



1. Wstęp do Inżynierii Lingwistycznej

1.1 Język, językoznawstwo i sztuczna inteligencja

Język naturalny jest chyba najbardziej wyraźnym przejawem ludzkiej inteligencji. Choć niektóre zwierzęta również wytworzyły sobie pewne systemy komunikacji¹, ich „języki” umożliwiają przekazanie bardzo ograniczonej i przede wszystkim skończonej liczby możliwych komunikatów. Nic więc dziwnego, że zaskakujący sposób komunikowania się ludzi oparty o **zasadę podwójnej artykulacji**, czyli łączenia skończonej liczby nic nieznaczących dźwięków w potencjalnie nieskończoną liczbę znaczących słów i zdań, od początku nas fascynował. Ta fascynacja nie wynika jedynie z unikalności języka dla rodzaju ludzkiego, ale także z tego że język jest nośnikiem myśli i stanowi on możliwy początek badań mających na celu rozwiązanie tajemnicy funkcjonowania ludzkiego umysłu.

1.1.1 Badania nad językiem

Choć już w czasach w których Leonidas toczył swój heroiczny bój w Termopilach, indyjski uczony Panini spisywał pierwsze na świecie opracowanie gramatyki² to początki rodzaje się w XIX w. językoznawstwa jako nauki wcale nie były łatwe. W tamtym czasie badania językoznawcze odbywały się jedynie na dwóch płaszczyznach: biologicznym i psychologicznym [3]. Badano więc fizjologiczny mechanizm mowy i działanie jego poszczególnych części w wydawaniu charakterystycznych dźwięków dla różnorodnych języków, analizowano etap rozumienia aktu mowy czy niezamierzone skojarzenia jako przyczyny językowych zmian. Jednak badanie języka samego w sobie w zasadzie nie istniało, gdyż wydawało się że jest to temat już dobrze wyeksplorowany.

Dopiero Ferdinand de Saussure (1851–1913) rozpropagował rozróżnienie pomiędzy

¹Wspominając choćby słynny pszczeli taniec https://en.wikipedia.org/wiki/Bee_learning_and_communication#Dance_communication

²Ashtadhyayi – tekst spisany w V wieku p.n.e. zawierający blisko cztery tysiące reguł gramatycznych dla sanskrytu, historycznego języka Indii; podzielony na osiem rozdziałów (tytuł oznacza „ośmioksiąg” [1]).

badaniem *parole* (mówienia) od *langue* czyli języka jako systemu (zespołu znaków) czy schematu (język jako czysta forma), który sam w sobie może (i powinien) być przedmiotem badań językoznawstwa. Najważniejsza praca de Saussure'a „Kurs językoznawstwa ogólnego” (1916)³ zainicjowała nurt badań który nazywamy **językoznawstwem strukturalnym**. Badania strukturalistów które traktowały język jako system norm społecznych umożliwiającą porozumiewanie się, zaowocowały uściśleniem wielu pojęć lingwistycznych.

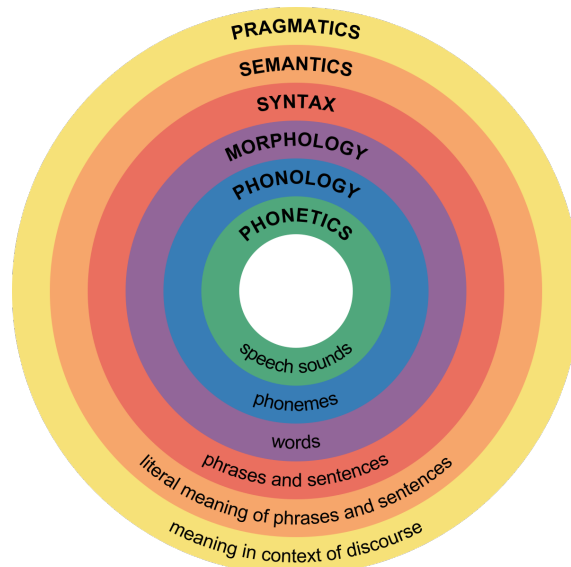
Język, w ujęciu strukturalnym system znaków i reguł ich tworzenia, możemy analizować na poziomie różnych struktur lingwistycznych, które możemy ułożyć od reprezentacji fizycznej (dźwięk, używając analogii do języków programowania: niska abstrakcja) do reprezentacji umysłowej (znaczenie, wysoka abstrakcja). Badania lingwistyczne możemy podzielić na:

- *fonetykę* i *fonologię* zajmujące się językiem na poziomie odpowiednio głosek i fonemów. Głoska to najmniejszy dźwięk języka który słyszymy, podczas gdy fonem jest zgrupowaniem głosek podobnych do siebie⁴. Jako przykład można przytoczyć słowo „rower” wymawiane zgodnie z polskimi zasadami wymowy poprzez Polaka i Francuza. Francuz wymówi słowo używając francuskiego „r” ale nadal nie zmieni to znaczenia wyrazu. Każdy z nich wyda inną głoskę, ale sprowadzi się to do tego samego fonemu⁵. Przeciwna sytuacja nastąpi gdy do głoski bezdźwięcznej w języku polskim dodamy dźwięczność np. „mak” zamienimy na „mag”. Taka zamiana sprowadzi się do odrębnych fonemów i dalej różnych znaczeń. Zwróć uwagę, że dany fonem wymawiany przez dwóch różnych ludzi, nawet stosujących prawidłową wymowę i tak brzmi różnie (czyli są to inne głoski) - jakie cechy tych dźwięków sprawiają że interpretujemy je w ten sam sposób (ten sam fonem) ? Na to pytanie odpowiada fonologia podczas gdy fonetyka opisuje sposoby produkcji tych dźwięków.
- *morfologia* - czyli dział językoznawstwa zajmujący się budową wyrazów, zarówno odmianami (robić - robię) jaki i zasadami tworzenia nowych słów (robić - odrobić). Ponieważ pojęcie wyrazu jest bardzo ogólne („don't” to jeden czy dwa wyrazy?), morfologia operuje pojęciami morfemu lub leksemu. Morfem to najmniejsza jednostka języka posiadająca znaczenie. Wyraz „dom” będzie jednym morfemem podczas gdy „domek” będzie już zawierał dwa morfemy: „dom” (miejsce w którym się mieszka) i „ek” (zdrobnienie).
- *składnia* (inaczej syntaktyka) zajmuje się budową wypowiedzi czyli łączeniem wyrazów w zdania. „Ala mieć kotka” to sekwencja słów zawierająca prawidłowo skonstruowane z punktu widzenia morfologii wyrazy. Dopiero zasady składni określają że nie jest to prawidłowe zdanie gdyż forma gramatyczna podmiotu i orzeczenia powinna być spójna.
- *semantyka* zajmuje się znaczeniem wyrazów i wypowiedzi. Bada np. relację pomiędzy znakiem a rzeczywistością czy transformacje znaczenia pewnych słów w czasie. Jednemu ze strukturalistów, Romanowi Jakobsonowi (znanego ze szkoły średniej ze swojego schematu komunikacji językowej) przypisujemy zdanie „The study of

³W rzeczywistości dzieło zostało wydane pośmiertnie przez uczniów de Saussure'a na podstawie notatek z wykładów (trwa spór badaczy w jakim stopniu jest to dzieło de Saussure'a na ile owych uczniów).

⁴Ponieważ nie jest to pełnowymiarowy kurs z językoznawstwa, pozwalam sobie tutaj i dalej na spore uproszczenia.

⁵Czyli obie głoski są alofonami



Rysunek 1.1: Główne poziomy struktury lingwistycznej.

language without meaning is meaningless”, choć nie brakuje lingwistów którzy wolą zostawić badania nad semantyką filozofom.

- *pragmatyka* opisuje stosunek między użytkownikiem języka i jego komunikatem. Jako przykład weźmy wypowiedź „Ale tutaj zimno!” wypowiedzianą przez studenta w czasie moich zajęć. Na gruncie semantyki przekazał on informację, że odczuwa chłód, podczas gdy z punktu widzenia pragmatyki zdanie to nie różniło się od „Czy można wyłączyć klimatyzację?”. Pragmatyka będzie też zajmowała się wypowiedziami które zmieniają stan faktyczny np. „ja mianuję Ciebie przewodniczącym” (przed tym zdaniem byłeś zwykłym Kowalskim, teraz Panem Przewodniczącym).

Powyższy podział nie tylko daje nam obraz jak złożonym systemem jest język, ale także pokazuje poziomy na których musimy operować konstruując systemy język przetwarzające. Zaczniemy jednak od innego pytania: Dlaczego poznanie języka i jego przetwarzanie powinno nas w ogóle interesować?

Oczywiście jednym z argumentów może być fakt, że głównym przedmiotem zainteresowania informatyki jest w istocie informacja, a ta zawsze zapisana jest w jakimś języku (jako systemie znaków). Z tego powodu poznanie struktury i możliwości języka wydaje się elementem niezbędnym aby skutecznie przetwarzać informacje, które przecież najczęściej wyraża się w języku naturalnym. Jednakże, żyjąc w erze sztucznej inteligencji, język może nas interesować z innego powodu: jako nośnik myśli, nieodzownego elementu inteligencji ludzkiej.

1.1.2 Język a myślenie

Myślenie ma charakter językowy, a sam język wpływa i kształtuje to co możemy pomyśleć i jak myślimy. Zgodnie z *hipotezą Sapira-Whorfa*⁶ każdy język rozczłonkowie rzeczywistość na pewne składowe i narzuca to rozczłonkowanie wszystkim którzy się nim posługują [8]. Nie jest więc możliwe doskonałe porozumienie pomiędzy użytkownikami

⁶inaczej prawo relatywizmu językowego, istnieje również pogląd odmienny – teorie uniwersalistyczne

różnych języków⁷, gdyż tworzone w językach słowa grupują na różne sposoby zjawiska czy przedmioty w jedną kategorię lingwistyczną.

Fakt, że w danym języku pewne elementy świata rzeczywistego zostały zgrupowane w słowa w taki a nie inny sposób jest rezultatem potrzeby wykorzystania języka do opisu rzeczywistości w której znajdują się jego użytkownicy. Z tego powodu w języku polskim występuje jedno słowo określające śnieg, a w języku eskimosów yup'ik istnieje co najmniej 30 słów określających śnieg. Śnieg spadający z nieba, śnieg leżący na ziemi, śnieg uciosany w blok, śnieg płynący na wodzie czy świeży śnieg doczekały się w nim osobnych słów [12]. Taka różnica w szczegółowości języka polskiego i języka yup'ik względem tego konceptu wynika oczywiście z drastycznie innego klimatu na obszarach zamieszkałych przez użytkowników tych języków. Fakt wprowadzenia bardziej szczegółowego rozróżnienia śniegu na różne rodzaje w języku wpływa na sposób myślenia jego użytkownika o tym zjawisku. Dla Polaka śnieg to śnieg, dla Eskimosa nie ma takiego pojęcia jak śnieg, bo śnieg sypki różni się od śniegu lepiącego się tak samo jak dla nas różnią się krzesło i fotel (choć skądinąd oba przedmioty służą do tego samego i wyglądają podobnie). Polak zapamięta że kilometr wcześniej był „śnieg”, a Eskimos że był tam „śnieg klejący się” czyli dobry do zbudowania igloo.

Możemy także zaobserwować istotne różnice w sposobach wyrażania informacji o ilości jakichś rzeczy. Jesteśmy przyzwyczajeni do języków europejskich w których istnieją określenia na dowolnie duże liczby, ale są też takie języki w których maksymalna liczba to 23 (dlaczego mielibyśmy kończyć na okrągłej liczbie w systemie dziesiętkowym?) a dalej jest już tylko „dużo”. Język pirahã, używany przez Indian w Brazylii, nie pozwala nawet na tak ograniczoną skalę liczb: istnieje jedynie słowo na „jeden” i „dwa”. Choć niektórzy badacze uważają, że w języku tym nie ma w ogóle liczebników a słowa na dwie pierwsze cyfry w rzeczywistości oznaczają „mniejszą ilość” i „większą ilość”. Wpływa to w mocny sposób na podstawowe umiejętności zliczania elementów przez użytkowników tego języka. W przeprowadzonym eksperymencie [2], rodowici użytkownicy języka pirahã byli proszeni o postawienie na stole takiej samej liczby elementów ile położyli przed nimi demonstratorzy (nie było oczywiście możliwe powiedzenie im „połóż osiem elementów”, bo słowo „osiem” nie jest im znane). Z chwilą gdy elementy demonstratora stały przed nimi, każdy potrafił postawić odpowiadającą liczbę swoich elementów (np. przez wizualne dopasowywanie). Jednak gdy po chwili patrzenia elementy demonstratora zostały zasłonięte, aż 85% uczestników nie potrafiło wykonać prawidłowo zadania już przy dziewięciu elementach, a przy dziesięciu pomylili się już wszyscy badani. Wskazuje to na problemy z zapamiętaniem konkretnej małej liczby przez użytkowników pirahã, co równocześnie jest trywialne dla przeciętnego sześciolatniego dziecka posługującego się jednym z języków indoeuropejskich.

1.1.3 Język a sztuczna inteligencja

Język więc nie tylko przenosi myśli, ale także je kształtuje. Nie bez przyczyny więc skuteczne przetwarzanie języka naturalnego rozumie się często jako ostateczny cel całej sztucznej inteligencji. Wystarczy chociażby spojrzeć na sposoby weryfikacji „inteligencji” systemu by przekonać się, że są to zadania przetwarzania języka. Jednym z najpopularniejszych testów weryfikujących czy maszyna przejawia inteligentne zachowanie jest **test**

⁷„Światy, w których żyją różne społeczeństwa, są odrębnymi światami, nie zaś tym samym światem, tylko opatrzonym odmiennymi etykietkami.” [9]

Turinga. Test ten polega na rozmowie sędzi-człowieka z rozmówcą który może być albo innym człowiekiem albo maszyną. Komunikacja odbywa się drogą pisaną (np. w formie czatu). Jeżeli sędzia nie jest w stanie odróżnić rozmowy z człowiekiem od rozmowy z maszyną to test Turinga uznaje się za zaliczony. Należy zwrócić uwagę, że w zasadzie inteligencja systemu jest tutaj określana poprzez porównanie zdolności systemu do przetwarzania języka naturalnego ze zdolnościami człowieka. Pomimo tego klasycznie test Turinga uznaje się za test sztucznej inteligencji ogólnego przeznaczenia.

Również inne testy inteligencji maszynowej, ulepszające test Turinga, dotyczą umiejętności językowych. Za przykład może posłużyć zaproponowany w 2013 roku **test schematów Winograda**, który miał m.in. wyeliminować z natury subiektywnego człowieka-sędziego z procesu ewaluacji inteligencji maszyny. Test schematów Winograda polega na odpowiedzeniu na pytania dotyczące zdań stworzonych ze specyficznych schematów. Przykładowym takim schematem jest „Radni miasta odmówili protestującym wejścia do budynku, ponieważ oni [obawiali się/nawoływali do] przemocy.” [11]. Po stworzeniu obu zdań możemy zadać pytanie komputerowi: „Kto obawiał się przemocy?” oraz „Kto nawoływał do przemocy?”. Główna trudność zadania polega na określeniu kogo dotyczy wyraz „oni” - chociaż dla człowieka zadanie jest trywialne to dla maszyny stanowi ono wyzwanie. Oczywiście, wymaga ono zarówno wiedzy jak i zdroworozsądkowego rozumowania, niemniej jednak w gruncie rzeczy jest to problem przetwarzania języka naturalnego.

Rozwiązanie problemu skutecznego przetwarzania języka naturalnego przez komputer jawi się więc jako ostateczny cel całej sztucznej inteligencji.

1.2 Inżynieria lingwistyczna

Inżynieria lingwistyczna (IL) zajmuje się badaniem różnorodnych zagadnień związanych z przetwarzaniem języka naturalnego (czyli ludzkiego, w opozycji do np. języków programowania⁸). Konstrukcja systemów odpowiadających na pytania, prowadzących rozmowę z użytkownikiem czy umożliwiających tłumaczenie maszynowe to tylko nieliczne zastosowania badań nad inżynierią lingwistyczną.

Być może zauważyłeś, że są to zadania analogiczne do tych którymi zajmuje się przetwarzanie języka naturalnego (ang. *natural language processing*), przetwarzanie tekstu (ang. *text mining*) czy językoznawstwo komputerowe (ang. *computational linguistics*). O ile jeszcze przetwarzanie tekstu można rozgraniczyć od reszty dyscyplin zawężeniem badań do przetwarzania języka w postaci tekstu to określenie jednoznacznych różnic pomiędzy innymi terminami jest w zasadzie niemożliwe. Subiektywnie mogę jedynie zasygnalizować, że w krajach angielskojęzycznych częściej mówimy o *natural language processing* (NLP) podczas gdy w Polsce tradycyjnie używamy terminu „inżynieria lingwistyczna”.

1.2.1 Zarys historyczny

Rzeczywisty rozwój inżynierii lingwistycznej jest silnie powiązany z rozwojem samej sztucznej inteligencji. Zaczynając od lat 50 ubiegłego wieku i równolegle z popularnością systemów ekspertowych w SI, w IL rozkwiatały podejścia oparte na regułach. System przetwarzający

⁸w szczególności inżynieria lingwistyczna zajmuje się przetwarzaniem tych języków sztucznych, które przeznaczone są do komunikacji między ludźmi jak np. esperanto

tekst posiadał zestaw ręcznie zdefiniowanych reguł, które były do niego dopasowywane i następnie stosowane. Przykładem algorytmu regułowego jest chociażby słynny algorytm Porter'a do wykonywania stemmingu (czyli próby pozbycia się z tekstu odmiany) dla języka angielskiego. Algorytm składa się z całej serii reguł, ale zacytujmy tu dwie wybrane:

jeśli token kończy się na „ed” a wcześniejsza jego część zawiera samogłoskę to usuń „ed”
jeśli token kończy się na „ing” a wcześniejsza jego część zawiera samogłoskę to usuń „ing”

Algorytm przetwarzając słowo „monitored” przypasowywał do niego pierwszą regułę i zmieniał wyraz na „monitor”. Z kolei przetwarzając „sing” nie znajdzie żadnej reguły.

Systemy regułowe potrafią działać bardzo szybko i precyzyjnie, stąd do niektórych prostszych zadań, jak np. wcześniej wspomniany stemming, nadal są z sukcesem używane. Ponadto dla niektórych zadań lwią część najczęstszych przypadków można obsłużyć bardzo ograniczoną liczbą reguł. Jednakże koszt tworzenia kolejnych reguł, wymagający eksperta ludzkiego jest często drogi, a obsługa coraz to bardziej wyrafinowanych przypadków powoduje konieczność obsługi „wyjątków od reguły”. Wprowadza to do systemu regułowego coraz to bardziej skomplikowane reguły, które z czasem stają się trudne w utrzymaniu.

Kolejna fala rozwoju IL nastąpiła poprzez rozpoczęcia wykorzystywania danych do automatycznego uczenia się wykonywania zadania przez program komputerowy. Podejścia te, choć zwykle wymagają większych zasobów obliczeniowych i pamięciowych, dają szansę na obsłużenie znaczenie większej liczby przypadków, których nie trzeba ręcznie specyfikować. Ponadto można założyć, że człowiek rodzi się z tylko podstawowymi umiejętnościami uczenia się czy skojarzeń, co wystarcza do całego procesu nabywania języka. Taka analogia podejść statystycznego przetwarzania języka naturalnego (ang. *statistical NLP*) jest niezwykle atrakcyjna dla osób zajmujących się sztuczną inteligencją oraz w przyszłości daje nadzieję na osiągnięcie przez te metody poziomu operowania językiem zbliżonego do ludzkiego.

Obecnie faza statystyczna rozwoju IL przerodziła się w tzw. fazę głębokiego przetwarzania języka naturalnego (ang. *deep NLP*) w której klasyczne metody uczące się zastępuje się głębokimi architekturami neuronowymi. Podobnie jak w innych działach SI doprowadziło to do przełomowej poprawy działania mechanizmów IL na wielu zadaniach. Rozwój jest tak szybki że „najlepszy algorytm” dla zadania X potrafi się zmieniać kilka razy w miesiącu. Otwiera to dużo możliwości na nowe zastosowania mechanizmów IL w biznesie, a „State of AI Report 2019” przewiduje, że w roku 2020 fala nowych start-upów wykorzystująca najnowsze zdobycze IL zbierze ponad 100 milionów dolarów finansowania.

1.2.2 Specyfika pracy z danymi lingwistycznymi

Przetwarzanie języka naturalnego jest bardzo trudnym zadaniem z wielu powodów. Po pierwsze sam język jest bardzo złożonym systemem, posiadającym - co pokazaliśmy - wiele warstw. Nawet z pozoru łatwe i podstawowe operacje na zasobach językowych okazują się często wyzwaniem dla programisty. Za przykład weźmy podział tekstu na zdania – wydaje się, że wykrycie znaku kropki w tekście i ustanowienie w tym miejscu będzie wystarczające. Jednak szybko okaże się, że w tekście występują też liczby jak np. „1.000” czy w notacji anglosaskiej „\$4.99”. W takim razie przed podziałem sprawdźmy czy kropki

nie otaczają cyfry! Jednak w tekście występują adresy internetowe jak np. „put.poznan.pl” - sprawdzamy czy po kropce jest spacja. Ale za raz co ze skrótami jak „np.”? Stwórzmy listę skrótów? Co z sytuacją gdy ktoś po kropce zapomniał wstawić spacji? Co jeśli ktoś pisząc w pośpiechu pominął kropkę? Już dla rozwiązania takiego „trywialnego” zadania, w praktyce możemy używać wcale nietrywialnego systemu regułowego czy systemu uczącego się.

Kluczowym problemem w przetwarzaniu języka jest jego **niejednoznaczność** (ang. *ambiguity*) z którą ludzie radzą sobie używając wiedzy zdroworozsądkowej czy wiedzy ogólnej. Kiedy mówisz „Barcelona” masz na myśli: miasto albo prowincję w Hiszpanii, duże miasto w Wenezueli (liczbą mieszkańców zbliżone do Szczecina), klub sportowy (a jeśli tak to ten w Hiszpanii czy w Ekwadorze?), przewodnik Lonely Planet o takim tytule czy może singiel grupy Pectus? Nie jest to niestety zjawisko, które ma miejsce tylko dla takiego szczególnie wybranego słowa - jest to zjawisko powszechne! Badania przeprowadzone na najpopularniejszych czasownikach języka angielskiego (stanowiących 20% wystąpień wszystkich czasowników) wykazują, że posiadają one średnio 12 znaczeń, a analogicznie wybrane rzeczowniki miały średnio 7.8 znaczeń [7].

Ponadto istnieje problem ustalania **powiązań anaforycznych** czyli słów których interpretacja zależy od kontekstu: „Mateusz prowadzi wykład. Jak ja go nie lubię!” – do kogo odnosi się wyraz „go”?⁹ Jakie jest jego znaczenie? Taką trudność zaobserwujemy przy analizie użycia wszelakich zaimków osobowych (on, ona, wy), dzierżawczych (jego, jej) czy wskazujących (ów, tamten). Inną trudność jest możliwość zmiany znaczenia wypowiedzi nawet pozostawiając wszystkie wyrazy bez żadnych zmian: po prostu dodając czy usuwając znaki interpunkcyjne jak np. w zdaniu „Let’s eat[,] kids!”.



Rysunek 1.2: Nawet znaki interpunkcyjne mogą być ważne z punktu widzenia semantyki.

Przetwarzanie języka jest trudne również ze względu na swoją złożoną konstrukcję. Z jakiego powodu, by utworzyć czas przeszły w większości czasowników należy dodać „-ed” ale „go” dziwnym trafem nie zamienia się w „goed” ale w „went”? Jak poradzić się z

⁹Lub ważniejsze z punktu widzenia zaliczenia IL: „do kogo odnosi się „ja”?”

różnymi niespójnościami zapisu wynikającymi chociażby z literówek lub po prostu braku jednego, ogólnie przyjętego zapisu danego języka¹⁰? Język ma także dziwną zarówno ciągłą jak i dyskretną naturą, gdyż z jednej strony możemy zdanie wyrazić w postaci mowy (sygnał ciągły) lub w postaci pisanej (litery, sygnał dyskretny). Przy czym w postaci mowy będzie się on miał trochę inną charakterystykę niż w postaci pisanej np. obecność poprawiania się (przerwanie wypowiedzi w środku i zaczęcie od nowa), sygnały zastanawiania się („yyy”) czy wyrażenie zrozumienia lub jego braku (periodyczne „yhym” czy „Słucham?”).

Trudność w modelowaniu języka stanowi też jego olbrzymia różnorodność, mierzona chociażby liczbą możliwych słów np. baza DeriNet[10] zawiera ponad milion leksemów słów języka czeskiego! Oczywiście większość z nich to rzeczowniki odmieniane przez przypadki czy czasowniki odmieniane przez osoby, czasy itd. czyli w praktyce możliwych do natrafienia w tekście słów w odpowiedniej formie gramatycznej jest wielokrotnie więcej. Nic więc dziwnego, że gdy kilka lat temu popularny stał się termin masywnych danych (ang. *big data*), również definiowanych jako dane o wysokiej wymiarowości, osoby zajmujące się IL odkryły że zajmowały się „big data” od zawsze.

Pewną interesującą zależność odnoszącą się do liczby słów prezentuje **prawo Heapsa**. Mówi ono, że liczbę unikalnych słów w kolekcji dokumentów możemy wyrazić wzorem

$$|V| = kN^\beta$$

gdzie N to liczba tokenów w kolekcji tekstów, a $\beta \in (0, 1)$ oraz $k > 0$ to stałe charakterystyczne dla danego języka, a także typu dokumentów (np. słownictwo w literaturze pięknej będzie z natury bogatsze). Dla języka angielskiego typowe wartości to k pomiędzy 10 a 100 oraz β pomiędzy 0.4 a 0.6.

Szybka analiza tego wzoru ukazuje, że niezależenie od parametrów k i β jest to zależność rosnąca – im dłuższy tekst tym więcej słów. Dochodzimy więc do zatrważającego odkrycia: zbiór słów zdanych uczących nie będzie się pokrywał ze zbiorem słów z danych testowych. Musimy więc wyposażyć systemy w umiejętność radzenia sobie ze słowami, których nigdy wcześniej nie widzieli. Takie słowa, które nie występowały w czasie uczenia a występują w czasie predykcji nazywamy **słowami spoza słownika** (ang. *out-of-vocabulary*, *OOV*).

Jeśli już przeanalizowaliśmy rozmiar zbioru słów, zapytajmy jest tak liczny jest zbiór wszystkich zdań? Jest on najprawdopodobniej nieskończony, co sugeruje chociażby prawo Heaps’a modelujące liczbę unikalnych słów jako stale rosnącą funkcję¹¹ w miarę jak tekst jest coraz dłuższy. Jednak nawet zakładając rozmiar słownika za skończoną, stałą wartość, liczba możliwych sekwencji słów jest olbrzymia. Zakładając, że słownik zawiera 10 milionów słów¹², które możemy dowolnie ustawić w zdanie o długości 25 wyrazów, możliwe jest utworzenie 10^{175} zdań-sekwencji. Dla porównania liczba możliwości w słynnej grze w Go to „tylko” ok. $2 \cdot 10^{170}$, co i tak jest liczbą siedmiokrotnie większą niż liczba wszystkich atomów we wszechświecie¹³. Zwróć też uwagę, że nasza analiza

¹⁰W Europie np. luksemburski ma wiele wariacji zapisu[?]

¹¹nie jest więc możliwe postawienie górnego ograniczenia na liczbę elementów (słów) sekwencji (zdania)

¹²czyli zakładamy że każde ze słów z DeriNet średnio posiada 10 form gramatycznych

¹³Założenie o możliwości ustawienia słów w dowolnej kolejności, oczywiście sztucznie zawyża liczbę możliwych zdań. Zwróć jednak uwagę, że ustalenie długości zdania na konkretną wartość pomija wszystkie zdania o mniejszych i większych długościach. Drugie zdanie z tego akapitu ma aż 34 słowa, a dwa akapity wyżej możesz znaleźć zdanie zawierające aż 44 tokeny.

dotyczyła jednego zdania, a w praktyce mamy do czynienia z całymi dokumentami! (sekwencjami zdań!)

Natura sekwencyjna języka stawia także specyficzne wyzwania modelom uczącym się. Podczas zajęć z uczenia maszynowego zwykle operowałeś na problemach klasyfikacji czy regresji, których wynikiem była jedna zmienna. Z kolei chociażby zadanie tłumaczenia maszynowego wymaga na podstawie jednego zdania wygenerowanie drugiego – wynikiem jest więc sekwencja. Wymaga to dodatkowego modelowania statystycznego i użycia bardziej złożonych metod, które należą w ramach uczenia maszynowego do dziedziny predykcji struktur (ang. *structured prediction*). Nauka IL będzie więc wymagała poszerzenia twojego wachlarza metod uczących się o nowe algorytmy oraz stanięcia przed kolejnymi wyzwaniami w uogólnianiu wiedzy.

1.3 Organizacja zajęć

Zajęcia z „Inżynierii lingwistycznej” odbywają się dwa razy w tygodniu i mają formę wykładu oraz ćwiczeń. Ponieważ jest to kurs magisterski zakłada się znajomość studenta z różnymi pojęciami i technikami poznanymi w trakcie studiów. Student rozpoczynający ten przedmiot powinien posiadać podstawową wiedzę z rachunku prawdopodobieństwa i statystyki, a także pogłębioną wiedzę z uczenia maszynowego, w szczególności z uczenia głębokiego (architektury wielowarstwowe, sieci rekurencyjne, sieci splotowe, wsteczna propagacja błędu). Dodatkowo zakłada się podstawową wiedzę z zakresu przetwarzania tekstu, ekwiwalentną do przedmiotu „Przetwarzanie i wyszukiwanie informacji” lub „Przetwarzanie języka naturalnego” (wyrażenia regularne, stemming, lematyzacja, stopwords, model bag-of-words, miary podobieństwa tekstu). Student powinien posiadać umiejętność rozwiązywania podstawowych problemów ze statystyki oraz rachunku prawdopodobieństwa, programowania w co najmniej jednym języku obiektowym wraz z odpowiednią biblioteką do uczenia głębokiego oraz pozyskiwania informacji ze wskazanych źródeł.

1.3.1 Program przedmiotu

Program naszego przedmiotu, z powodu ograniczonego czasu, nie może oddać prawdy o całej inżynierii lingwistycznej. Wszak na wielu zachodnich uczelniach IL jest pełnoprawnym kierunkiem studiów, a nie jednym przedmiotem. W dodatku, niedawne pojawianie się całkowicie nowych podejść głębokich, które zrewolucjonizowały przemysł, i fakt dokonywania przełomowych odkryć w praktycznie każdym nowym kwartale, sprawił że zawartość większość podręczników stała się przestarzała. Stanęliśmy więc przed niełatwym zadaniem zdefiniowania czego i jak powinno się uczyć. Tego zadania z pewnością nie ułatwia też gorąca dyskusja w środowisku badaczy dotycząca tego czy powinno się tworzyć modele ogólne czy modelować w nich strukturę języka.

Przy określaniu programu przedmiotu zdecydowano pominąć się całkowicie podejścia regułowe do przetwarzania języka. Nie chcę przez to powiedzieć że są one nieciekawe czy nieużywane, ale były one już omawiane na innych przedmiotach. Ponadto kluczowe zadanie tworzenia i utrzymywania samych reguł jest, w mojej opinii, bardziej zadaniem dla językoznawców komputerowych niż informatyków.

Choć być może niektórych zdziwi ten wybór, będą omawiane nie tylko metody głębokie, coraz częściej wykorzystywane w przemyśle i osiągające lepsze wyniki empirycznie, ale także klasyczne metody statystyczne. Metody statystyczne są dalece prostsze koncep-

cyjnie, często interpretowane, a niektóre z ich składowych z powodzeniem wykorzystuje się w głębokim przetwarzaniu języka naturalnego. Znajomość ich działania pozwala na lepsze zrozumienie zjawiska które się modeluje, dokładniejsze przeanalizowanie rozwiązania oraz pozwala przeanalizować różne próby rozwiązywania konkretnych problemów. Intuicje zdobyte w ten sposób nie tylko pozwalają docenić działanie metod głębokich, ale także pozwalają lepiej zrozumieć ich działanie i zdać sobie sprawę z ich zalet oraz ograniczeń. Ponadto zdecydowałem się skoncentrować wykład na przetwarzaniu języka pisanego, pomijając technologie mowy.

Dodatkowo, całkowicie zrezygnowałem z próbowaniem nadążania za najnowszymi rozwiązaniami „state-of-the-art”. Gdyby tak było to pierwszy temat tego kursu – modelowanie języka – trzeba by omawiać praktycznie przez cały czas, bo jestem przekonany, że podczas trwania tego przedmiotu zostanie zaproponowanych kilka(naście) nowych rozwiązań tego problemu. Zamiast tego, chciałbym pokazać „klasyczne” architektury sieci neuronowych do przetwarzania języka naturalnego i ogólne zasady ich konstruowania i ulepszania. Moim celem jest aby osoba kończąca ten kurs była w stanie przeczytać najnowszą literaturę w dziedzinie i w krótkim czasie zrozumieć działanie najnowocześniejszych metod dla jakiegoś konkretnego zadania, którym np. zajmować się będzie jej przyszły pracodawca. Sam przedmiot nie służy jednak do umówienia iluś tam zastosowań, ale bardziej zbudowanie zrozumienia działania metod przetwarzania języka na klasycznych problemach (choć oczywiście wybrane zastosowania również będą omawiane).

Program przedmiotu składa się z następujących modułów:

1. Statystyczne modelowanie języka i klasyfikacja tekstu
2. Semantyka dystrybucyjna
3. Wykrywanie encji nazwanych i części mowy
4. Modelowanie semantyki zdań (parsing)
5. Tłumaczenie maszynowe
6. Transfer wiedzy lingwistycznej
7. Wybrane zastosowania inżynierii lingwistycznej

Program przedmiotu może ulegać modyfikacji (pominięciu jakiś tematów) w zależności od szybkości omawiania materiału przez grupę.

1.3.2 Zasady zaliczenia przedmiotu

Zasady obowiązujące studentów są następujące:

- zajęcia (łącznie wykład i ćwiczenia) są podzielone na moduły tematyczne trwające ok. dwa tygodnie (czyli liczbę modułów należy szacować na ok. 7)
- wykład zalicza się poprzez napisanie kolokwium zaliczeniowego obejmujące wszystkie moduły
- zaliczenie poprawkowe z wykładu będzie polegało na napisaniu jednego zbiorczego kolokwium obejmującego wszystkie moduły przedmiotu
- w ramach ćwiczeń będą udostępniane studentom zadania domowe dla modułu lub grupy modułów, których termin realizacji będzie określany przez prowadzącego w chwili udostępniania zadania. Będzie to termin nie krótszy niż jeden tydzień.
- jednym z zadań domowych będzie przygotowanie prezentacji wybranego, zaawansowanego zagadnienia z Inżynierii lingwistycznej na podstawie artykułów naukowych oraz przedstawienie jej na zajęciach
- średni wynik z zadań domowych z wagą 45% oraz wynik z prezentacji z wagą 65%,

będzie podstawą do wystawienia oceny z ćwiczeń

- w przypadku nieterminowego oddania zadania domowego ocenia się je na 0%
- w ramach całego przedmiotu student może w dowolnych proporcjach wykorzystać 3 dni spóźnienia, przedłużające indywidualny termin oddania zadania domowego (jest to furtka dla osób którym akurat coś ważnego wypadnie i potrzebują więcej czasu)
- zadań domowych nie można poprawiać, a zaliczenie poprawkowe polega na dostarczeniu rozwiązań wszystkich zadań domowych, które pojawiły się w czasie semestru, a także prezentacji w formie pisemnego opracowania zagadnienia.
- w zakresie ćwiczeń i wykładów stosuje się następującą skalę ocen: od 51% - dostateczny, próg każdej następnej oceny rośnie o 10% (np. 3.5 jest od 61%)
- dopuszcza się 2 nieusprawiedliwione nieobecności studenta na ćwiczeniach, każda kolejna nieusprawiedliwiona nieobecność powoduje obniżenie oceny
- w przypadku komunikacji mailowej z prowadzącym (mateusz.lango@cs.put.poznan.pl) uprzejmie proszę o rozpoczynanie tematu maila od skrótu przedmiotu: „[IL]”. Maile (w szczególności zawierające zadania domowe, usprawiedliwienia itd.) wysłane na inny adres email¹⁴ lub bez skrótu przedmiotu mogą nie być uwzględniane (bo np. trafią do folderu SPAM).
- materiały do przedmiotu są zamieszczane na ogólnodostępnej stronie internetowej www.cs.put.poznan.pl/mlango w zakładce „Teaching”, również ew. ogłoszenia będą pojawiać się na tej stronie internetowej.
- student przyłapany na ściąganiu lub plagiatowaniu zadań domowych otrzymuje ocenę niedostateczną, niezależnie od innych ocen i bez możliwości poprawy.

Dodatkowo, zgodnie z Regulaminem Studiów Politechniki Poznańskiej:

- nieobecności studenta, w tym usprawiedliwione, przekraczające 1/3 zajęć są podstawą do niezaliczenia zajęć
- student zobowiązany jest do usprawiedliwienia u prowadzącego nieobecności na zajęciach w ciągu dwóch tygodni

1.4 Modelowanie języka

Jedną z charakterystycznych dla ludzi intuicji językowych jest umiejętność określenia, że jakieś słowo w danej wypowiedzi po prostu nie pasuje czy też źle brzmi. Na przykład, porównując zdania „Rodzicielko, jestem głodny!” czy „Mamo, jestem głodny!” od razu stwierdzimy, że pierwsze zdanie jest raczej dziwaczne pomimo tego że oba zdania niosą z grubsza to samo znaczenie, gdyż słowa rodzicielka i matka są synonimami. Inny przykład: powiemy zarówno „pszenny chleb” jak i „biały chleb” mając na myśli dość podobne znaczenie, powiemy też „biały człowiek” ale już raczej nie powiemy „pszenny człowiek”. W naturalny sposób jesteśmy wychwycić niuanse takie jak poniosły styl „rodzicielki” nieprzystający do codziennego funkcjonowania czy zdroworozsądkowy fakt, że człowiek nie może być pszenny.

To takie intuicje językowe pozwalają nam na konstruowanie poprawnych zdań bez rozważania tysiąca zasad gramatyki czy też poradzenia sobie z trudnościami w komunikacji np. z niewyraźnie wypowiedzianym czy zapisanym słowem. Ich obecność jest też wykorzystywana w zadaniach polegających na uzupełnieniu brakującego elementu

¹⁴Dla ciekawych: tak, zdarza się że student wysłał maila do mnie na adres który do mnie nie należy... i oczywiście jest zdziwiony, że go nie dostałem.

zdania, szczególnie częstych w nauce języków obcych. Mając zdanie „Głębokie __ maszynowe zrewolucjonizowało przetwarzanie języka naturalnego” czy „Do pokoju wdarł się __ mroźnego powietrza” bez problemu potrafimy je uzupełnić.

Często jednak do tego typu uzupełnianki pasuje więcej niż jedno słowo np. do zdania „Zdecydowałem __ z pracy, by mieć więcej czasu na naukę inżynierii lingwistycznej” pasują zarówno „odejść”, „zrezygnować”, „zwoľnić się” czy „wracać”. Jednak nawet tutaj możemy wskazać słowa, które pasują nam „lepiej” – czyli możemy powiedzieć, że są „bardziej prawdopodobne”. Pierwszym problemem z którym zmierzmy się w czasie tego przedmiotu to modelowanie języka (ang. *language modelling*), które polega na nauczaniu komputera takich właśnie intuicji.

Definicja 1.1 — Model języka. Model języka to rozkład prawdopodobieństwa po sekwencjach składających się z elementów (słów) ze skończonego słownika V .



Dla zachowania prostoty opisu będę pisał o modelu języka jako rozkładzie prawdopodobieństwa po zdaniach, jednak definicja modelu języka nie jest do nich ograniczona. Sekwencją słów może być kilka zdań, dokument lub nawet cała książka.

1.4.1 Zastosowania

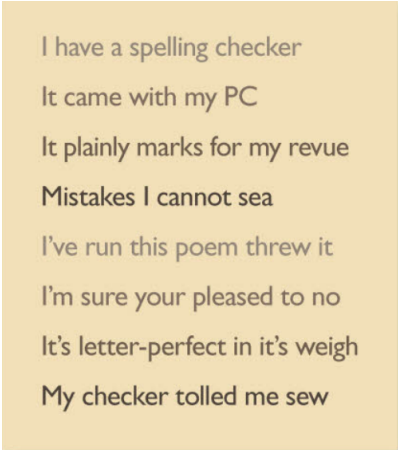
Na pierwszy rzut oka być w stanie nauczyć komputer rozkładu prawdopodobieństwa po zdaniach może się wydawać niespecjalnie przydatne – nic bardziej mylnego. Dzięki modelowi języka systemy rozpoznające mowę jest w stanie zdecydować czy zdanie które usłyszał to „Bread is made from flour” czy „Bread is made from flower”. Słowa „flour” i „flower” mają taki sam zapis fonetyczny¹⁵, więc wykrycie prawidłowego zapisu polega na znajomości kontekstu wypowiedzi czyli sprawdzenia które z tych zdań jest bardziej prawdopodobne wg. modelu języka. Analogicznie system rozpoznawania pisma odręcznego może napotkać niewyraźny wyraz i zwrócić kilka możliwości rozpoznania tekstu. Ta lista możliwości może być posortowana prawdopodobieństwem z modelu języka. Z tego powodu zadanie rozpoznania jednej pisanej litery jest czasami trudniejsze niż rozpoznanie całego słowa czy zdania.

Rozkład prawdopodobieństwa po elementach nie tylko przypisuje do nich znormalizowane do jedynki liczby, ale także oferuje możliwość losowania tych elementów z niego. Skoro nasz rozkład zdefiniowanych jest po zdaniach to elementy losowane z tego rozkładu też będą zdaniami! Model języka można więc wykorzystać do generowania tekstu: wygenerowania automatycznej odpowiedzi na mail, napisania opowiadania lub tzw. fake newsa.

W końcu, wracając do naszej początkowej motywacji polegającej na rozwiązywaniu uzupełnianek – model języka może służyć również do tego. Wyobraźmy, że chcemy policzyć prawdopodobieństwo że „ma” jest prawidłowym słowem do uzupełnienia zdania „Ala __ kota”. Korzystając z definicji prawdopodobieństwa warunkowego (i trochę nadużywając notacji) możemy je wyrazić jako:

$$P(X = ma | Ala X kota) = \frac{P(Ala ma kota)}{\sum_{X \in V} P(Ala X kota)}$$

¹⁵patrz słownik dictionary.cambridge.org. Inne homofony to np. serial i cereal czy piece i peace



I have a spelling checker
It came with my PC
It plainly marks for my revue
Mistakes I cannot sea
I've run this poem threw it
I'm sure your pleased to no
It's letter-perfect in it's weigh
My checker tolled me sew

Rysunek 1.3: Wierszyk (z błędami), który tutaj posłuży jako wyjaśnienie dlaczego modele języka są ważne w rozpoznawaniu mowy. [6]

czyli iloraz prawdopodobieństw, które możemy bezproblemowo odczytać z modelu języka (rozkładu prawdopodobieństwa sekwencji).

Ponadto modelowanie języka może mieć dla nas dużą wartość poznawczą. Całe gros statystycznych metod uczenia się polega na zbudowaniu na danych rozkładu prawdopodobieństwa oraz jego maksymalizacji. Umiejętność zamodelowania prawdopodobieństwa po zdaniach daje nam więc nadzieję na wykorzystanie nabytej wiedzy z uczenia maszynowego do przetwarzania języka naturalnego. Zadania modelowania języka nie w sposób przecenić, gdyż jest ono częścią rozwiązania większości problemów inżynierii lingwistycznej.

1.4.2 Model trzy-gramowy

Przystąpmy do opisu konstrukcji modelu językowego czyli zamodelowania rozkładu prawdopodobieństwa:

$$P(w_1, w_2, w_3, \dots, w_n)$$

gdzie $w_i \in V$, a n to długość zdania. Aby skonstruować taki model będziemy potrzebować danych, konkretnie zakładamy, że do dyspozycji mamy sporego rozmiaru korpus (ang. *corpus*, plural: *corpora*). Korpus to po prostu zbiór tekstów który służyć nam będzie do nauczania modelu języka. Taki korpus możemy prosto samemu stworzyć np. poprzez ściągnięcie wszystkich artykułów z Wikipedii, wyeksportowanie konwersacji z komunikatora internetowego czy zapisanie dużego zbioru maili. Wybór danych uczących (korpusu) ma oczywiście niebagatelne znaczenie bo model języka nauczony na treściach lektur szkolnych będzie dobrze szacował prawdopodobieństwo raczej staroświeckich zdań, a z kolei model wytrenowany na treściach maili będzie generował kolejne maile. W praktyce więc firmy często zbierają swoje własne korpusy dot. specyficznej dziedziny zastosowań (np. języka medycznego), albo – szczególnie do badań językoznawczych – używa się profesjonalnie zebranych korpusów, których autorzy starają się najwierniej zareprezentować cały język (m.in. różne rodzaje tekstów).



Najbardziej znanym korpusem języka polskiego jest Narodowy Korpus Języka Polskiego zbudowany zarówno z literatury pięknej, prasy jak i zapisów zwykłych rozmów. Długość korpusu to ponad półtora miliarda słów, jednak tylko jego mała część

jest dostępna publicznie. Cały korpus można natomiast przeszukiwać na stronie internetowej <http://nkjp.pl/>.

W jaki sposób mając dane możemy wyestymować rozkład prawdopodobieństwa? Gdyby chodziło o prawdopodobieństwo wyniku rzutu kostką i miałbyś do dyspozycji spisane wyniki całej serii rzutów to pewnie po prostu zliczyłybyś je i podzielił przez długość serii. Na przykład dla 10-elementowej serii 1, 1, 5, 2, 2, 6, 6, 3, 3, 4 stworzyłybyś rozkład:

$$P(1) = \frac{2}{10} \quad P(2) = \frac{2}{10} \quad P(3) = \frac{2}{10} \quad P(4) = \frac{1}{10} \quad P(5) = \frac{1}{10} \quad P(6) = \frac{2}{10}$$

Taki sam sposób można zastosować przy estymacji modelu języka: mając korpus złożony ze zdań s_1, s_2, \dots, s_N wystarczy zliczyć ile razy dane zdanie występuje w korpusie i podzielić przez jego długość.

Mając więc korpus składający się ze zdań: „Ala ma kota. Jurek ma kota i psa. Kamil ma psa.” otrzymalibyśmy estymaty:

$$P(\text{Ala ma kota}) = \frac{1}{3} \quad P(\text{Jurek ma kota i psa}) = \frac{1}{3} \quad P(\text{Kamil ma psa}) = \frac{1}{3}$$

Niestety, model ten jest całkowicie bezużyteczny. Po pierwsze, ze względu na różnorodność języka zdecydowana większość zdań w korpusie wystąpi tylko raz. Co oznacza, że pomijając bardzo krótkie i typowe zdania jak np. „Dzień dobry!” czy „Tak, poproszę”, model będzie zwracał jednorodny rozkład prawdopodobieństwa po sekwencjach, uniemożliwiając rozróżnienie ich (a w szczególności sortowanie ich, co było potrzebne w wymienionych wyżej zastosowaniach). Jednak dużo istotniejszy jest kolejny problem: całkowity brak generalizacji na danych testowych. Zdanie, którego nie zaobserwowaliśmy w korpusie będzie miało zawsze zerowe prawdopodobieństwo, a model generując tekst po prostu wylosuje któreś ze zdań korpusu, nigdy nie tworząc zdania nowego.

Należy także podkreślić, że uczenie w ten sposób modelu na naprawdę olbrzymim korpusie nie poprawiłoby jego przydatności do zastosowań praktycznych. Aby uzasadnić to stwierdzenie wystarczy chyba wspomnieć naszą dyskusję z podrozdziału 1.2.2, ale aby ostatecznie rozwiązać wszystkie wątpliwości przytoczmy interesujący fakt: firma Google podała że co siódme zapytanie kierowane do jej wyszukiwarki jest całkowicie nowe pomimo tego, że obsługuje ok. 3.5 miliarda zapytań każdego dnia i ponad tryliard rocznie. Oznacza to, że dysponując nawet olbrzymimi zasobami danych tej firmy, opisany wyżej model języka nauczony na zapytaniach przypisywałby zerowe prawdopodobieństwo ok. 14% zapytań!



Dla uproszczenia przez kilka następnych akapitów będziemy rozważać rozkład po zdaniach o stałej długości n .

Przystąpimy więc do wprowadzenia pewnych założeń, które spowodują ograniczenia na rozkładzie prawdopodobieństwa i wymuszą uogólnianie wiedzy. Zanim to zrobimy, zdekomponujmy rozkład prawdopodobieństwa na części używając wielokrotnie reguły łańcuchowej. Przypomnijmy, że z definicji rozkładu warunkowego mamy $P(x|y) = \frac{P(x,y)}{P(y)}$ co można przekształcić w regułę łańcuchową: $P(x,y) = P(x|y)P(y)$.

$$\begin{aligned} P(w_1, w_2, \dots, w_n) &= P(w_2, w_3, \dots, w_n | w_1) P(w_1) \\ &= P(w_3, w_4, \dots, w_n | w_1, w_2) P(w_2 | w_1) P(w_1) \\ &= P(w_4, w_5, \dots, w_n | w_1, w_2, w_3) P(w_3 | w_2, w_1) P(w_2 | w_1) P(w_1) \end{aligned} \tag{1.1}$$

Do rozpisanego wyżej modelu (nadal pełnego i równoważnego z wcześniejszym) dodamy założenie Markowa, które pozwoli na usunięcie wielu zmiennych z części warunkowej prawdopodobieństw, znaczenie je upraszczając. Konkretnie, wprowadzimy założenie Markowa trzeciego rzędu tj. założymy, że słowo na pozycji i -tej zależy tylko od słów na dwóch poprzednich pozycjach. Fakt wyboru założenia akurat trzeciego rzędu jest na razie arbitralny i będzie omówiony później.

$$\begin{aligned}
 P(w_1, w_2, \dots, w_n) &= P(w_4, w_5, \dots, w_n | w_1, w_2, w_3) P(w_3 | w_2, w_1) P(w_2 | w_1) P(w_1) \\
 &\approx P(w_4, w_5, \dots, w_n | w_2, w_3) P(w_3 | w_2, w_1) P(w_2 | w_1) P(w_1) \\
 &= P(w_5, \dots, w_n | w_2, w_3, w_4) P(w_4 | w_2, w_3) P(w_3 | w_2, w_1) P(w_2 | w_1) P(w_1) \\
 &\approx P(w_5, \dots, w_n | w_3, w_4) P(w_4 | w_2, w_3) P(w_3 | w_2, w_1) P(w_2 | w_1) P(w_1) \\
 &\approx \prod_{i=3}^n P(w_i | w_{i-1}, w_{i-2}) P(w_2 | w_1) P(w_1)
 \end{aligned} \tag{1.2}$$

By dalej uprościć zapis, ale bez wprowadzania dodatkowych założeń, ustalmy że każde zdanie na pozycjach 0 i -1 posiada specjalny token $w_0 = w_{-1} = \boxed{\text{START}}$. Ponieważ każde zdanie rozpoczyna się takimi tokenami, warunkowanie po tych tokenach nie zmienia prawdopodobieństwa np. $P(w_1) = P(w_1 | w_0 = \boxed{\text{START}}, w_{-1} = \boxed{\text{START}})$. Możemy więc zapisać powyższy model jako:

$$P(w_1, w_2, \dots, w_n) = \prod_{i=1}^n P(w_i | w_{i-1}, w_{i-2}) \tag{1.3}$$

Dzięki wprowadzeniu założenia Markowa, wyeliminowaliśmy konieczność modelowania rozkładów $P(w_i | w_{i-1}, w_{i-2}, w_{i-3}, \dots, w_1)$ zastępując je poprzez rozkłady $P(w_i | w_{i-1}, w_{i-2})$. Czy to znaczna różnica? Aby wyrazić rozkład słowa pod ustalonym warunkiem potrzebujemy $|V|$ (wielkość słownika) liczb¹⁶, niezależnie od warunku ponieważ oba rozkłady są jednowymiarowe. Różnica polega jednak na tym ile jest tych jednowymiarowych rozkładów. W modelu pełnym, bez założeń, tylko do zamodelowania ostatniej pozycji potrzebujemy $|V|^{n-1}$ takich rozkładów prawdopodobieństw (warunek jest $n-1$ elementowy) podczas gdy dla wyrażenia całego modelu z założeniem potrzebujemy tylko $|V|^2$ rozkładów. Z czego wynika, że do wyspecyfikowania całego modelu z założeniem Markowa trzeciego rzędu musimy wyestymować $|V|^3$ liczb.

Należy jednak podkreślić, że założenie Markowa jest przybliżeniem a z punktu widzenia językoznawstwa jest oczywiście wierutną bzdurą. Słowo w książce czasami zależy od słowa które pojawiło się nawet wiele rozdziałów wcześniej. Prosty przykład: „Ala mieszka z mamą i tatą. Ma też ciocię Zosię i wujka Staszka w Gnieźnie. (...) Ktoś puka do drzwi. Otwieram, a tam... ciocia __ z __!”. Niemniej jednak nasze założenie możemy motywować tym, że słowa bliżej siebie w zdaniu są zwykle *mocniej* powiązane ze sobą niż słowa dalekie. Po prostu aby umożliwić efektywne uczenie się musieliśmy pójść na pewne kompromisy.

¹⁶Uważny czytelnik zauważy, że tak naprawdę potrzebujemy $|V| - 1$ liczb. Prawdopodobieństwo ma tę własność, że sumuje się do wartości 1. Jeśli więc nauczymy się wartości prawdopodobieństwa dla wszystkich możliwych słów z wyjątkiem jednego to i tak możemy wyznaczyć to brakujące prawdopodobieństwo. Wynosi ono tyle ile brakuje sumie prawdopodobieństw pozostałych sum do wartości 1.

W końcu zauważmy, że nauka powyższego modelu wymaga od nas wyestymowania rozkładów $P(w_i|w_{i-1}, w_{i-2})$, które możemy bardzo prosto wyestymować podobnie jak wcześniej, przez zliczanie.

$$P(kota|ma, Ala) = \frac{\text{ile razy wystąpiła podsekwencja „Ala ma kota”}}{\text{ile razy wystąpiła podsekwencja „Ala ma”}}$$

Kontynuując pracę nad naszym przykładowym korpusie: „Ala ma kota. Jurek ma kota i psa. Kamil ma psa.” poniżej podano kilka przykładowych estymat:

$$P(Ala|\boxed{\text{START}}, \boxed{\text{START}}) = \frac{1}{3} \quad P(ma|Ala, \boxed{\text{START}}) = \frac{1}{1} \quad P(kota|ma, Ala) = \frac{1}{1}$$

Nasze rozważania, zaczynając od równania 1.1 zakładały, że sekwencja ma stałą i z góry znaną długość n . W szczególności gdybyśmy zsumowali prawdopodobieństwo po wszystkich o ustalonej długości otrzymalibyśmy 1 – nie jest to więc rozkład prawdopodobieństwa po wszystkich zdaniach, ale rozkład o zdaniach o z góry zadanej długości.

Zakończmy ten podrozdział zdefiniowaniem już pełnoprawnego trzygramowego modelu języka (ang. *trigram language model*), który używa założenia Markowa trzeciego rzędu ale oczywiście modeluje prawdopodobieństwo sekwencji o różnych rozmiarach. Na obsługę sekwencji o dowolnej długości pozwoli nam prosty trick: dodanie do końca każdej sekwencji specjalnego tokenu $w_{n+1} = \boxed{\text{STOP}}$. W przeciwieństwie do tokenu $\boxed{\text{START}}$, który był pewną umową notacyjną i nie zmieniał rozkładu prawdopodobieństwa, token $\boxed{\text{STOP}}$ powoduje wydłużenie sekwencji o jeden element i dodatkowe mnożenie we wzorze 1.3. Token $\boxed{\text{STOP}}$ jest więc dodatkowym, sztucznym elementem słownika V , a model zapiszemy jako

$$P(w_1, w_2, \dots, w_n) = P(\boxed{\text{STOP}}|w_n, w_{n-1}) \prod_{i=1}^n P(w_i|w_{i-1}, w_{i-2}) = \prod_{i=1}^{n+1} P(w_i|w_{i-1}, w_{i-2}) \quad (1.4)$$

Wykonanie tej prostej operacji powoduje, że prawdopodobieństwo po wszystkich możliwych zdaniach (o dowolnej długości) zaczyna sumować się do 1. Dzieje się tak, ponieważ model uczy się prawdopodobieństwa $P(\boxed{\text{STOP}}|w_n, w_{n-1})$ czyli prawdopodobieństwa zakończenia sekwencji w danym momencie. Wcześniej nie było to modelowane, bo rozkład był po sekwencjach n -wymiarowych – tutaj model sam musi sekwencję zakończyć, ucząc się charakterystyki słów będących na końcu zdania.

Wykonajmy następujący eksperyment myślowy, który mam nadzieję pozwoli nam na zrozumienie, że wykonanie tego triku daje nam rozkład po sekwencjach o różnych długościach. Wyobraźmy sobie, że każde zdanie uzupełniamy całą serią tokenów $\boxed{\text{STOP}}$, aż do osiągnięcia pewnej stałej, potencjalnie bardzo dużej, wspólnej długości M . W ten sposób rozmiar sekwencji przestaje mieć znaczenie, gdyż każde zdanie ma długość M . Modelując takie sekwencje modelem trzygramowym otrzymujemy: $P(w_1, w_2, \dots, w_M) = \prod_{i=1}^M P(w_i|w_{i-1}, w_{i-2})$, ale tak naprawdę różnorodne słowa zmieniają się do jakiejś pozycji n -tej, a potem następuje sekwencja dodatkowych tokenów. Możemy więc to zapisać jako:

$$P(w_1, w_2, \dots, w_M) = \prod_{i=1}^{n+1} P(w_i|w_{i-1}, w_{i-2}) P(\boxed{\text{STOP}}|\boxed{\text{STOP}}, w_n) \prod_{i=n+3}^M P(\boxed{\text{STOP}}|\boxed{\text{STOP}}, \boxed{\text{STOP}})$$

Jednak szybko zauważamy, że $P(\boxed{\text{STOP}}|\boxed{\text{STOP}}, \boxed{\text{STOP}})$ musi równać się 1, gdyż $\boxed{\text{STOP}}$ jest znakiem specjalnym, występującym tylko na końcu sekwencji. Jeżeli więc na poprzednich

pozycjach był $\boxed{\text{STOP}}$ to na każdej kolejnej też musi być. Takie samo rozumowanie względem $P(\boxed{\text{STOP}}|\boxed{\text{STOP}}, w_{i-2})$ wskazuje, że ono również równa się 1. Model więc sprowadza się efektywnie do modelu 1.4.

Problem 1.1 Modelowanie języka jest problemem uczenia nadzorowanego czy nienadzorowanego?

Generacja i ocena zdań modelem trzy-gramowym

Spróbujmy przypisać prawdopodobieństwo dla przykładowego zdania.

■ **Przykład 1.1** Dany jest korpus „Ala ma kota. Jurek ma kota i psa. Kamil ma psa.”. Jakie prawdopodobieństwo zostanie przypisane zdaniu $s = \text{„Ala ma kota i psa”}$ przez model trzygramowy?

$$\begin{aligned} P(s) &= \prod_{i=1}^6 P(w_i | w_{i-1}, w_{i-2}) \\ &= P(\text{Ala} | \boxed{\text{START}}, \boxed{\text{START}}) P(\text{ma} | \text{Ala}, \boxed{\text{START}}) P(\text{kota} | \text{ma}, \text{Ala}) P(\text{i} | \text{kota}, \text{ma}) P(\text{psa} | \text{i}, \text{kota}) P(\boxed{\text{STOP}} | \text{psa}, \text{i}) \\ &= \frac{1}{3} \cdot \frac{1}{1} \cdot \frac{1}{1} \cdot \frac{1}{2} \cdot \frac{1}{1} \cdot \frac{1}{1} = \frac{1}{6} \end{aligned}$$

Zdanie „Ala ma kota i psa” nie było obecne w korpusie, pomimo tego model prawidłowo potrafił uogólnić wiedzę z danych i przypisać jej niezerowe. ■

Można zauważyć, że dłuższe zdania będą miały w modelu trzygramowym tendencję do niższego prawdopodobieństwa niż zdania krótkie. Technicznie rzecz biorąc mnożymy co raz to większą liczbę czynników mniejszych równych 1. Nie jest to jednak powód do dużego niepokoju, gdyż nie oznacza to że zdania o określonej wysokiej długości mają niskie prawdopodobieństwo (jeśli by zsumować prawdopodobieństwo wszystkich zdań o danej długości). Po prostu im dłuższe zdanie tym zdania mogą być bardziej różnorodne. Jest więcej możliwości więc masa prawdopodobieństwa implicite przypisana do danej długości zdań zaczyna rozlewać się na więcej sekwencji. Ponadto, intuicyjnie spodziewamy się że powyżej pewnej długości (30?) prawdopodobieństwo coraz dłuższych zdań powinno systematycznie maleć.

Model trzygramowy może też służyć do generowania tekstu. Wystarczy rozpocząć losowanie od rozkładu $P(w | \boxed{\text{START}}, \boxed{\text{START}})$ i zgodnie z tym rozkładem wybrać pierwsze słowo tekstu w_1 . Kolejne słowo wybieramy z rozkładu $P(w | w_1, \boxed{\text{START}})$ i tak dalej, aż w końcu wylosujemy token $\boxed{\text{STOP}}$ i zakończymy proces.

■ **Przykład 1.2** Prześledźmy proces generowania tekstu z modelu opisanego w poprzednim przykładzie.

1. Losujemy z rozkładu $P(w | \boxed{\text{START}}, \boxed{\text{START}})$, które jest u nas rozkładem jednorodnym po słowach „Ala”, „Jurek” i „Kamil”. Los sprawił, że wybraliśmy Jurka.
2. Losujemy z rozkładu $P(w | \text{Jurek}, \boxed{\text{START}})$, który przypisuje 100% prawdopodobieństwa słowu „ma”
3. Losujemy z rozkładu $P(w | \text{ma}, \text{Jurek})$, który przypisuje 100% prawdopodobieństwa słowu „kota”

4. Losujemy z rozkładu $P(w|kota, ma)$, który przypisuje połowę prawdopodobieństwa słowu „i” i połowę tokenowi specjalnemu STOP. Rzut monetą wskazał na token kończący.
5. Koniec procedury. Wygenerowane zdanie to „Jurek ma kota”.

Zauważ, że model potrafił wygenerować sensowne zdanie, które nie było obecne w korpusie.

Problem 1.2 Jakie jedno zdanie należałoby dodać do korpusu z powyższych przykładów, aby model trzygramowy miał szansę wygenerować zdanie o nieskończonej długości? Jak zmienia się prawdopodobieństwo wygenerowania zdania o różnych długościach? Rozważ długości 7, 8 i 9.

Dlaczego »trzy« gram?

Liczba 3 nie jest w żaden sposób magiczna i oprócz modelu trzygramowego definiuje się też modele:

- jednogramowe (ang. *unigram*)

$$P(w_1, w_2, \dots, w_n) = \prod_{i=1}^n P(w_i)$$

- bigramowe

$$P(w_1, w_2, \dots, w_n) = \prod_{i=1}^n P(w_i | w_{i-1})$$

- czterogramowe

$$P(w_1, w_2, \dots, w_n) = \prod_{i=1}^n P(w_i | w_{i-1}, w_{i-2}, w_{i-3})$$

i więcej gramowe. W praktyce rzadko używa się modeli o licznosciach innych niż te wyżej wymienione, a wynika to z liczby n -gramowych prawdopodobieństw, które należałoby przechowywać i estymować. Jak wynika z naszej analizy na stronie 21, liczba parametrów modelu trzy-gramowego to liczba możliwych tokenów do potęgi trzeciej. Model bigramowy ma ich znacznie mniej - potęga kwadratowa, a czterogramowy znacznie więcej – potęga czwarta. Jeśli więc przy korpusie o stałej długości zaczniemy zwiększać długość n -gramu, to coraz to więcej estymat będzie równych 0, uniemożliwiając uogólnianie. Zaś niezerowe estymaty będą liczne na coraz to mniejszej liczbie przykładów. W korpusie ok. tryliona słów ze stron internetowych dostępnych w sieci (indeks firmy Google lub jego część) znajduje się 977 milionów unikalnych trzy-gramów i 1.176 milionów unikalnych pięciogramów. Biorąc pod uwagę, że przechodząc z modelu trzygramowego do pięciogramowego liczba wszystkich parametrów modelu eksponencjalnie urosła to liczba zawartych w korpusie unikalnych 3-gramów i 5 gramów tylko delikatnie się różni. I to pomimo tego, że mówimy o astronomicznie dużym korpusie: skompresowane estymaty modelu 5-gramowego zajmują aż 24 GB. Nawet używając modeli 5-gramowych na bardziej standardowych (i mniejszych) korpusach, trzeba zastosować sporo optymalizacji i pamięciowych tricków aby zmieścić je w pamięci RAM komputera. Z tego powodu, rzadko słyszy się o modelach używających czegoś więcej niż (szaleńcze) 7-gramy, a to i tak chyba tylko w celach naukowych.

Możemy mówić o pewnym przetargu pomiędzy dokładnością modelowania a jej pewnością. Im model używa dłuższych n -gramów tym potrafi uchwycić dłuższe zależności i lepiej zamodelować tekst. Jednak, w naturalny sposób coraz dłuższe n -gramy coraz rzadziej powtarzają się w korpusie przez co zwiększa się liczba estymat opartych na bardzo małej liczbie wystąpień n -grama (w szczególności na zaledwie jednym wystąpieniu). Takie estymaty są coraz mniej pewne – mają większą wariancję. Najczęściej używany model trzy-gramowy jest empirycznie dobrym praktycznym wyborem, równoważącym dokładność i pewność. Oczywiście, gdy pracujesz na dużym korpusie możesz pokusić się o skorzystanie z modelu z dłuższymi n -gramami, a z kolei pracując na małym korpusie, model bigramowy czy unigramowy może zadziałać lepiej.

Warto też wspomnieć, skąd wzięła się nazwa: modele n -gramowe. Podczas wyznaczania estymat dla np. modelu trzy-gramowego musimy zliczać wystąpienia wszystkich trójek słów występujących w tekście obok siebie. Taka n -elementowa zbitka słów czy też fragment tekstu składający się z n słów nazywamy właśnie n -gramem.

Problem 1.3 Zakładając, że w korpusie występuje $|V| = 40.000$ unikalnych słów – ile parametrów będzie miał model trzygramowy? A bigramowy i unigramowy?

Problem 1.4 Czy prawdopodobieństwa z modelu unigramowego $P(w)$ jest równe prawdopodobieństwu rozpoczęcia zdania w modelu trzygramowym $P(w_1)$? Dlaczego?

Dodatki

Materiały powtórkowe

Wprowadzanie do inżynierii lingwistycznej wraz z zarysem historycznym można znaleźć w rozdziale 1 książki [5] – darmowy dostęp online z konta bibliotecznego. Omówienie modeli n -gramowych można znaleźć w rozdziale 3 książki [4], która jest dostępna za darmo w internecie https://web.stanford.edu/~jura/slp3/old_oct19/edbook_oct162019.pdf. W szczególności osoby, które nie pamiętają podstaw przetwarzania tekstów powinny również przeczytać rozdział 2 tej książki.

Materiały dla chętnych

Dla zainteresowanych polecam wykład dr. Lery Boroditsky pt. „How the Languages We Speak Shape the Ways We Think” dostępny pod adresem <https://www.youtube.com/watch?v=iGuuHwbuQ0g> lub dla mniej wytrwałych jego krótszą wersję <https://www.youtube.com/watch?v=RKK7wGAYP6k>.

Bibliografia

- [1] Panini – Wikipedia, wolna encyklopedia. [https://pl.wikipedia.org/wiki/Panini_\(gramatyk\)](https://pl.wikipedia.org/wiki/Panini_(gramatyk)). Dostęp: 2020-01-24.
- [2] Michael C Frank, Daniel L Everett, Evelina Fedorenko, i Edward Gibson. Number as a cognitive technology: Evidence from pirahã language and cognition. *Cognition*, 108(3):819–824, 2008.
- [3] L. Hjelmslev. „Langue” i „parole”. In Halina Kurkowa i Adam Weinsberg, editors, *Językoznawstwo strukturalne. Wybór tekstów.*, pages 9–17. Polskie Wydawnictwo Naukowe, Warszawa, 1979.

- [4] Dan Jurafsky i James H. Martin. *Speech and Language Processing (3rd ed. draft, 16 Oct. 2019)*. 2019.
- [5] Uday Kamath, John Liu, i James Whitaker. *Deep Learning in Natural Language Processing*. Springer, 1st edition, 2019.
- [6] Christina Latham-Koenig i Clive Oxenden. *English File Advanced*. Oxford University Press, 3 edition, 2015.
- [7] Hwee Tou Ng i Hian Beng Lee. Integrating multiple knowledge sources to disambiguate word sense: An exemplar-based approach. In *34th Annual Meeting of the Association for Computational Linguistics*, pages 40–47, Santa Cruz, California, USA, June 1996. Association for Computational Linguistics. doi: 10.3115/981863.981869. URL <https://www.aclweb.org/anthology/P96-1006>.
- [8] Władysław Panas. Semiotyka kultury. *Znak*, 260, 1976.
- [9] Edward Sapir. *Kultura, język, osobowość*. Państwowy Instytut Wydawniczy, Warszawa, 1978.
- [10] Magda Ševčíková i Zdeněk Žabokrtský. Word-formation network for czech. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 1087–1093, 2014.
- [11] Terry Winograd. Understanding natural language. *Cognitive psychology*, 3(1):1–191, 1972.
- [12] Anthony C Woodbury. Counting eskimo words for snow: A citizen's guide. *Linguist List*, 5:1239, 1991.