Inżynieria lingwistyczna

Łukasz Kobyliński, Maciej Ogrodniczuk
Instytut Podstaw Informatyki PAN

1. Inżynieria lingwistyczna – co to takiego?

Inżynieria lingwistyczna (ang. *linguistic/language engineering*), określana także jako „przetwarzanie języka naturalnego" (ang. *natural language processing, NLP*) to dziedzina badań w informatyce związana z komputerowym przetwarzaniem (analizowaniem, generowaniem, rozumieniem) zapisów ludzkiej komunikacji – mowy, tekstu, sygnałów niewerbalnych. Z jej zdobyczy korzystamy wszyscy nie zdając sobie nawet z tego sprawy: wyszukując informacje w sieci, rozmawiając z asystentem głosowym czy przekazując tekst do tłumaczenia jednemu z automatycznych serwisów. „Proste" (z punktu widzenia użytkownika) skorzystanie z wyszukiwarki wymaga po stronie komputera wielu złożonych działań – dopasowania tekstu zapytania do zindeksowanych wcześniej dokumentów, zwykle z uwzględnieniem odmiany wyrazów czy ich tłumaczenia, uporządkowania wyników, czasem uwzględnienia niuansów semantycznych wynikających z wbudowanej w strukturę języka wieloznaczność oraz wymaganą do jego dekodowania wiedzę o kontekście i świecie. Z tego względu przetwarzanie języka jest zadaniem nietrywialnym i złożonym.

Określenie „inżynieria lingwistyczna" bywa także używane zamiennie z pojęciem „lingwistyki komputerowej" (ang. *computational linguistics*), warto jednak mieć na uwadze uwypuklenie czynnika informatycznego w pierwszym określeniu i lingwistycznego w drugim. Niedawne sukcesy komputerów w zadaniach związanych z przetwarzaniem języka, takich jak tłumaczenie maszynowe (ang. *machine translation*), asystenty głosowe (ang. *voice assistants*) czy czatboty/agenty dialogowe (ang. *chatbots*) wybiły na pierwszy plan aspekt techniczny, ukrywając lingwistykę w jego cieniu. Złożoność problemu każe jednak wierzyć, że językoznawstwo nie powiedziało jeszcze ostatniego słowa w tej globalnej rewolucji, być może najważniejszej w historii świata, której próg właśnie zdajemy się przekraczać.

2. Zadania i metody inżynierii lingwistycznej

Polem działania inżynierii lingwistycznej jest w pierwszej kolejności analiza zapisów języka – od przekształcenia sygnału akustycznego na tekst, przez wyodrębnienie z tego tekstu wielopoziomowej struktury (np. wyrazów, zdań) po przypisywanie elementom tej struktury pewnych własności (jak charakterystyka morfoskładniowa słów czy kategoryzacja nazw własnych) oraz wykrywanie związków pomiędzy nimi (np. struktur składniowych, odwołań anaforycznych itp.). Coraz częściej w analizie tej pojawiają się materiały audiowizualne, umożliwiające uwzględnienie takich elementów jak wyraz twarzy mówiącego czy wskazówki prozodyczne. Wymienione elementy procesu przetwarzania języka można także uporządkować w tradycyjnie rozumiane „warstwy" – mówimy wtedy o analizie morfologicznej, składniowej, semantycznej czy pragmatycznej. Możemy też mówić o złożonych zadaniach analizy, rozumienia i generowania tekstu działających na pojedynczych dokumentach lub ich zbiorach. Przykładami takich zadań jest tworzenie streszczeń czy wykrywanie plagiatów.

Obecnie „potokowy" sposób przetwarzania języka (od wyodrębnienia charakterystyki słów, poprzez frazy aż po związki międzyzdaniowe) jest coraz częściej wypierany przez wykorzystanie gotowych modeli językowych (ang. *language models*) wykonujących wiele zadań jednocześnie. Najbardziej efektywnym sposobem działania jest skorzystanie z modeli wstępnie przygotowanych (ang. *pre-trained*) na bazie zasobów językowych o wielkości wielu miliardów słów, a następnie dostosowanie ich (ang. *fine-tuning*) do konkretnego zadania. Modele tego rodzaju zawierają wiedzę językową zakodowaną w sposób niejawny, wydedukowaną przez model na podstawie zależności statystycznych wykrytych w dostarczonych tekstach. Z tego względu punkt ciężkości z tworzenia specjalizowanych narzędzi językowych przenosi się ostatnio na tworzenie danych językowych – opisanych lingwistycznie w pożądany sposób (np. z oznaczonymi w nich wskaźnikami emocji czy relacjami metatekstowymi), służącymi do uczenia narzędzi w sposób nadzorowany, a więc na wzorcowych przykładach. Inny scenariusz przygotowania danych językowych polega na tworzeniu ukierunkowanych zbiorów danych bez dodatkowej anotacji, które można wykorzystywać do analizy z wykorzystaniem nienadzorowanych metod uczenia maszynowego (ang. *machine learning*), a więc takich, które są w stanie samodzielnie rozpoznawać wzorce i zależności w danych, bez dostarczania wzorców przez człowieka. W tym kontekście szczególnego znaczenia nabiera potrzeba tworzenia stale aktualizowanych korpusów językowych, reprezentatywnych dla różnych gatunków języka i potrafiących odzwierciedlać najnowsze zjawiska językowe.

Nie sposób nie zauważyć też związanej z powyższym tendencji globalizacji przetwarzania języka naturalnego. Obecne wielkoskalowe modele językowe są coraz częściej wielojęzyczne i wielomodalne, co wydaje się być pożądanym kierunkiem rozwoju dziedziny mogącej wzbogacać działanie narzędzi tworzonych dla języków o mniejszej dostępności zasobów (ang. *less-resourced languages*) o wyniki badań ogólnoświatowych.

3. Zwięzła historia inżynierii lingwistycznej

Początki inżynierii lingwistycznej sięgają połowy wieku XX, kiedy to Alan Turing zaproponował znany eksperyment, w którym inteligencja maszyny oceniana była właśnie za pomocą komunikacji językowej (Turing 1950). Pierwsze wykonane w tym duchu eksperymenty dotyczyły tłumaczenia maszynowego z rosyjskiego na angielski, rozpoczynając erę systemów regułowych, których działanie opierało się na serii stworzonych ręcznie procedur. Działanie na zasadzie ręcznie tworzonych reguł w stoi w opozycji do metod uczenia maszynowego, których celem jest indukcja reguł lub (innych metod podejmowania decyzji) automatycznie, na podstawie wzorcowych danych, bez konieczności formułowania ich przez człowieka. Najbardziej znanym systemem z tego okresu jest z całą pewnością ELIZA (Weizenbaum 1966). Podwalinami dla działań informatycznych tego rodzaju stały się teorie Chomsky'ego, z gramatyką generatywno-transformacyjną na czele (Chomsky 1957). W latach 80. XX w. rozpoczęły się także prace nad regułowymi parserami (analizatorami składniowymi) wykorzystującymi gramatyki formalne (np. HPSG – *Head-driven phrase structure grammar*; Pollard i Sag 1994), algorytmami semantycznymi (jak algorytm Leska; Lesk 1986) czy wykorzystującymi teorie referencji (np. teorię centrowania; Grosz i in. 1995).

W latach 90. XX w. w związku z coraz powszechniejszą dostępnością komputerów i wzrostem ich mocy obliczeniowej nastąpił intensywny rozwój metod ilościowych, co rozpoczęło erę „statystyczną" przetwarzania języka naturalnego, w której do gry weszły algorytmy uczenia maszynowego. Jej zdobyczami były znów w pierwszej kolejności systemy tłumaczenia maszynowego takie jak Moses (Koehn 2007) opracowane na bazie ogólnodostępnych danych Parlamentu Europejskiego (Koehn 2005). Okres

ten charakteryzował się też działaniami zmierzającymi do tworzenia szeroko zakrojonych anotowanych (czyli zawierających oznaczenia dodatkowych informacji lingwistycznych) zasobów językowych służących uczeniu statystycznych systemów metodą nadzorowaną oraz powstaniem ontologii i baz wiedzy modelujących wiedzę praktyczną i ogólną takich jak np. CYC (od ang. *encyclopaedia*; Lenat 1986). Rozwój Internetu spowodował natomiast zwiększenie ilości dostępnych danych cyfrowych, co z kolei przyczyniło się do rozwoju nowych algorytmów uczenia bez nadzoru.

Rok 2010 przyniósł radykalną zmianę technologiczną, która zrewolucjonizowała całą dziedzinę. Kolejny wzrost mocy obliczeniowej spowodował upowszechnienie się metod wykorzystujących tzw. uczenie głębokie (ang. *deep learning*). Zastosowanie wielowarstwowych sieci neuronowych (ang. *deep neural networks*) w połączeniu z jeszcze większymi i jeszcze łatwiej dostępnymi zbiorami danych pozwoliło na osiągnięcie najlepszych jak dotąd wyników w wielu zadaniach przetwarzania języka naturalnego. Systemy stworzone w ten sposób przewyższają niekiedy sprawność człowieka-specjalisty.

4. Trendy w inżynierii lingwistycznej

Jednym z podstawowych zagadnień w kontekście inżynierii lingwistycznej jest kwestia reprezentacji tekstu, który ma podlegać przetwarzaniu. Przez to sformułowanie rozumiemy takie odwzorowanie znaków, słów, zdań, czy dłuższych fragmentów tekstu, które z jednej strony zachowa jak najwięcej informacji powiązanych z oryginalną jego formą, a z drugiej umożliwi przetwarzanie tej informacji przez metody obliczeniowe. Przykładem jednej z najprostszych metod reprezentacji tekstu jest metoda „*bag of words*" (BoW), która polega na tym, że zachowujemy jedynie informację o tym, jakie słowa wystąpiły w tekście, ignorując ich dodatkowe własności – pozycję w zdaniu, znaczenie, czy nawet częstotliwość występowania. Słowa wrzucamy niejako do „worka" – stąd nazwa – i przyporządkowujemy im identyfikatory, które będą mogły być wykorzystane w metodzie obliczeniowej. Taki sposób reprezentacji pozwala, na przykład, na dokonanie porównania tekstów pod względem używanego słownictwa: skoro w tekście A i B występują słowa o identyfikatorach 123 i 103, a w tekście C występują słowa o identyfikatorze 170, to być może teksty A i B są do siebie bardziej podobne niż tekst C do któregokolwiek z pozostałych.

Ten prosty sposób reprezentacji nie jest jednak najczęściej wystarczający. Po pierwsze, w takiej reprezentacji tekstu dominować będą bardzo częste słowa posiłkowe występujące w języku, a więc spójniki, zaimki, przyimki itp., które nie wydają się wnosić zbyt wiele do odwzorowania pełnej informacji zawartej w tekście. Po drugie, jeśli nie uwzględnimy krotności występowania tych samych słów w tekście, to możemy utracić ważną cechę potencjalnie odróżniającą teksty od siebie. Z drugiej jednak strony, uwzględnienie krotności niesie ze sobą problem porównywania tekstów o różnej długości – wówczas różnica w krotności wystąpień może być silniej związana z długością tekstu, a nie z jego treścią.

Pewnym rozwiązaniem tego rodzaju problemów jest wykorzystanie reprezentacji typu TF-IDF (ang. *term frequency – inverse document frequency*). W tej reprezentacji zliczamy każde ze słów występujących w tekście, a więc uwzględniamy krotność wystąpień i dzielimy ją przez liczbę wszystkich słów danego dokumentu. Tę względną częstość słów w dokumencie odnosimy dodatkowo do częstości występowania poszczególnych słów we wszystkich analizowanych dokumentach (lub innym korpusie odniesienia, na przykład ogólnym korpusie języka). Dzięki użyciu obu składowych w reprezentacji unikamy problemów, które były udziałem „worka słów", tzn. z jednej strony uwzględniamy krotność wystąpień słów, a z drugiej ignorujemy wpływ długości tekstów oraz słów nieistotnych dla treści. Dzieje się tak, gdyż wpływ słów, które występują podobnie często w badanym tekście, jak i we wszystkich pozostałych, na reprezentację

badanego tekstu będzie niewielki. Zakładając więc, że jeśli w analizowanym tekście częstość wystąpień słów nieznaczących będzie podobna jak we wszystkich innych tekstach (lub korpusie referencyjnym), nie będzie to miało istotnego wpływu na reprezentację używaną przez metodę obliczeniową.

W obu opisanych wyżej przypadkach, reprezentacja tekstu opiera się wyłącznie na poziomie jego reprezentacji ortograficznej, a więc bez jakiegokolwiek uwzględnienia kwestii semantycznych. Nie wiemy zatem, czy słowa występujące w dwóch różnych tekstach, choć mające tę samą formę ortograficzną, faktycznie odnoszą się do tego samego znaczenia. Dodatkowy problem pojawia się w przypadku analizy tekstów w językach fleksyjnych, takich jak język polski. Zliczając osobno słowa pojawiające się w tekście w różnych formach fleksyjnych różnicujemy je według kryterium najczęściej nieistotnego z punktu widzenia analizy. Pewnym rozwiązaniem tego problemu jest dokonanie wcześniejszej lematyzacji tekstu, a więc sprowadzenia słów do ich form podstawowych, co jednak redukuje reprezentację składniową zdań w tekście.

Przełomem w tym względzie była reprezentacja *word2vec* (Mikolov i in. 2013) pozwalająca odwzorować faktyczne zależności semantyczne pomiędzy słowami w postaci krótkiego wektora liczb, bez ograniczania się jedynie do statystycznego zliczania częstości słów. Dodatkowo zależności te reprezentacja „wychwytuje" samodzielnie na podstawie analizy odpowiednio dużych, nieanotowanych zbiorów danych. Wystarczy więc korpus języka o odpowiedniej wielkości, aby utworzyć reprezentację semantyczną poszczególnych słów. Często cytowaną cechą tego rodzaju reprezentacji jest możliwość wykonywania operacji na wektorach słów, których wyniki są najczęściej zupełnie zgodne z intuicją człowieka na temat ich znaczenia. Spójrzmy na przykład arytmetyki na reprezentacjach poszczególnych słów, gdzie *R(słowo)* to reprezentacja *word2vec* danego słowa.

$$R(\textit{Berlin}) - R(\textit{Niemcy}) + R(\textit{Polska}) \cong R(\textit{Warszawa})$$

$$R(\textit{król}) - R(\textit{mężczyzna}) + R(\textit{kobieta}) \cong R(\textit{królowa})$$

Przykłady te można interpretować jako zapytania o analogie: co jest dla Polski tym samym, czym jest Berlin dla Niemiec? Odpowiedź: Warszawa (stolica). Drugi przykład pokazuje, że reprezentacja ta jest również w stanie dobrze wyrazić koncepcję płci - słowo "królowa" odpowiada słowu "król", o ile "odejmiemy" od tego słowa koncepcję "mężczyzny", a dodamy "kobiety".

Możliwość wykonywania takich operacji na słowach w poszczególnych językach była punktem zwrotnym w metodach inżynierii lingwistycznej. Tworzenie reprezentacji uwzględniającej znaczenie słów w tekście stało się bowiem relatywnie łatwe. W związku z tym metody obliczeniowe, które na nich operowały mogły w dużo większym stopniu odpowiadać intuicji człowieka i jego oczekiwaniom. W ślad za metodą *word2vec* powstały inne „wektorowe" reprezentacje tekstu, które zachowały jej korzyści wprowadzając kolejne udoskonalenia. Jedną z nich jest *FastText* (Bojanowski i inl. 2017), w której jednostką podstawową są części słów (n-gramy literowe, ang. *sub-words*), a nie całe słowa. Dzięki temu możliwe jest reprezentowanie znaczenia segmentów zawartych w słowach (np. form aglutynacyjnych leksemu *być*, partykuł *-że* czy *-by* itd.), a także słów, których model wcześniej nie znał (nie znajdowały się w zbiorze treningowym). Jest to szczególnie korzystne w przypadku języka polskiego (i innych języków fleksyjnych), gdzie z jednej strony mamy do czynienia z bogactwem form słów o tym samym rdzeniu i chcielibyśmy, aby znajdowały się one blisko siebie w przestrzeni wektorowej, a z drugiej strony zależy nam również na odwzorowaniu znaczenia prefiksów i sufiksów pojawiających się w słowniku (np. jak na znaczenie słowa wpływa dodanie do niego sufiksu *-ów*).

Kolejnym przełomem był zaproponowany rok później model kontekstowy *ELMo* (ang. *Embeddings from Language Model*; Peters i in. 2018). O ile w przypadku reprezentacji typu *word2vec* mówimy zasadniczo o statycznie wyliczonych „tabelach" – wektorach reprezentujących słowa, które zostały jednokrotnie określone na podstawie dużych korpusów tekstowych, o tyle model ELMo jest w stanie tworzyć reprezentację słów na bieżąco, na podstawie kontekstu, w którym dane słowo się znalazło. Dzięki temu jeśli w tekście ta sama forma tekstowa występuje np. dwukrotnie, to reprezentacja każdego z tych wystąpień może być inna. Oznacza to również, że model może na podstawie kontekstu odróżniać homonimy. Istotność tego zagadnienia można zilustrować na przykładzie tekstów prawniczych: w tekstach tego gatunku pojawiać się może słowo *powód* w dwóch, zupełnie różnych znaczeniach (przyczyna czegoś oraz osoba wnosząca do sądu pozew). Model kontekstowy poprawnie odróżni od siebie oba znaczenia, co przełoży się na lepszą analizę tekstu.

Kolejny postęp przyniosła architektura BERT (ang. *Bidirectional Encoder Representations from Transformers*; Devlin i in. 2019), która opiera się na wykorzystaniu transformerów. To modele służące do przekształcania jednego ciągu znaków w inny (ang. *sequence-to-sequence*), oryginalnie zaprojektowane na potrzeby tłumaczenia maszynowego. Modele typu BERT trenowane są na zasadzie usuwania niektórych słów ze zdań w korpusie (zastępowania ich specjalnym znacznikiem maski), a celem modelu jest przewidzenie, które ze słów pasują do luk. Po tego rodzaju treningu z modelu usuwa się warstwę odpowiedzialną za przewidywanie słów otrzymując model tworzący reprezentację wektorową każdego ze słów. Model BERT odróżnia od ELMo możliwość dalszego dostrajania. O ile w ELMo parametry sieci są zamrożone, o tyle w przypadku modelu BERT możliwe jest zastosowanie go do bardzo wielu różnych zadań inżynierii lingwistycznej poprzez dotrenowanie (a więc dostosowanie wag) na potrzeby konkretnego zadania. Model BERT można więc wykorzystać do klasyfikacji pojedynczych segmentów (np. w zadaniu ujednoznaczniania morfoskładniowego), pojedynczych zdań (np. oceny ich wydźwięku), par zdań (np. analizując ich wynikanie logiczne) czy oznaczania w tekście fragmentów stanowiących odpowiedź na pytanie.

Modele typu BERT są dalej rozwijane; w szczególności w chwili tworzenia tego tekstu najlepsze efekty w wielu zadaniach praktycznych daje zastosowanie wersji RoBERTa (Liu i in. 2019), zoptymalizowanej w celu umożliwienia jej efektywnego trenowania na bardzo dużych zbiorach danych. Postępy w rozwoju i skuteczności tego rodzaju modeli można śledzić na stronach agregujących dokładność ich działania dla poszczególnych zadań NLP. W przypadku języka angielskiego jest to na przykład strona rankingu GLUE (https://gluebenchmark.com/; Wang i in. 2018), a w przypadku języka polskiego strony rankingów KLEJ (https://klejbenchmark.com/; Rybak i in. 2020) i LEPISZCZE (https://lepiszcze.clarin-pl.eu/; Augustyniak i in. 2022).

Inną grupą modeli są tzw. modele generatywne, w szczególności model GPT (*Generative Pre-Trained Transformer*; Radford i in. 2018). Modele te trenowane są w sposób nienadzorowany na dużych korpusach tekstowych, a następnie dotrenowywane na mniejszych korpusach anotowanych, w celu dostosowania do konkretnego zadania. Działanie modeli typu GPT, w odróżnieniu od modeli dyskryminacyjnych, takich jak BERT, nie polega na przewidywaniu pewnych decyzji dotyczących danych (takich jak np. etykietowanie fragmentów tekstu), lecz na tworzeniu ciągłego tekstu w języku naturalnym. Tekst ten uzależniony jest od tzw. zachęty (ang. *prompt*), która stanowi wejście dla modelu. Podejście to okazało się bardzo obiecujące w wielu zadaniach, takich jak streszczanie tekstu, odpowiadanie na pytania czy tłumaczenie maszynowe. Możliwe jest również wykorzystanie modeli generatywnych do zadań realizowanych typowo przez modele dyskryminacyjne, np. analizy wydźwięku czy wykrywania w tekstach nazw własnych.

Obiecujące wyniki zastosowania modeli generatywnych były inspiracją dla twórców modelu T5 (Raffel i in 2020). Jest to model typu *text-to-text*, a więc przekształcający pewien ciąg wejściowy w inny ciąg. Główną korzyścią takiego podejścia jest jego uniwersalność, nie wymagająca trenowania osobnych modeli dla każdego z zadań inżynierii lingwistycznej. Model T5 okazał się skuteczny w tak różnorodnych zadaniach jak automatyczne streszczanie, tłumaczenie maszynowe, ale też odpowiadanie na pytania czy klasyfikacja tekstu. Z założenia modele typu T5 są lepiej dostosowane do rozwiązywania problemów typowych dla modeli dyskryminacyjnych (np. klasyfikacji) niż modele generatywne, ale zachowują ich główną cechę wyróżniającą, jaką jest uniwersalność.

Modele GPT były rozwijane stopniowo, w szczególności poprzez trenowanie ich na coraz większych zbiorach danych. W ten sposób powstały kolejno: GPT-2 (Radford i in. 2019), GPT-3 (Brown i in. 2020), GPT-4 (OpenAI 2023). W roku 2022 zaproponowany został model ChatGPT, który zoptymalizowany został pod kątem prowadzenia rozmowy (ang. *chat*) z użytkownikiem[1], a w roku 2024 model GPT-4o pozwalający na jednoczesne wnioskowanie na podstawie danych o wielu modalnościach, takich jak tekst, obraz i dźwięk. Ze względu na fakt, że modele te trenowane są na ogromnych zbiorach tekstowych (od pewnego czasu również multimodalnych), zyskały one również nazwę *wielkich model językowych* (ang. *Large Language Models*, LLM). Obecnie, obserwujemy wzmożone prace nad nowymi modelami, prowadzone przez większe ośrodki naukowe oraz firmy inwestujące w rozwiązania sztucznej inteligencji, trudno więc je przytoczyć w tekście bez narażania się na brak aktualności. Warto jednak wspomnieć, że prace nad dużymi modelami językowymi prowadzone są również w Polsce. W chwili pisania tego tekstu rozwijane są przynajmniej dwa modele priorytetyzujące zastosowanie dla języka polskiego, a mianowicie model Bielik, realizowany poprzez społeczność zbudowaną wokół projektu SpeakLeash (https://speakleash.org) oraz projekt PLLuM, realizowany przez konsorcjum instytucji naukowych (https://pllum.org.pl).

Ze względu na bardzo szybki rozwój tej dziedziny warto obserwować zmiany, jakie następują na listach agregujących dokładność działania poszczególnych metod, np. na wymienionych wcześniej stronach tworzących rankingi modeli, a także w konkursach (ang. *shared tasks*) organizowanych w powiązaniu z konferencjami naukowymi. Dla języka polskiego takim corocznie organizowanym konkursem dla rozwiązań z obszaru inżynierii lingwistycznej jest PolEval[2].

5. Zastosowania inżynierii lingwistycznej

Wraz z upowszechnieniem się dużych modeli językowych (LLM) znacznie poszerzył się zakres zastosowań, które możemy przypisywać inżynierii lingwistycznej. Niemniej, już przed tym przełomem takich zastosowań było wiele. Typowe przykłady to tłumaczenie maszynowe, czy klasyfikacja tekstu (np. klasyfikacja korespondencji wpływającej do firmy, aby automatycznie przydzielić dokument do właściwego działu lub osoby, zajmującej się sprawą). Innymi szeroko wykorzystywanymi zastosowaniami metod inżynierii lingwistycznej są: analiza wydźwięku (na przykład w badaniach konsumenckich, w recenzjach, w zapisach rozmów z klientami), identyfikacja nazw własnych (na przykład w celu ułatwienia ekstrakcji nazw firm, organizacji, osób, produktów, czy miejsc w tekstach), czy automatyczna detekcja mowy nienawiści (np. w sieciach społecznościowych, czy na forach dyskusyjnych). Z korektą tekstu mamy do czynienia na co dzień, gdy korzystamy z edytora tekstu, a automatyczna detekcja i klasyfikacja spamu

[1] https://openai.com/blog/chatgpt/
[2] http://poleval.pl/

pozwala efektywnie korzystać z poczty elektronicznej. W kontekście badań nad językiem, wiele narzędzi zostało wytworzonych w ramach prac konsorcjum CLARIN[3]. W ramach tych prac powstało m.in. narzędzie Korpusomat[4], umożliwiające tworzenie własnych korpusów tekstowych, ich automatyczne znakowanie, analizę i przeszukiwanie.

Obecnie, korzystając z dużych modeli językowych, możemy dzięki inżynierii lingwistycznej realizować tak szeroką gamę zadań, że używa się coraz częściej określenia „agentów sztucznej inteligencji", którzy pomagają w codziennej pracy, automatyzacji procesów, czy w życiu prywatnym. Duże modele językowe pozwalają bowiem nie tylko realizować klasyczne, wymienione wcześniej zadania, ale problemy bardziej złożone, do których wcześniej wykorzystywano wieloetapowe przetwarzanie lub analizę, aby osiągnąć pożądanych skutek. W szczególności, duże modele mogą odpowiadać na pytania, których zakres wciąż się poszerza. W chwili pisania tego tekstu modele typu GPT dobrze radzą sobie z pytaniami o fakty, pytaniami wymagającymi logicznego wnioskowania, czy analizy danych. Nie radzą sobie natomiast zbyt dobrze z operacjami matematycznymi i wnioskowaniem, które wymaga wielu etapów (np. złożone zagadki matematyczne, czy algorytmiczne). W kontekście pracy z tekstem, modele mogą posłużyć do streszczania tekstu, ale też jego rozwijania na podstawie zalążkowego pomysłu. Dobrze sprawdzają się w roli recenzenta, czy asystenta podczas pisania tekstu, wspomagają więc istotnie na pracę autora, czy redaktora, sugerując zmiany czy uzupełnienia. Szeroką gamę zastosowań umożliwia również połączenie modeli generatywnych z narzędziami do automatyzacji, czy kodem wykonywalnych programów. Modele stają się wówczas sercem „agentów", którzy realizują na przykład automatyczną obsługę klienta (rozmowę poprzez chat lub telefon, identyfikację sprawy i jej rozwiązanie), automatyczne przetwarzanie dokumentów, tworzenie kodu programów (asystenci programowania), czy asystenci pomagający realizować codzienne zadania, funkcjonujący w telefonie komórkowym.

6. Kierunki dalszych prac

Jak już wspominaliśmy, przetwarzanie języka naturalnego wykorzystuje dziś najnowsze metody sztucznej inteligencji, a postęp wyznaczają dziś w większym stopniu kolejne konfiguracje głębokich sieci neuronowych (ang. *deep neural networks*) niż izolowane rozwiązania dla poszczególnych zadań. Czy oznacza to, że głównie globalni gracze, dysponujący wielką mocą obliczeniową i zindeksowanymi zasobami Internetu będą nadawać ton rozwiązaniom NLP w przyszłości? Takie obawy podziela też Komisja Europejska, która z jednej strony wdraża regulacje zapewniające możliwość rozwoju sztucznej inteligencji z poszanowaniem praw człowieka, takie jak rozporządzenie w sprawie sztucznej inteligencji[5], a z drugiej próbuje uniezależnić się od zagranicznych dostawców technologii językowej tworząc własne rozwiązania w rodzaju systemu eTranslation[6], wspierając inicjatywy zmierzające do budowy europejskiej przestrzeni danych językowych (*Language Data Space*, LDS)[7] czy infrastruktury danych i usług dla technologii językowych (w ramach *Alliance for Language Technologies*, ALT-EDIC)[8].

[3] https://clarin-pl.eu
[4] https://korpusomat.eu
[5] https://artificialintelligenceact.eu/
[6] https://commission.europa.eu/resources-partners/etranslation_pl
[7] https://language-data-space.ec.europa.eu/
[8] https://language-data-space.ec.europa.eu/related-initiatives/alt-edic_en

Jakie wnioski płyną z tego faktu dla badaczy w Polsce? Rozwój inżynierii lingwistycznej w naszym kraju wydaje się postępować wraz z rozwojem światowym. Natomiast rosnąca rola danych w zadaniach związanych ze sztuczną inteligencją sprawia, że oprócz działającego już dość sprawnie programu otwierania danych publicznych szczególnie istotna wydaje się organizacja stałego modelu rozwoju korpusu referencyjnego dla języka polskiego. Projekt Narodowego Korpusu Języka Polskiego (Przepiórkowski i in. 2012) zakończył się ponad 10 lat temu, nowsza inicjatywa Korpusu Współczesnego Języka Polskiego[9] dotyczyła zbierania danych tekstowych do roku 2020, a kolejne prace realizowane w trybie projektowym (czyli z określoną datą zakończenia prac i zbierania danych) będą zawsze generować wyniki nieaktualne już w momencie ich zakończenia. W ramach krajowych infrastruktur CLARIN-PL i DARIAH-PL udaje się co prawda utrzymywać korpusy monitorujące (zbierające dane na bieżąco ze stron internetowych w danym języku), takie jak Monco[10], PAJONK[11], czy PAWUK[12] (język ukraiński), ale zapewnienie środowisku naukowemu dostępu do aktualnych danych referencyjnych wydaje się coraz bardziej palące i wymaga systemowego rozwiązania.

Inną obserwowaną tendencją jest coraz powszechniejsze wykorzystywanie modeli wielojęzycznych, co przenosi punkt ciężkości z niskopoziomowego rozwoju rozwiązań specyficznych dla danego języka na adaptację istniejących modeli. Dla krajowych badaczy oznacza to szansę na skupienie się na zaawansowanych własnościach lingwistycznych języka polskiego, tworzeniu synergii między językoznawstwem a inżynierią i możliwość uzyskania wsparcia międzynarodowego. Dla społeczności lokalnych – perspektywę wykorzystania tego trendu w efektywnym przetwarzaniu dialektów i języków regionalnych takich jak kaszubski czy śląski.

Na najważniejszych konferencjach z dziedziny przetwarzania języka takich jak ACL (Annual Meeting of the Association for Computational Linguistics) czy EMNLP (Conference on Empirical Methods in Natural Language Processing) najczęściej prezentowane są obecnie badania, których autorzy zastanawiają się, jak pytać modele językowe o pożądaną odpowiedź. Trend ten może być przejawem wyłaniania się nowego sposobu działania i używania narzędzi językowych, opartego na zasadach współpracy z „usługą językową" realizującą pełne spektrum zadań wymagających do tej pory specjalizowanych narzędzi. Być może zwiastuje to zmianę roli badacza-lingwisty, do którego zadań będzie należeć postawienie właściwego pytania, dostarczenie danych do nauki i weryfikacji hipotez – a potem pozostawienie wyboru sposobu przeprowadzenia analizy maszynie. Warto już dziś zadbać o to, by wykorzystać te nowe możliwości jak najefektywniej.

[9] https://kwjp.pl
[10] https://monco-pl.clarin-pl.eu
[11] https://pajonk.ipipan.waw.pl (planowane udostępnienie w roku 2025)
[12] https://pawuk.ipipan.waw.pl

Lektura uzupełniająca

Jurafsky, Daniel, Martin, James H. (2022). *Speech and Language Processing*. Wyd. 3 (wersja robocza dostępna online :https://web.stanford.edu/~jurafsky/slp3/).

Lane, Hobson, Howard, Cole, Hapke, Hannes Max, Kobyliński, Łukasz, Tuora, Ryszard (2021). *Przetwarzanie języka naturalnego w akcji: rozumienie, analiza i generowanie tekstu w Pythonie na przykładzie języka angielskiego*. Wydawnictwo Naukowe PWN.

Bibliografia

Bojanowski, Piotr, Grave, Edouard, Joulin, Armand, Mikolov, Tomas (2017). *Enriching Word Vectors with Subword Information*. Transactions of the Association for Computational Linguistics 5:135--146.

Brown i in. (2020).Language Models are Few-Shot Learners. Preprint. https://arxiv.org/abs/2005.14165

Chomsky, Noam (1957). *Syntactic Structures*. Mouton & Co.

Devlin, Jacob, Chang, Ming-Wei, Lee, Kenton, Toutanova, Kristina (2019). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. W: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), s. 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Grosz, Barbara J., Joshi, Aravind K., Weinstein, Scott (1995). *Centering: A framework for modeling the local coherence of discourse*. Computational Linguistics, 21(2):202–225.

Koehn, Philipp (2005). *Europarl: A Parallel Corpus for Statistical Machine Translation.* In Proceedings of Machine Translation Summit X: Papers, s. 79–86, Phuket, Tajlandia.

Koehn, Philipp, Hoang, Hieu, Birch, Alexandra, Callison-Burch, Chris, Federico, Marcello, Bertoldi, Nicola , Cowan, Brooke, Shen, Wade, Moran, Christine, Zens, Richard, Dyer, Chris, Bojar, Ondrej, Constantin, Alexandra, Herbst, Evan (2007). *Moses: Open Source Toolkit for Statistical Machine Translation*, Annual Meeting of the Association for Computational Linguistics (ACL), demonstration session, Praga, Czechy.

Lenat, Doug, Prakash, Mayank, Shepherd, Mary (1986). *CYC: Using Common Sense Knowledge to Overcome Brittleness and Knowledge Acquistion Bottlenecks*". AI Magazine. 6(4):65–85. https://doi.org/10.1609/aimag.v6i4.510

Lesk, Michael E. (1986). *Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone*. In SIGDOC '86: Proceedings of the 5th Annual International Conference on Systems Documentation, s. 24–26, Nowy Jork. ACM.

Liu, Yinhan, Ott, Myle, Goyal, Naman, Du, Jingfei, Joshi, Mandar, Chen, Danqi, Levy, Omer, Lewis, Mike, Zettlemoyer, Luke, Stoyanov, Veselin (2019). *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. https://arxiv.org/pdf/1810.04805.pdf.

Mikolov, Tomas, Chen, Kai, Corrado Greg, Dean, Jeffrey (2013). *Efficient estimation of word representations in vector space*" https://arxiv.org/pdf/1301.3781.pdf.

Peters, Matthew E., Neumann, Mark, Iyyer, Mohit, Gardner, Matt, Clark, Christopher, Lee, Kenton, Zettlemoyer, Luke (2018). *Deep Contextualized Word Representations*. W: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), s. 2227–2237, Nowy Orlean, Louisiana. Association for Computational Linguistics. https://aclanthology.org/N18-1202.pdf.

Pollard, Carl J, Sag, Ivan A. (1994). *Head-Driven Phrase Structure Grammar*. Studies in Contemporary Linguistics, Chicago, IL/Londyn: The University of Chicago Press.

Przepiórkowski, Adam, Bańko, Mirosław, Górski, Rafał L., Lewandowska-Tomaszczyk, Barbara (red.). (2012). *Narodowy Korpus Języka Polskiego*. Wydawnictwo Naukowe PWN.

Radford, Alec, Narasimhan Karthik, Salimans, Tim, Sutskever, Ilya (2018). *Improving Language Understanding by Generative Pre-Training*. Preprint. https://openai.com/blog/language-unsupervised/

Radford, Alec, Wu, Jeffrey, Child, Rewon, Luan, David, Amodei, Dario, Sutskever, Ilya (2019). *Language Models are Unsupervised Multitask Learners.* Raport techniczny, OpenAI. https://openai.com/blog/better-language-models/

Raffel, Colin, Shazeer, Noam, Roberts, Adam, Lee, Katherine, Narang, Sharan, Matena, Michael, Zhou, Yanqi, Li, Wei, Liu Peter J. (2022). *Exploring the limits of transfer learning with a unified text-to-text transformer*. The Journal of Machine Learning Research 21(1):5485–5551.

Rybak, Piotr, Mroczkowski, Robert, Tracz, Janusz, Gawlik, Ireneusz (2020). *KLEJ: Comprehensive Benchmark for Polish Language Understanding*. W: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, s. 1191–1201, Online. Association for Computational Linguistics.

Turing, Alan M. (1950). *Computing Machinery and Intelligence*. Mind, New Series 59(236):433–460. Oxford University Press. http://www.jstor.org/stable/2251299

Wang, Alex, Singh, Amanpreet, Michael, Julian, Hill, Felix, Levy, Omer, Bowman, Samuel (2018). *GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding*. W: Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, s. 353–355, Bruksela, Belgia. Association for Computational Linguistics.

Weizenbaum, Joseph (1966). *ELIZA – A Computer Program for the Study of Natural Language Communication Between Man and Machine*. Communications of the ACM 9(1):36–45, https://doi.org/10.1145/365153.365168