Maciej Piasecki
Centrum Informatyczne Politechniki Wrocławskiej

Automatyczne modelowanie semantyki zdań twierdzących języka polskiego.

Mimo iż, oficjalnie pracuję naukowo dopiero od roku, z przetwarzaniem języka naturalnego jestem związany już od 1991 roku, czasów studenckich. Specjalnym zainteresowaniem i fascynacją darzę problematykę automatycznego tłumaczenia tekstu oraz analizę znaczenia tekstów w języku naturalnym. Jedynym większym ukończonym przeze mnie dotychczas dziełem jest eksperymentalny system generujący w sposób automatyczny formułę logiczną stanowiącą model znaczenia zdania. Stanowi on pewien krok badawczy w kierunku języka pośredniego - języka formalnego wyrażającego znaczenie zdania. W dalszej częsci artykułu skoncentruję się głównie na krótkiej prezentacji charakterystycznych cech systemu.

W celu uczynienia opisu bardziej przejrzystym, podzielę go trzy poziomy, plany:

- podstawowej teoria lingwistyczna,
- jej praktycznego rozwinięcia pod kątem zastosowania informatycznego,
- "implementacji" teorii w postaci systemu informatycznego.

Jako podstawę lingwistyczną mojej pracy przyjąłem gramatykę Richarda Montague, a szczególnie jej najbardziej praktyczną wersję - PTQ (*ang. The Proper Treatment of Quatification in Ordinary English*) [1,2,3,4]. Spełnia ona dobrze dwa podstawowe wymagania:

- generuje dość szeroki wycinek języka (przynajmniej jak na potrzeby mojej pracy),
- oraz posiada spójne i ściśle zdefiniowane mechanizmy generujące strukturę znaczeniową zdania.

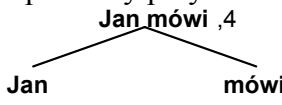
Również zakres wyznaczony przez PTQ stał się naturalną granicą mojej pracy - określa ono zarówno wycinek języka, jak ograniczenie się jedynie do analizy zdań twierdzących.

Gramatyka Montague stanowi ciekawe połączenie pozornych przeciwieństw: cech charakterystycznych dla gramatyki transformacyjnej (n.p. budowa reguł produkcji), gramatyki kategorialnej (n.p. konstrukcja systemu kategorii syntaktycznych) oraz semantyki generatywnej. Definiuje zarówno zbiór reguł syntaktycznych, jak i też ściśle z nimi powiązanych reguł semantycznych, przypisujących każdej wytworzonej frazie, jej reprezentację semantyczną zapisaną w postaci formuły logiki intensjonalnej (logika intensjonalna został stworzona przez R. Montague poprzez rozszerzenie logiki tradycyjnej o aspekty temporalne, modalne, operator lambda [12] oraz zdefiowane przez niego operatory intensji i ekstensji, wywodzące się z teorii możliwych światów

[3]). Powiązanie to jest definiowane w ramach roszerzonej wersji zasady Fregego (nazywanej też zasadą kompozycyjności): "znaczenie zdania jest funkcją znaczeń jego części i sposobu ich połączenia". W gramatyce Montague definiuje ona jednoznacze powiązanie pomiędzy kategoriami syntaktycznymi a typami logicznymi, regułami syntaktycznymi a semantycznymi oraz dla każdego wyrażenia podstawowego języka (leksemu) definiuje jego reprezentację w postaci formuły logiki intensjonalnej (LI).

Dzięki zasadzie kompozycyjności, jeżeli wygenerujemy dla pewnego zdania drzewo derywacji, to traktując numery reguł syntaktycznych w węzłach jako numery odpowiadających im reguł semantycznych, oraz przypisując wyrażeniom w liściach (czyli wyrażeniom podstawowym) odpowiadające im formuły LI, jesteśmy w stanie, w prosty sposób, wyliczyć formułę LI. Możemy ją traktować, z pewnym przybliżeniem, jako reprezentację semantyczną, model znaczenia zdania, utożsamianego z sądem logicznym - asercją (oczywiście interesuje nas tylko znaczenie informatywne zdania).

Proces ten, jest w dużym skrócie zilustrowany przez poniższy przykład:

zdanie: *Jan mówi*, otrzymuje drzewo derywacji: 

.Indeks skojarzony z węzłem sygnalizuje użytą regułę syntaktyczną (a tym samym semantyczną). Na podstawie drzewa derywacji możemy wyliczyć formułę LI.

1) **Jan** $\Rightarrow \lambda P[P\{j\}]$ translacja wyrażenia podstawowego,
2) **mówić** $\Rightarrow$ **mówić'** translacja wyrażenia podstawowego,
3) $\lambda P[P\{j\}](\hat{}$**mówić'**) z 1) i 2) przez zastosowanie T4,

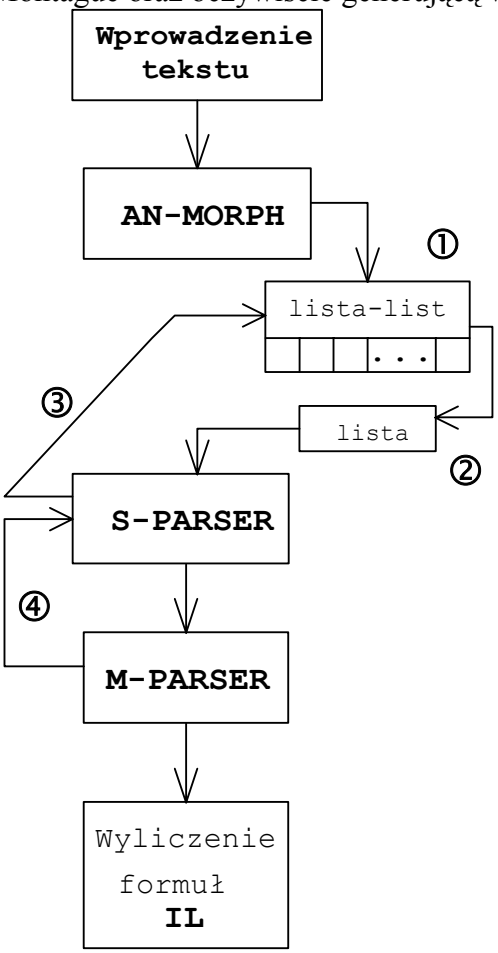
upraszczając otrzymujemy $\check{}\hat{}$**mówić'**(j), oraz ostatecznie **mówić'**(j).

(j oznacza pewną stałą ze zbioru istnienień, bytów realnych bądź abstrakcyjnych, $\hat{}$ operator intensji oraz $\check{}$ ekstensji)

Oczywiście oryginalne PTQ generuje wycinek języka angielskiego, musiałem dlatego też sformułować jego polską wersję. Po odpowiednim dobraniu wycinka języka, wyłonił się tylko jeden poważniejszy problem płynności szyku zdań.

Oryginalna gramatyka Montague nie nadaje się niestety do bezpośredniej implementacji w zastosowaniach informatycznych. Posiada dwie podstawowe wady: ma charakter wybitnie generacyjny oraz operuje na ciągach symboli ("stringach") bardzo niewdzięcznej strukturze danych do przetwarzania. Nie chcąc wywarzać otwartych już drzwi szukałem zrealizowanych rozwiązań tego problem. Za najciekawsze uznałem gramatykę izomorficzną M-grammar prof. Jana Landsbergena, stanowiącą podstawę systemu automatycznego tłumaczenia *Rosetta* [5, 6, 7, 8, 9]. Operuje ona na zdaniach zapisanych w postaci struktury drzew S-tree oraz jej konstrukcja umożliwia łatwą budowę zarówno analizatora (ang. *parser*), jak i

generatora. Wymaga skojarzenia z prostą gramatyką wstępną, której zadaniem jest wygenerowanie dla zdania wejściowego skończonej liczby drzew S-tree (nie wszystkie muszą być akceptowalne). Gramatyka ta, ukierunkowana na specyficzne zastosowanie w systemie automatycznego tłumaczenia, gubi niestety pewne korzystne z punktu widzenia mojej pracy powiązania pomiędzy syntaktyką i semantyką. Zostały między innymi wyeliminowane elementy pochodzące z gramatyki kategorialnej (powiązania pomiędzy kategoriami i typami LI). Spowodowało to konieczności jej przetworzenia w gramatykę PM-grammar, przywracającą wszystkie pożądane cechy gramatyki Montague oraz oczywiście generującą wycinek języka polskiego.



W ten sposób zostały skompletowane podstawy lingwistyczne systemu - przedstawiam obok jego poglądowy schemat. Analizator morfologiczny AN-MORPH (niezbędny element wejściowy) pracuje w oparciu o algorytm posługujący się słownikiem tematów i tablicami odmian. Podstawę do jego konstrukcji stanowi świetna praca prof. J. S. Bienia [10]. Produktem finalnym analizatora jest zbiór wszystkich prawdopodobnych ciągów wyrazów morfologicznych - zapisanych jako terminalne drzewa S-tree (sam węzeł zawierający informację morfologiczną i syntaktyczną rozpoznanego wyrazu). Każdy z ciągów jest poddawany dalszej analizie poprzez parser gramatyki wstępnej S-PARSER. Jest on oparty o opracowanie prof. S. Szpakowicza [11], formalizujące składnię języka polskiego w postaci gramatyki DCG z kontekstem. Efektem jego działania jest struktura drzewa S-tree, która następnie zostaje przekazana do parsera gramatyki PM-grammar - M-PARSER. Jeżeli struktura zostanie rozpoznana jako poprawna to na podstawie wygenerowanego drzewa derywacji (lub drzew w przypadku niejasności semantycznej) można dokonać wyliczenia formuły (formuł) LI.

Gramatyka Montague pozwala jedynie na formalny zapis struktury znaczeniowej analizowanego zdania, zatrzymuje się na poziomie znaczenia leksemów, analizowane są jedynie izolowane zdania a nie tekst. Wygenerowane formuły logiki intensjonalnej są

ciągle trudne do maszynowej interpretacji, prawdopodobnie sama logika intensjonalna jest zbyt mocnym narzędziem. Te przesłanki wyznaczają drogę moich dalszych poszukiwań. Możliwe, że ciekawe rozwiązania przyniosło by syntetyczne wykorzystanie niektórych osiągnięć sztucznej inteligencji (np. semantyki proceduralnej).

Bibliografia (wybór najistotniejszych pozycji):

[1] Montague Richard "English as a Formal Language" zawarte w B. Visentini et al , eds., "Linguaggi nella societá e nella technica", Milan: Edizioni di Comunitá, str. 189-224.

[2] Montague Richard "The Proper Treatment of Quantification in Ordinary English", zawarte w Hintikka J., Moravcsik J., Suppes P. "Approaches to Natural Language", Dordrecht: D. Reidel, str 221-242.

[3] Dowty D. R., Wall, R. E., Peters S. "Introduction to Montague Semantics", Dordrecht: D. Reidel, 1981

[4] Dowty D. R. "Montague Grammar and Word Meaning", Dordrecht: D. Reidel

[5] Landsbergen Jan, "Montague Grammar and Machine Translation", artykuł w zbiorczym wydaniu "Linguistic Theory and Computer Applications" Academic Press Limited, 1987.

[6] Landsbergen Jan, Odijk Jan, Schenk André, "The Power of Compositional Translation", artykuł w "Literary and Linguistic Computing", Vol. 4, No. 3, 1989, Oxford University Press.

[7] Landsbergen Jan, Appelo Lisette, Fellinger Carel, "Subgrammars, Rule Classes and Control in the Rosetta Translation System", materiały na "3rd Conference ACL, European Chapter", kwiecień 1987

[8] Landsbergen Jan, "Adaptation of Montague Grammar to The Requirements of Parsing" artykuł w zbiorczej publikacji "Formal Methods in the Study of Language" część 2, pod redakcą Groenendijk J.A.G., Janssen T.M.V., Stokhof M.B.J., MC Track 136, Mathematical Centre, Amsterdam, 1981, strony 399-420.

[9] Landsbergen Jan, "Isomorphic Grammars and their Use in the Rosetta Translation System", w zbiorczej publikacji "Machine Translation Today", pod redakcją King M., Edinburg: Edinburg University Press, 1985.

[10] Bień Janusz S. "Koncepcja słownikowej informacji morfologicznej i jej komputerowej weryfikacji", Wydawnictwo Uniwersytetu Warszawskiego, Warszawa 1991.

[11] Szpakowicz Stanisław "Formalny opis składiowy zdań polskich", Wydawnictwo Uniwersytetu Warszawskiego, Warszawa, 1983

[12] Brady J.M "Informatyka teoretyczna w ujęciu programistycznym" (tytuł oryginału *"The Theory of Computer Science A Programming Approach"*), Wydawnictwa Naukowo-Techniczne, Warszawa, 1983