B. INFORMATYKA

Dariusz Ceglarek

SKN Informatyków

ZASTOSOWANIE METOD STATYSTYCZNYCH I INFORMATYCZNYCH DO BADAŃ NAD JĘZYKIEM NATURALNYM

I. Słowo wstępne

W ostatnich latach znacznie wzrosło zainteresowanie postępem w komunikacji człowiek-maszyna cyfrowa za pomocą języków coraz bliższych językowi naturalnemu. Warunkiem koniecznym tego postępu jest lepsze poznanie struktur i związków występujących w języku, którym posługujemy się między sobą na co dzień. W artykule przedstawiono zastosowanie niektórych metod statystycznych, kombinatorycznych, informatycznych oraz grafowych dla analizy języka na dwóch jego poziomach: znakowym (morfologicznym) oraz semantycznym.
Pierwsza część pracy omawia wykorzystanie teorii informacji do optymalizacji kodów oraz zależności entropijne w języku polskim w porównaniu z innymi językami europejskimi. Przedstawiono także niektóre ciekawe metody statystyczne wykorzystywane w językoznawstwie. Dalej zaprezentowano niektóre istniejące oraz skonstruowane przez autora algorytmy analizy związków morfologicznych języka, w których jest on rozumiany jako system znaków (postać statyczna), a także algorytmy, których zadaniem jest przekroczenie bariery entropii (np. algorytm II). W tych ostatnich algorytmach język jest funkcjonującym układem o postaci dynamicznej. Ostatnia część artykułu stanowi opis metody analizy języka na poziomie związków semantycznych między słowoformami tworzącymi pełne zdania. Wszystkie przedstawione w artykule algorytmy napisane zostały w języku Fortran i sprawdzone na maszynie cyfrowej ODRA 1305.

II. Wprowadzenie: modele języka

W artykule tym pojawiają się dwa podejścia do badań języka naturalnego: „przyrodnicze" oraz „matematyczno-logiczne". Pierwsze polega na rozpatrywaniu zbiorów tekstów pisanych jako rzeczywistości, której prawidłowości bada się za pomocą obserwacji i matematycznego opracowania ich rezultatów. Stosuje się tu rachunek prawdopodobieństwa, kombinatorykę, statystykę matematyczną i teorię informacji. Drugie podejście polega na tym, że konstruujemy sztuczne modele języka naturalnego,

posiadające pewne jego własności i pełniące role analogiczne jak konstrukcje matematyczne.

Przez model języka będziemy rozumieli algorytm wytwarzający z danego zbioru elementów wejściowych (słownika pojęć) ciągi (łańcuchy) tych elementów. Łańcuchy tworzyć będą zdania w danym języku, a zbiór wszystkich łańcuchów (skończonych) składać będzie się na język tworzony przez dany model. Będziemy starać się traktować język formalnie, tzn. nie odwołując się do sensu, ponieważ sens, w odróżnieniu od struktury, jest bardo trudno określić jednoznacznie i precyzyjnie.

Budowa modeli języka jest pożądana z kilku względów. Po pierwsze dlatego, że budowanie modeli, tworzących języki dostatecznie bliskie językom naturalnym (ich dobre przybliżenia), rozwiązuje podstawowy problem lingwistyki strukturalnej: w jaki sposób z elementarnych jednostek danego języka (np. wyrazów), za pomocą skończonej ilości formalnych reguł operowania tymi jednostkami, można stworzyć dowolny tekst w tym języku? Po drugie, budowanie modeli języka jest pożyteczne przy rozwiązywaniu zadań związanych z automatycznym tłumaczeniem tekstu zapisanego w jednym języku na tekst zapisany w innym języku. Wreszcie, budowanie modeli języka pozwala wysunąć pewne hipotezy dotyczące tego w jaki sposób tworzona jest mowa przez ludzi.

III. Zastosowanie teorii informacji i statystyki matematycznej do badania języka naturalnego

Z punktu widzenia teorii informacji badanie charakterystyki pisma jest niezmiernie ważne, ponieważ jest ono przekazywane rozmaitymi kanałami łączności. Przykładowo stwierdzono nieoptymalność kodu telegraficznego Morse´a, proponując w zamian kodowanie na podstawie częstości występowania liter w danym języku (np. metoda Shannona-Fano) lub co jest tego uogólnieniem – kodowanie blokowe, czyli kodowanie nie oddzielonych liter, a kodowanie par, trójek itd. (por. [1], [20], lub [23]).

Stopień zorganizowania języka pisanego można rozważać na dwóch jego poziomach: morfologicznym i semantycznym. W ujęciu morfologicznym za najmniejszy nośnik informacji można uważać znak alfabetu. Oczywiście jest to uproszczenie, które można ominąć poprzez transkrypcję fonetyczną przyporządkowując litery lub kombinacje liter w sposób wzajemnie jednoznaczny[1].

Średnia ilość informacji przypadająca na jedną literę tekstu, przy znajomości prawdopodobieństw $p_1, p_2, \ldots, p_n$ występowania liter, gdzie n oznacza ilość liter alfabetu, równa się

$$I_1 = -\sum_{i=1}^{n} p_i \log_2 p_i$$

Przyjmujemy tu założenie o niezależności pojawiania się kolejnych liter. Oczywiste jest, że średnia ilość informacji I_0, przypadająca na jedną literę tekstu, przy założeniu, że wszystkie litery występują z jednakowym prawdopodobieństwem (co zresztą nie ma miejsca w żadnym ze znanych języków) równa jest $I_0=\log_2 n$, gdyż za p_i podstawiamy

---

[1] Pojęcia z zakresu językoznawstwa oraz stronę teoretyczną odwzorowań fonetyczno-znakowych omówiono w pracy: Bańczrowski J., Pogonowski J., Zgółka T.: Wstęp do językoznawstwa, Wyd. UAM, 1982. Uwagi praktyczne umieszczone są np. w [7] oraz [8].

1/n. Wielkości I_0 i I_1 nazywamy odpowiednio entropią zerowego i entropią pierwszego rzędu. Ogólnie entropia r-tego rzędu

$$I_r = -\sum_{i=1}^{l_r} p_i \log_2 p_i \ (*),$$

gdzie: l_r – liczba związków r-literowych

p_i – prawdopodobieństwo pojawienia się w tekście związku r-literowego o numerze i.

Entropia (miara nieokreśloności) systemu zorganizowanego jest zawsze mniejsza od entropii systemu niezorganizowanego. Entropia jest zatem miarą stopnia organizacji systemu, a takim jest między innymi język naturalny. Uwzględniając wszystkie zależności między kolejnymi literami zmniejszamy wartość entropii przypadającej na literę tekstu i, co jest tego prostą implikacją, zmniejszamy nieokreśloność każdej litery.

Ponieważ znalezienie częstotliwości występowania wszystkich kombinacji pięcio, sześcio i więcej literowych w zasadzie jest obecnie technicznie niewykonalne, w 1950 roku C. Shannon spróbował ustalić wielkość I_5 i I_6 innym sposobem. Metoda Shannona opierała się na starej zabawie, w której chodzi o odgadnięcie k-tej litery tekstu przy znajomości k-1 liter poprzedzających ją.

Załóżmy, że chcemy wyliczyć I_8, tj. informację warunkową, zawartą w jednej literze tekstu przy znajomości siedmiu liter ją poprzedzających. W eksperymencie uczestniczą dwie osoby. Pierwsza czyta drugiej losowo wybrany ciąg kolejnych siedmiu liter tekstu, druga zgaduje ósmą literę, przy czym ustala się ilość prób. Doświadczenie powtarza się dostatecznie dużo razy. Granice, w których zawiera się wielkość I_8 Shannon (por. 29) określił następującym wzorem:

$$\sum_{i=1}^{n} i(q_i - q_{i+1}) \log_2 i \leqslant I_8 \leqslant -\sum_{i=1}^{n} q_i \log_2 q_i$$

gdzie: q_i – prawdopodobieństwo odgadnięcia litery w i-tej próbie,

n – ilość liter alfabetu (a zarazem maksymalna liczba prób)

Wzór ten określa granice, w których zawiera się I_8, w bardzo szerokim przedziale, nie określa stopnia ufności i nie uwzględnia wielkości próby. W miejsce wzoru Shannona można (dla dużej próby) zastosować wzór pochodzący ze statystyki matematycznej (zob. [10]), szacujący precyzyjniej I_8 i uwzględniający wielkość próby:

$$P\left\{\frac{m}{y} - U_\alpha \sqrt{\frac{\frac{m}{y}\left(1-\frac{m}{y}\right)}{y}} \leqslant I_8 \leqslant \frac{m}{y} + U_\alpha \sqrt{\frac{\frac{m}{y}\left(1-\frac{m}{y}\right)}{y}}\right\} = 1-\alpha$$

gdzie: m – ilość odgadniętych liter w y próbach

y – ilość prób

$U\alpha$ – wartość krytyczna standaryzowanego rozkładu normalnego przy poziomie istotności α.

Próby przeprowadzone przez autora tego artykułu dla ustalenia I_2, I_3, oraz I_8 dla języka polskiego dały następujące rezultaty:

$$I_2 = 3.97 \quad I_3 = 3.09 \quad I_8 = 1.35 \quad \text{dla } y = 3000.$$

Widoczna jest tu zbieżność z wynikami uzyskanymi metodą badania częstości połączeń

międzyliterowych (zob. tablicę I). Uwzględniając zależności statystyczne w coraz dłuższych grupach liter otrzymujemy coraz lepsze przybliżenia poprawnych morfologicznie wyrażeń danego języka, gdyż entropia dla coraz dłuższych związków szybko maleje. Bauer i Goos w [4] podają, że dla języka niemieckiego, w wyniku przeprowadzonych obliczeń, kodując całymi wyrazami otrzymano I=1.6 bita/znak. Doświadczenia językoznawców wykazują, że poczynając od k=15, I_k praktycznie przestaje maleć, stając się stałe, tzn. $I_{15} \approx I_{16} \approx \ldots \approx I_\infty$, gdzie $I_\infty = \lim_{k\to\infty} I_k$ jest realną informacją przypadającą na literę tekstu. Dla języka rosyjskiego $I_8 = 1.36$, a I_∞ oceniana jest jako wielkość z przedziału (1.0, 1.2).

Tabela 1

Entropia	I_0	I_1	I_2	I_3
język angielski	4.76	4.03.	3.32	3.10
język niemiecki	4.76	4.11	–	–
język polski	5.04	4.52	4.02	3.28
język rosyjski	5.00	4.35	3.52	3.01

Entropia rzędów od 0 do 3 dla wybranych języków[2].
(symbol "-" oznacza brak danych)

Ponieważ litery występują z różnymi częstościami a litery poprzedzające mają istotny wpływ na częstości następnych, rzeczywista informacja I_∞ zawarta w literze bliska jest 1. Wobec tego każda litera zawiera $I_0/I_\infty \approx 5$ razy mniej informacji niż mogłaby zawierać, gdyby wszystkie litery były równoprawdopodobne i niezależne.
Jeśliby udało się zatem zakodować tekst polski w taki sposób, aby wszystkie symbole kodu były równoprawdopodobne i niezależne, to taki tekst okazałby się pięciokrotnie krótszy od rzeczywistego. Wprowadza się dlatego specjalną wielkość: nadmiar języka (redundancja):

$$R = 1 - \frac{I_\infty}{I_0}$$

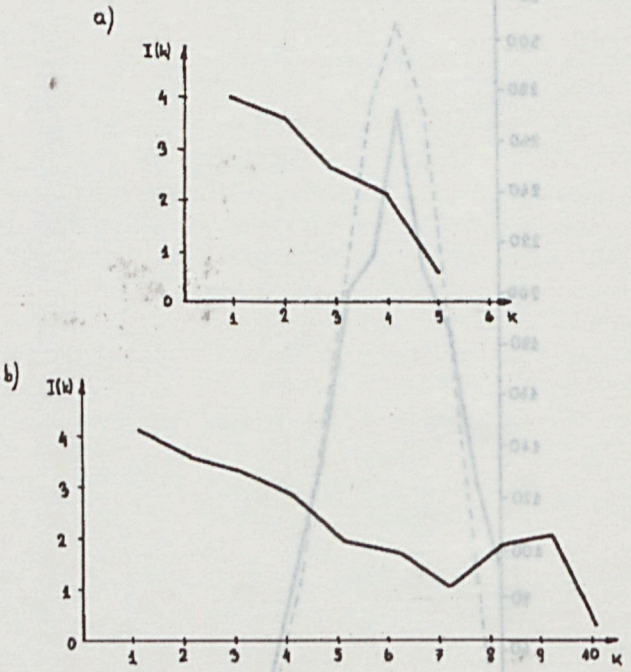
Przykładowo dla języka polskiego $R = 1 - 1/5.04 \approx 0.8$
Wynika z tego, że około 80% liter dowolnego tekstu polskiego jest „zbyteczne". Redundancja wszystkich współczesnych i starożytnych (poznanych) języków naturalnych jest prawdopodobnie nie mniejsza od 60%. Stanowi to pewnego rodzaju barierę użycia metod statystycznych dla modelowania związków międzyliterowych dowolnego rzędu. Jednakże oszacowania dokonane przez Kołmogorowa dla poezji rosyjskiej wykazały, że wypełnienie warunków, nakładanych formą poetycką, zmniejsza „nieokreśloność" jednej litery tekstu do mniej więcej połowy wartości I, obliczonej dla

---

[2] Dane dla języka niemieckiego pochodzą z [4], dla języka rosyjskiego i angielskiego z [29], natomiast dane dla języka polskiego pochodzą z obliczeń własnych wykonanych przy zastosowaniu wzoru (*).

przeciętnego utworu literackiego. Zrozumiałe jest także, że ilość informacji przypadająca na literę tekstu będzie mniejsza dla tekstów technicznych czy naukowych, niż dla języka potocznego. Każdy bowiem język naukowy i techniczny charakteryzuje się stosunkowym ubóstwem słownika oraz występowaniem wielkiej ilości specjalnych terminów i zwrotów.

Interesujące jest również pytanie: jaki jest rozkład informacji wewnątrz słowa, tj. które litery w słowie niosą większą, a które mniejszą informację? To zagadnienie także można rozwiązać za pomocą doświadczenia ze zgadywaniem liter. Jeśli k−ta litera w słowie (oczywiście k−ta od lewej) została odgadnięta w l przypadkach na sto, to średni nadmiar k−tej litery $R(k) = 1 - \frac{I(k)}{I_0}$, gdzie $I(k)$ oznacza informację, przypadającą na k −tą literę słowa. Stąd otrzymujemy $I(k) = I_0(1-R(k)) = 5.04\,(1-1/100)$.



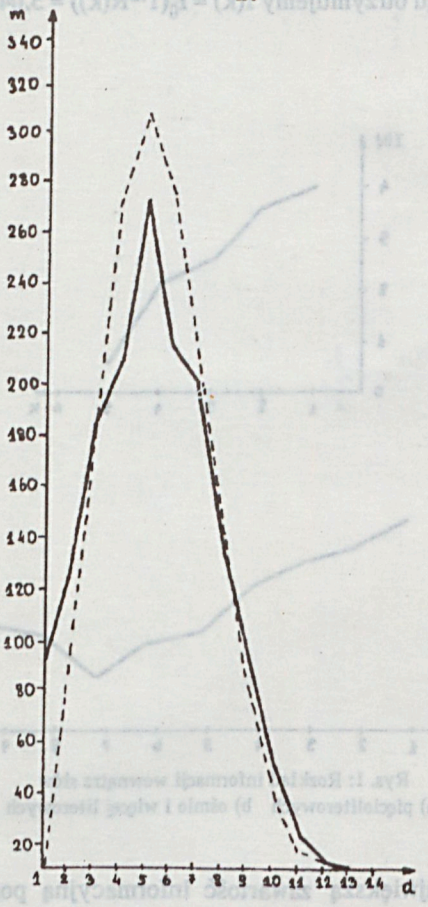
Rys. 1: Rozkład informacji wewnątrz słów
a) pięcioliterowych b) ośmio i więcej literowych

W słowach krótkich największą zawartość informacyjną posiadają pierwsze litery słowa, co jest wykorzystywane przy układaniu skrótów: „zob.", „inż.", „prof.", „itp.". W dłuższych słowach rozkład informacji ma nieco inną postać (por. rys. 1), co tłumaczyć należy zwiększeniem nieokreśloności wyboru liter w związku z rozmaitością możliwych przyrostków i zakończeń słów. Znajduje to zastosowanie przy układaniu skrótów postaci: „W-wa" (Warszawa), „k-ny" (kolejny), „w-pa" (wyspa), to jest w

zachowaniu w dłuższych słowach jedynie początku i końca, czyli części niosących najwięcej informacji.

W ramach niniejszej pracy przeprowadzono próbę ustalenia rozkładu liczby liter w słowach polskich. W losowo wybranej 1600 elementowej próbie zanotowano średnią liczbę liter wyrazu $\bar{x}=5.42$, a więc oszacowano, że przeciętny polski wyraz ma od pięciu do sześciu liter. Dla porównania, średnia liczba liter wyrazu w języku niemieckim wynosi 5.93, a w języku angielskim 4.50. Stwierdzono ponadto ogromną zbieżność rozkładu w owej próbie z liczebnościami wynikającymi z przyjęcia rozkładu Poissona, przemnożonego przez liczbę elementów próby, czyli przez 1600 (czego ilustrację zawiera rys. 2). Dla rozkładu Poissona funkcja prawdopodobieństwa wyraża się wzorem

$$P(d,\lambda) = \frac{\lambda^d}{d!} e^{-\lambda}$$



Rys. 2: Rozkład liczby liter w słowach polskich w losowej 1600 elementowej próbie (oznaczony linią ciągłą). Linią przerywaną oznaczono liczebności wynikające z przyjęcia rozkładu Poissona (dla $\lambda = 4.42$), przemnożonego przez liczbę elementów próby.
Oznaczenia: m– liczba wyrazów d-literowych
d– ilość liter wyrazu

Wzór ten podaje prawdopodobieństwo tego, że wylosowany z tekstu w języku, w którym przeciętny wzór posiada $\lambda+1$ liter, wyraz, będzie miał d liter. Wartość parametru λ jest mniejsza od $\bar{x}$ o 1, gdyż wyraz 0–literowe nie istnieją, a rozkład Poissona jest określony dla $d > 0$ oraz dla $d = 0$.

Zastosowanie statystyki do badania języka jest możliwe dzięki temu, że dostatecznie długi tekst musi zawierać powtarzające się elementy: litery, słowa, formy gramatyczne itd. Badanie powtarzania się tych elementów i częstości, z jakimi występują w tekście, pozwala na znalezienie prawidłowości statystycznych, dotyczących języka jako całości. Niezmiernie użytecznym prawem statystycznym jest prawo Zipfa. Rozpatruje się tekst zawierający N słów, przy czym N jest wystarczająco dużą liczbą; buduje się tzw. słownik frekwencyjny tego tekstu, tj. wypisuje się z tekstu wszystkie różne od siebie słowa i układa je w kolejności zależnej od tego, z jaką częstością wystąpiły one w tekście. Ostatecznie każdemu słowu przypisuje się jego numer k w słowniku frekwencyjnym i prawdopodobieństwo pojawienia się tego słowa w tekście. Według prawa Zipfa prawdopodobieństwo to kształtuje się w przybliżeniu jak

$$p_k = ck^{-\gamma}$$

Stwierdzono, że stałe c oraz γ są bliskie odpowiednio 0.1, 1.0; zależą one w pewnym stopniu od rodzaju tekstu i od języka, w którym tekst jest napisany. Im większy jest numer słowa w słowniku frekwencyjnym, tym mniejsze jest prawdopodobieństwo napotkania tego słowa. Z powyższego wzoru wynika, że chociaż pełny słownik dowolnego języka zawiera setki tysięcy słów, to na tysiąc najczęściej używanych słów przypada około 80% tekstu, co jest wykorzystywane przy układaniu słowników dla przekładu maszynowego lub przy ustaleniu zbioru słów dla podręczników do nauki języków obcych.

Metody statystyczne celowe są także wtedy, gdy chodzi o określenie czasu, w którym powstały, przez rozszczepienie jakiegoś pierwotnego jednolitego języka, języki spokrewnione ze sobą. M. Swadesh stworzył metodę opartą na tym, że w każdym języku istnieje tzw. podstawowy zasób słów; wchodzą doń między innymi nazwy elementarnych czynności (np. patrzeć, chodzić, spać), przedmiotów (np. kamień, woda, słońce), zjawisk przyrody (wiatr, ogień, tęcza) itp. Obliczenia wykazały, że tempo zmian podstawowego zasobu słów w ciągu długich okresów czasu jest w przybliżeniu jednakowe.

Mianowicie po tysiącu lat z podstawowego zasobu słów zachowuje się około 86% słów. Załóżmy, że w ciągu czasu dt w podstawowym zasobie ulegnie zmianie dN słów z ogólnej liczby N; liczba zmieniających się słów ze względu na jednostajną szybkość zmiany ich zasobu podstawowego jest proporcjonalna do czasu dt oraz do ogólnej ilości słów N, czyli $-dN = \lambda\, Ndt$ (znak minus oznacza, że chodzi o zmniejszanie się podstawowego zasobu słów), gdzie λ oznacza współczynnik proporcjonalności. Przez całkowanie otrzymujemy $N = N_0 e^{\lambda t}$, gdzie stała całkowania N_0, to pierwotna ilość słów należących do podstawowego zasobu. Ponieważ $N = N_0$ dla $t = 0$ więc $f = N/N_0$ to część słów, która zachowała się bez zmian w zasobie podstawowym w ciągu czasu t. Stąd otrzymujemy $t = -\frac{f}{\lambda} \ln \frac{N}{N_0}$, ale przy $t = 1$ (za jednostkę czasu przyjmujemy tysiąclecie) $\frac{N}{N_0} = f_0 = 0.86$ Dlatego $\lambda = -\ln f_0$ oraz $t = \frac{\ln f}{\ln f_0}$

Np. rozpatrujemy podstawowy zasób słów jakiegoś języka starożytnego, składający się z 200 słów, oraz składający się z odpowiednich 200 słów słownik jego współczesnego potomka, i jeśli 60% słów obu tych słowników jest identyczne lub prawie identyczne, to zgodnie z otrzymamym wzorem stwierdzamy, że od czasu istnienia starożytnego języka minęło $t = \frac{\ln 0.60}{\ln 0.86} \approx$ 3 tysiące lat.

Zazwyczaj trzeba jednak porównać nie starożytny język z jego potomkiem, lecz dwa współczesne języki posiadające wspólnego przodka (np. polski i rosyjski). Jeśli w każdym z tych języków część zachowanych słów należących do podstawowego zasobu (w porównaniu z językiem - przodkiem) równa się f, to jak pokazują rozważania oparte o rachunek prawdopodobieństwa, część F słów wspólnych dla podstawowych zasobów tych języków równa jest obecnie f^2. Stąd otrzymujemy wzór na czas jaki upłynął od rozdzielenia się języka przodka na dwa języki: $t = \frac{\ln F}{2 \ln f}$.

Wzór ten został potwierdzony przez liczne badania historyczne. Należy również zwrócić uwagę na to, że podobne badania statystyczne są niezwykle przydatne do ustalania autorstwa danego utworu literackiego (zob. [3], [19], [24] lub [25]).

IV. Analiza związków morfologicznych języka naturalnego i próba ich syntezy

Dla potrzeb tego rozdziału przyjmijmy poniższą definicję:

DEF.: Język to układ przechodzący ze stanu A_0 w inne stany. Ciągi elementów wytworzonych przy dozwolonych przejściach, tworzą zbiór ciągów dopuszczalnych „języka" zadanego układem. Ze stanu A_0 przechodzi się przez wiele położeń pośrednich, aby na koniec znaleźć się w stanie A_k (po k przejściach), gdzie proces kończy się [3].

Przy użyciu związków N–literowych język–algorytm, przechodząc ze stanu A_1 w stan A_{i+1}, produkuje 1 literę tekstu na podstawie znajomości N–1 liter poprzednich, czyli jakby posiadał N–1 literową pamięć. Oczywiście pierwsza produkcja algorytmu to N liter.

Dla ustalenia wpływu redundancji na możliwości syntezy tekstu na poziomie związków literowych opracowano i wykorzystano algorytm:

Algorytm I

Wstępnym etapem konstrukcji algorytmu było wyznaczenie tablicy częstości występowania połączeń dwuliterowych. Tablicę tę ustalono badając ponad 4500 słów z wybranych tekstów książkowych. Oznaczmy przez $f_{i_j, i_{j+1}}$ częstość połączenia litery o i_j-tym numerze w alfabecie z literą o i_{j+1}-szym numerze w alfabecie, gdzie i_j=1, ..., n dla j=1, ..., k . Symbol j oznacza, że litera jest j-tą literą wyrazu k-literowego. Symbol n natomiast oznacza ilość liter w alfabecie.

---

[3] Pojęcia teorii automatów i lingwistyki matematycznej ujęto np. w pracach [5] lub [16].

Istota algorytmu:

Chodzi o wyznaczenie L wyrazów k-literowych, dla których wyrażenie $\prod_{j=1}^{k-1} f_{i_j, i_{j+1}}$ jest największe. Program tworzy L wyrazów o największej wartości iloczynu częstości połączeń międzyliterowych, które to iloczyny przyjęto za przybliżenie „prawdopodobieństwa" wystąpienia danego wyrazu w słowniku języka polskiego.

Na 400 wygenerowanych przez program wyrazów pięcioliterowych, 72 z nich to wyrazy występujące w języku polskim, co stanowi 18% wszystkich wyrazów. Tak więc sprawność programu wynosi 18%. Przeciętna sprawność dla związków k-literowych e_k dana jest wzorem

$$e_k = \frac{I_0 - I_k}{I_0}.$$

Za lewą stronę wzoru wstawiamy iloraz 1/L gdzie 1 oznacza ilość wyrazów istniejących w języku polskim spośród wszystkich L wyrazów wygenerowanych przez program. Następnie przekształcamy wzór do postaci $I_k = I_0 (1 - 1/L)$ i otrzymujemy 5.04 (1 – 0.18)= 4.13. Wynik 4.13 pokazuje, że program w dużym stopniu wykorzystał możliwości tworzenia wyrazów poprawnych, gdyż możliwości te ograniczone są od góry entropią I_2=4.02. Należy przypuszczać, że duży wpływ na efektywność algorytmu miałoby wprowadzenie danych uzyskanych z wyspecjalizowanego języka. Przy danych częstościach połączeń dwuliterowych istota algorytmu sprowadza się do „wyłapania" możliwie największej liczby „błędnych" związków wyższych rzędów.

Np. W związku SZI; związki składowe SZ oraz ZI posiadają duże częstości występowania w języku polskim, ale ich połączenie SZI w języku polskim pojawia się niezmiernie rzadko.

Nie uwzględnienie tych zależności powoduje znaczne pogorszenie wyników. Oczywisty jest fakt, że wyrazy o różnych funkcjach językowych (rzeczowniki, czasowniki itd.) posiadają różną strukturę międzyliterowych. Tak więc ograniczenie się do generowania np. rzeczowników powoduje spadek „nieokreśloności" znaku i ułatwia wyszukanie wspomnianych „błędnych" związków.

W niniejszej pracy zaproponowano następujący algorytm syntezy języka na poziomie związków międzyliterowych:

Algorytm II (opis)

Dla ustalonej dwuwymiarowej tablicy częstości połączeń 2–literowych algorytm generuje takie słowa N–literowe, że wartość iloczynu częstości połączeń 2–literowych jest większa od ustalonej a priori liczby. Dla każdego uzyskanego słowa określa się czy istnieje ono w słowniku danego języka, czyli czy jest ono poprawne.

Następnie wydziela się zbiory wszystkich połączeń w wyrazach poprawnych i połączeń w wyrazach niepoprawnych. Wartości elementów tablicy częstości połączeń, odpowiadające połączeniom, których procentowy udział w wyrazach niepoprawnych w stosunku do całej liczby ich wystąpień (tak w poprawnych jak i w niepoprawnych wyrazach), będzie najwyższy zostają zmniejszone o ustaloną wielkość. Następnym krokiem algorytmu jest obniżenie dolnej granicy iloczynu, powyżej której mają znaleźć się wyrazy poprawne. Gdyby granica ta nie została obniżona, to w następnej iteracji w

zbiorze uzyskanych wyrazów znalazłyby się wyrazy z poprzedniej iteracji, uszczuplone o wyrazy, które zawierały połączenia „ukarane". Jeżeli w zbiorze wygenerowanych wyrazów są niepoprawne, to należy rozpocząć następną iterację. Skuteczność tego algorytmu opiera się na tym, że zmniejszenie częstości występowania „szkodliwych" połączeń powoduje zmniejszenie prawdopodobieństwa pojawienia się następnej iteracji wyrazów niepoprawnych zawierających owe połączenia. Załóżmy, że pierwotna struktura tablicy połączeń międzyliterowych została ustalona na podstawie słów polskich. Możliwa jest sytuacja, że użytkownik programu napisanego w oparciu o algorytm II po kilku lub kilkunastu iteracjach, uznając za poprawne te wyrazy, które istnieją w słowniku języka angielskiego, ostatecznie otrzyma wyłącznie wyrazy (poprawne) angielskie.

Algorytm II można zastosować np. do ustalania słownika pojęć dla nauki języka obcego, dzięki temu, że wygenerowane słowa są łatwo przyswajalne, co z kolei związane jest z tym, że związki międzyliterowe jako kolejne produkcje algorytmu są w większości unikalne (dany związek pojawia się najczęściej tylko w jednym słowie). Jak wykazały testy realizacji algorytmu II, ostateczny układ tablicy połączeń międzyliterowych pozwala na to, że niektóre produkcje (przejścia do następnej litery wyrazu) algorytmu nie są jednoznacznie określone, a mimo to prowadzą do poprawnych wyrazów. Dla wyrazów sześcioliterowych maksymalna liczba wyrazów, dla których ich połączenia literowe są jednoznacznie określone i dla n=28 (tj. uwzględniając w danych 28 liter alfabetu) wynosi dla tablicy trójwymiarowej, operując oczywiście związkami 3–literowymi, 6227, a dla tablicy dwuwymiarowej– 189.

Innym kierunkiem zastosowań tego rodzaju badań nad językiem może być weryfikacja hipotezy, że dany wyraz czy zwrot jest słowem czy zwrotem polskim. Znając częstości połączeń międzyliterowych dla kilku języków, możliwe jest ustalenie z jakiego języka dany wyraz, funkcjonujący w naszym języku, jest zapożyczony. Chodzi tu oto, z jakim prawdopodobieństwem wyraz ten pochodzi z danego języka.

Przykład wykorzystania opisanych metod dla języka naturalnego:

Dysponujemy dwuwymiarową tablicą $F=[f_{ij}]$, gdzie i, j=1, ..., n, częstości występowania połączeń dwuliterowych. Załóżmy, że dany jest wyraz ABCDE i z pewnego powodu brakuje w nim litery oznaczonej przez C. Naszym zadaniem jest zastąpienie niewiadomej C właściwym znakiem. Otóż najrozsądniej będzie wziąć ten znak, dla którego zachodzi

$$\max_{C\in\Omega}\{f(B, C)*f(C, D)\}$$

gdzie Ω jest zbiorem znaków alfabetu, a f(i, j) –częstością występowania związku dwuliterowego, którego pierwszym elementem jest i, a drugim j.

Jeżeli tablica F jest trójwymiarowa, to szukamy

$$\max_{C\in\Omega}\{f(A, B, C)*f(B, C, D)*f(C, D, E)\}.$$

Stwierdzono, że metoda ta aczkolwiek często prowadzi do poprawnych odpowiedzi, to na skutek znacznej entropii drugiego i trzeciego rzędu nie da się zastosować do automatycznego (komputerowego) zgadywania niewiadomych znaków. Natomiast odpowiedzi w postaci zbioru alternatywnych możliwości gwarantują poprawną odpowiedź zawsze.

Jako przykład innego algorytmu analizującego związki międzyliterowe można podać algorytm (morfologiczny) Suchotina. Jest to najbardziej znany algorytm tego typu, który z reguły prowadzi do poprawnych wyników, lecz z uwagi na olbrzymią liczbę obliczeń i absorbcję pamięci maszyny cyfrowej, jest trudny do zastosowania na niej.

Algorytm morfologiczny Suchotina (cyt. za [25])

Suchotin zaproponował następujący algorytm podziału tekstu na segmenty, czyli ciągi liter:

Załóżmy, że dany jest w tekście ciąg literowy ABCDEF.

Wszystkie możliwe segmentacje tego ciągu na dwa podciągi przedstawiają się tak:

A–BCDEF
AB–CDEF
ABC–DEF
ABCD–EF
ABCDE–F

Zatem wskutek podziału ciągu m elementów (m=6) na dwa podciągi otrzymuje się m–1 podziałów, w wyniku których powstaje 2(m–1) podciągów. Każdy z otrzymanych 2(m–1) podciągów ma w tekście jakąś częstość występowania. Dla 10 ciągów składowych liczymy w tekście 10 różnych częstości otrzymując:

A	f_1
AB	f_2
.	.
.	.
.	.
f	f_{10}

Licząc średnią częstość ciągu składowego $\bar{x} = \sum_{i=1}^{10} f_i/10$ otrzymujemy „stałość" ciągu ABCDEF. Zdaniem Suchotina częstość f_1 wskazuje na udział podciągu A w ciągu ABCDEF; jest to jakby warunkowe prawdopodobieństwo występienia ciągu ABCDEF przy założeniu, że pojawił się w tekście podciąg A. To samo dla f_2, f_3, ..., f_{10}. Każdy z podciągów wskazuje swoją częstością na prawdopodobieństwo pojawienia się całego ciągu ABCDEF[4]).

4) Tak rozumiana stałość ciągu może prowadzić do błędnych wyników ponieważ średnia $\bar{x}$ nie uwzględnia rozproszenia związków np. podciąg A może pojawiać się bardzo często, a pozostałe stosunkowo rzadko, co w wyniku da zawyżoną stałość ciągu ABCDEF, tzn. podciąg A jest składową wielu ciągów. Poza tym warunkowe prawdopodobieństwo wystąpienia ciągu ABCDEF przy założeniu, że pojawił się w tekście podciąg A, to nie f_1 lecz f_1/f_{11}, gdzie f_{11} oznacza częstość występowania całego ciągu ABCDEF. W rezultacie można zauważyć, że ciąg o większej stałości bynajmniej nie „przepowiada" silniej całości niż ciąg o stałości elementów mniejszej. Wzór na stałość ciągu można by uzyskać przez odpowiednie zmodyfikowanie warunków z algorytmu V.

Algorytm składa się z dwóch części. W pierwszej ustala się zbiór wszystkich ciągów liter w tekście, których f>1, oblicza się ich częstości i średnią jako stopień ich stałości. W części drugiej algorytmu dokonuje się wyboru ciągów, dla których wystąpienie jest najbardziej prawdopodobne przy pojawieniu się ich części składowych. Inaczej mówiąc, rozwiązaniem dopuszczalnym jest tu dowolny podział t tekstu na segmenty, rozwiązaniem optymalnym natomiast jest taki podział, w wyniku którego otrzyma się ciągi liter wykazujące maksymalną stałość (maksymalnie stałe następstwa liter).

Algorytm IV

Na użytek tej pracy opracowano algorytm, który na podstawie danej macierzy F połączeń 3–literowych, tworzy wyrazy o zadanej liczbie liter (większej od 3) w następujący sposób:

W kroku pierwszym algorytm dla 300 związków 3–literowych o największych częstościach tworzy związki 4–literowe w następujący sposób: do związku o indeksach I, J, K przyłącza związek o indeksach J, K, L tak, aby utworzyć związek 4–literowy o indeksach I, J, K, L. Jeżeli A (I, J, K) oznacza związek 3–literowy o indeksach I, J, K, A (I, J, K, L) związek 4–literowy o indeksach I, J, K, L, a ∇ – operacja przyłączania (konkatenacji), to działanie można przedstawić jako: A(I, J, K) ∇ A (J, K, L) → A (I, J, K, L). Algorytm w kroku pierwszym jednocześnie wybiera 200 związków 4–literowych, dla których iloczyn częstości ich dwóch składowych związków 3–literowych jest największy. W następnym kroku tworzone są związki 5–literowe przez przyłączenie do związków 4–literowych (dwustu) jednej litery ze związków 3–literowych:

A (I, J, K, L) ∇ A(K, L, M) → A(I, J, K, L, M) i wybierane jest 200 połączeń 5 –literowych o największych iloczynach częstości składowych. Postępuje się w ten sposób, aż do wygenerowania wyrazów o żądanej liczbie liter.

Algorytm V

W niniejszej pracy zaproponowano ponadto algorytm, który sprawdza dla wszystkich związków poniższe dwa warunki:

$$\frac{\sum_{L\in\Omega} f(I,J,L)}{f(I,J,K)} \leqslant \sigma \quad \text{oraz} \quad \frac{\sum_{L\in\Omega} f(L,J,K)}{f(I,J,K)} \leqslant \sigma$$

gdzie σ jest zadanym z góry parametrem. Im σ bliższe jest jedności tym większe jest prawdopodobieństwo, że związek A (I, J, K) doprowadzi nas do wyrazu poprawnego (istniejącego w języku), gdyż $p=1/\sigma$. Program, będący realizacją tego algorytmu, wyszukuje wszystkie związki spełniające którykolwiek z dwóch warunków. Istotę algorytmu można wyjaśnić na następującym przykładzie:

Załóżmy, że chcemy wygenerować słowo siedmioliterowe, zaczynające się od liter PRO. Załóżmy jeszcze, że jedynym związkiem, który łączy się z PRÓ, czyli zaczynający się od RÓ, jest RÓC, a ten z kolei jako jedyny następnik posiada ÓCH. Mamy już zatem PRÓ RÓC ÓCH, czyli PRÓCH.

Przyjmijmy, że wybór związku następnego, który ma pierwsze litery CH, nie jest jednoznaczny i że mamy : CHA f (CHA) = 2
CHN f (CHN) = 6
CHO f (CHO) = 1,

gdzie f (CHA) oznacza częstość występowania związku CHA.

Sprawdzamy który z wymienionych związków zostanie najłatwiej zaakceptowany i jaką wartość wtedy powinien przyjąć parametr σ.

$$\frac{f(CHA)+f(CHN)+f(CHO)}{f(CHN)} = \frac{2+6+1}{6} = 1.5$$

(sprawdzaliśmy ten warunek dla CHN ponieważ miał on największą częstość)
Zatem σ musiałaby być większa od 1.5, aby program wygenerował połączenie PRÔCHN i jakiekolwiek połączenie sześcioliterowe zaczynające się od PRÔ. Następnie program szuka związków zaczynających się od HN. Powiedzmy, że są to HNO f(HNO)=3 oraz HNI f(HNI)=1.

Dla HNO mamy $\frac{f(HNO)+f(HNI)}{f(HNO)} = \frac{3+1}{3} = 1.333$

Przyłączenie do PRÓCHN związku HNO nie wymaga więc podwyższenia parametru σ powyżej 1.5, czyli aby wygenerować wyraz PRÓCHNO (i jak wynika z przykładu również, aby utworzyć wyraz 7-literowy zaczynający się od PRÓ) należałoby przyjąć σ=1.5. W innych przypadkach przyjęcie tak wysokiej σ może doprowadzić do utworzenia wyrazów niepoprawnych.
Jeżeli znane są częstości f (I, J, K, L), to można sprawdzić dla nich warunek

$$\frac{f(I,J,K)+f(J,K,L)}{f(I,J,K,L)} \leqslant \sigma.$$

Innym sposobem wychwycenia wszystkich „trwale związanych" połączeń jest sprawdzenie dla wszystkich A (I, J, K) warunku

$$r = \frac{f(I,J)+f(J,K)}{f(I,J,K)} \leqslant \sigma$$

Im mniejsze jest r, tym większe jest prawdopodobieństwo, że związek jest prawidłowy statystycznie.

V. Analiza syntaktyczna języka naturalnego – algorytm syntaktyczny Suchotina

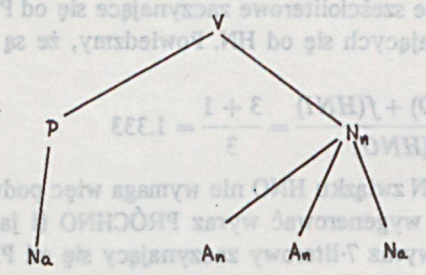
Algorytm Suchotina jest jednym z najlepszych algorytmów analizy języka na poziomie słowoform. Daje on w wyniku zbiór elementarnych reguł syntaktycznych danego języka (w części pierwszej) – otrzymuje się więc na wyjściu gramatykę języka, wyabstrahowaną z tekstu[5]. W części drugiej uzyskane reguły stosuje się do analizy konkretnego zdania.

Załóżmy, że mamy ustalić związki syntaktyczne słowoform w zdaniu „Do pokoju wpadło jasne wiosenne światło poranka". Zdanie to ma schemat PN V A_nN_nN, gdzie:P–przyimek, N–przypadek (dopełniacz) rzeczownika, V–czasownik w formie osobowej, A_n–przymiotnik, N_n rzeczownik lub zaimek w mianowniku.

---

5) Zagadnienia semantyki omówiono np. w Schaff A.: Wstęp do semantyki, PWN, 1960 oraz w sposób wysoce sformalizowany w pracy Bańczerowski J., Pogonowski J., Zgółka T.: Wstęp do językoznawstwa, Wyd. UAM, 1982.

Oczekuje się, że słowoformy wzajemnie zależne powinny pojawiać się często w tym samym zdaniu (uchwycenie składnika systematycznego). Schemat analizowanego zdania (w formie drzewa) pokazano na rysunku 3.
W wyniku pracy algorytmu winniśmy otrzymać graf spójny i acykliczny[6].
(Uwaga : dla danej słowoformy bierze się pod uwagę tylko jeden typ związku; związki zgody bez związków rzędu)



Rys. 3: Schemat zdania: „Do pokoju wpadło jasne wiosenne światło poranka" w postaci drzewa.

Punktem wyjścia algorytmu jest pełny graf zdania, w którym sieć łuków grafu to sieć wszystkich możliwych związków syntaktycznych między słowoformami tego zdania. Tylko niektóre z nich są istotne, zadanie polega więc na tym, aby usunąć z grafu łuki nieistotne, a zostawić istotne, tak aby graf był spójny i acykliczny.

Krok I.

Bierzemy tekst, na który składa się dość duży zbiór zdań prostych. Tworzymy tabelę związków między słowoformami. Każdorazowe wystąpienie związku powoduje zwiększenie się wartości w odpowiednim miejscu tabeli o jedność. W efekcie dostajemy tablicę częstości współwystępowania słowoform w zdaniu prostym, np. związek typu $N_n - V$ będzie miał dużą częstość, a rzadszy będzie związek typu $A_n - V$ itd.

Krok II.

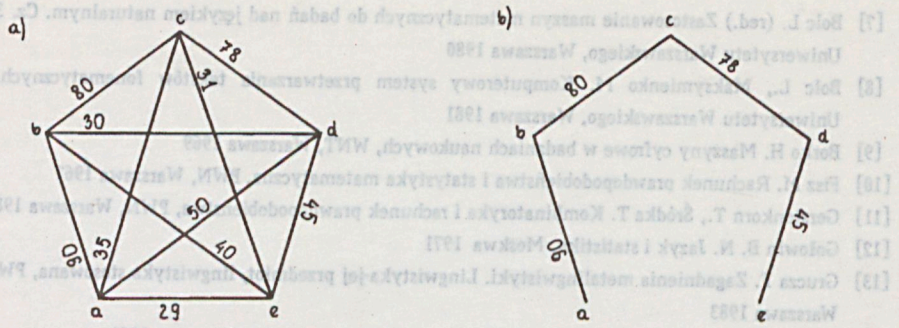
Jeżeli dla danych połączeń obliczymy średnie arytmetyczne częstości, np. $\frac{N_a V + V N_a}{2}$, to otrzymamy tabelę, w której dane liczbowe będą symetryczne względem przekątnej.

Część II (praktyczna)

Początkowy wygląd grafu naszego zdania pokazano na rysunku 4a, gdzie wielkości liczbowe umieszczone na łukach grafu przyjęto umownie. Chodzi nam o wybór takich łuków, które wskazują na związki najsilniejsze. Postępowanie ma doprowadzić do drzewa, czyli grafu spójnego i acyklicznego. Usuwamy kolejne łuki o liczbach a–e (29), b–d (30), c–e (31). Doszedłszy do kolejnego łuku d–e (45) widzimy, że jego

---

[6] Terminologię z zakresu teorii grafów omówiono np. w pracy Deo N.: Teoria grafów i jej zastosowanie w technice i informatyce, PWN, 1980 lub Korzan B.: Elementy teorii grafów i sieci. Metody i zastosowania, WNT, 1978



Rys. 4 Graf zdania „Do pokoju wpadło jasne wiosenne światło poranka"
a) pełny b) po zredukowaniu do postaci drzewa.

usunięcie izolowałoby słowoformę e i graf stałby się niespójny; usuwamy więc a–d (50) i otrzymujemy ostatecznie drzewo zdania, co ilustruje rysunek 4b.

VI. Uwagi końcowe

W pracy podjęto zasadniczo dwa zagadnienia: analizy związków morfologicznych oraz analizy związków semantycznych języka naturalnego. Przedstawione tu metody rozwiązania obu tych zagadnień z powodzeniem mogą być zastosowane w praktycznych badaniach językowych. Ponieważ zaprezentowane w pracy algorytmy analizy morfologicznej języka są nieporównywalne ze sobą ze względu na odmienne cele, które mają osiągnąć, więc mogą być traktowane niezależnie i jako metody uzupełniające, co pozwoli na traktowanie badanych struktur językowych niejako w sposób wielowymiarowy.

Podsumowując należy stwierdzić, że zastosowanie metod statystyczno – matematycznych i informatycznych pozwala na ujawnienie pewnych prawidłowości w strukturze języka. Wskazuje to na konieczność stosowania metod formalnych w językoznawstwie, a szczególnie jeśli chodzi o dialog człowieka z maszyną cyfrową.

Bibliografia

[1] Abramson N. Teoria informacji i kodowania, PWN, Warszawa 1969

[2] Baborski A., Duda M., Forlicz S. Elementy cybernetyki ekonomicznej, PWE, Warszawa 1977

[3] Bartkowiakowa A. O rozkładzie określeń w zdaniach opisowych Żeromskiego i Sienkiewicza, Zastosowania Matematyki 1962/1963

[4] Bauer F. L., Goos G. Informatyka, WNT, Warszawa 1977

[5] Blikle A. Automaty i gramatyki. Wstęp do lingwistyki matematycznej, PWN, Warszawa 1971

[6] Bolc L., Cichy M., Różańska L. Przetwarzanie języka naturalnego, WNT, Warszawa 1982

[7] Bolc L. (red.) Zastosowanie maszyn matematycznych do badań nad językiem naturalnym. Cz. 3, Wyd. Uniwersytetu Warszawskiego, Warszawa 1980

[8] Bolc L., Maksymienko M. Komputerowy system przetwarzania tekstów fonematycznych, Wyd. Uniwersytetu Warszawskiego, Warszawa 1981

[9] Borko H. Maszyny cyfrowe w badaniach naukowych, WNT, Warszawa 1969

[10] Fisz M. Rachunek prawdopodobieństwa i statystyka matematyczna, PWN, Warszawa 1967

[11] Gerstenkorn T., Śródka T. Kombinatoryka i rachunek prawdopodobieństwa, PWN, Warszawa 1980

[12] Gołowin B. N. Jazyk i statistika, Moskwa 1971

[13] Grucza F. Zagadnienia metalingwistyki. Lingwistyka-jej przedmiot, lingwistyka stosowana, PWN-PAN, Warszawa 1983

[14] Jagłom A. M. Jagłom I. M. Prawdopodobieństwo i informacja, KiW, Warszawa 1963

[15] Kierzkowski Z. Elementy informatyki, PWN, Warszawa-Poznań 1978

[16] Kowalski S., Mostowski A. W. Teoria automatów i lingwistyka matematyczna, PWN, Warszawa 1979

[17] Lewin J., Gastiew J., Rozanow J. Język. Matematyka. Cybernetyka, PWN, Warszawa 1967

[18] Lindgren B. W. Elementy teorii decyzji, WNT, Warszawa 1977

[19] Mayenowa M. R. (red.) Poetyka i matematyka, PWN, Warszawa 1965

[20] Nowakowski J., Sobczak W. Teoria informacji, WNT, Warszawa 1970

[21] Pachelski W. Fortran dla maszyn ODRA serii 1300, WNT, Warszawa 1980

[22] Plucińska A., Pluciński E. Elementy probabilistyki, PWN, Warszawa 1979

[23] Rachunek prawdopodobieństwa w zadaniach i problemach, WNT, Warszawa 1975

[24] Sambor J. Badania statystyczne nad słownictwem (na materiale „Pana Tadeusza"), PWN-PAN, Warszawa 1969

[25] Sambor J. Słowa i liczby, PWN, Warszawa 1972

[26] Targowski A. Informatyka-modele systemów i rozwoju, PWE, Warszawa 1980

[27] Turski W. M. Propedeutyka informatyki, PWN, Warszawa 1977

[28] Wolfowitz J. teoremy kodirovanija teorii informacji, Moskwa 1967

[29] Współczesna matematyka i cybernetyka, PWN, Warszawa 1969