Leszek Gajecki, Ryszard Tadeusiewicz

Modelowanie języka polskiego z wykorzystaniem gramatyki struktur frazowych

Naszym celem jest modelowanie języka. Model językowy odgrywa ważną rolę w systemach automatycznego rozpoznawania mowy dużego słownika. W wielu do tej pory stosowanych modelach kolejność słów jest istotna. Jednakże w przypadku języków słowiańskich często kolejność słów nie ma dużego znaczenia. To jest przyczyną tego, że zastosowane modele są niedokładne dla języków słowiańskich.

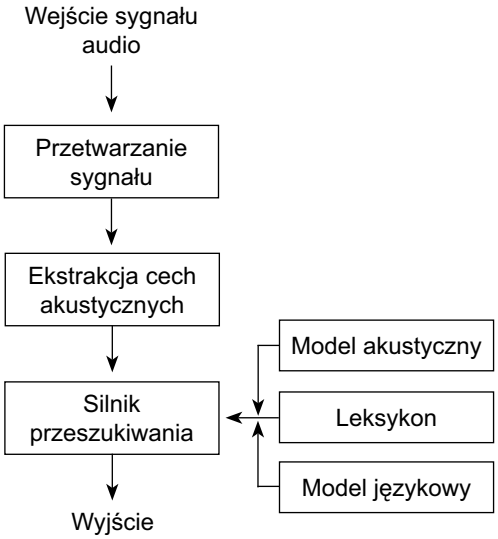
Naszym zdaniem lepszym rozwiązaniem jest zastosowanie gramatyki struktur frazowych. Przedstawiamy tutaj prostą gramatykę oraz nasze podejście do automatycznej ekstrakcji reguł za pomocą sieci neuronowej. Reguły te obejmują zależności między słowami według ich kategorii gramatycznych i zostaną użyte do generowania leksykonu gramatyki, który opisuje także zależności specyficzne dla poszczególnych słów.

1. Budowa systemu ciągłego rozpoznawania mowy oraz zastosowanie modelu językowego

Na początku będziemy rozważać architekturę typowego systemu ciągłego rozpoznawania mowy (rycina 1.). Przykładami takiego systemu mogą być: system rozpoznawania mowy K. U. Leuven [Duchateau 1998; PSI 2006], HTK [Young i in. 2006], a także dla języka polskiego system opisany przez Daniela Koržinka i Łukasza Brockiego [2007] albo system rozpoznawania mowy języka polskiego przeznaczony dla wąskiej domeny słów [Hnatkowska, Sas 2008].

W rozważanym systemie na początku sygnał audio poddajemy przetworzeniu, tak aby mógł zostać przekazany do przetwornika A/C. Po przekształceniu na postać cyfrową widmo sygnału cyfrowego zostaje poddane normalizacji. Moduł ekstrakcji cech akustycznych wydobywa cechy istotne z punktu widzenia rozpoznawania mowy – otrzymujemy wektor cech akustycznych $X = [x_1, x_2, ..., x_n]$.

Nim zostanie przeprowadzone rozpoznawanie mowy, silnik wyszukiwania konstruuje siatkę słów – jest to sieć opisująca budowę słów w postaci podstawowych jednostek mowy (fonemy, sylaby, trifony). Każda podstawowa jednostka mowy jest reprezentowana za pomocą Niejawnych Modeli Markowa (HMM). Dodatkowo siatka słów może być połączona z modelem językowym, który będzie określał możliwe słowa, które wystąpią po danym słowie. Mając zbudowaną siatkę słów, moduł może rozpocząć zadanie rozpoznawania mowy.



Rycina 1. Architektura typowego systemu ciągłego rozpoznawania mowy

Na jego wejście zostaje podany ciąg wektorów cech akustycznych $W_1^K = w_1, \ldots, w_K$. Sam moduł dokonuje przeszukiwania optymalnej sekwencji słów W_{opt} pośród wszystkich możliwych sekwencji słów, co możemy zapisać:

$$W_{opt} = \arg\max_{w_i \subset W_1^K} P\,(w_i|X) \tag{1.1}$$

Najczęściej jest stosowane przeszukiwanie Beam Search, z odrzucaniem najgorszych hipotez. Każda hipoteza posiada koszt – miarę heurystyczną jej jakości uwzględniającą:

a. Podobieństwo wektorów cech akustycznych do wzorców podstawowych jednostek mowy, podawane przez Model Akustyczny jako miara prawdopodobieństwa.
b. Poprawność w sensie wymowy. Leksykon przechowuje informacje o wymowie każdego słowa, więc $P(X|W_1^K)$ stanowi efekt współpracy z Modelem Akustycznym. Każda hipoteza jest budowana jako złożenie wypowiedzi słów zawartych w Leksykonie, co znacznie ogranicza przestrzeń przeszukiwania.
c. Poprawność w sensie modelu językowego. Może to być prawdopodobieństwo określające możliwość wystąpienia danej sekwencji słów, np. $P(W_1^K)$.

Koszt każdej hipotezy można określić następująco:

$$f(X, W_1^K) = \log P\,(X|W_1^K) + CA\,\log P\,(W_1^K) + CC \tag{1.2}$$

gdzie:
CA – waga prawdopodobieństwa uzyskanego przez model językowy
CC – wartość ujemna rozumiana jako kara za rozpoczęcie nowego słowa

X – otrzymany wektor cech akustycznych
W_1^K – sekwencja słów

Dalsze szczegółowe informacje o systemach rozpoznawania mowy są opisane w pracach Duchateau [1998], a także Benesty, Sondhi, Huang [2008], Younga [2006] i Markowitz [1996], a charakterystyka sygnału mowy również w książce Tadeusiewicza [1988].

2. Model językowy

2.1. Model trigramowy

Model N-gramowy wyznacza prawdopodobieństwo wystąpienia słów jako iloczyn prawdopodobieństwa wystąpienia ciągu K – 1 ostatnich słów i prawdopodobieństwa nowego słowa w_k pod warunkiem wystąpienia ciągu N – 1 ostatnich słów:

$$P(W_1^K) = P\,(W_1^{K-1})\;P\,(w_k \mid W_{K-N+1}^{K-1}) \tag{2.1}$$

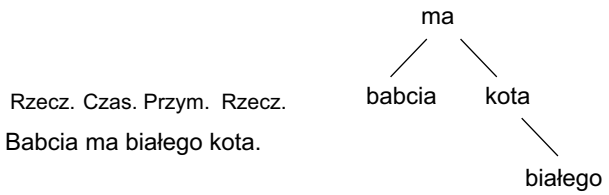
W przypadku modelu trigramowego N = 3, więc prawdopodobieństwo wystąpienia nowego słowa w_k pod warunkiem wystąpienia N – 1 poprzedzających je słów może zostać wyliczone za pomocą funkcji zliczającej:

$$P\,(w_k \mid W_{K-2}^{K-1}) = \frac{c\,(w_{K-2}, w_{K-1}, w_K)}{c\,(w_{K-2}, w_{K-1})} \tag{2.2}$$

Modelowanie języka naturalnego dobrze opisuje Winograd [1983], a w szczególności zagadnienie modelu N-gramowego przedstawia w swoich pracach Jelinek [np. 1985].

2.2. HPSG – gramatyka struktur frazowych

HPSG (ang. Head-driven Phrase Structured Grammar) jest opisana w pracy Pollarda [1994], a jej zastosowanie dla języka polskiego – w książce Przepiórkowskiego, Kupść, Marciniak, Mykowieckiej [2002]. Jest to teoria lingwistyczna bazująca na formalizmie unifikacji (albo inaczej mówiąc ograniczeń). Składa się z dwóch części: niewielkiego zbioru reguł (ogólnych ograniczeń) oraz dużego zbioru pozycji leksykalnych, który opisuje zależności specyficzne dla każdego słowa. Przykład jest pokazany na rycinie 2.



Rycina 2. Przykład zastosowania gramatyki HPSG

Omówimy teraz zdanie „Babcia ma białego kota". Na początku rzeczownik „kota" łączy się z przymiotnikiem „białego" zgodnie z regułą uzgodnienia (przymiotnik jest tu argumentem dla rzeczownika). Tak otrzymana fraza „białego kota" ma cechy rzeczownika „kota" (np. liczba, rodzaj gramatyczny według reguły elementu głównego) i staje się argumentem czasownika „ma" (orzeczenia), tworząc z nim grupę orzeczenia. Następnie dochodzi do połączenia z grupą podmiotu (tylko rzeczownik „babcia"), co tworzy zdanie.

2.3. Uproszczona gramatyka

Utworzyliśmy prostą gramatykę, która umożliwiła nam rozpoczęcie badań nad zastosowaniem HPSG w modelu językowym [Gajecki, Tadeusiewicz 2008]. Zaimplementowaliśmy reguły elementu głównego, pozycji leksykalnych (*word entry*) oraz uzgodnienia (tylko rzeczowniki i przymiotniki). Sposób generowania pozycji leksykalnych:

- Aby było możliwe połączenie grupy orzeczenia z grupą podmiotu w zdanie, czasownik jako orzeczenie łączy się z rzeczownikami w pozostałych przypadkach niż mianownik, stając się grupą orzeczenia. Ta z kolei łączy się z grupą podmiotu, której elementem głównym jest rzeczownik w mianowniku.
- Rzeczownik może się łączyć z przymiotnikami jako argumentami (zachowując regułę uzgodnienia) i staje się elementem głównym takiej frazy, która mając cechy elementu głównego, może być łączona z czasownikiem lub grupą orzeczenia jak wyżej.
- Pozostałe słowa mają pustą listę argumentów i modyfikatorów.

Tak prosta gramatyka ze względu na swoje niezbyt duże pokrycie rzadko dawałaby informację o poprawności całego zdania, co w niewielkiej liczbie przypadków pozwoliłoby nam wybrać właściwe zdanie spośród hipotez. Z tego powodu będziemy używać określenia częściowej poprawności zdania (wypowiedzi), co nawiązuje do np. komunikacji słownej z codziennego życia, w którym niekiedy dla wygody i sprawności komunikacji używamy zdań niezupełnie poprawnych. W przypadku naszych badań tworzymy listę słów, z których każde łączy się przynajmniej z jednym innym słowem, a to podejście heurystyczne pozwala nam uzyskać miarę poprawności częściowej zdania.

2.4. Zastosowanie HPSG

Zastosowanie gramatyki HPSG w modelu językowym systemu rozpoznawania mowy jest wspomniane w pracy Kaufmanna i Pfister [2007]. W naszym przypadku zamiast informacji o pełnej poprawności zdania wykorzystujemy miarę częściowej poprawności

ności. Po uzyskaniu N najlepszych hipotez poddajemy je parsowaniu, a następnie w zależności od wyników tej operacji modyfikujemy koszt każdej z nich, po czym wybieramy najlepszą hipotezę (o najniższym koszcie). Funkcja kosztu:

$$h\,(W_1^K) = f\,(X,\, W_1^K) - HC \cdot \frac{l_p}{l} \tag{1.2}$$

gdzie:

$f\,(X,\, W_1^K)$ – wartość funkcji kosztu obliczona przez silnik wyszukiwania – wzór (1.2)
HC – waga dla miary zwracanej przez moduł gramatyki HPSG
l_p – łączna liczba wszystkich słów $w_i \in WP_1^K$, które łączą się przynajmniej z innym takim słowem; ciąg takich słów $WP_1^K \subset W_1^K$ jest podciągiem ciągu słów W_1^K tworzących całe zdanie (wypowiedź)
l – całkowita długość ciągu słów W_1^K (czyli zdania/wypowiedzi)

3. Badania nad HPSG

Oprogramowanie, którego użyliśmy, jest oprogramowaniem symulującym działanie systemu rozpoznawania mowy [Gajecki, Tadeusiewicz 2008] – tutaj rozpoznawaliśmy słowa pochodzące z ciągu liter. Ciągi te powstały przez połączenie liter tworzących wyrazy. Aby uzyskać efekt niedokładności – innej artykulacji danej głoski niż zostało to zapisane w modelu, zastosowaliśmy „zaszumianie" tych ciągów. Takie podejście, mimo że jest pewnym uproszczeniem, pozwala nie tylko na przyśpieszenie eksperymentów, lecz także na zbadanie wpływu modelu językowego na cały system rozpoznawania mowy. Celem porównania uzyskanych wyników z rezultatami dalszych prac, przytoczymy je tutaj za powyższą pracą.

HC	WER [%] (+trigramy)	WER [%] (bez trigramów)
0	77,5	77,5
1		77,5
10	79,3	77,5
100	85	77,5
1000	86,8	79,3
–2000	80	78,7

Tabela 1. Procent błędnych słów (%WER – ang. *word error rate*) dla różnych wartości wagi HC za pracą Tadeusiewicza i Gajeckiego [2008]

Wykorzystaliśmy część Korpusu IPI PAN [IPI PAN 2006; Przepiórkowski 2004]. Jako zbiór testowy (symulowane zadanie rozpoznawania mowy) użyliśmy zbioru 12 zdań (160 słów) zamkniętego słownika. Leksykon zawiera pozycje pochodzą-

ce z 2800 zdań (50 000 słów). Model trigramowy był trenowany na zbiorze 58 000 słów. Drugi zbiór trigramowy był trenowany na zbiorze 3,2 mln słów, ale jego zastosowanie nie spowodowało zmian. Spośród spostrzeżeń najważniejsze jest to, że kiedy uwzględniamy wyniki parsowania HPSG (HC > 0), jakość rozpoznawania przez większy zakres wag pozostaje niezmieniona w porównaniu do przypadku, kiedy nie uwzględniamy gramatyki HPSG (HC = 0).

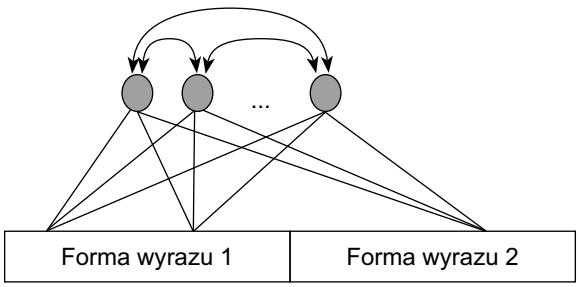
4. Rozbudowa modelu

Ręczne wprowadzanie dalszych reguł HPSG oraz informacji specyficznych o pozycjach leksykalnych wymaga dużego nakładu pracy. Z naszego punktu widzenia interesująca jest możliwość automatycznego bądź półautomatycznego wygenerowania niezbędnych reguł.

Zdecydowaliśmy się wybrać do tego celu sztuczne sieci neuronowe. Opis sieci neuronowych znajduje się np. w pracy Tadeusiewicza, Gąciarza, Borowik, Lepera [2007]. Na samym początku musimy określić sposób kodowania danych. Podjęliśmy decyzję, że w obecnej fazie będziemy uczyć sieć relacji zależących tylko od cech gramatycznych wyrazu (co byłoby wspólne dla innych wyrazów), a nie od niego samego. Istota kodowania dla jednego wyrazu jest przedstawiona na rycinie 3. Mamy tutaj do czynienia z sześcioma cechami. Każda cecha będzie zapisana w formie kodu 1 z N. Daje to ciąg 52 wartości. Cechy, które nie pojawiają się przy danym wyrazie (np. rzeczownik nie zawiera informacji o stopniu), będą kodowane jako ciąg zer (w procesie uczenia), a w czasie pracy sieci nie będą brane do obliczenia odległości.

Fleksem	Liczba	Rodzaj	Przypadek	Osoba	Stopień
0 1 0 ... 0	0 1	0 1 0	0 0 1 ... 0	0 0 0	0 0 0

Rycina 3. Kodowanie cech jednego słowa



Rycina 4. Sieć samoorganizująca użyta w eksperymentach

Zdecydowaliśmy także, że będziemy uczyć sieć reguł występujących między dwoma kolejnymi wyrazami, więc wektor danych wejściowych będzie się składał z kodu odpowiadającego pierwszemu słowu, jak również drugiemu. Zastosowaliśmy tutaj sieć samoorganizującą (rycina 4).

W celu określenia odległości wektora wejściowego od wektora odpowiadającego wagom danego neuronu stosujemy odległość Euklidesa:

$$d(x, w) = \sqrt{\sum_{i=1}^{M} (x_i - w_i)^2} \tag{4.1}$$

gdzie:
x – wektor wejściowy
w – wektor wag neuronu
M – wymiar wektora danych wejściowych

Taka sieć samoorganizująca po procesie uczenia będzie reprezentować reguły, według których łączą się kolejne dwa wyrazy. Reguły te będą określone za pomocą wag neuronów. Jak jednak pokazały nasze późniejsze eksperymenty, tak otrzymane reguły będą rzadko reprezentowane w zbiorze testowym, natomiast często będziemy mieli wektory wejściowe dla zbioru testowego częściowo różniące się od wektorów reprezentowanych przez neurony. W takim przypadku dokonamy podziału neuronów na takie, które reprezentują reguły mające odpowiednie pokrycie w zbiorze walidacyjnym – będą reprezentować środki obszarów decyzyjnych odpowiadających poprawnym parom słów (reguły pozytywne) – oraz na neurony, które mają niewielkie pokrycie w zbiorze walidacyjnym – będą reprezentować niepoprawne zestawienia słów (reguły negatywne). W wyniku przetwarzania zbioru testowego, sieć neuronowa określi, który spośród wektorów, a w konsekwencji neuronów wejściowych jest najbardziej podobny do wektora danych wejściowych. Jeżeli neuron ten będzie oznaczony wcześniej jako odpowiadający regule pozytywnej, to para słów zostanie uznana jako poprawna, w przeciwnym wypadku – niepoprawna. Ostateczny koszt hipotezy będzie wyrażony następująco:

$$h\,(W_1^K) = f\,(X, W_1^K) - HC \cdot \frac{l_p}{l} \tag{4.2}$$

gdzie:

$f\,(X, W_1^K)$ – wartość funkcji kosztu obliczona przez silnik wyszukiwania – wzór (1.2)
HC – waga dla miary zwracanej przez moduł sieci neuronowej
l – ilość par kolejnych słów uznanych jako poprawne
l_p – całkowita długość ciągu słów W_1^K (czyli zdania/wypowiedzi)

Drugi (alternatywny) sposób wyliczenia końcowej wartości kosztu uwzględnia sumę odległości między kolejnymi wektorami wejściowymi a wagą odpowiadających im aktywnych neuronów:

$$h\,(W_1^K) = f\,(X, W_1^K) - HC \cdot s \tag{4.3}$$

Przy czym wspomniana wyżej suma wyraża się wzorem:

$$s = \sum_{i=1}^{k} \min d(X^1, w) \tag{4.4}$$

gdzie:

$f(X, W_1^K), HC$ – opisano wcześniej
w – wagi neuronów
X^i – i-ty wektor wejściowy w
d – odległość opisana wzorem (4.1)

Docelowo planujemy generowanie reguł HPSG z reguł nauczonych przez sieć neuronową. Pozwoliłoby to na połączenie uogólniającego działania sieci neuronowych oraz formalizmu HPSG w celu dokładnego modelowania języka naturalnego.

5. Wyniki i wnioski

Sieć neuronowa zawiera 100 neuronów i była trenowana na zbiorze 130 000 par słów wchodzących w skład kolejnych zdań należących do korpusu IPI PAN (wykonano dwie iteracje). Zbiór walidacyjny zawierał 40 000 par i jego pokrycie (w sensie aktywności neuronów odpowiadających regułom pozytywnym) wynosiło > 99%. Dla każdego testu dla zbioru walidacyjnego stwierdzono ok. 40–50 aktywnych neuronów. Zbiór testowy zawierał 80 000 par wyrazów i miał pokrycie w ilości 47% par dwójek poprawnych oraz 27% zdań całkowicie poprawnych (czyli zawierających tylko poprawne dwójki).

W celu rozpoznawania mowy wykorzystano wcześniejszy program symulujący rzeczywisty system rozpoznawania mowy. Zbiór testowy na rozpoznawanie mowy, tak samo jak w sekcji 3, składał się z 12 zdań (160 słów) zamkniętego słownika. Leksykon zawierał pozycje pochodzące z 2800 zdań (50 000 słów). Nie używano tutaj modelu trigramowego. Wyniki rozpoznawania mowy były takie same z użyciem sieci neuronowej jak i bez niej.

Stwierdzamy, że potencjalnie HPSG może przynieść poprawę jakości rozpoznawania mowy, podobnie jak zastosowanie sieci neuronowej. Interesującym rozwiązaniem jest nauka reguł HPSG przez sieć neuronową, gdyż reguły te byłyby czytelne dla człowieka, co pozwala na obserwację procesu ich tworzenia. Obecne prace pozostawiają do zbadania szeroki zakres zagadnień związanych z rozmiarem sieci oraz innymi możliwymi architekturami sieci neuronowej. Przypuszczamy, że zwiększenie zbioru treningowego sieci może spowodować poprawę jakości działania sieci neuronowej i jej widoczny wpływ na proces rozpoznawania mowy. Jesteśmy obecnie w fazie przenoszenia badań na rzeczywisty system rozpoznawania mowy.

BIBLIOGRAFIA

Benesty J., Sondhi M.M., Huang Y. (2008). *Springer Handbook of Speech Processing*. Berlin Heidelberg: Springer-Verlag.

Duchateau J. (1998). *HMM Based acoustic Modeling in Large Vocabulary Speech Recognition*, PhD thesis, Belgium: Catholic University of Leuven.

Gajecki L., Tadeusiewicz R. (2008). *„Modeling of Polish language for Large Vocabulary. Continuous Speech Recognition" in Speech and Language Technology*. Volume 11. ed. G. Demenko, K. Jassem (red.), Poznań: Polish Phonetic Association.

Hnatkowska B., Sas J. (2008). *Application of Automatic Speech Recognition to medical reports*. „Journal of Medical Informatics and Technologies" Vol 12.

IPI PAN (2006). *Corpus of Polish* [dostęp: 12–03–2009] Dostępne w Internecie: <http://korpus.pl>.

Jelinek F. (1985). *The development of an Experimental Discrete Dictation Recognizer*. Proceedings of the IEEE 73(11).

Kaufmann T., Pfister B. (2007). *An HPSG Parser Supporting Discontinuous Licenser Rules*. International Conference on HPSG, Stanford.

Koržinek D., Brocki L. (2007). *Grammar Based Automatic Speech Recognition System for the Polish Language* [w:] R. Jabłoński, M. Turkowski, R. Szewczyk (red.), *Recent Advances in Mechatronics*, s. 87–91.

Markowitz J.A. (1996). *Using Speech Recognition*. Prentice Hall PTR.

Pollard C.J., Sag I.A. (1994). *Head-Driven Phrase Structure Grammar*. Chicago: The University of Chicago Press.

Przepiórkowski A. (2004). *Korpus IPI PAN. Wersja wstępna*. Warszawa: IPI PAN.

Przepiórkowski A., Kupść A., Marciniak M., Mykowiecka A. (2002). *Formalny opis języka polskiego – Teoria i implementacja*. Warszawa: Akademicka Oficyna Wydawnicza EXIT.

PSI (2006). *Description of the ESAT speech recognition system*. (PSI – Speech Group Catholic University of Leuven, Belgium). [dostęp: 12–03–2009] Dostępne w Internecie: <http://www.esat.kuleuven.be/psi/spraak/>.

Tadeusiewicz R. (1988). *Sygnał mowy*. Warszawa: Wydawnictwo Komunikacji i Łączności.

Tadeusiewicz R., Gąciarz T., Borowik B., Leper B. (2007). *Odkrywanie właściwości sieci neuronowych przy użyciu programów w języku C#*. Kraków: Wydawnictwo Polskiej Akademii Umiejętności.

Winograd T. (1983). *Language as a Cognitive Process*, Volume I *Syntax*. Reading, MA: Addison-Wesley Publishing Company.

Young S., Evermann G., Kershaw D., Moore G., Odell J., Ollason D., Povey D., Valtchev V., Woodland P. (2006). *HTK Book*. Cambridge University Engineering Department.