmb}_{i+N} = \tau$ dla $N \in \{-2, -1, 0, 1, 2\}$
9. $s_i = \sigma \wedge \texttt{agr}(i-1, i)$
10. $s_i = \sigma \wedge \texttt{agr}(i, i+1)$
11. $s_i = \sigma \wedge \texttt{agr}(i+N-1, i+N, i+N+1)$ dla $N \in \{-1, 0, 1\}$
12. $s_i = \sigma \wedge \texttt{islower}(i)$
13. $s_i = \sigma \wedge \texttt{isupper}(i)$

Tabela 4.5. Propozycja szablonów cech dla znakowania fraz w języku polskim

W powyższym zapisie używamy tej samej konwencji, którą przedstawiliśmy przy okazji opisu szablonów cech stosowanych przez Sha i Pereira (2003), por. str. 87. Dodatkowo, używamy symboli `class`, `cas`, `gnd`, `nmb`, które oznaczają, odpowiednio, wartości klasy gramatycznej, przypadka, rodzaju i liczby. Symbol `agr` oznacza test na uzgodnienie gramatyczne co do libczy, rodzaju i przypadka (testy te rozumiane są tak samo jak w punkcie 4.6.3), natomiast symbole `islower` i `isupper` oznaczają testy na „kształt" formy wyrazowej: czy zaczyna się małą literą, czy zaczyna się wielką literą.

Powyższe szablony i cechy zamieszczono w punkcie C.3 dodatku C.

Nasza implementacja algorytmu znakowania fraz CRF korzysta z pakietu CRF++ (Kudo, 2005) z domyślnymi wartościami parametrów.

4.9. Podsumowanie

W tym rozdziale omówiliśmy problem znakowania fraz w kontekście języka polskiego i innych języków słowiańskich. Przedstawiliśmy dostępne dla języka polskiego korpusy oznakowane frazami. Omówiliśmy problem definicji (płaskich) fraz w kontekście języków słowiańskich, a także dokonaliśmy przeglądu stosowanych na potrzeby takich języków definicji i wytycznych znakowania. Przedstawiono również przegląd metod znakowania fraz, zarówno stosowanych dla języka angielskiego, jak i dla języków słowiańskich. Przedstawiono także trzy algorytmy znakowania fraz znane z literatury dostosowane do języka polskiego, a także sposób dostosowania wyjścia płytkiego parsera Spejd do przyjętej przez nas definicji zadania znakowania fraz.

Elementami oryginalnymi tego rozdziału są:

1. przegląd definicji fraz i wytycznych znakowania korpusów frazami stosowanych dla języków słowiańskich[9];
2. wytyczne znakowania fraz w Korpusie Języka Polskiego Politechniki Wrocławskiej, będące wynikiem współpracy autora rozprawy z dwoma lingwistami: Markiem Maziarzem oraz Janem Wieczorkiem (Radziszewski i inni, 2012);
3. dostosowanie trzech metod znakowania fraz opartych na technikach maszynowego uczenia do specyfiki języka polskiego; dostosowanie to polegało na zaproponowaniu zestawu cech korzystającego z pozycyjnego charakteru tagsetu, a także uwzględniającego ważną rolę składniową pełnioną w języku polskim przez kategorie gramatyczne liczby, rodzaju i przypadka.

[9] Wszystkie dostępne opracowania, do których udało się dotrzeć autorowi rozprawy, ograniczone są do jednego języka słowiańskiego.

Rozdział 5

Eksperymentalna ocena algorytmów znakowania fraz

5.1. Cel

Zasadniczym celem eksperymentów opisanych w tym rozdziale jest porównanie osiągów kilku algorytmów znakowania fraz na dwóch korpusach: przygotowanym przy udziale autora tej rozprawy Korpusie Języka Polskiego Politechniki Wrocławskiej (KPWr) oraz Narodowym Korpusie Języka Polskiego (NKJP). Porównanie to umożliwia wskazanie algorytmu, który daje najlepsze wyniki. Eksperymenty przeprowadzone na korpusie NKJP powzwalają na porównanie osiągów testowanych algorytmów opartych na technikach maszynowego uczenia z osiągami polskiego parsera Spejd wyposażonego w gramatykę pisaną ręcznie (gramatykę NKJP).

Definicje fraz przyjęte w obu korpusach różnią się w istotny sposób (por. rozdział 4.2). Mimo to, wyniki te pozwalają na poglądową ocenę skali problemu znakowania fraz w języku polskim, a także na (również poglądowe) porównanie wyników osiągniętych dla języka polskiego z wynikami parserów opartych o gramatyki pisane ręcznie opracowanych dla innych języków słowiańskich (por. rozdz. 4.4.3). Co więcej, o ile nam wiadomo, opisane tu eksperymenty są pierwszymi próbami oceny algorytmów znakowania fraz opartych na technikach maszynowego uczenia na danych z języka słowiańskiego[1]. Podane tu wyniki mogą zatem stanowić punkt odniesienia dla przyszłych eksperymentów.

Dodatkowym celem przeprowadzonych badań jest wskazanie istotnej roli pełnionej przez tager w procesie znakowania fraz. Pokazujemy, że zmiana algorytmu tagera przynosi istotne różnice w wynikach znakowania fraz. Pokazujemy również, że jeśli ocena analizatora składniowego ma być rzetelna, nie może zaniedbywać ani błędów modułu ujednoznaczniania morfosyntaktycznego, ani nawet modułu segmentacji i analizy morfologicznej. Testy wykazują również, że użycie zaproponowanego w rozdziale 2 algorytmu znakowania morfosyntaktycznego w oparciu o uczenie pamięciowe przynosi istotną statystycznie poprawę w osiągach analizatora w stosunku do wyników

[1] Wstępne eksperymenty ze znakowaniem fraz rzeczownikowych w języku polskim zostały opisane w pracy (Radziszewski i Piasecki, 2010). Eksperymenty te stanowiły badania wstępne w ramach prac nad tą rozprawą i dlatego nie omawiamy ich tutaj bardziej szczegółowo.

osiągniętych przy pomocy tagera PANTERA. Oceniamy też wpływ zastosowania proponowanego modułu odgadywania słów nieznanych na wyniki znakowania fraz.

5.2. Kryterium oceny i stosowane zbiory danych

Eksperymenty przeprowadzono na dwóch zbiorach danych:

1. Ręcznie oznakowanej frazami części korpusu KPWr z dnia 30.03.2012 (w trakcie pisania rozprawy proces znakowania korpusu był w toku, użyto więc dostępnej części).
2. Części podkorpusu milionowego NKJP 1.0, która została oznakowana płytką strukturą składniową.

W tabeli 5.1 podsumowano statystyki związane z tymi korpusami: liczbę segmentów, zdań, a także liczbę wystąpień fraz podanych typów. Omawiana tutaj część korpusu NKJP jest nieco mniejsza niż cały *podkorpus milionowy NKJP*, który został oznakowany na poziomie morfosyntaktycznym (1,2 mln segmentów; por. tabela 3.1 ze strony 57). Wynika to z faktu, że nie wszystkie dokumenty korpusu, które są oznakowane na poziomie morfosyntaktycznym, zostały oznakowane na poziomie składniowym (choć i tak część oznakowana na obu poziomach zawiera ponad milion segmentów). W przypadku korpusu KPWr podane tutaj nazwy fraz odnoszą się bezpośrednio do przyjętych definicji (por. punkt 4.2.2); w przypadku NKJP nazwy te odnoszą się do wyniku scalenia „drobnoziarnistych" grup składniowych z NKJP zgodnie z opisem z punktu 4.5. Definicje tych samych fraz w obu korpusach są więc do siebie podobne (osiągięcie tego podobieństwa było głównym celem wspomnianego scalenia), lecz mimo to występują istotne różnice.

W obu korpusach zdarzają się nieciągłe frazy. Jak wspomnieliśmy na stronie 77, każda fraza nieciągła traktowana jest tutaj jako zbiór dwóch lub więcej fraz — jako osobną frazę traktujemy każdy ciągły jej fragment. Co więcej, jeśli wystąpią dwie lub więcej frazy, które częściowo nakładają się na siebie, arbitralnie przyporządkujemy segmenty konfliktowe (należące do więcej niż jednej frazy) do frazy, której początek znajduje się najbliżej początku zdania. Rozwiązanie to jest oczywiście bardzo prymitywną heurystyką, którą uznajemy za sensowne rozwiązanie tylko dlatego, że frazy nakładające się występują bardzo rzadko.

Dla uproszczenia, te właśnie zbiory danych nazywamy w tym rozdziale skrótowo korpusami KPWr i NKJP.

	KPWr	**NKJP**
Segmentów	29 116	1 064 370
Zdań	1 842	72 720
AgP	10 930	*brak*
NP	5 722	294 287
AdjP	390	25 823
VP	2 321	325 898

Tabela 5.1. Statystyki korpusów oznakowanych frazami: KPWr oraz NKJP.

Warto zwrócić też uwagę na rozmiary korpusów KPWr i NKJP, w szczególności w kontekście dotychczas przeprowadzonych badań dla języków słowiańskich (tabela 4.4 ze strony 91). Korpus KPWr jest prawie 3 razy większy niż największy korpus używany w przedstawionych tam badaniach (korpus języka chorwackiego), natomiast korpus NKJP jest od niego ponad 100 razy większy.

Jako kryterium oceny algorytmów uznajemy miarę F, tj. średnią harmoniczną między *dokładnością* a *kompletnością* (zgodnie z definicjami z punktu 4.3). Oprócz tego podajemy wartości dokładności oraz kompletności, gdyż umożliwiają one wyciągnięcie dodatkowych wniosków.

Jak wspomnieliśmy w punkcie 4.3, rzetelna ocena parsera zakładającego użycie tekstu oznakowanego morfosyntaktycznie wymaga uwzględnienia wpływu błędów tagera na osiągi parsera. Ocena taka jest stosunkowo prosta, jeśli wzorcowy korpus oznakowany frazami został przygotowany przy pomocy tagera. Sytuacja taka miała miejsce w przypadku korpusu KPWr: zebrano dokumenty w postaci czystego tekstu podzielonego jedynie na akapity, po czym tekst ten poddano znakowaniu za pomocą tagera WMBT. Mimo oczywistej obecności błędów tagera, oznakowanie to nie było poddane żadnym korektom. Zadaniem lingwistów było jedynie wzbogacenie istniejącego korpusu o wzorcowe oznakowanie frazami. Zatem ocena parsera uczonego i testowanego na korpusie uprzednio oznakowanym przez tager będzie uwzględniać wpływ błędów tagera. Testy parsera przeprowadzone na korpusie KPWr opisane poniżej przeprowadzono przy takim właśnie założeniu.

W przypadku korpusu NKJP sytuacja jest bardziej skomplikowana: korpus NKJP został ręcznie oznakowany zarówno na poziomie morfosyntaktycznym (zadaniem lingwistów była nawet ręczna korekta segmentacji), jak i na poziomie płytkiej analizy składniowej. Co więcej, korpus ten jest jednocześnie jedynym dostępnym korpusem zawierającym wzorcowe oznakowanie morfosyntaktyczne w tagsecie NKJP. Aby uniknąć sytuacji, w której tager uczony i testowany jest na tych samych danych (testowany co prawda nie wprost, lecz jako składnik ciągu przetwarzania zakończonego modułem znakowania fraz), konieczne było włączenie uczenia tagera w schemat oceny analizatora. Idea jest następująca: dany podział korpusu na część uczącą i testową wykorzystywany jest zarówno przez tager, jak i moduł znakowania fraz. Faza uczenia polega na uczeniu obu komponentów na części uczącej. Test polega na sprowadzeniu części testowej do czystego tekstu, oznakowaniu za pomocą wyuczonego tagera, po czym uruchomieniu na wynikowym korpusie modułu znakowania fraz. Dopiero tak powstały wynik jest porównywany z pierwotną częścią testową, zawierającą wzorcowe oznakowanie. Tę metodę oceny nazywać będziemy **testami uwzględniającymi błędy znakowania**. Łatwo zauważyć analogię między tą metodą oceny, a oceną tagerów proponowaną w punkcie 2.5. Idea jest ta sama: chcemy w ten sposób uwzględnić błędy popełniane na wszystkich etapach, począwszy od segmentacji. Warto też podkreślić, że prawidłowe wykonanie tej procedury wymaga porównania oznakowania frazami między korpusem wzorcowym a wynikowym również w sytuacjach, gdzie segmentacja uległa zmianie. Porównanie takie jest łatwiejsze do wykonania, jeśli oba korpusy sprowadzimy do postaci uproszczonej, zawierającej jedynie czysty tekst, tj. ciąg znaków, a oznakowanie frazami zapiszemy jako indeksy początku i końca fraz. Postać taka jest wygodna, gdyż czysty tekst pochodzący z obu korpusów jest identyczny, a zatem możliwe jest bezpośrednie porównanie pośredniej reprezentacji fraz.

Dla pokazania, jak ważne jest przeprowadzenie oceny całego systemu rozpatrujemy jeszcze podejście pośrednie, gdzie uwzględniamy co prawda użycie tagera, lecz zaniedbujemy błędy analizatora morfosyntaktycznego i modułu segmentacji. Pokazujemy, że podejście takie jest jedynie półśrodkiem. Procedura wykonania tego testu była następująca: uczenie przebiegało identycznie, jak w testach całego systemu, tj. część ucząca posłużyła do wyuczenia zarówno tagera, jak i modułu znakowania fraz. Test przeprowadzono w uproszczony sposób, tj. użyto wzorcowej segmentacji i analizy morfosyntaktycznej z części testowej, zadaniem tagera był natomiast jedynie wybór prawidłowej interpretacji (ujednoznacznianie morfosyntaktyczne). Tę metodę testów nazywać będziemy **testami uwzględniającymi błędy ujednoznaczniania**.

Tam, gdzie test wymagał użycia tagera, stosujemy opisany w rozdziale 2 tager WMBT wraz z modułem odgadywania słów nieznanych; tager uczony jest z zastosowaniem zaproponowanej techniki ponownej analizy morfosyntaktycznej danych uczących.

5.3. Metodyka analizy wyników

Podobnie jak w rozdziale 3, wszystkie eksperymenty przeprowadzono w oparciu o 10-krotny sprawdzian krzyżowy. Kryterium porównania osiągów analizatorów składniowych były wartości **miary F**. Ocenę istotności różnic przeprowadzono przy pomocy testu t-Studenta dla prób zależnych, zakładając poziom istotności $\alpha = 0,05$ (czyli w sposób identyczny z tym zastosowanym w rozdziale 3, por. str. 58).

5.4. Ocena analizatorów na korpusie KPWr

W tabeli 5.2 przedstawiamy wyniki oceny analizatorów składniowych na korpusie KPWr. Podano wartości używanych miar (P — dokładność, R — kompletność, F — miara F) dla wszystkich czterech typów fraz: AgP (fraz uzgodnionych), NP (fraz rzeczownikowych), AdjP (przymiotnikowych) i VP (czasownikowych). Oznakowanie morfosyntaktyczne korpusu KPWr zostało wykonane w oparciu o tager WMBT wyuczony na całym NKJP 1.0. Testy algorytmów znakowania fraz uwzględniają zatem wpływ błędów tagera.

Alg.	**AgP**			**NP**		
	P	R	F	P	R	F
DT	75,84%	77,88%	76,84%	46,86%	56,62%	51,27%
MBL	81,49%	83,71%	82,58%	60,87%	66,44%	63,52%
CRF	84,52%	84,96%	84,74%	71,13%	67,88%	69,46%

Alg.	**AdjP**			**VP**		
	P	R	F	P	R	F
DT	7,74%	16,14%	10,41%	74,65%	83,36%	78,75%
MBL	21,40%	23,95%	22,33%	80,71%	84,17%	82,39%
CRF	61,54%	34,50%	43,73%	85,08%	83,13%	84,09%

Tabela 5.2. Porównanie algorytmów znakowania fraz na korpusie KPWr.

Osiągi algorytmów różnią się znacznie w zależności od frazy. Można zaobserwować związek między wartościami miary F a oczekiwaną trudnością rozpoznawania fraz: frazy AgP są stosunkowo prostymi konstrukcjami gramatycznymi, podczas gdy rozpoznanie fraz rzeczownikowych (NP) może wymagać podjęcia trudnych decyzji (por. rozdział 4.2). Prawdopodobną przyczyną niskiej skuteczności rozpoznawania fraz przymiotnikowych (AdjP) jest niewielka ich liczba w korpusie.

Ciekawą obserwacją są dobre osiągi algorytmu CRF (warunkowe pola losowe) — osiągi te są w większości przypadków istotnie lepsze od osiągów pozostałych algorytmów. We wszystkich przypadkach osiągi algorytmu opartego na uczeniu pamięciowym (MBL) przewyższają osiągi analizatorów opartych na indukcji drzew decyzyjnym (DT). Szczegółowe porównanie różnic pomiędzy obserwowanymi wartościami miary F przedstawiamy poniżej. Do oznaczenia różnic istotnych statystycznie używamy symbolu '$\gg$'; pozostałe różnice oznaczamy symbolem $>$.

1. Dla fraz *AgP*, *NP* i *AdjP* wszystkie różnice są istotne statystycznie:
   — $F_{\mathrm{CRF}} \gg F_{\mathrm{MBL}}$
   — $F_{\mathrm{MBL}} \gg F_{\mathrm{DT}}$
   — $F_{\mathrm{CRF}} \gg F_{\mathrm{DT}}$
2. Dla frazy *VP* różnica między algorytmem CRF i MBL jest nieistotna:
   — $F_{\mathrm{CRF}} > F_{\mathrm{MBL}}$
   — $F_{\mathrm{MBL}} \gg F_{\mathrm{DT}}$
   — $F_{\mathrm{CRF}} \gg F_{\mathrm{DT}}$

5.5. Ocena analizatorów na NKJP

W tabeli 5.3 przedstawiono wyniki oceny analizatorów składniowych na frazach pozyskanych z korpusu NKJP: NP (rzeczownikowych), AdjP (przymiotnikowych) i VP (czasownikowych). Testy przeprowadzono w sposób opisany w punkcie 4.3: korpus testowy zamieniono na czysty tekst, po czym dokonano segmentacji, analizy morfologicznej i ujednoznaczniania za pomocą modelu tagera wyuczonego na danych uczących. Przypomnijmy, że zabiegi te były konieczne, by uzyskać przybliżenie rzeczywistego odsetka błędów analizatora składniowego, uwzględniając wpływ błędów tagera na wyniki analizatora składniowego (korpus NKJP zawiera wzorcowe oznakowanie morfosyntaktyczne wykonane przez lingwistów, zabieg ten miał na celu symulację przetwarzania tekstu nieoznakowanego).

W eksperymentach uwzględniliśmy również wyniki osiągnięte dzięki zastosowaniu parsera regułowego Spejd (Przepiórkowski, 2008) oraz zestawu reguł napisanych ręcznie na potrzeby znakowania składniowego NKJP (Waszczuk i inni, 2010). Sprowadzenie wyjścia Spejda do płaskich fraz zostało przeprowadzone zgodnie z opisem z punktu 4.5.

Przedstawione wartości miar są stosunkowo wysokie, w szczególności wyraźnie wyższe niż w przypadku korpusu KPWr. Należy podkreślić istotne różnice w przyjętych definicjach fraz między korpusami. Przyjęta tutaj definicja fraz NP ma stopień złożoności pomiędzy frazami AgP a NP z korpusu KPWr (por. rozdział 4.2). Z drugiej strony, użyty w eksperymentach podkorpus milionowy NKJP jest 37 razy większy od dostępnego korpusu KPWr. Jest to prawdopodobnym wytłumaczeniem dużo lepszych

wyników uzyskanych na NKJP niż KPWr. Drugim prawdopodobnym powodem tej różnicy był większy rygor w procesie znakowania NKJP — jest prawdopodobne, że oznakowanie KPWr zawiera znacząco większą liczbę błędów i niespójności.

Alg.	**NP**		
	P	R	F
DT	75,07%	81,01%	77,93%
MBL	75,14%	81,06%	77,99%
CRF	86,94%	86,75%	86,85%
Spejd	78,22%	81,48%	79,82%

Alg.	**AdjP**			**VP**		
	P	R	F	P	R	F
DT	58,77%	62,64%	60,64%	87,96%	89,53%	88,74%
MBL	58,96%	71,97%	64,82%	89,33%	90,39%	89,85%
CRF	75,42%	81,75%	78,45%	97,26%	97,58%	97,42%
Spejd	56,11%	83,06%	66,97%	96,21%	96,91%	96,56%

Tabela 5.3. Porównanie algorytmów znakowania fraz na korpusie NKJP.

Podobnie jak w przypadku testów na KPWr, osiągi algorytmów na korpusie NKJP różnią się znacznie w zależności od frazy. Zdecydowanie najlepiej wypada rozpoznawanie fraz czasownikowych (VP). Wartości miary F dla rozpoznawania fraz przymiotnikowych (AdjP) są znowu niskie — najlepszy wynik to 78,45%, choć znacznie wyższe niż w przypadku korpusu KPWr (43,73%). Można podejrzewać, że zadecydował tu głównie rozmiar korpusu, ale w pewnym stopniu także prostsza definicja tych fraz w NKJP.

Najciekawszą obserwacją wydają się bardzo dobre osiągi algorytmu CRF. Wartości miary F nie tylko przewyższają wartości odnotowane dla pozostałych testowanych algorytmów opartych na maszynowym uczeniu (DT i MBL), lecz przewyższają również wyniki parsera Spejd wyposażonego w reguły pisane ręcznie. Wyniki te są zaskakujące, gdyż reguły dla parsera Spejd pisane były pod kątem tego właśnie korpusu NKJP, a korekta reguł przeplatała się z korektą oznakowania korpusu (por. punkt 4.5) — można powiedzieć, że w tym teście Spejd był faworyzowany. Należy w tym miejscu zaznaczyć, że Spejd nie jest typowym narzędziem znakującym frazy, a jego wyjście zawiera więcej użytecznej informacji niż same granice fraz (m.in. *słowa składniowe* z przypisanymi tagami oraz nadrzędniki fraz, por. rozdział 4.5). Dlatego też w przypadku niektórych zastosowań może być mimo wszystko lepszym wyborem niż algorytm CRF w postaci przez nas rozważanej.

Szczegółowe porównanie różnic pomiędzy obserwowanymi wartościami miary F przedstawiamy poniżej.

1. Dla fraz *VP* i *AdjP* wszystkie różnice są istotne statystycznie:
   — $F_{\mathrm{CRF}} \gg F_{\mathrm{MBL}}$, $F_{\mathrm{CRF}} \gg F_{\mathrm{Spejd}}$, $F_{\mathrm{CRF}} \gg F_{\mathrm{DT}}$
   — $F_{\mathrm{Spejd}} \gg F_{\mathrm{MBL}}$, $F_{\mathrm{Spejd}} \gg F_{\mathrm{DT}}$
   — $F_{\mathrm{MBL}} \gg F_{\mathrm{DT}}$
2. Dla frazy *NP* różnica między algorytmem MBL i DT jest nieistotna:
   — $F_{\mathrm{CRF}} \gg F_{\mathrm{MBL}}$, $F_{\mathrm{CRF}} \gg F_{\mathrm{Spejd}}$, $F_{\mathrm{CRF}} \gg F_{\mathrm{DT}}$

— $F_{\text{Spejd}} \gg F_{\text{MBL}}$, $F_{\text{Spejd}} \gg F_{\text{DT}}$
— $F_{\text{MBL}} > F_{\text{DT}}$

Kierunki nierówności są identyczne, jak te obserwowane w testach przeprowadzonych na KPWr. W szczególności, wartości miary F osiągnięte za pomocą opisywanego tutaj algorytmu CRF przewyższają lub są niegorsze niż osiągi uzyskane za pomocą pozostałych testowanych algorytmów na obu korpusach dla wszystkich typów fraz. Biorąc pod uwagę szerokie spektrum przetestowanych definicji fraz (eksperymenty objęły łącznie 4 definicje fraz w ramach korpusu KPWr oraz 3 definicje w ramach NKJP), możemy wysnuć hipotezę, że jeśli dążymy do uzyskania możliwie dobrego oznakowania frazami tekstu polskiego, to prawdopodobnie najlepszym wyborem będzie właśnie algorytm CRF.

5.6. Wpływ tagera na wyniki analizatora

Przeprowadziliśmy eksperymentalną ocenę wpływu doboru tagera na wyniki analizatora składniowego. Ocena ta została przeprowadzona na frazach NP z korpusu NKJP. Przeprowadzono test całego systemu składającego się z tagera (wraz z segmentacją i analizą morfologiczną) i modułu znakowania fraz — a więc jest to *ocena uwzględniająca błędy znakowania* zgodnie z opisem na s. 101.

Wyniki testów przedstawiono w tabeli 5.4. Pośród testowanych tagerów uwzględniliśmy tager oparty na uczeniu pamięciowym (WMBT) z modułem odgadującym słowa nieznane (rozdział 2.8) i bez niego, a także warszawski tager PANTERA. Tager WMBT był uczony przy użyciu techniki ponownej analizy morfosyntaktycznej danych uczących (por. rozdział 2.9). Ponieważ tager PANTERA nie był tworzony z myślą o użyciu tej techniki, przetestowaliśmy dwa warianty: z jej zastosowaniem oraz bez jej zastosowania.

Tager	**Algorytm**	P	R	F
WMBT +A +O	DT	75,07%	81,01%	77,93%
	MBL	75,14%	81,06%	77,99%
	CRF	86,94%	86,75%	86,85%
WMBT +A -O	DT	72,65%	78,74%	75,57%
	MBL	73,27%	79,05%	76,05%
	CRF	85,40%	84,66%	85,03%
PANTERA +A	DT	72,25%	78,36%	75,18%
	MBL	72,68%	78,57%	75,51%
	CRF	84,44%	84,06%	84,25%
PANTERA -A	DT	72,20%	78,30%	75,13%
	MBL	72,63%	78,49%	75,44%
	CRF	84,43%	84,04%	84,24%

Tabela 5.4. Wpływ wybranego tagera na wyniki algorytmów znakowania fraz NP obserwowany na danych z NKJP. Oznaczenie ±A określa, czy użyto techniki ponownej analizy morfosyntaktycznej danych uczących. Oznaczenie ±O określa, czy użyto wersji algorytmu WMBT z odgadywaniem słów nieznanych.

Najlepsze wyniki znakowania fraz (mierzone poprzez miarę F) osiągnięto przy zastosowaniu tagera WMBT wraz z modułem rozpoznawania słów nieznanych. Poprawa osiągów wszystkich trzech algorytmów znakowania fraz uzyskana dzięki wprowadzeniu do tagera WMBT modułu rozpoznawania słów nieznanych jest istotna statystycznie. Co więcej, różnica między osiągami trzech algorytmów przy udziale tagera WMBT a osiągami tych samych algorytmów przy udziale tagera PANTERA jest istotna statystycznie. Co ciekawe, stwierdzenie to jest prawdziwe zarówno dla WMBT z modułem odgadywania słów nieznanych, jak i bez niego. Wyniki eksperymentu potwierdzają zatem praktyczną wartość opracowanego algorytmu znakowania morfosyntaktycznego. Użycie ponownej analizy morfosyntaktycznej w przypadku tagera PANTERA nie przyniosło istotnej poprawy wyników żadnego z trzech algorytmów znakowania fraz.

Tabela 5.5 przedstawia porównanie trzech metod oceny algorytmów znakowania fraz. Wszystkie podane liczby są wartościami miary F dla problemu znakowania fraz NP. Pierwsza kolumna (*„Wzorcowe”*) odpowiada testom na danych zawierających wzorcowe oznakowanie morfosyntaktyczne z NKJP. Przedstawione wartości uwzględniają zatem jedynie błędy popełnione przez sam moduł znakowania fraz. W kolumnie drugiej (*„Ujednoznacznienie”*) przedstawiono wyniki osiągnięte przy użyciu wzorcowej segmentacji i analizy morfosyntaktycznej z NKJP, lecz ujednoznaczniania (wyboru spośród dostępnych w korpusie wzorcowym interpretacji) wykonanego automatycznie za pomocą tagera WMBT. Taki test uwzględnia zatem błędy ujednoznaczniania, lecz pomija błędy popełnione na wcześniejszych etapach przetwarzania. Wyniki z kolumny trzeciej (*„Oznakowanie”*) odpowiadają testowi całego systemu, podobnie jak w przypadku poprzedniego eksperymentu (z tabeli 5.4). Wartości miary F osiągnięte w ten sposób są najlepszym przybliżeniem rzeczywistego błędu popełnianego przez system płytkiej analizy składniowej oparty na danym algorytmie i dostępnych komponentach.

Algorytm	Wzorcowe	Ujednoznacznienie	Oznakowanie
DT	84,13%	79,96%	77,93%
MBL	85,06%	80,25%	77,99%
CRF	92,31%	88,61%	86,85%
Spejd	87,94%	81,75%	79,82%

Tabela 5.5. Trzy metody oceny algorytmów znakowania fraz na korpusie NKJP. Wszystkie podane wyniki są wartościami miary F osiągniętymi w danym teście.

Porównanie pokazuje duże rozbieżności między wynikami testów przeprowadzone zgodnie z różnymi założeniami. W przypadku algorytmu CRF, ocena uwzględniająca błędy ujednoznaczniania (kolumna *„Ujednoznacznienie”*) wykazuje prawie 1,5 razy większy odsetek błędu niż ocena na wzorcowych danych morfosyntaktycznych. Ocena uwzględniająca błędy znakowania (a więc najbardziej rzetelna) wykazuje zaś ok. 1,7 razy większy odsetek błędów niż ocena na wzorcowych danych morfosyntaktycznych.

Pomiędzy wartościami miary F osiągniętymi przy użyciu różnych analizatorów składniowych zachodzą te same nierówności, niezależnie od przyjętej metody oceny. Wszystkie różnice są istotne statystycznie z wyjątkiem nierówności $F_{\mathrm{MBL}} > F_{\mathrm{DT}}$ dla testu całego systemu (*„Oznakowanie”*) — ta różnica jest nieistotna.

5.7. Podsumowanie

W rozdziale zostały opisane badania kilku metod znakowania fraz dla języka polskiego. Przebadano trzy metody korzystające z technik maszynowego uczenia, a także metodę zakładającą użycie płytkiego parsera Spejd wyposażonego w gramatykę pisaną ręcznie. Eksperymenty wykazały, że użycie metody opartej na warunkowych polach losowych pozwala na osiągnięcie najlepszych wyników, przewyższających nie tylko pozostałe dwie metody korzystające z technik maszynowego uczenia (mianowicie, indukcji drzew decyzyjnych oraz uczenia pamięciowego), lecz również użycie płytkiego parsera Spejd. Tej ostatniej obserwacji dokonano na podstawie badań przeprowadzonych na korpusie NKJP. Jest ona szczególnie interesująca, gdyż pisanie reguł gramatyki dla parsera Spejd wspomagane było analizą danych z tego właśnie korpusu — a więc parser ten był w pewien sposób faworyzowany w tym teście.

Wyniki eksperymentów sugerują również, że przyjęte definicje fraz mają duży wpływ na trudność ich automatycznego rozpoznawania. Frazy rzeczownikowe rozumiane zgodnie z wytycznymi KPWr są w ogólności bardziej złożone niż frazy rzeczownikowe w NKJP (por. rozdział 4.2). Znajduje to odzwierciedlenie w wynikach rozpoznawania tych fraz: w przypadku korpusu KPWr udało się osiągnąć wartość miary F 69,46%, natomiast dla fraz rzeczownikowych z NKJP wartość ta wynosi 86,85%. Stosunkowo niskie wyniki osiągnięte dla korpusu KPWr mogą sugerować, że przyjęta tam definicja fraz uwzględnia konstrukcje zbyt skomplikowane jak na zadanie płytkiej analizy składniowej.

Mimo to, eksperymenty dowiodły, że proponowane metody znakowania fraz w oparciu o techniki maszynowego uczenia cechuje duża elastyczność. Ta sama metoda została użyta z powodzeniem do znakowania fraz zdefiniowanych w różny sposób. Do przeprowadzenia eksperymentów użyto tej samej implementacji bez konieczności wprowadzania jakichkolwiek modyfikacji. Pod tym względem metody te przewyższają podejście regułowe.

Pokazano także, że jakość użytego tagera ma istotny wpływ na obserwowane wyniki modułu znakowania fraz. Badania potwierdziły wartość praktyczną zaproponowanego w rozdziale 2.8 algorytmu WMBT: jego użycie przyniosło poprawę w stosunku do wyników osiągniętych za pośrednictwem konfiguracji, w której skład wchodził tager PANTERA. Pokazano również, że wszystkie algorytmy znakowania fraz wykazują znacznie wyższe wartości miary F, gdy mają dostęp do wzorcowego oznakowania morfosyntaktycznego. Takie warunki odbiegają jednak od typowego scenariusza użycia modułu znakowania fraz, a zatem istotne jest, by ocenę metod znakowania fraz przeprowadzać w sposób uwzględniający wpływ błędów tagera.

Rozdział 6

Ocena płytkiego analizatora składniowego jako narzędzia wspomagającego systemy przetwarzania języka polskiego

W tym punkcie przedstawiamy dodatkowe eksperymenty polegające na zastosowaniu zaproponowanego w rozdziale 4 modułu znakowania fraz jako narzędzia wspomagającego rozwiązywanie dwóch praktycznych zadań przetwarzania języka naturalnego. Celem tych eksperymentów jest pokazanie przydatności praktycznej modułu znakowania fraz wykonanego w oparciu o proponowany algorytm. Wybór tych zastosowań został w dużej mierze podyktowany harmonogramem projektów badawczych prowadzonych w Instytucie Informatyki Politechniki Wrocławskiej oraz Instytucie Podstaw Informatyki Polskiej Akademii Nauk — takie badania były akurat prowadzone w momencie pisania rozprawy, co znacznie ułatwiło uzyskanie potrzebnych danych i przeprowadzenie eksperymentów w rozsądnym czasie. Warto podkreślić jednak, że kolejne działania zaplanowane w ramach tych projektów również zakładają użycie modułu znakowania fraz; moduł ten znajdzie zastosowanie m.in. w systemie automatycznego odpowiadania na pytania zadane w języku polskim, a także w systemie znajdującym powiązania anaforyczne w tekście (oba systemy opracowywane są obecnie w ramach projektu NEKST, por. str. 45).

Przedstawione poniżej eksperymenty zakładają, że stosowane będą dwie metody znakowania fraz:

1. znakowanie fraz za pomocą algorytmu CRF (rozdział 4.8) — gdyż pozwolił osiągnąć najlepsze wyniki w eksperymentach z rodziału 5,
2. pozyskiwanie fraz z wyjścia płytkiego parsera Spejd (rozdział 4.5) — jako punkt odniesienia, gdyż parser Spejd i gramatykę NKJP uznajemy za wyznacznik dotychczasowego stanu badań związanych z płytką analizą składniową języka polskiego.

6.1. Wydobywanie terminów z korpusu dziedzinowego

Zadanie **wydobywania terminów dziedzinowych** (ang. *terminology extraction*, *automatic term recognition*) polega na wydobyciu listy terminów specyficznych dla danej dziedziny na podstawie automatycznej analizy korpusu reprezentującego dziedzinę (Marciniak i Mykowiecka, 2012). Pozyskana w ten sposób lista terminów może służyć jako podstawa do budowy słownika bądź leksykonu związanego z daną dziedziną. Listy terminów dziedzinowych przydają się również w innych systemach przetwarzania języka

naturalnego, m.in. systemach tłumaczenia maszynowego, indeksowania dokumentów na potrzeby bibliotek cyfrowych oraz pozyskiwania ontologii dziedzinowych (Korkontzelos i inni, 2008).

Typowe metody wydobywania terminów dziedzinowych składają się z dwóch etapów (Marciniak i Mykowiecka, 2012):

1. wydobycie z korpusu fraz rzeczownikowych,
2. statystyczny ranking fraz rzeczownikowych i odfiltrowanie fraz, które nie są terminami.

Badania nad wydobywaniem terminów dziedzinowych prowadzone są od niedawna również dla języka polskiego. Prace te prowadzone są przez Małgorzatę Marciniak i Agnieszkę Mykowiecką z Instytutu Podstaw Informatyki Polskiej Akademii Nauk. W pracy Marciniak i Mykowiecka (2012) autorki przedstawiają metodę wydobywania terminów dostosowaną do tekstów języka polskiego oraz eksperymenty przeprowadzone na korpusie tekstów o tematyce ekonomicznej. W celu rozpoznania fraz rzeczownikowych zastosowano napisaną na tę potrzebę gramatykę. Gramatyka ta ma charakter płytki i uwzględnia specyfikę fraz rzeczownikowych występujących w roli terminów ekonomicznych — m.in. duży nacisk położono na prawidłowe rozpoznanie skrótów oraz wyrażeń zawierajacych łączniki (np. społeczno-ekonomiczny).

Eksperyment przedstawiony w tym punkcie polegał na zastąpieniu wspomnianej gramatyki przez metody znakowania fraz rozważane w tej rozprawie. W szczególności, zastosowano zaproponowaną w tej pracy metodę znakowania fraz korzystającą z warunkowych pól losowych (por. punkt 4.8; odtąd nazywać będziemy ją skrótowo *metodą CRF*) oraz parser regułowy Spejd wyposażony w gramatykę NKJP (por. punkt 4.5). Celem eksperymentu była ocena metody CRF pod kątem możliwości zastosowania w problemie wydobywania terminów dziedzinowych, a także porównanie pod kątem tego zastosowania jej osiągów z osiągami parsera Spejd. Eksperyment opisany poniżej został przeprowadzony we współpracy z Agnieszką Mykowiecką.

Gramatyka opisana w pracy Marciniak i Mykowiecka (2012) składa się z sześciu podzbiorów reguł oraz dodatkowych reguł poprawiających typowe błędy tagera. Reguły operują na tekście ujednoznacznionym morfosyntaktycznie i odwołują się do informacji zawartych w tagach. Gramatyka zakłada, że nadrzędniki fraz rzeczownikowych mogą być rzeczownikami, odsłownikami (gerundiami) lub skrótami. Frazy mogą oprócz tego zawierać określenia przymiotnikowe (do pięciu przymiotników, opcjonalnie oddzielonych przymiotnikami lub spójnikami) oraz przysłówkowe określenia przymiotników. Bardziej złożone frazy mogą również zawierać określenia będące rzeczownikami w dopełniaczu (po takich rzeczownikach mogą następować ich określenia przymiotnikowe), a także apozycje (por. str. 4.2.1). Co więcej, gramatyka dopuszcza także frazy rzeczownikowe zawierające w sobie frazy przyimkowe — np. (6.1) i (6.2).

(6.1) $[_{NP}$ cena na nowy produkt]
(6.2) $[_{NP}$ cena równowagi kształtowana przez relację podaży]

Gramatyka ta pozwala na rozpoznanie granic fraz rzeczownikowych, a także wybranych ich podfraz będących też frazami rzeczownikowymi. Przykładowo, w ramach frazy (6.3) gramatyka rozpoznaje podfrazy (6.4)–(6.6). Pozwala to na rozpoznanie terminów, które należą do większych fraz. Warto tu zaznaczyć, że zastosowanie modułu znakowania fraz nie daje takich możliwości, gdyż przyjęta definicja zadania znakowania fraz

wyklucza takie zagnieżdżenia. Z drugiej strony, jest prawdopodobne, że wygenerowanie zbyt dużej liczby podfraz może spowodować zaakceptowanie przez metodę zbyt dużej liczby podfraz, które w rzeczywistości nie będą terminami.

(6.3) [$_{NP}$ współczesna struktura systemu transportowego]
(6.4) [$_{SubNP}$ struktura systemu]
(6.5) [$_{SubNP}$ współczesna struktura systemu]
(6.6) [$_{SubNP}$ struktura systemu transportowego]

Wspomniane reguły korygujące błędy tagera wprowadzono celem dostosowania oznakowania morfosyntaktycznego do dziedziny ekonomicznej. Przykładowo, jedna z reguł nadaje segmentom wyróżnionym w ciągu Dz.U. interpretacje zgodne z rozwinięciem skrótu (Dziennik Ustaw).

Drugim etapem algorytmu jest statystyczny ranking fraz rzeczownikowych. Marciniak i Mykowiecka (2012) stosują w tym celu statystykę zwaną *wartością C* (ang. *C-value*) poddaną drobnym modyfikacjom. Wartość C opisana jest wzorem (6.7), gdzie funkcja lc to uogólnienie logarytmu, pozwalające na wyliczenie wartości dla fraz jednowyrazowych — zgodnie z wzorem (6.8). Zapis length(p) oznacza liczbę segmentów, z której składa się fraza p. Zbiór LP oznacza zbiór wszystkich nadfraz rozpatrywanej frazy p. Marciniak i Mykowiecka (2012) uznają, że lp jest nadfrazą frazy p, jeśli gramatyka pisana ręcznie w ramach frazy lp wyodrębniła w niej podfrazę p.

$$C(p) = \begin{cases} \mathrm{lc}(p)\ \mathrm{freq}(p) - \frac{1}{||LP||} \sum \mathrm{freq}(lp), & ||LP|| > 0, lp \in LP \\ \mathrm{lc}(p)\ \mathrm{freq}(p), & ||LP|| = 0, lp \in LP \end{cases} \tag{6.7}$$

$$lc(p) = \begin{cases} \log_2(\mathrm{length}(p)), & \mathrm{length}(p) > 1 \\ 0,1 & w\ pp. \end{cases} \tag{6.8}$$

Opisana wartość C służy do rankingu orzymanej listy terminów: pozycje o wysokich wartościach C uznane są za wiarygodne propozycje terminów dziedzinowych. Wszystkie frazy rozpatrywane są jako ciągi lematów.

Marciniak i Mykowiecka (2012) stosują dodatkowy zabieg, którego celem jest poprawienie jakości pozyskanej listy terminów. Zabieg ten polega na porównaniu listy terminów uzyskanej na podstawie analizy korpusu tekstów ekonomicznych z listą terminów pozyskaną z korpusu języka ogólnego (w tym wypadku zastosowano korpus NKJP) i usunięcia z listy terminów dziedzinowych tych pozycji, które na liście pozyskanej z korpusu ogólnego występują z większą wartością C. Badania przedstawione w tym punkcie miały charakter pilotażowy i w przeprowadzonych przez nas eksperymentach pominęliśmy ten etap (pozwoliło to na zmniejszenie nakładu pracy).

Istnieje kilka możliwości oceny pozyskanych automatycznie terminów. Najbardziej wiarygodną metodą jest prawdopodobnie ręczna ocena każdego z terminów dokonana przez lingwistę: zadaniem lingwisty jest wtedy ocena, które z pozycji na liście są prawidłowymi terminami ekonomicznymi.

Definicja *terminu dziedzinowego* przysparza pewnych problemów. W pracy Marciniak i Mykowiecka (2012) przyjęto następującą definicję:

> Termin jest frazą rzeczownikową, która występuje w tekstach dziedzinowych wystarczająco często, by uwiarygodnić hipotezę, że reprezentuje ona coś ważnego, czego szukać mogą internauci.

Lingwista, któremu przydzielono zadanie ręcznej oceny terminów, uznał tę definicję za trudno rozstrzygalną w praktyce, gdyż odwołuje się ona do zachowania internautów. Wynikiem dalszych dyskusji było przyjęcie ostatecznie nieco innej „definicji":

> Termin jest frazą rzeczownikową, którą uwzględniłbyś w słowniku z dziedziny ekonomicznej (np. polsko-niemieckim) lub encyklopedii pojęć ekonomicznych.

Przyjęta przez nas „definicja" jest również bardzo subiektywna i nieścisła, lecz jej zaletą z punktu widzenia lingwisty było odwołanie się do jego oceny, a nie hipotetycznej oceny innych (internautów). Liczymy na to, że dzięki temu uzyskana ocena przeprowadzona została w sposób konsekwentny, mimo jej oczywistej subiektywności. Celem tego eksperymentu było sprawdzenie przydatności opracowanej metody znakowania fraz pod kątem zastosowań praktycznych — a za takie można uznać zarówno wyszukiwarki internetowe, jak i narzędzia wspomagające tworzenie słowników.

Marciniak i Mykowiecka (2012) przyjmują, że wiarygodne terminy zajmują pierwsze 500 pozycji pozyskanej listy terminów o największych wartościach C. Nasz eksperyment zakładał wygenerowanie czterech takich list:

1. listy pozyskanej dzięki zastosowaniu oryginalnej gramatyki z pracy Marciniak i Mykowiecka, 2012 (odtąd: *Gramatyka*),
2. listy pozyskanej dzięki zastosowaniu metody CRF, gdzie model wyuczono na danych z korpusu KPWr (odtąd: *CRF-KPWr*),
3. jak wyżej, lecz model wyuczono na korpusie NKJP (odtąd: *CRF-NKJP*),
4. za pomocą fraz rzeczownikowych pozyskanych za pomocą parsera Spejd (*Spejd-NKJP*).

Wyjście parsera Spejd zostało przetworzone na płaskie frazy przy użyciu procedury, którą opisaliśmy w rozdziale 4.5. Dane z NKJP, na których nauczono moduł CRF, zostały przetworzone w ten sam sposób. Procedura ta zakłada, że do fraz rzeczownikowych (NP) należą też frazy rozpoczynające się przyimkami. Podobne założenie ma miejsce w przypadku fraz w korpusie KPWr. W zadaniu wydobywania terminów jest to niepożądane. Dlatego też zastosowaliśmy prostą regułę, która z fraz rozpoczynających się przyimkami usunęła te przyimki.

W przypadku zastosowania metod innych niż *Gramatyka* konieczna była zmiana interpretacji zbioru LP, gdyż nie mieliśmy dostępu do informacji na temat podfraz. Uznaliśmy, że w takiej sytuacji lp jest nadfrazą frazy p, jeśli p tekstowo zawiera się we frazie lp oraz obie frazy zostały rozpoznane przez moduł znakowania fraz jako samodzielne frazy występujące w korpusie.

Do przeprowadzenia eksperymentu użyto tego samego korpusu tekstów ekonomicznych. Korpus składa się z 1219 artykułów o tematyce ekonomicznej pobranych z polskiej Wikipedii, co daje 458 819 segmentów.

Do porównania istotności statystycznej użyliśmy testu z dla dwóch proporcji i dużych prób (Ogunnaike, 2009, s. 606). Wszystkie próby mają rozmiar $n = 500 > 50$. Porównywaną proporcją był odsetek pozycji na listach uznanych przez lingwistę za prawidłowe terminy ekonomiczne. Podobnie jak w pozostałych testach istotności statystycznej przeprowadzanych w rozprawie, przyjmujemy poziom istotności $\alpha = 0,05$.

Wyniki eksperymentu prezentujemy w tabeli 6.1. Wartości podane w kolumnie *Terminy ekonomiczne* oznaczają odsetek pozycji na liście (500-elementowej) zwróconej

przez daną metodę, który został uznany przez lingwistę za prawidłowe terminy ekonomiczne (prawidłowe terminy muszą być jednocześnie prawidłowymi frazami rzeczownikowymi).

Metoda	Terminy ekonomiczne
Gramatyka	61,2%
Spejd-NKJP	51,8%
CRF-NKJP	45,6%
CRF-KPWr	52,4%

Tabela 6.1. Terminy ekonomiczne wydobyte przy pomocy płytkich parserów w ocenie lingwisty

Wyniki osiągnięte przez nas są wyraźnie gorsze niż te przedstawione w pracy Marciniak i Mykowiecka (2012), gdzie jeden z dwóch lingwistów dokonujących oceny uznał 89% zwróconych pozycji za prawidłowe terminy, a drugi aż 96%. Różnica ta może wynikać zarówno z pominięcia przez nas etapu filtrowania fraz, które pojawiły się często w korpusie języka ogólnego, jak i z przyjęcia innej „definicji" terminu. Jest prawdopodobne, że lingwista uznał część pozycji za zbyt oczywiste lub zbyt łatwe w interpretacji i jako takie mogły zostać uznane za niewarte uwzględnienia w słowniku bądź encyklopedii. Można przypuszczać, że pierwotne kryterium oceny fraz w oparciu o hipotetyczne zachowanie internautów skłoniłoby do zaakceptowania większej liczby fraz, choćby luźno powiązanych z ekonomią, jako terminy ekonomiczne.

Originalna gramatyka z pracy Marciniak i Mykowiecka (2012) pozwoliła osiągnąć najlepsze wyniki. Procent fraz pozyskanych przy udziale tej gramatyki, które lingwista uznał za terminy ekonomiczne, jest wyższy niż taki procent w przypadku pozostałych metod analizy składniowej (różnica ta jest istotna statystycznie). Prawdopodobnie duży wpływ na tę przewagę miały wspomniane reguły korygujące błędy tagera. W przypadku list uzyskanych przez metody inne niż *Gramatyka*, stosowany był tager WMBT (por. punkt 2.8) bez użycia reguł korygujących. Tager uczony był na korpusie języka ogólnego (NKJP), a stosując go do tekstów dziedzinowych musimy liczyć się ze wzrostem liczby błędów. W przypadku konfiguracji *Gramatyka* użyto narzędzi wcześniej używanych przez Małgorzatę Marciniak i Agnieszkę Mykowiecką, tj. tagera PANTERA i wspomnianych reguł korygujących. Drugą prawdopodobną przyczyną przewagi oryginalnej gramatyki jest rozpoznawanie nie tylko całych fraz, ale także podfraz.

Na drugim miejscu plasuje się zastosowanie metody *CRF-KPWr*, a nieco gorsze wyniki otrzymano przy użyciu parsera Spejd (choć różnica między nimi nie jest istotna statystycznie). Najgorsze wyniki osiągnięto przy użyciu metody *CRF-NKJP* (procent fraz uznanych za terminy ekonomiczne uzyskany przy pomocy tej metody jest niższy niż wszystkie pozostałe, różnice te są istotne).

Przeprowadzony eksperyment potwierdza możliwość praktycznego zastosowania metody CRF: wyniki rozpoznawania terminów ekonomicznych osiągnięte przy pomocy wariantu *CRF-KPWr* są nie gorsze niż te osiągnięte przy pomocy parsera Spejd wyposażonego w płytką gramatykę ogólnego języka polskiego.

Nieco zaskakujące są słabe osiągi wariantu *CRF-NKJP* na tle metody metody *Spejd-NKJP*. Obserwacja ta stoi w sprzeczności z gorszymi wynikami Spejda w testach z punktu 5.5. Analiza list zwróconych przez obie metody wykazała, że na liście

CRF-NKJP częściej pojawiły się skróty (np. P.) oraz liczby pisane cyframi niż miało to miejsce w przypadku listy *Spejd-NKJP*. Możliwą przyczyną takiego stanu rzeczy jest rzadsze popełnianie błędów przez parser Spejd wyposażony w gramatykę NKJP w sytuacjach, gdy rozróżnienie ma charakter typowo gramatyczny — zaś mniej konsekwentne przestrzeganie wytycznych w sytuacjach, gdzie człowiek uznał to za mniej istotne (np. traktowaniem znaków interpunkcyjnych, symboli itp.). Warto jednak zaznaczyć, że nasza metoda oceny skupia się na *dokładności* (oceniamy, jaki odsetek zwróconych przez nas pozycji jest terminami), natomiast z przyczyn praktycznych nie oceniamy *kompletności* (tj. udziału prawidłowo rozpoznanych terminów pośród wszystkich, które były do rozpoznania w danym korpusie). Możliwe jest, że w przypadku tekstów dziedzinowych gramatyka Spejda rozpoznaje mniej fraz niż metoda CRF, lecz frazy, które już zostały rozpoznane, są na ogół prawidłowe.

Lepsze wyniki uzyskane przy pomocy konfiguracji *CRF-KPWr* w stosunku do *CRF-NKJP* wynikają prawdopodobnie z różnic w definicji fraz przyjętych w korpsach KPWr i NKJP. W szczególności, wytyczne NKJP nie pozwalają na włącznie fraz przyimkowych do fraz rzeczownikowych, podczas, gdy w KPWr taka sytuacja często ma miejsce (por. punkt 4.2.2 i 4.2.3). Struktura składniowa części terminów ekonomicznych wymaga, by frazy przyimkowe były włączane. Opisana sytuacja wskazuje na praktyczną korzyść płynącą z dostępu do dwóch korpusów języka polskiego oznaczonych frazami według różnych wytycznych: w zależności od zastosowania można wybrać jeden z nich.

6.2. Wydobywanie relacji powiązania znaczeniowego

Drugie z prezentowanych zastosowań modułu znakowania fraz wiąże się z innym przedsięwzięciem realizowanym na Politechnice Wrocławskiej, mianowicie budową *Słowosieci*. Słowosieć, czyli tzw. *wordnet* języka polskiego, jest wielką leksykalną bazą wiedzy. Baza ta pełni m.in. rolę komputerowego słownika wyrazów bliskoznacznych. Zawiera również inne relacje semantyczne między wyrazami, np. *hiponimię* (np. pistolet jest rodzajem broni), *meronimię* (lufa jest częścią pistoletu) (Piasecki i inni, 2009).

Budowa Słowosieci jest przedsięwzięciem wymagającym ogromnych nakładów pracy lingwistów. By proces ten przyspieszyć, wprowadzono metody przetwarzania języka naturalnego, których zadaniem jest podpowiadanie lingwistom prawdopodobnych instancji relacji semantycznych między wyrazami znalezionych na podstawie analizy wielkich korpusów językowych. Ta analiza tekstu prowadzi do pozyskania wiedzy sformalizowanej w postaci tzw. **miary powiązania znaczeniowego** (ang. *measure of semantic relatedness*, *MSR*). Miarę tę można zdefiniować jako funkcję:

$$\mathrm{MSR} : L \times L \to \mathbb{R} \tag{6.9}$$

gdzie L to zbiór jednostek leksykalnych (w uproszczeniu — pojedynczych wyrazów lub konstrukcji wielowyrazowych), a $\mathbb{R}$ — liczby rzeczywiste (Broda i Piasecki, 2008). Interpretacją miary jest siła powiązania znaczeniowego między dwoma jednostkami leksykalnymi. Przykładowo, można się spodziewać, że wartość $\mathrm{MSR}(\textsf{pistolet}, \textsf{lufa})$ będzie znacznie większa niż $\mathrm{MSR}(\textsf{pistolet}, \textsf{doktorant})$. Miara abstrahuje od konkretnego rodzaju relacji semantycznej, w szczególności jej zadaniem nie jest rozróżnienie między relacją synonimii, a np. wspomnianej hiponimii. Mimo to, pozyskanie takiej funkcji

ma duże znaczenie praktyczne, gdyż umożliwia to podanie lingwiście trafnych podpowiedzi, które umożliwiają szybkie rozszerzanie Słowosieci (każda instancja relacji zasugerowana przez system jest weryfikowana przez lingwistę, por. Piasecki i inni, 2009). Co więcej, można sobie wyobrazić inne zastosowania praktyczne dla miary podobieństwa znaczeniowego, w szczególności dla języków, dla których nie opracowano jeszcze odpowiedników Słowosieci.

Broda i Piasecki (2008) opisują system *SuperMatrix* stosowany do wydobycia takiej miary dla języka polskiego. Metoda zaimplementowana w tym systemie opiera się na założeniu, że współwystępowanie dwóch jednostek leksykalnych w podobnych *kontekstach* jest dowodem na ich powiązanie znaczeniowe. Realizacja metody sprowadza się do zliczania częstości wystąpienia pary jednostek leksykalnych w danym *kontekście* (realizowane jest to za pomocą macierzy, której wierszami są jednostki leksykalne, a kolumnami — konteksty). Drugim istotnym założeniem jest zliczanie tych częstości na podstawie analizy ogromnych korpusów, składających się z wielu milionów segmentów — zwiększa to szanse znalezienia informacji istotnej wśród wielu przypadkowych współwystąpień. Informacja o częstościach występowania par poddawana jest potem dalszym przekształceniom matematycznym, które prowadzą ostatecznie do uzyskania funkcji MSR. Kontekst można różnie definiować: może być nim pojedyncze zdanie, okno tekstowe o stałej szerokości (liczonej w liczbie segmentów), a także — konteksty o charakterze składniowym, np. występowanie jednostek jako uczestniczących w danej relacji składniowej. Eksperymenty wykazały, że ten ostatni typ kontekstu daje szczególnie dobre rezultaty.

W pracy (Broda i inni, 2009) przedstawiono wyniki takich badań dla języka polskiego. Eksperymenty zostały ograniczone do wydobywania miary podobieństwa między rzeczownikami. Autorzy ze względu na brak dostępnego w owym czasie parsera, który spełniałby przyjęte założenia, zdecydowali się na ręczne napisanie kilku predykatów w formalizmie JOSKIPI (por. 2.4.1 oraz 2.6). Zadaniem predykatów było znalezienie par wyrazów, między którymi zachodzą następujące związki składniowe:

— przymiotnik, który jest określeniem rzeczownika,
— dwa rzeczowniki w szeregowej frazie rzeczownikowej,
— rzeczownik w dopełniaczu, który jest określeniem innego rzeczownika,
— rzeczownik, który może być dopełnieniem czasownika.

Powyższe predykaty miały charakter heurystyczny; celem nadrzędnym nie była wysoka precyzja analizy składniowej, lecz osiągnięcie dobrych wyników wydobywania miary powiązania znaczeniowego. Z drugiej strony, obecna postać predykatów jest wynikiem kilku badań eksperymentalnych i serii korekt, więc należy się spodziewać stosunkowo dobrej jakości pozyskiwanej miary powiązania znaczeniowego.

W tym punkcie przedstawiamy eksperymenty[1] przeprowadzone w sposób analogiczny, w których poza zastosowaniem powyższych predykatów (przepisanych na formalizm WCCL[2]), rozpatrujemy także zastosowanie modułu znakowania fraz jako narzędzia pozwalającego na znalezienie takich par wyrazów powiązanych składniowo.

[1] Eksperymenty opisane w tym punkcie zostały przeprowadzone przez Bartosza Brodę, Dominika Piaseckiego oraz autora tej rozprawy.

[2] Formalizm JOSKIPI ograniczony jest do tagsetu KIPI. Jego następca — WCCL — może działać na dowolnym tagsecie pozycyjnym (Radziszewski i inni, 2011c). Aby umożliwić przetwarzanie korpusu oznakowanego morfosyntaktycznie w tagsecie NKJP, oryginalne predykaty przepisano na formalizm WCCL.

Eksperymenty przeprowadziliśmy na danych pozyskanych z całego zrównoważonego korpusu NKJP (por. punkt 2.2). Dane wejściowe zostały użyte w postaci czystego tekstu, który oznakowaliśmy morfosyntaktycznie za pomocą tagera WMBT wraz z modułem odgadywania słów nieznanych (por. 2.8). Oznakowany w ten sposób korpus składał się z prawie 265 mln segmentów.

Eksperymenty zakładały wydobycie miary podobieństwa przy pomocy par wyrazów pozyskanych na sześć różnych sposobów:

1. za pomocą ręcznie napisanych predykatów WCCL,
2. przy pomocy fraz pozyskanych z parsera Spejd (bez łączenia fraz),
3. przy pomocy fraz pozyskanych z parsera Spejd (z łączeniem fraz),
4. przy pomocy fraz oznakowanych za pomocą algorytmu CRF wyuczonego na danych z oznakowanego frazami podkorpusu milionowego NKJP (z łączeniem fraz),
5. połączenie par uzyskanych przy pomocy predykatów (1) z parami ze Spejda (3),
6. połączenie par uzyskanych przy pomocy predykatów (1) z parami z CRF (4).

Wspomniane wyżej łącznie fraz rozumiane jest jako zabieg opisany w punkcie 4.5, którego wynikiem jest scalenie oryginalnych grup składniowych z NKJP we frazy rzeczownikowe (NP), przymiotnikowe (AdjP) i czasownikowe (VP). Ponieważ nasze eksperymenty ograniczone są do badania podobieństwa między rzeczownikami, pomijamy frazy czasownikowe (nie zawierają one rzeczowników). W przypadku eksperymentu przeprowadzonego bez łączenia fraz (2), pod uwagę brane były te grupy składniowe zwrócone przez Spejd, które włączamy do naszych definicji NP i AdjP (4.5), lecz zachowywaliśmy informację, do której oryginalnej grupy składniowej należała dana para.

System SuperMatrix wymaga, by każda para wyrazów (reprezentowanych przez lematy) wzbogacona była o etykietę nazwy relacji. W przypadku predykatów WCCL etykietami tymi były nazwy relacji składniowych rozpoznawanych przez poszczególne predykaty, np. `AdjC` oznaczało, że para składa się z rzeczownika i przymiotnika, przy czym przymiotnik jest określeniem rzeczownika. W przypadku fraz pozyskanych z parsera Spejd etykieta ta była oryginalną nazwą grupy składniowej (np. `PrepAdjG`, `NG`) lub zbiorczą nazwą frazy reprezentującą złączone frazy (`NP` lub `AdjP`). W przypadku fraz oznakowanych za pomocą algorytmu CRF etykieta ta była również nazwą frazy (`NP` lub `AdjP`).

Każda fraza została zamieniona na zbiór par — dla każdej frazy rozpatrywane były wszystkie pary (*centrum semantyczne frazy*, *słowo należące do frazy niebędące centrum semantycznym*). Parser Spejd znakuje centra semantyczne (por. 4.2.3). W przypadku, gdy centrum składało się z więcej niż jednego segmentu, arbitralnie wybierany był pierwszy z nich (centrum składniowym może być całe słowo składniowe, które może składać się z kilku segmentów). Generowanie par ilustruje przykładowa fraza (6.10) i wydobyte z niej pary (6.11–6.13).

(6.10) [$_{NP}$ przez **Ministra** Edukacji Narodowej]
(6.11) Ministra, przez, `NP`
(6.12) Ministra, Edukacji, `NP`
(6.13) Ministra, Narodowej, `NP`

Jako że wyjście algorytmu CRF nie zawiera informacji o centrach semantycznych (ani nadrzędnikach składniowych), napisano prosty skrypt, który znajduje takie centra w frazach rzeczownikowych i przymiotnikowych za pomocą bardzo prostych reguł

napisanych ręcznie. Reguły te opierają się na założeniu, że typowym centrum semantycznym fraz NP są rzeczowniki lub gerundia, natomiast typowymi centrami fraz AdjP są przymiotniki bądź zaimki przymiotne. Drugim założeniem jest, że centra semantyczne na ogół położone są bliżej początku frazy, więc jeśli występuje więcej elementów spełniających kryteria, wybierany jest pierwszy od lewej.

Ocenę wydobytych miar powiązania przeprowadzamy zgodnie z metodyką przedstawioną w pracy (Piasecki i inni, 2009). Stosowane są dwa *testy synonimii*: HWBST (łatwiejszy) i EWBST (trudniejszy). Testy zakładają użycie podzbioru rzeczowników, które wystąpiły jednocześnie w Słowosieci i w analizowanym korpsusie NKJP (w przeprowadzonych przez nas testach rzeczowników tych było prawie 50 tys.). Każdemu z tych rzeczowników przypisywany jest losowy synonim ze Słowosieci (prawidłowa odpowiedź w teście) oraz trzy inne słowa niebędące synonimami. Miary HWBST i EWBST różnią się kryterium wyboru słów niebędących synonimami — w przypadku miary EWBST słowa te są wybierane z fragmentu grafu Słowosieci, który znajduje się w ustalonej odległości od danego rzeczownika. Poniżej przedstawiamy przykład automatycznie wygenerowanego testu HWBST (6.14) oraz EWBST (6.15). Prawidłowe odpowiedzi oznaczono znakiem ✓.

(6.14) **gruźlica**
- annalista
- lilia
- poidło
- suchoty ✓

(6.15) **gruźlica**
- dna
- koklusz
- suchoty ✓
- szkarlatyna

W ten sposób wygenerowane testy „rozwiązywane są" przez automatycznie wydobyte miary powiązania znaczeniowego: zgodnie z wartością miary wybierany jest najlepszy kandydat dla danego rzeczownika. Jako ostateczny wynik testu podawana jest wartość **trafności** rozumianej jako procent testów (rzeczowników, których synonim jest poszukiwany), gdzie miara pozwoliła wybrać prawidłową odpowiedź.

Oceny istotności statystycznej dokonano w sposób zbieżny z wcześniejszymi badaniami na tym polu (Broda i Piasecki, 2008; Piasecki i inni, 2009), tj. przy pomocy testu χ^2. Podobnie jak w przypadku pozostałych testów statystycznych przeprowadzanych w tej rozprawie, przyjmujemy poziom istotności $\alpha = 0,05$.

Wyniki eksperymentów przedstawiono w tabeli 6.2. Kolumny *HWBST* i *EWBST* przedstawiają wartości trafności osiągnięte w tych testach. Użycie par wyrazów pochodzących z obu parserów dało nieco niższe wartości trafności niż osiągnięte przy użyciu predykatów pisanych ręcznie. Różnica ta jest istotna statystycznie w przypadku obu testów. Różnicę tę może tłumaczyć fakt, że predykaty były pisane specjalnie pod to zadanie, a użyty tutaj wybór predykatów został poparty kilkoma wcześniejszymi eksperymentami (Broda i inni, 2009; Piasecki i inni, 2009).

Z drugiej strony, połączenie par pochodzących z parsera z parami uzyskanymi za pomocą predykatów WCCL pozwoliło osiągnąć lepsze niż dotychczas wyniki. Co wię-

Źródło par	HWBST	EWBST
Predykaty WCCL	73,22%	53,30%
CRF	71,24%	52,40%
Spejd bez łączenia	68,67%	51,53%
Spejd z łączeniem	70,71%	52,32%
WCCL + CRF	74,72%	55,38%
WCCL + Spejd	74,54%	54,76%

Tabela 6.2. Wartości trafności osiągniętej w testach synonimii przy użyciu kilku źródeł par słów.

cej, zastosowanie samego parsera opartego na algorytmie CRF daje lepsze wyniki niż zastosowanie samego parsera Spejd (różnica jest istotna statystycznie). Szczegółowe porównanie trafności osiągniętych w testach EWBST i HWBST przedstawiamy poniżej.

1. Dla obu testów następujące różnice są istotne statystycznie:
   — $\mathrm{Acc}_{\mathrm{WCCL}} \gg \mathrm{Acc}_{\mathrm{CRF}}$
   — $\mathrm{Acc}_{\mathrm{WCCL}} \gg \mathrm{Acc}_{\mathrm{Spejd}}$
   — $\mathrm{Acc}_{\mathrm{CRF}} \gg \mathrm{Acc}_{\mathrm{Spejd}}$
   — $\mathrm{Acc}_{\mathrm{WCCL+Spejd}} \gg \mathrm{Acc}_{\mathrm{WCCL}}$
2. Dla obu testów nieistotna jest różnica między trafnością uzyskaną przez układ WCCL + CRF a układ WCCL + Spejd:
   — $\mathrm{Acc}_{\mathrm{WCCL+CRF}} > \mathrm{Acc}_{\mathrm{WCCL+Spejd}}$
3. Wzrost trafności dzięku dodaniu do predykatów WCCL krotek pozyskanych algorytmem CRF jest istotny jedynie w teście EWBST:
   — $\mathrm{EWBST}_{\mathrm{WCCL+CRF}} \gg \mathrm{EWBST}_{\mathrm{WCCL}}$
   — $\mathrm{HWBST}_{\mathrm{WCCL+CRF}} > \mathrm{HWBST}_{\mathrm{WCCL}}$

Eksperymenty te można podsumować w następujący sposób. Moduły znakowania fraz można z powodzeniem zastosować jako narzędzia wspomagające wydobywanie relacji semantycznych. Wyniki osiągnięte przy ich pomocy są nieco niższe niż te uzyskane dzięki zastosowaniu predykatów o charakterze składniowym napisanych ręcznie pod to konkretne zastosowanie, choć różnica ta jest nieduża. Użycie modułów znakowania fraz wraz z tymi predykatami pozwala poprawić wyniki, co można zaobserwować w wynikach testu EWBST. Wreszcie, wszystkie przebadane konfiguracje korzystające z algorytmu CRF działają nie gorzej niż te, gdzie algorytm ten zastąpiono parserem Spejd wyposażonym w reguły pisane ręcznie (w niektórych sytuacjach przewaga algorytmu CRF nad Spejdem jest istotna statystycznie).

6.3. Podsumowanie

Przedstawiliśmy badania pokazujące możliwości zastosowania opisywanych metod znakowania fraz w dwóch praktycznych systemach przetwarzania języka polskiego: sys-

temie wydobywania terminów ekonomicznych z korpusu dziedzinowego oraz systemie wydobywania relacji powiązania znaczeniowego na podstawie analizy wielkich korpusów językowych. Metoda znakowania fraz w oparciu o technikę maszynowego uczenia, mianowicie warunkowe pola losowe, znalazła zastosowanie w obu tych systemach. Przeprowadzone badania pokazały, że osiągnięte przy jej pomocy wyniki są nie gorsze niż te uzyskane za pośrednictwem parsera Spejd wyposażonego w napisaną przez lingwistę gramatykę NKJP. Badania te wskazują na praktyczną wartość opracowanej metody.

Rozdział 7

Podsumowanie

Celem pracy było:

1. udoskonalenie znanych dotychczas metod znakowania morfosyntaktycznego języka polskiego korzystających z technik maszynowego uczenia oraz
2. opracowanie metody znakowania fraz w języku polskim, która uczyć się będzie na korpusie oznakowanym ręcznie.

Cele te zostały osiągnięte poprzez realizację zaplanowanych zadań badawczych:

1. **Przebadanie algorytmów znakowania morfosyntaktycznego stosowanych dla języka polskiego.**
   Dokonano krytycznej oceny popularnych metod oceny tagerów i zaproponowano metodykę oceny uwzględniającą często zaniedbywane błędy popełniane na etapie segmentacji i analizy morfologicznej. Zaproponowana metodyka została opracowana w bliskiej współpracy z Szymonem Acedańskim, twórcą tagera PANTERA. Szymon Acedański miał też udział w dyskusjach na temat różnic segmentacji w wyjściu tagera w stosunku do korpusu wzorcowego. Wyniki tej współpracy zostały opublikowane w artykule (Radziszewski i Acedański, 2012).
   Zastosowanie zaproponowanej metodyki pozwoliło dostrzec słabą stronę tagerów języka polskiego: bardzo niską trafność znakowania słów nieznanych. Problem ten nie był dotąd omawiany w literaturze związanej ze znakowaniem morfosyntaktycznym języka polskiego.
2. **Opracowanie ulepszonego algorytmu znakowania morfosyntaktycznego języka polskiego.**
   Opracowano metodę znakowania morfosyntaktycznego języka polskiego korzystającą z techniki znakowania warstwowego i uczenia na pamięć. Techniki te są znane z literatury, jednak nowością jest ich połączenie. Co więcej, w metodzie połączono także stosowane często dla języków słowiańskich użycie zewnętrznego analizatora morfosyntaktycznego z prostą techniką odgadywania słów, których nie ma w jego słowniku. Opis proponowanej metody, a także jej wyniki zostały również opublikowane w artykule (Radziszewski i Śniatowski, 2011b).
   Kolejną nowością jest propozycja zabiegu, który pozwala na zmiejszenie negatywnego wpływu rozbieżności między słownikiem analizatora morfosyntaktycznego a

danymi uczącymi, co prowadzi do poprawy wyników opracowanej metody znakowania morfosyntaktycznego.

3. **Przegląd praktyk i przyjmowanych definicji fraz stosowanych dla zadania znakowania fraz w językach słowiańskich.**
   Dokonano przeglądu definicji fraz zaproponowanych na potrzeby znakowania fraz w językach słowiańskich oraz praktyk stosowanych podczas znakowania frazami korpusów. Przegląd jest nowością, gdyż dostępne opracowania związane z tym tematem ograniczają się do jednego języka (autorowi rozprawy nie udało się dotrzeć do żadnych prac przeglądowych poświęconych problemowi definicji płaskich fraz dla języków słowiańskich).
4. **Opracowanie wytycznych znakowania fraz w języku polskim**.
   We współpracy z dwoma lingwistami — Markiem Maziarzem i Janem Wieczorkiem — opracowano wytyczne znakowania tekstu polskiego płaskimi frazami (Radziszewski i inni, 2012). Wytyczne uwzględniają frazy rzeczownikowe, przymiotnikowe, czasownikowe, a także proste frazy rzeczownikowe lub przymiotnikowe wykazujące uzgodnienie gramatyczne. Na podstawie wytycznych lingwiści oznakowali na poziomie składniowym fragment Korpusu Języka Polskiego Politechniki Wrocławskiej (KPWr; Broda i inni, 2012).
5. **Dostosowanie znanych metod znakowania fraz opartych na maszynowym uczeniu do specyfiki języka polskiego**.
   Wybrano trzy znane z literatury metody znakowania fraz korzystające z technik maszynowego uczenia: przy pomocy warunkowych pól losowych, uczenia na pamięć oraz indukcji drzew decyzyjnych. Dostosowano te metody do specyfiki języka polskiego poprzez zaproponowanie zestawu cech, który korzysta z pozycyjnego charakteru tagsetu oraz ważnej roli składniowej pełnionej w językach słowiańskich przez kategorie gramatyczne przypadka, liczby i rodzaju.
   Dostosowane w ten sposób metody spełniają przyjęte w pracy założenia: dzięki skorzystaniu z technik maszynowego uczenia, metody te są w stanie dostosować się do różnych definicji fraz. Zgodnie ze stanem wiedzy autora rozprawy, są to pierwsze tego typu prace przeprowadzone dla języka słowiańskiego. Aby umożliwić porównanie tych metod z płytkim parserem Spejd (Przepiórkowski, 2008) wyposażonym w reguły napisane dla języka polskiego, opracowano metodę konwersji wyjścia parsera Spejd do postaci zgodnej z przyjętą definicją zadania znakowania fraz.
6. **Badania eksperymentalne opracowanych metod znakowania morfosyntaktycznego i znakowania fraz.**
   Przeprowadzono badania eksperymentalne metod znakowania morfosyntaktycznego na Narodowym Korpusie Języka Polskiego (NKJP), które wykazały, że zaproponowana metoda znakowania morfosyntaktycznego pozwala osiągnąć lepsze wyniki niż tager PANTERA, uznawany dotąd za wyznacznik stanu badań dla języka polskiego (Acedański, 2010; Radziszewski i Acedański, 2012). Dokonano też porównania zaproponowanej metody ze znanym wcześniej prostszym modelem znakowania przy użyciu pojedynczego klasyfikatora pamięciowego (tager MBT, Daelemans i inni, 2010b). Porównanie wykazało, że poprawa uzyskana dzięki wprowadzeniu znakowania warstwowego jest duża.
   Przeprowadzono również badania, które pozwoliły ocenić opracowane metody zna-

kowania fraz na dwóch korpusach: KPWr oraz NKJP. Badania wykazały, że metoda korzystająca z warunkowych pól losowych pozwala na uzyskanie lepszych wyników znakowania fraz niż pozostałe testowane metody. Badania przeprowadzone na korpusie NKJP wykazały, że metoda ta osiąga lepsze wyniki znakowania fraz niż metoda zakładająca użycie parsera Spejd (eksperymenty te opublikowano również w artykule Radziszewski i Pawlaczek, 2012). Eksperymenty dowiodły również, że użycie zaproponowanej metody znakowania morfosyntaktycznego wpływa korzystnie na wyniki znakowania fraz (w stosunku do konfiguracji, gdzie stosowany był tager PANTERA).

Przeprowadzono również badania nad możliwością zastosowania opracowanej metody jako narzędzia wspomagającego dwa wybrane systemy przetwarzania języka naturalnego: system wydobywania terminów ekonomicznych oraz system wydobywania miary powiązania znaczeniowego. Pokazano, że metoda oparta na warunkowych polach losowych daje wyniki nie gorsze niż te osiągnięte przy pomocy parsera Spejd. Co więcej, użycie proponowanej metody w połączeniu z dotychczas stosowanymi predykatami składniowymi napisanymi ręcznie pozwoliło uzyskać lepsze niż dotychczas wyniki wydobywania miary powiązania znaczeniowego.

Cele pracy zostały osiągnięte, gdyż:

1. Opracowano nową metodę znakowania morfosyntaktycznego języka polskiego, która łączy kilka znanych technik (uczenie pamięciowego, znakowanie warstwowego, analizę morfosyntaktyczną). Metoda pozwoliła osiągnąć lepsze wyniki niż znane dotąd metody znakowania morfosyntaktycznego języka polskiego.
2. Dostosowano do specyfiki języka polskiego trzy metody znakowania fraz w języku polskim, w tym metodę znakowania fraz przy pomocy warunkowych pól losowych. Dostosowana metoda pozwala osiągnąć wyniki znakowania fraz porównywalne z osiąganymi dzięki zastosowaniu płytkiego parsera Spejd wyposażonego w gramatykę powierzchniową języka polskiego (wyniki porównania z korpusem wzorcowym wskazują na istotną przewagę proponowanej metody, zaś wyniki badań w kontekście dwóch konkretnych aplikacji sugerują, że oba rozwiązania przynoszą podobne rezultaty).
3. Obie zaproponowane metody (tj. znakowania morfosyntaktycznego oraz znakowania fraz) opierają swoje działanie na technikach maszynowego uczenia.

W pracy postawiono następującą tezę:

> Metody znakowania morfosyntaktycznego i płytkiej analizy składniowej oparte na technikach maszynowego uczenia umożliwiają budowę praktycznych systemów przetwarzania języka polskiego.

Teza została wykazana, gdyż wykazaliśmy użyteczność opracowanych metod opartych na technikach maszynowego uczenia w dwóch systemach przetwarzania języka polskiego: systemie budującym słownik terminologiczny na podstawie automatycznej analizy korpusu dziedzinowego oraz systemie wydobywania relacji semantycznych.

Realizacja celu pracy pozwoliła na odkrycie nowych problemów badawczych. Ciekawym kierunkiem dalszych prac może być ocena prezentowanych metod znakowania

fraz na korpusach innych języków słowiańskich. Szczególnie obiecujący wydaje się pod tym względem korpus języka chorwackiego CW100 (Vučković i inni, 2010) ze względu na jego rozmiar oraz przyjętą definicję fraz rzeczownikowych zbliżoną do stosowanej w korpusie NKJP.

Eksperymenty ze znakowaniem fraz wykazały przewagę metody zakładającej wykorzystanie warunkowych pól losowych nad techniką uczenia na pamięć. Ze względu na problem złożoności obliczeniowej i dużego tagsetu języka polskiego warunkowych pól losowych nie przebadaliśmy pod kątem znakowania morfosyntaktycznego języka polskiego. W momencie pisania rozprawy trwają dalsze prace w tej dziedzinie. Wstępne eksperymenty wykazały, że zastosowanie techniki znakowania warstwowego pozwala zredukować skalę tego problemu, a co za tym idzie, prawdopodobnie możliwe będzie zastosowanie warunkowych pól losowych do znakowania morfosyntaktycznego języka polskiego przy użyciu pełnego tagsetu NKJP.

W ramach dalszych badań planowane jest także opracowanie metody rozpoznawania wybranych relacji składniowych pomiędzy frazami, w szczególności relacji *podmiot* i *dopełnienie*. *De facto*, wytyczne znakowania KPWr uwzględniają już ten poziom znakowania (Radziszewski i inni, 2012), a część korpusu, którą oznakowano frazami, oznakowano także tymi relacjami (Broda i inni, 2012). Wprowadzenie tego etapu przetwarzania pozwoli uzyskać strukturę składniową nieco bliższą analizie głębokiej.

Dodatek A

Oprogramowanie

Do przeprowadzenia badań eksperymentalnych konieczna była implementacja opracowanych metod, a także kilku pomocniczych narzędzi. Ponieważ powstałe w ten sposób oprogramowanie ma wartość praktyczną, omówimy je w skrócie w tym dodatku. Przy opisie poszczególnych narzędzi podajemy także odsyłacze do stron internetowych, gdzie można znaleźć szczegółową dokumentację, a także uzyskać dostęp do kodów źródłowych.

Wszystkie poniżej opisywane pakiety oprogramowania zostały udostępnione na otwartych licencjach na zasadzie wolnego oprogramowania.

A.1. Maca — system analizy morfosyntaktycznej

Analizator morfosyntaktyczny Morfeusz (Woliński, 2006) jest dostarczany jako biblioteka programistyczna oraz proste narzędzie uruchamialne. Niestety, narzędzie to spełnia jedynie rolę interaktywnej demonstracji analizatora, lecz nie pozwala na analizę tekstu dowolnej długości. Co więcej, na wyjściu Morfeusza pojawiają się struktury grafowe, podczas gdy badane algorytmy wymagają struktury liniowej — ciągu segmentów (por. str. 14).

Potrzebne było zatem narzędzie, które umożliwia analizę morfosyntaktyczną czystego tekstu przy pomocy Morfeusza i wygenerowanie w ten sposób wyjścia zawierającego ciąg segmentów z przypisanymi interpretacjami morfosyntaktycznemi. Potrzebny był również podział tekstu na zdania.

W tym celu zrealizowano system **Maca** (*Morphological Analysis Converter and Aggregator*). System może spełniać następujące funkcje:

1. analiza morfosyntaktyczna czystego tekstu lub pliku XML, wynikiem czego jest korpus zapisany w formacie XCES (lub innym),
2. wparcie dla obu wersji Morfeusza: SGJP lub SIAT,
3. obsługa prostej heurystyki wyboru najkrótszej ścieżki w grafie Morfeusza celem zwrócenia liniowej struktury,
4. analiza morfosyntaktyczna w oparciu o słownik napisany przez użytkownika,

5. budowę bardziej złożonych potoków analizy, gdzie w zależności od wstępnie wyodrębnionego rodzaju segmentu (np. liczba, słowo, symbol), może być stosowana inna strategia analizy,
6. podział teksty na segmenty w oparciu o reguły napisane ręcznie w prostym pliku INI,
7. podział tekstu na zdania w oparciu o reguły w standardzie SRX,
8. proste konwersje tagsetu.

Implementacja dostarczana jest pod postacią biblioteki programistycznej (dostępny interfejs w języku C++ i Python) oraz narzędzia uruchamialnego `maca-analyse`. Narzędzie uruchamialne jest w pełni funkcjonalne i pozwala na przetwarzanie strumieniowe wielkich korpusów.

System jest konfigurowalny: konkretne zachowanie analizatora można opisać w prostym pliku INI. Plik może zawierać odwołania do zewnętrznych zasobów (słowników morfosyntaktycznych, reguł podziału na segmenty i zdania). Podczas przeprowadzanych w tej pracy eksperymentów używana była konfiguracja `morfeusz-nkjp-official-guesser`.

System Maca opisano w artykule (Radziszewski i Śniatowski, 2011a).

Strona projektu: `http://nlp.pwr.wroc.pl/redmine/projects/libpltagger/wiki/`.

A.2. WCCL — narzędzie generowania cech morfosyntaktycznych

W rozdziale 2.6 opisaliśmy formalizm WCCL, który pozwala na zapis wyrażeń funkcyjnych wartościowanych na zdaniach oznakowanych morfosyntaktycznie. Wyrażenia te używane są głównie jako cechy dla zadań przetwarzania języka naturalnego w oparciu o techniki maszynowego uczenia, w tym znakowania morfosyntaktycznego (patrz punkt 2.7.3) oraz znakowania fraz (4.8.3).

Formalizm WCCL został zaimplementowany w systemie o tej samej nazwie. Implementacja składa się z biblioteki programistycznej (dostępne są dwa interfejsy: w języku C++ i Python) oraz narzędzia uruchamialnego `wccl-run`, umożliwiającego przetworzenie korpusu poddanego analizie morfosyntaktycznej (opcjonalnie, także ujednoznacznieniu morfosyntaktycznemu) w prosty plik tekstowy zawierający wartości wyrażeń funkcyjnych.

Formalizm i system WCCL zostały opisane w pracy (Radziszewski i inni, 2011c).

Strona projektu (zawiera również formalną specyfikację formalizmu): `http://nlp.pwr.wroc.pl/redmine/projects/joskipi/wiki/`.

A.3. WMBT — warstwowy tager pamięciowy języka polskiego

Zaproponowana w punktach 2.7–2.8 metoda została zaimplementowana w postaci praktycznego tagera języka polskiego o nazwie WMBT (*Wrocław Memory-Based Tagger*. Tager został napisany w języku Python przy użyciu interfejsów pythonowych opisanych w poprzednich punktach systemów — Maca i WCCL.

Tager pozwala na oznakowanie czystego tekstu (w tym celu używana jest Maca), a także na ujednoznacznienie tekstu poddanego już analizie morfosyntaktycznej.

Tager jest w pełni konfigurowalny. Użytkownik może określić stosowany tagset, zdefiniować cechy w formalizmie WCCL, włączyć lub wyłączyć moduł rozpoznawania słów nieznanych, a także zmienić kolejność stosowania warstw (możliwość ta nie została przebadana w pracy — zastosowano kolejność zgodną z kolejnością zdefiniowania w tagsecie atrybutów). Konfiguracje tagera zapisywane są w pliku INI, który wskazuje także na plik w składni WCCL. W pracy testowane były dwa warianty, odpowiadające konfiguracjom `nkjp-noguess.ini` (bez odgadywania słów nieznanych) oraz `nkjp-guess.ini` (z odgadywaniem).

Wraz z kodami źródłowymi dostarczany jest skrypt `reanalyse`, który pozwala przeprowadzić ponowną analizę morfosyntaktyczną danych uczących (punkt 2.9).

Tager został opisany w pracy (Radziszewski i Śniatowski, 2011b).

Strona projektu: `http://nlp.pwr.wroc.pl/redmine/projects/wmbt/wiki/`.

A.4. Disaster i IOBBER — moduły znakowania fraz

Dostosowane do specyfiki języka polskiego metody znakowania fraz (punkty 4.6–4.8) zostały zaimplementowane w sytemie płytkiej analizy składniowej **Disaster** (*DISAmbiguator and STatistical chunkER*). Przyjęto założenie, że analizowany składniowo tekst został już oznakowany morfosyntaktycznie. System jest konfigurowalny. W pliku INI można zdefiniować ciąg etapów przetwarzania.

Strona projektu: `nlp.pwr.wroc.pl/trac/disaster`

System Disaster z założenia miał charakter prototypowy i zmieniała się niejednokrotnie jego wizja. Negatywnie odbiło się to na jakość kodu. Uzyskanie dobrych wyników znakowania fraz za pomocą warunkowych pól losowych (por. rozdział 5) było powodem do opracowania wersji wdrożeniowej systemu. System wdrożeniowy nosi nazwę **IOBBER** (nazwa pochodzi od znaczników IOB2, por. str. 83). Nowa implementacja jest szybsza i pozwala na kilka rozszerzeń, m.in. umożliwia grupowanie fraz w „warstwy" (frazy należące do tej samej warstwy znakowane są jednocześnie i z założenia są rozłączne). Oba systemy korzystają z tej samej implementacji warunkowych pól losowych — pakietu CRF++ (Kudo, 2005).

Strona projektu IOBBER: `http://nlp.pwr.wroc.pl/redmine/projects/iobber/wiki/`.

Dodatek B

Tagset NKJP

Każdy tag tagsetu NKJP składa się z *klasy gramatycznej* oraz listy wartości przypisanych poszczególnym *atrybutom* właściwym dla danej klasy gramatycznej. W przypadku niektórych klas lista ta jest pusta. Każda klasa gramatyczna wyznacza więc dwa zbiory atrybutów:

1. zbiór atrybutów wymaganych, których wartość musi być określona;
2. zbiór atrybutów opcjonalnych, których wartość można pominąć.

Zarówno wartości atrybutów, jak i klasy gramatyczne reprezentowane są przez tekstowe symbole (mnemoniki). Zbiór mnemoników klas gramatycznych oraz zbiór mnemoników wartości atrybutów są rozłączne.

W tabeli B.1 przedstawiono przypisanie atrybutom zbiorów możliwych wartości wraz z ich symbolami (mnemonikami). Tabela B.2 przedstawia przypisanie klasom gramatycznym zbiorów atrybutów. Symbolem • oznaczono atrybuty wymagane dla danej klasy, zaś symbolem ∘ — atrybuty opcjonalne.

Atrybut	Możliwe wartości
Liczba	pojedyncza `sg`, mnoga `pl`
Przypadek	mianownik `nom`, dopełniacz `gen`, celownik `dat`, biernik `acc`, narzędnik `inst`, miejscownik `loc`, wołacz `voc`
Rodzaj	męski osobowy `m1`, męski zwierzęcy `m2`, męski rzeczowy `m3`, żeński `f`, nijaki `n`
Osoba	pierwsza `pri`, druga `sec`, trzecia `ter`
Stopień	równy `pos`, wyższy `com`, najwyższy `sup`
Aspekt	niedokonany `imperf`, dokonany `perf`
Negacja	niezanegowana `aff`, zanegowana `neg`
Akomodacyjność	uzgadniająca `congr`, rządząca `rec`
Akcentowość	akcentowana `akc`, nieakcentowana `nakc`
Poprzyimkowość	poprzyimkowa `praep`, niepoprzyimkowa `npraep`
Aglutynacyjność	aglutynacyjna `agl`, nieaglutynacyjna `nagl`
Wokaliczność	wokaliczna `wok`, niewokaliczna `nwok`
Wymaganie kropki	z kropką `pun`, bez kropki `npun`

Tabela B.1. Tagset NKJP: atrybuty i ich wartości

Klasa	Symbol	Liczba	Przypadek	Rodzaj	Osoba	Stopień	Aspekt	Negacja	Akomodacyjność	Akcentowość	Poprzyimkowość	Aglutynacyjność	Wokaliczność	Wymaganie kropki
Przymiotnik	`adj`	•	•	•		•								
Przym. przyprzymiotnikowy	`adja`													
Przym. poprzyimkowy	`adjp`													
Przym. predykatywny	`adjc`													
Spójnik współrzędny	`conj`													
Spójnik podrzędny	`comp`													
Predykatyw	`pred`													
Przysłówek	`adv`					○								
Bezosobnik	`imps`						•							
Bezokolicznik	`inf`						•							
Przyimek	`prep`		•										○	
Zaimek *siebie*	`siebie`		•											
Rzeczownik	`subst`	•	•	•										
Forma deprecjatywna	`depr`	•	•	•										
Odsłownik	`ger`	•	•	•			•	•						
Zaimek nietrzecioosobowy	`ppron12`	•	•	•	•					○				
Zaimek trzecioosobowy	`ppron3`	•	•	•	•					○	○			
Liczebnik główny	`num`	•	•	•					•					
Liczebnik zbiorowy	`numcol`	•	•	•					•					
Imiesłów przysł. uprzedni	`pant`						•							
Imiesłów przysł. współcz.	`pcon`						•							
Imiesłów przym. czynny	`pact`	•	•	•			•	•						
Imiesłów przym. bierny	`ppas`	•	•	•			•	•						
Czasownik typu *winien*	`winien`	•		•			•							
Pseudoimiesłów	`praet`	•		•			•					○		
Forma przyszła *być*	`bedzie`	•			•		•							
Forma nieprzeszła	`fin`	•			•		•							
Rozkaźnik	`impt`	•			•		•							
Aglutynant *być*	`aglt`	•			•		•						•	
Wykrzyknik	`interj`													
Burkinostka	`burk`													
Kublik	`qub`												○	
Skrót	`brev`													•
Ciało obce	`xxx`													
Interpunkcja	`interp`													
Forma nierozpoznana	`ign`													

Tabela B.2. Tagset NKJP: klasy gramatyczne i przypisane im atrybuty

Dodatek C

Zestawy cech zaproponowane dla języka polskiego

Przedstawiamy tutaj zestawy cech zaproponowane na potrzeby znakowania morfosyntaktycznego oraz znakowania fraz w języku polskim. Zestawy cech zapisane są w formalizmie WCCL. Przytoczone zapisy użyte są bezpośrednio w implementacji opracowanych algorytmów.

C.1. Tager WMBT

Poniższy zestaw cech stosowany jest przez tager WMBT, zarówno w konfiguracji zakładającej użycie modułu odgadującego nieznane słowa, jak i w konfiguracji z niego nie korzystającej.

```
@ "default" (
   class[-3]; class[-2]; class[-1]; class[0]; class[1]; class[2];
   cas[-3]; cas[-2]; cas[-1]; cas[0];      cas[1]; cas[2];
   gnd[-3]; gnd[-2]; gnd[-1]; gnd[0]; gnd[1]; gnd[2];
   nmb[-3]; nmb[-2]; nmb[-1]; nmb[0]; nmb[1]; nmb[2];
   lex(lower(orth[-3]), "fq");
   lex(lower(orth[-2]), "fq");
   lex(lower(orth[-1]), "fq");
   lex(lower(orth[0]), "fq");
   lex(lower(orth[1]), "fq");
   lex(lower(orth[2]), "fq");
   agrpp(-1,0,{nmb,gnd,cas});
   agrpp(0,1,{nmb,gnd,cas});
   and(inside(-2), wagr(-2,0,{nmb,gnd,cas}));
   and(inside(-1), inside(1), wagr(-1,1,{nmb,gnd,cas}));
   and(inside(2), wagr(0,2,{nmb,gnd,cas}));
   affix(orth[0], -1); affix(orth[0], -2); affix(orth[0], -3);
   regex(orth[0], "\\P{Ll}.*"); regex(orth[0], "\\P{Lu}.*")
)
```

C.2. Moduł znakowania fraz w oparciu o algorytm MBL

Poniższy zestaw cech zastosowano zarówno w algorytmie MBL, jak i DT.

```
@ "default" (
   class[-3]; class[-2]; class[-1]; class[0]; class[1]; class[2];
   cas[-3]; cas[-2]; cas[-1]; cas[0];      cas[1]; cas[2];
   gnd[-3]; gnd[-2]; gnd[-1]; gnd[0]; gnd[1]; gnd[2];
   nmb[-3]; nmb[-2]; nmb[-1]; nmb[0]; nmb[1]; nmb[2];
   lex(lower(orth[-3]), "fq");
   lex(lower(orth[-2]), "fq");
   lex(lower(orth[-1]), "fq");
   lex(lower(orth[0]), "fq");
   lex(lower(orth[1]), "fq");
   lex(lower(orth[2]), "fq");
   agrpp(-1,0,{nmb,gnd,cas});
   agrpp(0,1,{nmb,gnd,cas});
   and(inside(-2), wagr(-2,0,{nmb,gnd,cas}));
   and(inside(-1), inside(1), wagr(-1,1,{nmb,gnd,cas}));
   and(inside(2), wagr(0,2,{nmb,gnd,cas}));
   regex(orth[0], "\\P{Ll}.*"); regex(orth[0], "\\P{Lu}.*")
)
```

C.3. Moduł znakowania fraz w oparciu o algorytm CRF

Implementacja algorytmu CRF zakłada użycie pakietu CRF++. Pakiet ten wymaga, by dane wejściowe wzbogacone były o cechy opisujące wystąpienie każdego segmentu. Cechy te są potem rozwijane poprzez szablony cech, których rolą jest zarówno wygenerowanie na podstawie wielowartościowych cech funkcji charakterystycznych, jak i uzyskanie wielu instancji danej cechy, które odnosić się będą do pozycji sąsiadujących z bieżącym segmentem. Szablony pozwalają również na złączenie kilku cech z danych wejściowych w pojedynczą cechę (zanim zostanie ona rozbita na funkcje charakterystyczne). Przedstawiamy najpierw zestaw cech, zapisany w formalizmie WCCL, który odpowiada „podstawowym" cechom dodanym do danych wejściowych. W dalszej kolejności prezentujemy szablony cech odwołujące się do tych „podstawowych cech".

Zastosowany zestaw cech „podstawowych" wygląda następująco (w komentarzach podano numery cech, do których odwołują się szablony):

```
@ "default" (
   orth[0];  // 0
   class[0]; // 1
   cas[0];   // 2
   gnd[0];   // 3
   nmb[0];   // 4
   agrpp(0,1,{nmb,gnd,cas}); // 5
   and(inside(-1), inside(1), wagr(-1,1,{nmb,gnd,cas})); // 6
   regex(orth[0], "\\P{Ll}.*"); regex(orth[0], "\\P{Lu}.*") // 7, 8
```

)

Powyższy zestaw cech rozwijany jest zgodnie z poniższymi szablonami. Szablony podano w składni przyjmowanej przez implementację CRF++ (Kudo, 2005).

```
# Unigram
# orth
U00:%x[-2,0]
U01:%x[-1,0]
U02:%x[0,0]
U03:%x[1,0]
U04:%x[2,0]
U05:%x[-1,0]/%x[0,0]
U06:%x[0,0]/%x[1,0]

# class
U10:%x[-2,1]
U11:%x[-1,1]
U12:%x[0,1]
U13:%x[1,1]
U14:%x[2,1]
U15:%x[-2,1]/%x[-1,1]
U16:%x[-1,1]/%x[0,1]
U17:%x[0,1]/%x[1,1]
U18:%x[1,1]/%x[2,1]

# cas
U20:%x[-2,2]
U21:%x[-1,2]
U22:%x[0,2]
U23:%x[1,2]
U24:%x[2,2]

# gnd
U30:%x[-2,3]
U31:%x[-1,3]
U32:%x[0,3]
U33:%x[1,3]
U34:%x[2,3]

# nmb
U40:%x[-2,4]
U41:%x[-1,4]
U42:%x[0,4]
U43:%x[1,4]
U44:%x[2,4]
```

```
# agr
U50:%x[-1,5] # agr(0,1) -> agr(-1,0)
U51:%x[0,5] # agr(0,1)
U52:%x[-1,6] # agr..(-1,1) -> agr(-2,0)
U53:%x[0,6] # (-1,1)
U54:%x[1,6] # ... -> (0,2)

# regex feats
#U60:%x[-1,7]/%x[-1,8]
U61:%x[0,7]/%x[0,8]
#U62:%x[1,7]/%x[1,8]

# wordclass trigrams
U80:%x[-2,1]/%x[-1,1]/%x[0,1]
U81:%x[-1,1]/%x[0,1]/%x[1,1]
U82:%x[0,1]/%x[1,1]/%x[2,1]

# Bigram
B
```

Bibliografia

Abney, S. (1991). Parsing by chunks. W: *Principle-Based Parsing*, strony 257–278. Kluwer Academic Publishers.

Abney, S. (1995). Chunks and dependencies: Bringing processing evidence to bear on syntax. W: *Computational Linguistics and the Foundations of Linguistic Theory*, strony 145–164. CSLI.

Abney, S. (1996a). Chunk stylebook. http://www.vinartus.net/spa/96i.pdf.

Abney, S. (1996b). Partial parsing via finite-state cascades. *Natural Language Engineering*, 2(4):337–344, Cambridge University Press.

Acedański, S. (2010). A morphosyntactic Brill tagger for inflectional languages. W: Loftsson, H., Rögnvaldsson, E., i Helgadóttir, S., red., *Advances in Natural Language Processing*, volume 6233 of *Lecture Notes in Computer Science*, strony 3–14. Springer Berlin / Heidelberg.

Acedański, S. i Gołuchowski, K. (2009). A morphosyntactic rule-based Brill tagger for Polish. W: *Proceedings of Intelligent Information Systems*, strony 67–76.

Acedański, S. i Przepiórkowski, A. (2010). Towards the adequate evaluation of morphosyntactic taggers. W: *Proceedings of the 23rd International Conference on Computational Linguistics (COLING)*, strony 1–8, Pekin, Chiny. Coling 2010 Organizing Committee.

Aho, A. V. i Ullman, J. D. (1972). *The Theory of Parsing, Translation and Compiling*, volume 1. Prentice-Hall, Englewood Cliffs, NJ.

Appelt, D. E., Hobbs, J. R., Bear, J., Israel, D., Kameyama, M., i Tyson, M. (1993). SRI: description of the JV-FASTUS system used for MUC-5. W: *Proceedings of the 5th conference on Message understanding*, strony 221–235, Stroudsburg, USA. Association for Computational Linguistics.

Babarczy, A., Carroll, J., i Sampson, G. (2005). Definitional, personal, and mechanical constraints on part of speech annotation performance. *Natural Language Engineering*, 12(1):77–90, Cambridge University Press.

Bień, J. S. i Woliński, M. (2003). Wzbogacony korpus *Słownika frekwencyjnego polszczyzny współczesnej*. W: Linde-Usiekniewicz, J. i Huszcza, R., red., *Prace językoznawcze dedykowane Profesor Jadwidze Sambor*, strony 6–10. Uniwersytet Warszawski, Wydział Polonistyki, Warszawa.

Bird, S., Klein, E., i Loper, E. (2009). *Natural Language Processing with Python*. O'Reilly Media.

Brants, T. (2000). Tnt – a statistical part-of-speech tagger. W: *Proceedings of the Sixth Conference on Applied Natural Language Processing*, strony 224–231, Seattle, USA. Association for Computational Linguistics.

Brill, E. (1992). A simple rule-based part of speech tagger. W: *Proceedings of the Third Conference on Applied Natural Language Processing*, strony 152–155, Morristown, USA. Association for Computational Linguistics.

Broda, B., Marcińczuk, M., Maziarz, M., Radziszewski, A., i Wardyński, A. (2012). KPWr: Towards a free corpus of Polish. W: Calzolari, N., Choukri, K., Declerck, T., Doğan, M. U., Maegaard, B., Mariani, J., Odijk, J., i Piperidis, S., red., *Proceedings of LREC'12*, Stambuł, Turcja. ELRA.

Broda, B. i Piasecki, M. (2008). SuperMatrix: a general tool for lexical semantic knowledge acquisition. W: *Proceedings of the International Multiconference on Computer Science and Information Technology — 3rd International Symposium Advances in Artificial Intelligence and Applications (AAIA'08)*, strony 345–352.

Broda, B., Piasecki, M., i Szpakowicz, S. (2009). Rank-based transformation in measuring semantic relatedness. W: *Advances in Artificial Intelligence: 22nd Canadian Conference on Artificial Intelligence*, volume 5549 of *LNCS*, strony 187–190. Springer Verlag.

Charniak, E. (2000). A maximum-entropy-inspired parser. W: *Proceedings of NAACL-2000*.

Church, K. W. (1988). A stochastic parts program and noun phrase parser for unrestricted text. W: *Proceedings of the second conference on Applied natural language processing*, ANLC '88, strony 136–143, Teksas, USA. Association for Computational Linguistics.

Clark, S., Curran, J., i Osborne, M. (2003). Bootstrapping pos-taggers using unlabelled data. W: Daelemans, W. i Osborne, M., red., *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, strony 49–55.

Cohn, T. (2007). *Scaling conditional random fields for natural language processing*. rozprawa doktorska, Department of Computer Science and Software Engineering, University of Melbourne, Australia.

Collins, M. (1999). *Head-Driven Statistical Models for Natural Language Parsing*. rozprawa doktorska, University of Pennsylvania.

Cortes, C. i Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3):273–297, Springer.

Cunningham, H., Maynard, D., i Tablan, V. (2000). JAPE: a Java Annotation Patterns Engine (second edition). Research Memorandum CS–00–10, Department of Computer Science, University of Sheffield.

Cussens, J. (1997). Part-of-speech tagging using progol. W: Lavrač, N. i Džeroski, S., red., *Inductive Logic Programming, 7th International Workshop*, volume 1297 of *LNCS*, strony 93–108. Springer.

Cussens, J., Dzeroski, S., i Erjavec, T. (1999). Morphosyntactic tagging of Slovene using Progol. W: *Inductive Logic Programming, 9th International Workshop*, volume 1634 of *LNCS*, strony 68–79, Bled, Słowenia. Springer.

Cussens, J., Page, D., Muggleton, S., i Srinivasan, A. (1997). Using inductive logic

programming for natural language processing. W: *Proceedings of ECML'97*, strony 25–34, Praga, Czechy. Springer-Verlag.

Cutting, D., Kupiec, J., Pedersen, J., i Sibun, P. (1992). A practical part-of-speech tagger. W: *Proceedings of the Third Conference on Applied Natural Language Processing*, strony 133–140, Trento, Włochy. Association for Computational Linguistics.

Daelemans, W., Buchholz, S., i Veenstra, J. (1999). Memory-based shallow parsing. W: *Proceedings of the CoNLL 1999*. Association for Computational Linguistics.

Daelemans, W. i van den Bosch, A. (2005). *Memory-Based Language Processing*. Cambridge University Press.

Daelemans, W., Zavrel, J., i Ko van der Sloot, A. V. d. B. (2010a). TiMBL: Tilburg Memory Based Learner, version 6.3, reference guide. Raport nr 10-01, ILK.

Daelemans, W., Zavrel, J., Van den Bosch, A., i van der Sloot, K. (2010b). MBT: Memory-Based Tagger, version 3.2. Raport nr 10-04, ILK.

Degórski, Ł. i Przepiórkowski, A. (2012). Ręcznie znakowany milionowy podkorpus NKJP. W: Przepiórkowski i inni (2012).

Déjean, H. (2000). Learning syntactic structures with xml. W: Cardie, C., Daelemans, W., Nedellec, C., i Tjong Kim Sang, E., red., *Proceedings of CoNLL-2000 and LLL-2000*, strony 133–135. Lisbon, Portugal.

Derwojedowa, M., Gałczyńska, A., Gruszczyński, W., Kopcińska, D., Linde-Usiekniewicz, J., i Winiarska-Górska, I. (2005). *Język polski. Kompendium*. Świat Książki.

Dietterich, T. G. (1998). Approximate statistical test for comparing supervised classification learning algorithms. *Neural Computation*, 10(7):1895–1923.

Dębowski, L. (2001). Tagowanie i dezambiguacja morfosyntaktyczna. przegląd metod i oprogramowania. Raport nr 934, IPI PAN, Warszawa.

Dębowski, L. (2004). Trigram morphosyntactic tagger for Polish. W: *Intelligent Information Processing and Web Mining, Proceedings of the International IIS: IIPWM'04 Conference*, strony 409–413, Zakopane. Springer.

Dojchinova, V. i Mihov, S. (2004). High performance part-of-speech tagging of bulgarian. W: *Artificial Intelligence: Methodology, Systems, and Applications, 11th International Conference*, strony 246–255. Springer.

Džeroski, S., Erjavec, T., i Zavrel, J. (1999). Morphosyntactic tagging of Slovene: Evaluating taggers and tagsets. Raport nr IJS-DP 8018, Instytut Josefa Stefana, Lublana, Słowenia.

Etzioni, O., Banko, M., Soderland, S., i Weld, D. S. (2008). Open information extraction from the web. *Communications of the ACM — Surviving the data deluge*, 51(12):68–74, ACM.

Federici, S., Montemagni, S., i Pirrelli, V. (1997). Shallow parsing and text chunking: a view on underspecification in syntax. W: *Proceedings of the Eight European Summer School In Logic, Language and Information*, Praga, Czechy.

Feldman, A. i Hana, J. (2010). *A resource-light approach to morpho-syntactic tagging*. Rodopi, Amsterdam/Nowy Jork.

Grác, M., Jakubíček, M., i Kovář, V. (2010). Through low-cost annotation to reliable parsing evaluation. W: *Proceedings of the 24th Pacific Asia Conference on Language, Information and Computation*, strony 555–562, Tokio. Waseda University.

Grefenstette, G. (1996). Light parsing as finite state filtering. W: *Proceedings of the*

ECAI '96 workshop on Extended finite state models of language. Budapeszt.

Grzegorczykowa, R. (2006). *Wykłady z polskiej składni*. PWN, Warszawa.

Górski, R. L. i Łaziński, M. (2012). Reprezentatywność i zrównoważenie korpusu. W: Przepiórkowski i inni (2012).

Głowińska, K. (2011). Granice frazy nominalnej. Dokument roboczy otrzymany od autorki 10.01.2012.

Głowińska, K. (2012). Anotacja składniowa. W: Przepiórkowski i inni (2012).

Hajič, J. (2000). Morphological tagging: Data vs. dictionaries. W: *Proceedings of the 6th Applied Natural Language Processing and the 1st NAACL Conference*, strony 94–101.

Hajič, J., Krbec, P., Květoň, P., Oliva, K., i Petkevič, V. (2001). Serial combination of rules and statistics: A case study in Czech tagging. W: *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, strony 268–275. Association for Computational Linguistics.

Hajič, J. i Vidová-Hladká, B. (1998a). Czech language processing — PoS tagging. W: *Proceedings of the 1st International Conference on Language Resources and Evaluation*, strony 931–936.

Hajič, J. i Vidová-Hladká, B. (1998b). Tagging inflective languages: Prediction of morphological categories for a rich, structured tagset. W: *Proceedings of the COLING - ACL Conference*, strony 483–490. ACL.

Hajnicz, E. i Kupść, A. (2001). Przegląd analizatorów morfologicznych dla języka polskiego. Raport nr 937, IPI PAN.

Hindle, D. (1989). Acquiring disambiguation rules from text. W: *Proceedings of the 27th Annual Meeting of the Association for Computational Linguist*, strony 118–125.

Hobbs, J. R. (1992). Fastus: A system for extracting information from natural-language text. Raport nr 519, AI Center, SRI International, 333 Ravenswood Ave., Menlo Park, CA 94025.

Hobbs, J. R. i Riloff, E. (2010). Information extraction. W: Indurkhya, N. i Damerau, F. J., red., *Handbook of Natural Language Processing*. Chapman & Hall/CRC Press, Taylor & Francis Group, wydanie drugie.

Hollingshead, K., Fisher, S., i Roark, B. (2005). Comparing and combining finite-state and context-free parsers. W: *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, strony 787–794. Association for Computational Linguistics.

Hulth, A. (2003). Improved automatic keyword extraction given more linguistic knowledge. W: Collins, M. i Steedman, M., red., *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, strony 216–223.

Jakubíček, M., Horák, A., i Kovář, V. (2009). Mining phrases from syntactic analysis. W: *Proceedings of the 12th International Conference on Text, Speech and Dialogue*, TSD '09, strony 124–130, Berlin. Springer-Verlag.

Jinguji, D., Lewis, W., Efthimiadis, E. N., Minor, J., Bertram, A., Eggers, S., Johanson, J., Nisonger, B., Yu, P., i Zhou, Z. (2006). The University of Washington's UWclmaQA system. W: Voorhees, E. M. i Buckland, L. P., red., *Proceedings of TREC 2006*, Gaithersburg, Maryland, USA. NIST.

Johansson, C. (2000). A context sensitive maximum likelihood approach to chunking. W: *Proceedings of CoNLL-2000 and LLL-2000*. Lisbon, Portugal.

Karlsson, F. (1990). Constraint Grammar as a Framework for Parsing Running Text. W: Karlgren, H., red., *Proceedings of the 13th Conference on Computational Linguistics*, volume 3, strony 168–173, Helsinki, Finlandia.

Karolak, S. (1999a). Grupa nominalna. W: Polański (1999a), strony 225–227.

Karolak, S. (1999b). Grupa werbalna. W: Polański (1999a), strony 229–230.

Karwańska, D. i Przepiórkowski, A. (2010). On the evaluation of two Polish taggers. W: Goźdź-Roszkowski, S., red., *The proceedings of Practical Applications in Language and Computers PALC 2009*, Frankfurt, Niemcy. Peter Lang.

Kilgarriff, A., Rychlý, P., Smrž, P., i Tugwell, D. (2004). The sketch engine. W: *Proceedings of EURALEX*, Lorient, Francja.

Koeling, R. (2000). Chunking with maximum entropy models. W: Cardie, C., Daelemans, W., Nedellec, C., i Tjong Kim Sang, E., red., *Proceedings of CoNLL-2000 and LLL-2000*, strony 139–141. Lisbon, Portugal.

Korkontzelos, I., Klapaftis, I., i Manandhar, S. (2008). Reviewing and evaluating automatic term recognition techniques. W: *Proceedings of the 6th International Conference on Natural Language Processing, GoTAL 2008*, Gothenburg, Szwecja.

Koronacki, J. i Ćwik, J. (2005). *Statystyczne systemy uczące się*. Wydawnictwo Exit, Warszawa.

Kotsyba, N., Radziszewski, A., i Derzhanski, I. (2009). Integrating the Polish language into the MULTEXT-Eeast family: Morphosyntactic specifications, converter, lexicon and corpus. W: Erjavec, T., red., *Research Infrastructure for Digital Lexicography: Proceedings of MONDILEX Fifth Open Workshop*, strony 37–55, Lublana, Słowenia.

Krenn, B. i Samuelsson, C. (1997). The linguist's guide to statistics - don't panic.

Kudo, T. (2005). CRF++: Yet another CRF toolkit. Instrukcja i implementacja dostępne na stronie http://crfpp.googlecode.com/svn/trunk/doc/index.html.

Kudoh, T. i Matsumoto, Y. (2000). Use of support vector learning for chunk identification. W: *Proceedings of CoNLL-2000 and LLL-2000*. Lisbon, Portugal.

Kudoh, T. i Matsumoto, Y. (2001). Chunking with support vector machines. W: *Proceedings of NAACL 2001*. Pittsburgh, USA.

Kuta, M. (2010). *Tagging and Corpus based Methods for improving Natural Language Processing of Polish*. rozprawa doktorska, Wydział Elektrotechniki, Automatyki, Informatyki i Elektroniki, Akademia Górniczo-Hutnicza, Kraków.

Lafferty, J., McCallum, A., i Pereira, F. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. W: *Proceedings of the Eighteenth International Conference on Machine Learning (ICML-2001)*.

Lavrač, N. i Džeroski, S. (1994). *Inductive Logic Programming: Techniques and Applications*. Ellis Horwood, Nowy Jork.

Manicki, L. (2009). Płytki parsing języka francuskiego. praca magisterska, Wydział Matematyki i Informatyki Uniwersytetu im. Adama Mickiewicza.

Manning, C. D. i Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. MIT Press.

Marciniak, M. (2010). Wyodrębnianie prostych fraz. W: Marciniak, M., red., *Anotowany korpus dialogów telefonicznych*, rozdział 6. Akademicka Oficyna Wydawnicza EXIT.

Marciniak, M. i Mykowiecka, A. (2012). Terminology extraction from domain texts in Polish. W: *Intelligent Tools for Building a Scientific Information Platform: Advanced*

Architectures and Solutions. Springer Verlag.

Màrquez, L. (1999). *Part-of-speech Tagging: A Machine Learning Approach based on Decision Trees*. rozprawa doktorska, Universitat Politécnica de Catalunya.

Mastalerz, R. (2011). Tager maksimum entropii dla języka polskiego. praca magisterska, Uniwersytet Warszawski, Wydział Matematyki, Informatyki i Mechaniki.

McCallum, A., Freitag, D., i Pereira, F. C. N. (2000). Maximum entropy markov models for information extraction and segmentation. W: Langley, P., red., *Proceedings of the Seventeenth International Conference on Machine Learning*, strony 591–598. Morgan Kaufmann.

McDonald, R., Crammer, K., i Pereira, F. (2005). Flexible text segmentation with structured multilabel classification. W: *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, strony 987–994, Stroudsburg, USA. Association for Computational Linguistics.

McEnery, T. i Wilson, A. (2001). *Corpus Linguistics: An Introduction*. Edinburgh Textbooks in Empirical Linguistics. Edinburgh University Press.

Mráková, E. i Sedláček, R. (2003). From Czech morphology through partial parsing to disambiguation. W: *Proceedings of the 4th international conference on Computational linguistics and intelligent text processing*, CICLing'03, strony 126–135, Berlin, Heidelberg. Springer-Verlag.

Müller, F. H. (2005). *A Finite-State Approach to Shallow Parsing and Grammatical Functions Annotation of German*. rozprawa doktorska, Universität Tübingen, Wilhelmstr. 32, 72074 Tübingen.

Mykowiecka, A., Marasek, K., Marciniak, M., Rabiega-Wiśniewska, J., i Gubrynowicz, R. (2007). Annotated corpus of Polish spoken dialogues. W: *Proceedings of the Third Language and Technology Conference, LTC 2007*, Lecture Notes in Computer Science, Poznań. Springer.

Nenadić, G. (2000). Local grammars and parsing coordination of nouns in Serbo-Croatian. W: Sojka, P., Matoušek, V., Pala, K., i Kopeček, I., red., *Proceedings of TSD 2000*, strony 57–62, Brno, Czechy. Springer.

Nenadić, G. i Vitas, D. (1998a). Formal model of noun phrases in Serbo-Croatian. *BULAG*, (23):297–311, Presses de l'Université de Franche-Comté.

Nenadić, G. i Vitas, D. (1998b). Using local grammars for agreement modeling in highly inflective languages. W: Sojka, P., Kopeček, I., i Pala, K., red., *Proceedings of TSD 1998*, strony 91–96, Brno, Czechy. Springer.

Nenadić, G., Vitas, D., i Krstev, C. (1999). Local grammars and compound verb lemmatization in Serbo-Croatian. *Current Issues in Formal Slavic Linguistics*, strony 469–477, Peter Lang.

Nepil, M. (2003). *Relational Rule Induction for Natural Language Disambiguation*. rozprawa doktorska, Wydział Informatyki, Uniwersytet Masaryka, Brno.

Nepil, M., Popelínský, L., i Žáčková, E. (2001). Part-of-speech tagging by means of shallow parsing, ILP and active learning. W: *Proceedings of 3rd Workshop on Learning Language in Logic (LLL)*, Strasburg, Francja.

Nivre, J. (2003). An efficient algorithm for projective dependency parsing. W: *Proceedings of the 8th International Workshop on Parsing Technologies (IWPT)*, strony 149–160.

Obrębski, T. i Stolarski, M. (2006). *UAM Text Tools v0.90*.

Ogunnaike, B. A. (2009). *Random Phenomena: Fundamentals of Probability and Statistics for Engineers*. CRC Press.

Osborne, M. (2000). Shallow parsing as part-of-speech tagging. W: Cardie, C., Daelemans, W., Nedellec, C., i Tjong Kim Sang, E., red., *Proceedings of CoNLL-2000 and LLL-2000*, strony 145–147. Lisbon, Portugal.

Osenova, P. (2002). Bulgarian nominal chunks and mapping strategies for deeper syntactic analyses. W: *Proceedings of the Workshop on Treebanks and Linguistic Theories, September 20-21 (TLT02)*, Sozopol, Bułgaria.

Osenova, P. i Simov, K. (2003). Between chunk ideology and full parsing needs. W: *Proceedings of the Shallow Processing of Large Corpora (SProLaC 2003) Workshop*, strony 78–87, Lancaster, Anglia.

Pedersen, T. (2008). Empiricism is not a matter of faith. *Computational Linguistics*, 34(3):465–470, Association for Computational Linguistics.

Piasecki, M. (2006). Hand-written and automatic rules for Polish tagger. W: *Text, Speech and Dialogue, 9th International Conference, Brno, Czechy*, volume 4188, strony 205–212. Springer.

Piasecki, M. (2007). Polish tagger TaKIPI: Rule based construction and optimisation. *Task Quarterly*, 11(1–2):151–167.

Piasecki, M. (2008). Cele i zadania lingwistyki informatycznej. W: *Metodologie językoznawstwa. Współółczesne tendencje i kontrowersje*. Lexis.

Piasecki, M. i Godlewski, G. (2006a). Effective architecture of the Polish tagger. W: *Text, Speech and Dialogue*, volume 4188, strony 213–220, Brno, Czechy. Springer.

Piasecki, M. i Godlewski, G. (2006b). Reductionistic, tree and rule based tagger for polish. W: *Intelligent Information Processing and Web Mining — Proceedings of the International IIS: IIPWM'06 Conference*, strony 531–540, Wisła. Springer.

Piasecki, M. i Radziszewski, A. (2009). Morphosyntactic constraints in acquisition of linguistic knowledge for Polish. W: *Aspects of Natural Language Processing (a festschrift for Professor Leonard Bolc)*, volume 5070, strony 163–190. Springer. Bolc Festschrift.

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

Pla, F., Molina, A., i Prieto, N. (2000). Improving chunking by means of lexical-contextual information in statistical language models. W: Cardie, C., Daelemans, W., Nedellec, C., i Tjong Kim Sang, E., red., *Proceedings of CoNLL-2000 and LLL-2000*, strony 148–150. Lisbon, Portugal.

Polański, K., red. (1999a). *Encyklopedia językoznawstwa ogólnego*. Wydawnictwo Zakładu Narodowego im. Ossolińskich, wydanie drugie.

Polański, K. (1999b). Nawiasowe znakowanie. W: Polański (1999a), strony 387–388.

Polański, K. (1999c). Wypowiedzenie. W: Polański (1999a), strona 645.

Przepiórkowski, A. (2004). *Korpus IPI PAN. Wersja wstępna*. Instytut Podstaw Informatyki Polskiej Akademii Nauk, Warszawa.

Przepiórkowski, A. (2005). The IPI PAN Corpus in numbers. W: Vetulani, Z., red., *Proceedings of the 2nd Language & Technology Conference*, Poznań.

Przepiórkowski, A. (2007). Slavic information extraction and partial parsing. W: *Proceedings of the Workshop on Balto-Slavonic Natural Language Processing*, strony 1–10, Praga, Czechy. Association for Computational Linguistics.

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

Przepiórkowski, A. i Woliński, M. (2003). A flexemic tagset for Polish. W: *Proceedings of* Morphological Processing of Slavic Languages*, EACL 2003*.

Przepiórkowski, A. (2009a). A comparison of two morphosyntactic tagsets of Polish. W: Koseska-Toszewa, V., Dimitrova, L., i Roszko, R., red., *Representing Semantics in Digital Lexicography: Proceedings of MONDILEX Fourth Open Workshop*, strony 138–144, Warszawa.

Przepiórkowski, A. (2009b). Towards the automatic acquisition of a valence dictionary for Polish. W: Marciniak, M. i Mykowiecka, A., red., *Aspects of Natural Language Processing*, volume 5070 of *LNCS*, strony 191–210. Springer Verlag, Berlin.

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

Przepiórkowski, A. i Szałkiewicz, Ł. (2012). Anotacja morfoskładniowa. W: Przepiórkowski i inni (2012).

Przepiórkowski, A. i Woliński, M. (2003). The unbearable lightness of tagging: A case study in morphosyntactic tagging of Polish. W: *Proceedings of the* 4th International Workshop on Linguistically Interpreted Corpora (LINC-03)*, EACL 2003*.

Quinlan, J. R. (1993). *C4.5: programs for machine learning*. Morgan Kaufmann Publishers Inc., San Francisco, USA.

Radziszewski, A. i Acedański, S. (2012). Taggers gonna tag: An argument against evaluating disambiguation capacities of morphosyntactic taggers. W: *Proceedings of the 15th International Conference on Text, Speech and Dialogue*, volume 7499 of *LNCS*, Brno, Czechy. Springer Verlag.

Radziszewski, A., Kilgarriff, A., i Lew, R. (2011a). Polish word sketches. W: *Proceedings of the 5th Language & Technology Conference, Poznań*.

Radziszewski, A., Marcińczuk, M., i Wardyński, A. (2011b). *Specyfikacja języka WCCL*. Wydział Informatyki i Zarządzania, Politechnika Wrocławska. Instrukcja dostępna on-line pod adresem http://nlp.pwr.wroc.pl/redmine/projects/joskipi/wiki/Specyfikacja.

Radziszewski, A. i Maziarz, M. (2011). Developing free morphological data for Polish. *Cognitive Studies — Etudes Cognitives*, 11:201–212, SOW, Warszawa.

Radziszewski, A., Maziarz, M., i Wieczorek, J. (2012). Shallow syntactic annotation in the Corpus of Wrocław University of Technology. *Cognitive Studies*, 12, SOW, Warszawa.

Radziszewski, A. i Pawlaczek, A. (2012). Large-scale experiments with NP chunking of Polish. W: *TSD 2012: Proceedings of the 15th International Conference on Text, Speech and Dialogue*, strony 143–149, Brno, Czechy. Springer-Verlag.

Radziszewski, A. i Piasecki, M. (2010). A preliminary noun phrase chunker for Polish. W: *Intelligent Information Systems*, strony 169–180. Springer.

Radziszewski, A. i Śniatowski, T. (2011a). Maca — a configurable tool to integrate Polish morphological data. W: *Proceedings of FreeRBMT11*.

Radziszewski, A. i Śniatowski, T. (2011b). A memory-based tagger for Polish. W: *Proceedings of the 5th Language & Technology Conference, Poznań*.

Radziszewski, A., Wardyński, A., i Śniatowski, T. (2011c). WCCL: A morpho-syntactic

feature toolkit. W: *Proceedings of the Balto-Slavonic Natural Language Processing Workshop*. Springer.

Ramshaw, L. A. i Marcus, M. P. (1995). Text chunking using transformation-based learning. W: *Proceedings of the Third ACL Workshop on Very Large Corpora*, strony 82–94. Cambridge, MA, USA.

Ratnaparkhi, A. (1996). A maximum entropy model for part-of-speech tagging. W: *Proceedings of the conference on empirical methods in natural language processing*, volume 1, strony 133–142.

Rudolf, M. (2004). *Metody automatycznej analizy korpusu tekstów polskich*. Uniwersytet Warszawski, Wydz. Polonistyki.

Sang, E. F. T. K. i Veenstra, J. (1999). Representing text chunks. W: *Proceedings of the ninth conference on European chapter of the Association for Computational Linguistics*, strony 173–179, Morristown, NJ, USA. Association for Computational Linguistics.

Schmid, H. (1995). Improvements in part-of-speech tagging with an application to german. W: *Proceedings of the ACL SIGDAT Workshop*, strony 47–50.

Schmid, H. i Laws, F. (2008). Estimation of conditional probabilities with decision trees and an application to fine-grained POS tagging. W: *Proceedings of COLING 2008*, volume 1, strony 777–784. Association for Computational Linguistics.

Sha, F. i Pereira, F. C. N. (2003). Shallow parsing with conditional random fields. W: *Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, Edmonton, Kanada.

Sharoff, S. (2004). What is at stake: a case study of Russian expressions starting with a preposition. W: *Proceedings of the Workshop on Multiword Expressions: Integrating Processing*, strony 17–23. Association for Computational Linguistics.

Shen, H. (2004). Voting between multiple data representations for text chunking. praca magisterska, Simon Fraser University, Kanada.

Shen, H. i Sarkar, A. (2005). Voting between multiple data representations for text chunking. W: *Canadian Conference on AI*, strony 389–400.

Silberztein, M. (2003). *NooJ manual*. Instrukcja użytkownika dostępna on-line pod adresem http://www.nooj4nlp.net.

Simov, K. i Osenova, P. (2001). A hybrid system for morphosyntactic disambiguation in Bulgarian. W: *Proceedings of the RANLP 2001 Conference*, strony 5–7, Cigow, Bułgaria.

Sjöbergh, J. (2003). Combining POS-taggers for improved accuracy on Swedish text. W: *Proceedings of NoDaLiDa 2003*.

Śniatowski, T. (2011). Metody analizy tekstu w języku polskim na poziomie wyrazowym. praca magisterska, Politechnika Wrocławska, Wydział Informatyki i Zarządzania.

Śniatowski, T. i Piasecki, M. (2011). Combining Polish Morphosyntactic Taggers. W: *Proceedings of the 2011 International Joint Conference on Security and Intelligent Information Systems*. Springer Berlin / Heidelberg.

Su, J. i Zhang, H. (2006). A fast decision tree learning algorithm. W: *Proceedings of the 21st national conference on Artificial intelligence — Volume 1*, AAAI'06, strony 500–505. AAAI Press.

Sun, X., Morency, L.-P., Okanohara, D., i Tsujii, J. (2008). Modeling latent-dynamic

in shallow parsing: a latent conditional model with improved inference. W: *Proceedings of the 22nd International Conference on Computational Linguistics - Volume 1*, strony 841–848.

Sussex, R. i Cubberley, P. (2006). *The Slavic Languages*. Cambridge University Press.

Sutton, C. i McCallum, A. (2011). An introduction to conditional random fields. W: *Foundations and Trends in Machine Learning*.

Świdziński, M. (1992). *Gramatyka formalna języka polskiego*. Wydawnictwo Uniwersytetu Warszawskiego, Warszawa.

Tanev, H. i Mitkov, R. (2002). Shallow language processing architecture for Bulgarian. W: *COLING 2002: The 17th International Conference on Computational Linguistics*, Tajpej, Tajwan.

Tjong Kim Sang, E. F. (2000). Text chunking by system combination. W: Cardie, C., Daelemans, W., Nedellec, C., i Tjong Kim Sang, E., red., *Proceedings of CoNLL-2000 and LLL-2000*, strony 151–153. Lisbon, Portugal.

Tjong Kim Sang, E. F. i Buchholz, S. (2000). Introduction to the CoNLL-2000 shared task: Chunking. W: *Proceedings of CoNLL-2000 and LLL-2000*, strony 127–132. Lisbon, Portugal.

Tufiş, D. (1999). Tiered tagging and combined language models classifiers. W: Matousek, V., Mautner, P., Ocelíková, J., i Sojka, P., red., *Text, Speech and Dialogue*, volume 1692 of *Lecture Notes in Computer Science*, strony 843–843. Springer Berlin / Heidelberg.

Tufiş, D. (2011). Natural language question answering in open domains. W: *Computer Science Journal of Moldova*, strony 70–85, Chisinau, Mołdawia. Institute of Mathematics and Computer Science.

Tzoukermann, E., Radev, D. R., i Gale, W. A. (1997). Tagging French without lexical probabilities — combining linguistic knowledge and statistical learning. W: *Natural Language Processing Using Very Large Corpora*. Kluwer.

van Halteren, H., red. (1999). *Syntactic Wordclass Tagging*. Kluwer.

van Halteren, H. (2000). Chunking with WPDV models. W: Cardie, C., Daelemans, W., Nedellec, C., i Tjong Kim Sang, E., red., *Proceedings of CoNLL-2000 and LLL-2000*, strony 154–156. Lisbon, Portugal.

Van Halteren, H., Daelemans, W., i Zavrel, J. (2001). *Improving accuracy in word class tagging through the combination of machine learning systems*, volume 27, strony 199–229. MIT Press.

Veenstra, J. (1998). Fast NP chunking using memory-based learning techniques. W: *Proceedings of the Eighth Belgian-Dutch Conference on Machine Learning BENELEARN'98*, strony 71–78, Wageningen, Holandia. ATO-DLO.

Veenstra, J. i van den Bosch, A. (2000). Single-classifier memory-based phrase chunking. W: Cardie, C., Daelemans, W., Nedellec, C., i Tjong Kim Sang, E., red., *Proceedings of CoNLL-2000 and LLL-2000*, strony 157–159. Lisbon, Portugal.

Vidová-Hladká, B. (2000). *Czech Language Tagging*. rozprawa doktorska, Uniwersytet Karola, Wydział Matematyki i Fizyki, Praga.

Viterbi, A. J. (1967). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, IT-13(2):260–269.

von Halteren, H. (2000). A default first order family weight determination procedure for

WPDV models. W: *Fourth Conference on Computational Natural Language Learning and the Second Learning Language in Logic Workshop*. Association for Computational Linguistics.

Šmerk, P. (2004). Unsupervised learning of rules for morphological disambiguation. W: Sojka, P., Kopecek, I., i Pala, K., red., *Text, Speech and Dialogue, 7th International Conference*, volume 3206 of *Lecture Notes in Computer Science*, strony 211–216, Brno, Czechy. Springer.

Vučković, K. (2009). *Model parsera za hrvatski jezik*. rozprawa doktorska, Wydział Filozoficzny, Uniwersytet w Zagrzebie, Zagrzeb, Chorwacja.

Vučković, K., Tadić, M., i Dovedan, Z. (2008). Rule-based chunker for Croatian. W: (ELRA), E. L. R. A., red., *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, Marakesz, Maroko.

Vučković, K., Željko Agić, i Tadić, M. (2010). Improving chunking accuracy on Croatian texts by morphosyntactic tagging. W: Chair), N. C. C., Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., Rosner, M., i Tapias, D., red., *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

Wallach, H. M. (2004). Conditional random fields: An introduction. Raport nr MS-CIS-04-21, Department of Computer and Information Science, University of Pennsylvania, USA.

Waszczuk, J., Głowińska, K., Savary, A., i Przepiórkowski, A. (2010). Tools and methodologies for annotating syntax and named entities in the National Corpus of Polish. W: *Proceedings of the International Multiconference on Computer Science and Information Technology (IMCSIT 2010): Computational Linguistics – Applications (CLA'10)*, strony 531–539, Wisła, Poland. PTI.

Woliński, M. (2004). *Komputerowa weryfikacja gramatyki Świdzińskiego*. rozprawa doktorska, Instytut Podstaw Informatyki Polskiej Akademii Nauk, Warszawa.

Woliński, M. (2006). Morfeusz — a practical tool for the morphological analysis of Polish. W: *Advances in Soft Computing 5. Proceedings of the IIS:IIPWM-2006*, strony 511–520. Springer-Verlag.

Woliński, M., Głowińska, K., i Świdziński, M. (2011). A preliminary version of Składnica — a treebank of Polish. W: *Proceedings of the LTC 2011*.

Wróblewska, A. i Woliński, M. (2011). Preliminary experiments in polish dependency parsing. W: *Proceedings of SIIS 2011*, Warszawa. Springer-Verlag.

Zabłocki, A. (2010). Optymalizacja programu Spejd z wykorzystaniem technik skończenie stanowych. praca magisterska, Wydział Matematyki, Informatyki i Mechaniki Uniwersytetu Warszawskiego.

Zavrel, J. i Daelemans, W. (1999). Recent advances in memory-based part-of-speech tagging. W: *Actas del VI Simposio Internacional de Comunicacion Social*, strony 590–597, Santiago de Cuba.

Zhou, G., Su, J., i Tey, T. (2000). Hybrid text chunking. W: Cardie, C., Daelemans, W., Nedellec, C., i Tjong Kim Sang, E., red., *Proceedings of CoNLL-2000 and LLL-2000*, strony 163–166. Lisbon, Portugal.