STUDIA INFORMATICA 2011
Volume 32 Number 4A (100)

Nina SUSZCZAŃSKA, Przemysław SZMAL
Politechnika Śląska, Instytut Informatyki

INŻYNIERIA JĘZYKA DLA SYSTEMU THETOS

Streszczenie. W artykule są opisywane prace badawcze i implementacyjne rozwijane głównie w ramach projektu *Thetos* tłumaczenia tekstów na język migowy. Celem prac jest zbudowanie informatycznego modelu wybranych aspektów języka polskiego. W tych ramach opracowano formalizmy składniowej i semantycznej reprezentacji struktury zdania, gramatyki odpowiadające tym formalizmom, procedury analizy składniowej i semantycznej, a także metodę modelowania tekstu. Szczególny nacisk położono na praktyczne zastosowanie opracowywanych metod przetwarzania języka polskiego w analizatorach *Polsyn* realizującym model składni oraz *Polsem* realizującym model semantyki. Na osobną uwagę zasługuje moduł *Polin* realizujący generowanie wypowiedzi w języku *Thel*.

Słowa kluczowe: inżynieria języka, SG-model języka polskiego, system Thetos, głęboki parser Polsyn, analizator semantyczny Polsem, linearyzator Polin

COMPUTER NATURAL LANGUAGE ENGINEERING FOR THETOS SYSTEM

Summary. The paper is devoted to a description of the research and implementation works developed in the framework of the *Thetos* project concerned with translation of texts into the sign language. The works are aimed at construction of a computer model of selected aspects of Polish. The elements elaborated in this framework are formalisms for syntactic and semantic representation of the sentence, grammars that correspond to those formalisms, syntactic and semantic analysis procedures. Special stress has been put on practical application of the considered methods for Polish language processing in the *Polsyn* analyzer implementing the syntax model, and in *Polsem* implementing the semantic one. A special attention deserves the Polin module responsible for generation of *Thel* language utterances.

Keywords: natural language engineering, SG-model for Polish, *Thetos* system, *Polsyn* deep parser, *Polsem* semantic analyzer, *Polin* linearizer

1. Wstęp

Internet oraz współczesne możliwości przechowywania i rozpowszechniania tekstów w postaci elektronicznej spowodowały falę wzrostu zainteresowania komputerowym przetwarzaniem tekstów w języku naturalnym[1]. Stymuluje to rozwój stosunkowo nowej dziedziny informatyki – inżynierii języka naturalnego. Inżynieria języka skupia się głównie na poznawaniu technik przetwarzania, możliwościach modelowania różnych przejawów językowych i metodach modelowania, algorytmach przetwarzania, problemach ich implementacji i weryfikacji.

W sferze przetwarzania języka inżynieria języka styka się z inżynierią oprogramowania; te dwie dziedziny wzajemnie oddziałują na siebie. Z jednej strony, metody informatyczne są szeroko wykorzystywane w inżynierii języka – przede wszystkim przy modelowaniu różnych zjawisk językowych. Z drugiej strony, osiągnięcia inżynierii języka są na tyle znaczące, że coraz częściej w systemach informatycznych znajdują swoje zastosowanie moduły przetwarzania języka. Jednym z takich systemów jest *Thetos*, system tłumaczenia tekstów na język migowy [2].

W artykule zostały krótko opisane prace teoretyczne i implementacyjne z zakresu inżynierii języka polskiego prowadzone od lat w Instytucie Informatyki Politechniki Śląskiej, głównie w ramach projektu *Thetos*. Ich celem jest zbudowanie informatycznego modelu wybranych aspektów języka polskiego – jest to tzw. SG-model. Na SG-model składa się zbiór modeli cząstkowych: morfologii, składni, semantyki i tekstu. W tych ramach rozpatruje się formalizmy składniowej i semantycznej reprezentacji struktury zdania, gramatyki odpowiadające tym formalizmom, procedury analizy składniowej i semantycznej, a także metodę modelowania tekstu. Szczególny nacisk został położony na praktyczne zastosowanie opracowywanych metod przetwarzania języka polskiego, w szczególności w głębokim parserze *Polsyn*, realizującym model składni, oraz w analizatorze *Polsem* realizującym model semantyki. Na osobną uwagę zasługuje moduł *Polin* realizujący generowanie wypowiedzi w języku *Thel*.

Możliwości praktycznego zastosowania SG-modelu odkrywają projekty, w których stosuje się uzyskane wyniki symulacyjne i rezultaty eksperymentów. Oprócz wspomnianego wyżej systemu *Thetos*, warto w pierwszej kolejności wymienić dwa projekty: Pierwszy to projekt *infomat-e*, kierowany przez Instytut Technik Innowacyjnych EMAG. W ramach tego projektu powstał kiosk informacyjny dostosowany do potrzeb osób niepełnosprawnych (http://www.infomat-e.pl). W wyniku drugiego – projektu *SIT* kierowanego przez Wydział

Nauk Geograficznych i Geologicznych Uniwersytetu im. Adama Mickiewicza w Poznaniu – wykonano prosty system informacyjny ułatwiający osobom niesłyszącym zwiedzanie ekspozycji muzealnych i poruszanie się po trasach turystycznych [3]. Są też różne projekty laboratoryjne, mniejsze i większe, na przykład: projekt *Liana* wspomagania lingwistycznego procesu projektowania systemów informatycznych [4], czatterbot *ANKA* [5], program *Polsumm* streszczania polskich tekstów [6] oraz kilka małych projektów związanych z przetwarzaniem języka polskiego.

2. Słowo o inżynierii języka naturalnego

Język naturalny od zawsze był trudnym obiektem badań, gdyż jego formalizacja i automatyczne przetwarzanie wymaga nie tylko dogodnych formalizmów i wydajnych algorytmów, ale także dużej mocy obliczeniowej. Rozwój techniki komputerowej przyczynił się między innymi do oddzielenia w latach siedemdziesiątych zeszłego stulecia lingwistyki komputerowej (ang. *Computational Linguistics; CL*) od lingwistyki matematycznej, która tradycyjnie zajmowała się formalnymi metodami badania języków. Lingwistyka komputerowa szybko stała się obszerną dziedziną wiedzy, która należy do nauk interdyscyplinarnych na pograniczu lingwistyki i informatyki.

Inżynieria języka naturalnego (ang. *Natural Language Engineering; NLE*) jako dziedzina badawcza powstała niedawno, wyodrębniła się jako osobna gałąź inżynierii lingwistycznej (ang. *Linguistic Engineering; LE*), ta z kolei powstała w końcu zeszłego stulecia jako część lingwistyki komputerowej. Do tej pory terminy *inżynieria lingwistyczna* i *inżynieria języka* są używane czasem jako równoważne. Obie dziedziny można uznać za dyscypliny informatyczne, jednak pierwsza zajmuje się opracowaniem modeli i algorytmów przetwarzania tekstów w języku naturalnym, a druga – ich implementacją. Wymienione wyżej dyscypliny pochodzą z bardzo obszernej dziedziny, noszącej nazwę przetwarzania języka naturalnego (ang. *Natural Language Processing; NLP*).

Trudno o ścisłą definicję NLP (jak zresztą każdej dziedziny naukowej); wszystkie definicje sprowadzają się do dość rozległego opisu spraw, którymi zajmuje się ta dziedzina, lub wskazaniem jej miejsca wśród innych kierunków badań. Jedni rozpatrują NLP jako dyscyplinę informatyczną, inni definiują NLP jako naukę na pograniczu lingwistyki i informatyki; A. Przepiórkowski [7] podaje definicję, wyliczając zakres prac powiązanych z „*automatycznym tworzeniem lub przetwarzaniem wypowiedzeń*”, podkreślając, że prace te są

[1] Przyjmujemy definicję języka naturalnego zaproponowaną przez Z. Vetulaniego: jest to język, w którym użytkownik wypowiada się w sposób nieskrępowany pod względem formalnym [1].

„*związane ze znaczeniem lub strukturą lingwistyczną tych wypowiedzeń*". Niniejszy artykuł jest poświęcony pracom implementacyjnym, dlatego przyjmujemy w nim najbardziej ogólną definicję NLP: NLP określamy jako naukę, której celem jest poznanie formalnych zasad budowy języka i jego rozumienia. Wszystkie prace w NLP są ukierunkowane na ich wykorzystanie w systemach informatycznych przetwarzania tekstów. Należy podkreślić, że problemy przetwarzania rozpatrujemy w odniesieniu do tekstów w postaci elektronicznej, bez względu na naturę tych tekstów: wprowadzonych ręcznie, uzyskanych w procesie rozpoznawania mowy lub wypowiedzeń w języku migowym, wygenerowanych automatycznie itp.

Inżynieria lingwistyczna jest zorientowana na zastosowania praktyczne. Głównym jej zadaniem jest opracowanie komputerowych modeli kompetencji językowej człowieka [1]. Zadaniem inżynierii języka natomiast jest opracowanie algorytmów i programów przetwarzania wypowiedzeń w języku naturalnym [8]. Przy tym zakłada się, że modelowanie opiera się na wynikach badań z dziedziny lingwistyki komputerowej (ponad sześćdziesiąt lat badań w tej dziedzinie wykazało bowiem nieskuteczność czysto inżynierskiego podejścia do przetwarzania języków naturalnych).

Rola inżynierii języka jako samodzielnej dziedziny badań naukowych jest bardzo ważna. W lingwistyce komputerowej, lub – trafniej – w badaniach języka, teoria i praktyka są równie istotne: hipotezy powinny być sprawdzone w eksperymentach komputerowych. Właśnie inżynieria językowa zajmuje się weryfikacją teoretycznych opisów poszczególnych zjawisk językowych. Sporządzone przez lingwistów-teoretyków opisy teoretyczne inżynier powinien przekształcić do postaci algorytmów modelujących te zjawiska, skonstruowawszy uprzednio odpowiednie modele. Co również ważne, modele powinny być adekwatnie zaimplementowane, tylko w tym przypadku wyniki eksperymentów można uważać za wiarygodne. Tylko w ten sposób teoria może być potwierdzona lub obalona. Z drugiej strony, inżynier języka występuje czasami w roli twórczej – badacza języka [1]. Przy tworzeniu konkretnego systemu informatycznego wymagającego przetwarzania języka, informatyk (czytaj: inżynier języka) zwykle zaczyna opracowanie algorytmów od gromadzenia i analizy danych językowych w celu wyboru właściwej teorii opracowanej przez lingwistów. Często jednak brakuje tej właściwej, wtedy niezbędne jest opracowanie teorii własnej, która pozwoliłaby na w miarę łatwą implementację modeli wybranego zjawiska językowego. Oczywiście opracowanie opisu teoretycznego modelowanego fragmentu języka, tworzenie modelu lingwistycznego wymaga odpowiedniej znajomości teorii przetwarzania języka. Wieloletnie doświadczenie pokazało, że nieznajomość szczegółów budowy języka i rozwiązanie problemów *ad hoc* nie może doprowadzić do sukcesu.

3. Organizacja automatycznego przetwarzania tekstów

Analizę tekstu zwykle prowadzi się w kilku etapach realizujących mniej lub bardziej szczegółowo zaproponowane przez Chomsky'ego [9] „linguistic levels". Liczba etapów, a także wybór formalizmów w nich stosowanych zależy od potrzeb systemu informatycznego, lecz u podstaw zazwyczaj leży schemat klasyczny: przetwarzanie morfologiczne, składniowe i semantyczne. Przetwarzanie morfologiczne – naturalnie – zajmuje najniższy poziom w tej hierarchii. Schemat ten można rozszerzyć o poziom interpretacji pragmatycznej, która często jest nazywana także analizą pragmatyczną, a polega na powiązaniu elementów reprezentacji semantycznej z rzeczywistością realną lub wirtualną. Ponadto, jeżeli zachodzi taka potrzeba, ciąg przetwarzań można poszerzyć o etapy przetwarzania tekstu jako całości: analizy składni i semantyki całości tekstu.

Za przetwarzanie na każdym z etapów jest odpowiedzialny osobny analizator, który z reguły przetwarza wynik etapu poprzedniego. Podejście to jest trafne nie tylko z punktu widzenia lingwistyki, ale również z punktu widzenia inżynierii oprogramowania, pozwala bowiem na implementację wieloskładnikowego systemu przetwarzania z podzieloną odpowiedzialnością (kompetencją). To ostatnie jest zgodne z teorią projektowania złożonych systemów oprogramowania, zwłaszcza przy podejściu obiektowym, a współczesne systemy przetwarzania tekstu są właśnie takimi systemami. Wielopoziomowość i wielomodułowość czyni system elastycznym i czytelnym, co zwiększa niezawodność oprogramowania.

Uważamy za słuszne podejście, przy którym każdy z analizatorów wykonuje przetwarzanie w miarę możliwości „samodzielnie", nie korzystając z wiedzy dziedziny stojącej wyżej w hierarchii etapów. Dopuszcza się jednak niektóre uzasadnione pragmatycznie odstępstwa. Na przykład, na poziomie składni w celu zmniejszenia niejednoznaczności leksykalnej może być stosowana semantyka leksykalna.

Problemowi rozwiązywania niejednoznaczności warto w tym miejscu poświęcić kilka słów. Z każdym etapem przetwarzania liczba konstrukcji niejednoznacznych rośnie wykładniczo. Na przykład, wieloznaczność cech morfosyntaktycznych w wynikach analizy morfologicznej powoduje powstanie wieloznacznych konstrukcji składniowych. Wyniki analizy składniowej stają się także niejednoznaczne. Konstrukcje w węźle homonimicznym mogą mieć prawidłową strukturę składniową, lecz nie wszystkie z nich są prawidłowe w kontekście struktury zdania lub są błędne znaczeniowo. Wyszukiwanie wśród nich właściwych wariantów często jest niemożliwe na poziomie składniowym. Rozstrzygnięcie niejednoznaczności w takim przypadku przekazuje się na poziom semantyczny.

Teoria twierdzi, że interpretacja semantyczna składników niejednoznacznych jest czasochłonna. W praktyce analizator często nie jest w stanie opracować tak dużej liczby niejedno-

znacznych danych, co powoduje wzrost czasu przetwarzania jednego zdania do kilku godzin, a nawet może doprowadzić do zawieszenia systemu. Dlatego bardzo ważny jest proces ich ujednoznacznienia za pomocą tzw. filtrów wyników wieloznacznych, których zadaniem jest redukcja wyników niewłaściwych w danym kontekście lub niewykorzystywanych w danej dziedzinie tekstu. Oczywisty jest fakt, że w taki sposób otwiera się możliwość podejścia subiektywnego dla określenia kategorii prawidłowości, jednak doświadczenie ostatnich 2-3 dekad wskazuje na to, że bez sensownych ograniczeń na wyniki produkowane formalnymi metodami niemożliwe jest otrzymanie jakkolwiek satysfakcjonującego rezultatu końcowego. Ponadto ograniczenia mogą być całkiem uzasadnione: przykładem tego może być zakaz produkowania na poziomie morfologii starosłowiańskich form wyrazów w przypadku analizy tekstów współczesnych itp.

Nasuwa się wniosek, że każdy etap przetwarzania powinien być wykonywany minimum w dwu podetapach. Na przykład, analiza morfologiczna lub składniowa może być dokonana najpierw przez analizę właściwą i dalej – przez filtrację wyników. Z reguły właśnie na podetapie filtracji może być zastosowana wiedza lingwistyczna sąsiadujących dziedzin, na przykład można na poziomie przetwarzania składniowego stosować elementy semantyki, a nawet wiedzę pozalingwistyczną (pragmatyczną). Stosowanie wiedzy pozalingwistycznej jest zależne od zakresu tematycznego tekstów, wymagań do systemu przetwarzania i innych warunków, które można nazwać zewnętrznymi. Dlatego zgodnie z regułami projektowania złożonych systemów informatycznych proces przetwarzania ma być podzielony na podprocesy: część stosunkowo niezmienną, która dotyczy rozwiązania problemów czysto lingwistycznych, oraz część zmienną, która w razie zmiany warunków zewnętrznych może być zamieniona na inną. Należy także uwzględniać ten fakt, że przy zamierzonych lub niezamierzonych zmianach w modelu przetwarzania, część lingwistyczna też musi ulec zmianom, wobec tego moduł „niezmienny” także ma być budowany jako wielomodułowy system oprogramowania zgodnie z zasadami projektowania systemów informatycznych [10,11].

4. Modelowanie języka naturalnego

Opracowanie gramatyk formalnych modelujących język jest jednym z najstarszych zagadnień informatyki, powstałym bodajże jednocześnie z powstaniem teorii oprogramowania, przy czym modele te są tworzone nie tylko dla języków formalnych, ale także i dla języków naturalnych. Zadaniem modelu jest opis języka w sposób formalny a przy tym adekwatny, nadający się do wykorzystania przy przetwarzaniu komputerowym tego języka.

Potocznie mówiąc, model ma opisać język tak, żeby program przetwarzania (np. program w C++, Javie lub C#) „rozumiał" i mógł reagować na treść tekstów w tym języku w zakresie określonym przez pragmatykę systemu przetwarzania. Języki naturalne w przeciwieństwie do języków formalnych są złożone i obszerne, do tego ciągle się rozwijają. Dlatego mimo wysiłków informatyków i lingwistów nie wydaje się możliwe zbudowanie ścisłego modelu obejmującego wszystkie aspekty języka dla jakiegokolwiek z języków naturalnych. Istnieje jednak zapotrzebowanie nawet na takie niepełne modele, które odzwierciedlają jedynie wybrane przejawy językowe, formalizując różne płaszczyzny wiedzy o przetwarzanym języku. Poziom opisu języka wyznacza się przez wymagania względem systemu przetwarzania; oczywiście, model jest tym lepszy, im więcej aspektów języka obejmuje. Opis języka w modelu może dotyczyć każdego z poziomów przetwarzania opisanych wyżej: morfologii, składni, semantyki lub pragmatyki.

W zależności od podejścia do modelowania, istniejące obecnie różnorodne modele języków naturalnych można podzielić na trzy klasy: modele klasyczne, tradycyjne dla inżynierii języka oparte na wiedzy o języku zgromadzonej w lingwistyce komputerowej; modele oparte na badaniach statystycznych korpusów tekstu, oraz mieszane, w których przeważa jakaś część: tradycyjna lub statystyczna. *Model tradycyjny* służy jako baza do opracowania odpowiedniego formalizmu do opisu języka, a ten z kolei jest podstawą do napisania reguł szczegółowych gramatyk formalnych. W tym przypadku opis języka na każdym poziomie buduje się zgodnie z pewnymi zasadami teoretycznymi, struktury językowe są przekształcane na struktury formalne, których forma jest zgodna z wybranym formalizmem. *Model statystyczny* opisuje język na podstawie analizy statystycznej dużych zbiorów tekstów reprezentatywnych dla wybranego zakresu pragmatycznego. Analiza ta daje podstawy do pewnego wnioskowania co do własności języka na tym lub innym poziomie. Zwykle analizę taką prowadzi się za pomocą systemu uczącego się – model oparty na uczeniu się także jest nazywany *uczącym się*. Nazwa *modelu mieszanego* mówi sama za siebie – są w nim wykorzystywane elementy obu modeli wymienionych wyżej. Na przykład, w modelu formalnym dla opracowania reguł szczegółowych gramatyki korzysta się z analizy statystycznej odpowiedniego korpusu tekstów.

Każdy typ modelu ma swoje wady i swoje zalety. Model formalny korzysta ze ścisłych metod opisu języka, ale żadna z metod ścisłych nie jest w stanie opisać języka w całości. Język naturalny ciągle rozwija się, więc przy zajściu zmian w języku, które powinny być uwzględnione w systemie przetwarzania, powinna być zmieniona gramatyka, co powoduje sekwencję czynności projektowych: zmiany w implementacji odpowiednich programów, ponowne testowanie systemu przetwarzania w całości itd. Ponadto, translatory struktur nieformalnych na struktury formalne są wyjątkowo czaso- i pracochłonne, w szczególności

dla klasy języków, do których należy język polski. Model statystyczny jest modelem bardziej elastycznym, zwykle jest modelem uczącym się, przy wystąpieniu wyżej wspomnianych zmian w języku generowanie nowych reguł przetwarzania wykonuje się przez ponowną analizę korpusu odpowiednich tekstów. Metody statystyczne mają także swoje wady: są to metody przybliżone, zależne od tekstów uczących, wyboru metod wnioskowania oraz doboru tekstów testowych. Niemniej modele te mają powodzenie w praktyce, w szczególności są wykorzystywane w systemach przetwarzania, które są opracowywane przy podejściu inżynierskim.

5. SG-model języka polskiego

Opracowany w Instytucie Informatyki Politechniki Śląskiej SG-model języka polskiego należy do klasycznych modeli formalnych. Modelowanie prowadzi się na czterech poziomach przetwarzania opisanych wyżej: morfologicznym, składni zdania, semantyki zdania, modelowania tekstu. Główna idea jest następująca: reprezentacją zdania na każdym poziomie modelowania jest graf, na którego topologię nakłada się pewne ograniczenia. Na każdym poziomie węzłami grafu są etykietowane grupy składniowe (SG), krawędziami są relacje między SG. Na węzłach grafu reprezentacji określono operacje. Wynik modelowania na każdym poziomie jest zapisywany za pomocą formalnego języka. Każdemu poziomowi przetwarzania odpowiada własna topologia grafu reprezentacji, odpowiednie zbiory etykiet dla węzłów, zbiory relacji między węzłami i języki reprezentacji. Poszczególne modele cząstkowe są omówione w dalszej części tego punktu.

5.1. Model morfologiczny

Model morfologiczny jako jedyny ma do czynienia z tekstem przedstawianym w postaci strumienia, reszta modeli przetwarza formalne struktury przygotowane przez poprzedni model. Za model morfologiczny został przyjęty model *a tergo* Tokarskiego [12], dostosowany do naszych celów z myślą o dalszym przetwarzaniu na poziomie składniowym. Model morfologiczny nie jest zbyt złożony: graf składa się z pojedynczych etykietowanych liści, zbiór relacji na poziomie morfologicznym jest pusty. Model ten został zaimplementowany w postaci analizatora *Polmorf*. Wynikiem modelowania morfologicznego jest reprezentacja morfologiczna tekstu wejściowego, przedstawiana jako zbiór otagowanych tokenów. Do zbioru tagów należy indeks tokenu w liniowym porządku tekstu, klasa tokenu, jego cechy morfosyntaktyczne, leksem i informacja uzupełniająca.

5.2. Model składniowy

Model składniowy bazuje na formalizmie SGS (Systemu Grup Składniowych) [13], zmodyfikowanym w celu zastosowania go do języków naturalnych. Ten oryginalny formalizm powstał jako uogólnienie dwóch znanych formalizmów: drzew zależności i składników bezpośrednich[2]. Główną zaletą SGS jest możliwość grupowania słów w tzw. grupy składniowe (SG), ustawiając relacje syntaktyczne między składowymi grup, a także między samymi SG.

W SG-modelu składni wykorzystano kilka innych formalizmów pomocniczych, na przykład na drugim i trzecim poziomie są stosowane schematy syntaktyczno-generatywne, które stanowią pewną odmianę schematów walencyjnych [14][3]. Na podstawie zmodyfikowanego SGS została opracowana gramatyka SGGP (*Syntactic Groups Grammar for Polish*) [15], zrealizowana następnie w parserze głębokim *Polsyn* [16]. Ogromny wpływ na kształt SG-modelu dla języka polskiego oraz na powstanie gramatyki SGGP w obecnej postaci miały prace polskich lingwistów, z których zaczerpnięta została wiedza o języku polskim, w szczególności o jego formalizacji. Do powstania i rozwoju opisywanych badań przyczyniły się w pierwszym rzędzie prace [12, 14, 17-22], później [1, 7, 23-28], a także wiele innych tutaj niewymienionych.

Składnia jest modelowana na pięciu poziomach; przetwarzanie każdego następnego poziomu wykorzystuje wyniki poziomu poprzedniego. Wynikiem modelowania jest reprezentacja syntaktyczna tekstu, na którą składa się zbiór reprezentacji zdań pojedynczych, przedstawianych jako otagowane drzewo z grupami składniowymi jako wierzchołkami, grupą czasownika (VG) jako korzeniem i relacjami składniowymi na krawędziach.

W swoich badaniach wychodzimy z założenia [29-31], że struktura składniowa zdania zależy od przekazywanej w nim treści i jest swoistym zapisem algorytmu rozpoznawania tej treści. To założenie rzutuje na rozumienie struktury SG i relacji zachodzących między nimi jako mechanizmów semantyki języka. Dlatego szczególną uwagę zwracamy na operacje przekształcenia SG i ich cech gramatycznych: operacje na grupie nie mogą spowodować jej wyjścia poza zakres SGS [32]. Analiza składniowa jest tylko jednym z pomocniczych etapów przetwarzania, jeśli jego celem jest rozumienie tekstu. Parser jest w tym przypadku narzędziem do formalnego przedstawienia zjawisk językowych, mając na celu dostarczanie danych dla następnego etapu formalizacji – analizy semantycznej, celem której jest odkrycie

[2] Szczegółowy opis tych formalizmów dany jest np. w [24].

[3] Uproszczony wariant schematów składniowych odzwierciedlających semantyczne otoczenie słów predykatywnych, zmodyfikowany w celu zastosowania nie tylko do słów, ale także do grup składniowych.

znaczenia struktur składniowych. Analiza semantyczna jest następnym etapem formalizacji, której wynikiem jest reprezentacja semantyczna tekstu.

5.3. Model semantyczny

Podobnie jak model syntaktyczny, model semantyczny także jest wielopoziomowy. W SG-modelu semantycznym użyto kilku formalizmów. Przede wszystkim jest to predykatowo-argumentowa struktura zdania, której generowanie opiera się na odmianie modelu walencyjnego czasowników – modelu otoczenia semantycznego VG. Ważną rolę w identyfikacji struktur semantycznych w analizowanym tekście odgrywa także semantyczny aspekt formalizmów składniowych. Do modelowania relacji predykatowo-argumentowych zaproponowano metodę semantycznej interpretacji grup składniowych i relacji składniowych wewnętrznych w tych grupach. Do odkrycia treści ukrytej w elementach struktur predykatowo-argumentowych zaproponowano modyfikację gramatyki Polańskiego, która polega na semantycznej interpretacji schematów otoczenia kontekstowego grup w zdaniu oraz na uzupełnieniu ich przez listę relacji sytuacyjnych. Osobny wątek modelowania semantycznego stanowi rozpoznawanie ładunków emocjonalnych w SG i w zdaniu.

Przy opracowaniu SG-modelu semantyki oparto się na fakcie ścisłego powiązania semantyki i składni języka. Zgodnie z tezą o tym, że SG jest składniową jednostką języka, która służy do przekazywania treści, na poziomie semantycznym badamy topologię grafu reprezentacji składniowej – mianowicie znaczenie grup składniowych i mechanizmy przekazywania przez nie treści, a także wpływ typów SG i relacji składniowych między SG na to przekazywanie. Zatem celem opisywanych badań jest „rozumienie" treści wypowiedzi. Jako „rozumienie", cytując Z. Vetulaniego [1], przyjmujemy formę tłumaczenia z języka naturalnego na pewien język sztuczny, sformalizowany, którego jednostki służą do przechowywania znaczeń zawartych w konstrukcjach składniowych. Przy tym formalne struktury semantycznej reprezentacji tekstu powinny mieć postać wygodną do dalszego przetwarzania komputerowego. Przez dalsze przetwarzanie rozumiemy tłumaczenie wypowiedzi z języka polskiego na język migowy [2, 33], udzielanie odpowiedzi w systemie ANKA [5], identyfikację kluczowych abstrakcji w dziedzinie problemowej [4], streszczanie tekstów [6], a także dalsze „rozumienie" treści na poziomie przetwarzania pragmatycznego. Określenie modelu jako informatycznego oznacza, że jest on opracowany w celu jego realizacji w postaci analizatora semantycznego. Takim analizatorem jest *Polsem* [34], opracowany na potrzeby systemu *Thetos*.

Szczególną rolę w SG-modelu odgrywają wyspecjalizowane bazy danych. Na każdym poziomie modelowania są wykorzystywane słowniki, mniejsze lub większe, których ogólna

liczba wynosi kilkadziesiąt. Jedne zawierają kilkadziesiąt tysięcy haseł słownikowych, niektóre kilka tysięcy, najmniejsze – jedynie kilka haseł.

5.4. Model generowania tekstu

Model generowania tekstu został opracowany na potrzeby systemu *Thetos*. Tłumaczenie tekstu w systemie Thetos wykonuje się zgodnie ze schematem:

Tekst wejściowy →
Analiza morfologiczna → *Analiza składniowa* → *Analiza semantyczna* →
Generowanie struktury semantycznej → *Generowanie struktury składniowej* →
Linearyzacja wypowiedzi → *Tekst w Thel*

Do pewnego stopnia podobną organizacją przetwarzania tekstu cechują się także systemy tradycyjnego tłumaczenia automatycznego – np. [35].

Zagadnieniem generowania wypowiedzi docelowej zajmuje się aplikacja *Polin* (ang. *Polish Linearizer*), ostatnia w szeregu aplikacji przetwarzania języka systemu *Thetos*. Do zapisu wyników tłumaczenia opracowano język *Thel* (*Thetos Language*); jest to język pośredniczący między częścią przetwarzania tekstu a częścią animacyjną. Więcej o *Thel* można znaleźć w [36]. Moduł *Polin* służy właśnie do generowania wypowiedzi w *Thel*. Zakres jego działania obejmuje ostatnie cztery punkty wymienionego wyżej schematu.

Na podstawie reprezentacji semantycznej dla każdego pojedynczego zdania budowana jest struktura semantyczna wypowiedzi wyjściowej. W najprostszych przypadkach jest ona tą samą strukturą predykatowo-argumentową co dla zdania wejściowego. Dla zdań biernych i innych przypadków bardziej złożonych struktura semantyczna jest generowana zgodnie z wymogami składni języka migowego: nie ma w nim konstrukcji biernych, wtrąconych, imiesłowowych, zdanie powinno być proste, z AKCJĄ i AKTOREM. Następnym krokiem jest generowanie struktury składniowej odpowiadającej strukturze semantycznej. Miejsce podmiotu zajmuje SG pełniąca rolę AKTOR, orzeczenia – AKCJA, reszta miejsc walencyjnych w tej strukturze jest wypełniana przez pasujące SG z reprezentacji składniowej zdania wejściowego. Tak otrzymana syntaktyczna reprezentacja wypowiedzi wyjściowej jest w następnym kroku linearyzowana. Przede wszystkim ustawia się szyk zdania, który w języku migowym jest stały: na pierwszym miejscu zawsze stoi podmiot, na drugim – orzeczenie, na trzecim – dopełnienie bliższe; wśród dalszych części zdania dopuszczona jest względna swoboda. Następnie każda SG jest linearyzowana z osobna i zapisywana do pliku wyjściowego zgodnie z ustalonym szykiem w tym zdaniu. Jak widać, generowanie tekstu także jest oparte na mechanizmie grup składniowych.

6. Zakończenie

Podkreślając charakter informatyczny SG-modelu, akcentujemy jego aspekt inżynierski, skierowany na algorytmizację i wykorzystanie w systemach informatycznych. Formalne modele języków są opracowywane w celu zastosowania ich w systemach przetwarzania tekstów, toteż cały mechanizm modelowania jest ukierunkowany na jego praktyczne wykorzystanie. SG-model nie stanowi w tym wyjątku: pierwotnie był skierowany na użycie w systemie tłumaczenia Thetos.

Nasz model, jak każdy formalny opis, uwypukla i odzwierciedla tylko niektóre cechy niezmiernie złożonego obiektu, jakim jest język naturalny, i w żadnym stopniu nie pretenduje do bycia opisem języka w całości; co więcej, uważamy, że zbudowanie globalnego modelu języka, a tym bardziej jego realizacja w postaci oprogramowania obecnie nie są możliwe.

Obecnie programy realizujące SG-model są udostępnione on-line na serwerze LAS (http://las.aei.polsl.pl/las2), dzięki czemu są intensywnie testowane zarówno przez autorów, jak i przez ekspertów zewnętrznych, do których zaliczamy kolegów, studentów, pracowników zaprzyjaźnionych firm, a także całkiem obcych ludzi zwracających uwagę na błędy i niedociągnięcia. Mimo niedoskonałości oprogramowania eksperymenty potwierdzają główne tezy badań: grupy składniowe są tymi cegiełkami, z których buduje się treść tekstu, a relacje składniowe wiążą elementarne treści w pewien spójny system.

Jak każdy model, SG-model jest niedoskonały i zapewne będzie rozwijany. Na pewno zmianom ulegnie gramatyka SGGP, w szczególności w części analizy zdań biernych, zdań zawierających zdania wtrącone, jak też skomplikowane struktury imiesłowowe. Ponadto, SGGP zostanie uzupełniona o ustanowienie relacji między zdaniami pojedynczymi – składowymi zdań złożonych. Modelu semantycznego w wersji obecnej nie można uważać za ostateczny i spójny. Mimo naszego przekonania, że nie należy szukać jedynego formalizmu do modelowania semantyki, lecz modelować oddzielne zjawiska semantyczne, korzystając z formalizmów odpowiednich do każdego zagadnienia, należy jednak ułożyć te formalizmy w pewien spójny system. Być może w przyszłości, po nabyciu doświadczenia w wykorzystaniu tych modeli da się ułożyć z nich spójny system formalizmów semantycznych.

BIBLIOGRAFIA

1. Vetulani Z.: Komunikacja człowieka z maszyną. Komputerowe modelowanie kompetencji językowej, Akademicka Oficyna Wydawnicza EXIT, Warszawa 2004.

2. Szmal P., Suszczańska N.: Selected problems of translation from the Polish written language to the sign language. Archiwum Informatyki Teoretycznej i Stosowanej, 13(1), 2001, p. 37÷51.
3. Zajadacz A., Szmal P., Suszczańska N., Grudziński T.: Programy multimedialne SITex i SITur jako udogodnienia w przekazywaniu informacji niesłyszącym kulturowo turystom. [W:] Młynarczyk Z., Zajadacz A.: Uwarunkowania i plany rozwoju turystyki. Tom VII – aspekty społeczne. Seria Turystyka i Rekreacja – Studia i Prace nr 7. Bogucki Wydawnictwo Naukowe, Poznań 2010, s. 107÷123.
4. Suszczańska N.: System LIANA wspomagania projektowania systemów informatycznych. [W:] Hnatkowska B. i Huzar Z. [red.]: Inżynieria oprogramowania – metody i narzędzia wytwarzania oprogramowania PWN, Warszawa 2008, s.146÷159.
5. Żurek J., Myrda Ł., Pokusa D. i in.: Raport działalności koła naukowego IPiIJ. Specyfikacja systemu ANKA. Gliwice 2010.
6. Szmal P., Kulików S.: System Thetos w serwisie tekstów i streszczeń z tłumaczeniem na język migowy. [W:] Demenko G., Izworski A., Michałek M. [eds.]: Speech Analysis, Synthesis and Recognition in Technology, Linguistics and Medicine. Uczelniane Wydawnictwa Naukowo-Dydaktyczne AGH, Kraków 2005, p.118-121.
7. Przepiórkowski A.: Powierzchniowe przetwarzanie języka polskiego. Akademicka Oficyna Wydawnicza EXIT, Warszawa 2008.
8. Piasecki M.: Cele i zadania lingwistyki informatycznej, [W:] Metodologie językoznawstwa. Współczesne tendencje i kontrowersje. Lexis, 2008
9. Chomski N.: Syntactic Structures. Walter de Gruyter GmbH & Co. KG, Berlin 1957, 2002.
10. Flasiński M.: Wstęp do analitycznych metod projektowania systemów informatycznych. WNT, Warszawa 1997.
11. Subieta K.: Obiektowość w projektowaniu i bazach danych. Akademicka Oficyna Wyd. PLJ, Warszawa 1998.
12. Tokarski J. -red., Saloni Z.: Schematyczny indeks a tergo polskich form wyrazowych. PWN, Warszawa1993.
13. Gladky A. V.: Sintaksiczeskie struktury jestestwennogo jazyka w awtomatizirowannych sistemach obszczenija. Nauka, Moskwa 1985.
14. Polański K. (red.): Słownik syntaktyczno-generatywny czasowników polskich. Wydawnictwo PAN, Warszawa-Wrocław-Katowice-Gdańsk 1980.

15. Suszczańska N.: GS-gramatyka języka polskiego. [W:] Demenko G., Izworski A., Michałek M. [eds.]: Speech Analysis, Synthesis and Recognition in Technology, Linguistics and Medicine. Uczelniane Wydawnictwa Naukowo-Dydaktyczne AGH, Kraków 2005, s. 113÷117.
16. Suszczańska N., Szmal P. and Simiński K.: The Deep Parser for Polish. LNAI, vol. 5603 (eds. Z Vetulani, H. Uszkorait), Springer, Berlin / Heidelberg 2009, s. 205÷217.
17. Bień J. S.: Koncepcja słownikowej informacji morfologicznej i jej komputerowej weryfikacji. Wydawnictwa Uniwersytetu Warszawskiego, Warszawa 1991.
18. Bolc L., Cichy M., Różańska L.: Przetwarzanie języka naturalnego (Natural Language Processing). WNT, Warszawa 1982.
19. Mykowiecka A.: Podstawy przetwarzania języka naturalnego. Akademicka Oficyna Wydawnicza ReadMe, Warszawa 1992.
20. Szafran K.: Analizator morfologiczny SAM-95, opis użytkowy. Technical Report TR 96-05, Instytut Informatyki Uniwersytetu Warszawskiego, Warszawa 1996.
21. Szpakowicz S.: Formalny opis składniowy zdań polskich. Wydawnictwa Uniwersytetu Warszawskiego, Warszawa 1986.
22. Saloni Z., Świdziński M.: Składnia współczesnego języka polskiego. PWN, Warszawa 1975.
23. Grzegorczykowa R.: Wprowadzenie do semantyki językoznawczej. PWN, Warszawa 2001.
24. Grzegorczykowa R.: Wykłady z polskiej składni. PWN, Warszawa 2004.
25. Moroz A., Wiśniewski M. -red.: Studia z gramatyki i semantyki języka polskiego. Wydawnictwo Uniwersytetu Mikołaja Kopernika, Toruń 2004.
26. Przepiórkowski A., Kupść A., Marciniak M., Mykowiecka A.: Formalny opis języka polskiego. Teoria i implementacja. Akademicka Oficyna Wydawnicza EXIT, Warszawa 2002.
27. Saloni Z., Świdziński M.: Składnia współczesnego języka polskiego, PWN, Warszawa 2007.
28. Mykowiecka A.: Inżynieria lingwistyczna. Komputerowe przetwarzanie tekstów w języku naturalnym. Wydawnictwo PJWSTK, Warszawa 2007.
29. Mel'čuk I. A.: Opyt teorii lingvisticzeskikh modelej „Smysl↔Tekst". Semantika, sintaksis. Wydawnictwo MGU, Moskva 1974.
30. Padučeva E. V.: O semantike sintaksisa. Nauka, Moskva 1974.
31. Wierzbicka A.: Dociekania semantyczne. Wydawnictwo PAN, Wrocław-Warszawa-Kraków 1969.

32. Suszczanska N.: On some universal algebras using in NL-semantics, Atlas Mathematical Conference Abstracts, Workshop on General Algebra AAA60 (60. Arbeitstagung Allgemeine Algebra), UT, Dresden 2000, http://at.yorku.ca/c/a/e/e/75.htm.
33. Suszczańska N., Szmal P., Francik J.: Translating Polish Texts into Sign Language in the TGT System, 20th IASTED International Multi-Conference Applied Informatics AI 2002, Innsbruck, Austria 2002, s. 282÷287.
34. Romaniuk J., Suszczańska N. and Szmal P.: Semantic Analyzer in the Thetos-3 System, LNAI vol. 6562 (eds. Z. Vetulani, H. Uszkoreit), Springer, Berlin/Heidelberg 2011, s. 234÷244.
35. Jassem K.: Przetwarzanie tekstów polskich w systemie tłumaczenia automatycznego POLENG. Wydawnictwo Naukowe UAM, Poznań 2007.
36. Suszczańska N., Szmal P.: Thel, a language for formalization of Polish Sign Language utterances. In: Proceedings of the 5th Language & Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics, Poznań 2011, s. 177÷181.

Recenzent: Dr Filip Graliński

Wpłynęło do Redakcji 5 grudnia 2011 r.

Abstract

The paper is devoted to a description of the research and implementation works developed in the framework of the *Thetos* project concerned with translation of texts into the sign language. The works are aimed at construction of a computer model of selected aspects of Polish. The elements elaborated in this framework are formalisms for syntactic and semantic representation of the sentence, grammars that correspond to those formalisms, syntactic and semantic analysis procedures. Special stress has been put on practical application of the considered methods for Polish language processing in the *Polsyn* analyzer implementing the syntax model, and in *Polsem* implementing the semantic one. A special attention deserves the Polin module responsible for generation of *Thel* language utterances.

Adresy

Nina SUSZCZAŃSKA: Politechnika Śląska, Instytut Informatyki, ul. Akademicka 16, 44-100 Gliwice, Polska, nina.suszczanska@polsl.pl

Przemysław SZMAL: Politechnika Śląska, Instytut Informatyki, ul. Akademicka 16, 44-100 Gliwice, Polska, przemyslaw.szmal@polsl.pl