Kraków, 18 kwietnia 2022r.

Prof. dr hab. Wiesław Lubaszewski
Profesor emerytowany
Instytut Informatyki AGH
Katedra Lingwistyki Komputerowej UJ

Recenzja pracy doktorskiej mgra inż. Sławomira Dadasa, pt. Iteracyjne metody wykrywania hierarchicznych jednostek nazewniczych

Recenzję przygotowano dla potrzeb przewodu doktorskiego z dyscypliny Informatyka techniczna i telekomunikacja prowadzonego przez Radę Naukową Instytutu Badań Systemowych PAN.

Recenzowana rozprawa składa się z przedmowy, 8 rozdziałów i bibliografii. *Przedmowa* (s. 6-9) szkicowo przedstawia problematykę tzw. *named entities* i ich automatycznego rozpoznawania. Autor w pracy używa terminu jednostki nazewnicze, choć lepiej byłoby powiedzieć jednostki nazywające. Jednostki nazywające to bardzo liczny zbiór jedno i wieloskładnikowych symboli (jednostek języka), które w odróżnieniu od tzw. wyrazów pospolitych nie mają znaczenia i tylko wskazują (referencja) na obiekt lub zbiór obiektów należący do rzeczywistości pozajęzykowej. Są to głównie nazwy własne (imiona, nazwiska, nazwy miejscowe i geograficzne, nazwy instytucji i firm itd.) oraz terminy. Jednostkami nazywającymi i mogą być także ciągi wyrazów pospolitych, które jednoznacznie wskazują na konkretną nazwę, np. *zwycięzca spod Trafalgaru* wskazuje nazwę *admirał Nelson*. Patrząc na strukturę jednostek nazywających możemy powiedzieć, że jest ona zbliżona do tych jednostek języka naturalnego, które mają znaczenie. W praktyce, każdy mający znaczenie wyraz (jednostka słownika) może być użyty jako nazwa własna, np. *Stodoła* może być nazwą restauracji, klubu, kabaretu itp. Dlatego często w strukturze nazw własnych pojawia się klasyfikator ułatwiający interpretację nazwy, a więc, np. *Restauracja Stodoła, Kabaret Stodoła* itp. Terminologia ma nieco inną strukturę niż nazwy własne. Można powiedzieć, że struktura terminów jest bardziej zbliżona do mających znaczenie wyrazów wielosegmentowych (multipart words), np. *panna młoda* (kobieta w dniu ślubu) wobec *komputer osobisty*. Trzeba przy tym dodać, że pewne terminy w odróżnieniu od nazw mogą mieć szyk zmienny *większość absolutna* albo *absolutna większość* itp. Ten krótki zarys problematyki pokazuje, jak trudnym zadaniem jest automatyczne rozpoznawanie jednostek nazywających za pomocą metod uczenia maszynowego (machine learning).

Rozdział 1 *Wstęp* (s. 10-17) przedstawia cele pracy oraz metodę, za pomocą której Autor zamierza osiągnąć założone cele. Celem zasadniczym jest „Zaproponowanie nowego iteracyjnego ujęcia wykrywania hierarchicznych struktur jednostek nazewniczych.” (s. 17). Sposób w jaki Autor pojmuje hierarchiczną strukturę jednostek nawanych ilustruje zaczerpnięty z literatury przykład*: Mr. Rapanelli met in August with US Assistant Treasury Secretary David Mullford* (s. 12). Hierarchię składników w prezentowanym przykładzie Autor przedstawia następująco. Jednostka nazywająca traktowana jako całość ma składniki zagnieżdżone, np. *US Assistant Treasury Secretary David Mullford* sklasyfikowany jako osoba, który ma własne składniki zagnieżdżone, takie jak np. *US Assistant Treasury Secretary,* który klasyfikowany jako całość otrzymuje klasyfikator Rola i można w nim wyróżnić *US* Nation oraz *Assistant Treasury Secretary* Rola, dający się podzielić na *Treasury Secretary* z klasyfikatorem Government i Rola. Można dyskutować z taką interpretacją i zapytać dlaczego

dokonano podziału stanowiącego niepodzielną całość terminu *Assistant Treasury Secretary* oznaczającego funkcję wiceministra. Jednak nie jest to aż tak istotne.

Jeśli chodzi o metodę, za pomocą której Autor chce osiągnąć zakładane cele, można - zgodnie z intuicją - przyjąć, że przytoczony w pracy przykład może być zinterpretowany stosowaną w pracy metodą tagowania (znakowania) sekwencyjnego, która ma tę zaletę, że nie nakłada ograniczeń na długość rozpoznawanej jednostki nazwanej. Autor zakłada, przy tym, iż zdoła wykazać, że „[..] iteracyjne metody charakteryzujące się poprawnością wyników porównywalną lub wyższą od stosowanych od stosowanych do tej pory rozwiązań dla szerokiego zakresu zadań związanych z wykrywaniem zagnieżdżonych struktur jednostek nazewniczych" (s. 14). W tym celu „Metody zaproponowane w tej pracy zostaną przetestowane na powszechnie używanych zbiorach danych dla tego problemu, różniących się od siebie m. in. językiem, domeną czy poziomem złożoności zagnieżdżonych struktur" (s. 14).

Z punktu widzenia recenzenta, który powinien ocenić przedstawiony w pracy algorytm rozpoznawania struktury jednostek nazywających, istotna jest odpowiedź na pytanie, czy zaproponowany w pracy algorytm zdoła poprawnie zinterpretować, np. następujące nazwy krakowskich restauracji:

Pod Aniołem – brak klasyfikatora
Chłopskie Jadło - brak klasyfikatora,
Restauracja Stodoła – klasyfikator jest pierwszym składnikiem nazwy,
Pierwszy Lokal na Stolarskiej Idąc od Strony Małego Rynku - klasyfikator to drugi składnik nazwy, zaś *Stolarska* i *Mały Rynek* są lokalizatorami .

Jeśli tak, można uznać, że Autor osiągnął zakładane cele. Jeśli nie, należy oczekiwać odpowiedzi wyjaśniających. Chcielibyśmy też wiedzieć, jak na wynik interpretacji wpłynęłoby zamiana dużych liter na małe? Odpowiedzi na postawione pytania będziemy szukać w opisie testów i we wnioskach.

Rozdziały 2,3 i 4 mają charakter przeglądowy. Autor dokonuje przeglądu zagadnień związanych z problematyką poruszaną w pracy. Rozdział 2 *Wprowadzenie do modelowania sekwencji* (s. 18-33) omawia modele grafowe, rekurencyjne sieci neuronowe, sieci Transformer i splotowe sieci neuronowe oraz możliwość integracji modeli na przykładzie integracji modeli grafowych z sieciami neuronowymi. Rozdział 3 Wprowadzenie do wektorowych reprezentacji tekstu omawia reprezentacje statyczne i reprezentacje kontekstowe oraz niespodziewanie dla recenzenta metody tokenizacji (wyróżniania jednostek) tekstu, które są zagadnieniem odrębnym i powinny być omówione wcześniej, np. przed rozdziałem 2. Trzeba przy tym dodać, że omówienie problematyki tokenizacji powinno być obszerniejsze i zrobione staranniej. W pracy lokującej się w obszarze przetwarzania języka naturalnego (NLP) nie powinny się zdarzać wpadki typu „… - słowo w przeciwieństwie do mniejszych jednostek tekstu, posiada znaczenie semantyczne" (s. 41). Powinno być znaczenie leksykalne lub tylko znaczenie. Dodatkowo, jak pisałem na wstępie, nazwy własne i terminy w odróżnieniu od wyrazów pospolitych nie mają znaczenia tylko referencję. Tak więc token jest jednostką tekstu niezależnie od tego czy reprezentuje w tekście konkretne znaczenie leksykalne , czy też odsyła (referencja) do konkretnego elementu rzeczywistości pozajęzykowej. Słowem, problem tokenizacji jest o wiele bardziej skomplikowany niż to wynika z pracy.

Rozdział 4 *Przegląd stosowanych metod* (s.44-54) także wywołuje uwagi krytyczne. Tym razem chodzi o tytuł rozdziału. Dopiero z tekstu (s. 45) dowiadujemy się, że chodzi o przegląd metod wykrywania hierarchicznych jednostek nazwanych. Tego typu błąd nie powinien się pojawić w rozprawie doktorskiej. Autor omawia metody statystyczne oraz metody oparte na hipergrafach i sieciach neuronowych. Omówieniu istotnych dla pracy metod interacyjnych poświęcono rozdział 5 *Metody iteracyjne* (s. 55-61), w którym Autor omawia problemy iteracyjnego tagowania sekwencji jednostek tekstu, iteracyjny model neuronowy oraz zagadnienie trenowania model iteracyjnych.

Wreszcie rozdział 6 *Dwukierunkowy algorytm iteracyjny* (s. 62-71) przynosi opis oryginalnego osiągnięcia Autora, tj. iteracyjnego algorytmu rozpoznawania jednostek nazywających, którym można przyporządkować strukturę hierarchiczną. Najogólniej mówiąc, model Autora to iteracyjnie stosowana sieć neuronowa, operująca na wektorowej reprezentacji jednostek tekstu (tokenów). Na stronie 59 Autor tak opisuje swój algorytm:

„Schemat całego modelu neuronowego przedstawiono na rysunku 5.2 . Utworzone dla każdego elementu sekwencji reprezentacja wektorowa jest transformowana przez blok składający się z dwukierunkowych warstw LSTM. Wynikiem tych warstw są reprezentacje ukryte tokenów, na podstawie których dokonywana jest predykcja modelu. Jako warstwy predykcyjnej użyto *linear-chain CRF*, co pozwala na uwzględnienie strukturalnych zależności pomiędzy poszczególnymi elementami sekwencji wyjściowej modelu. Wyjściem modelu jest sekwencja tagów, z której po dokonanej predykcji ekstrahowane są wykryte jednostki nazewnicze, a następnie na ich podstawie aktualizowany jest stan modelu. Jeżeli stan nie uległ zmianie, to znaczy zbiór wykrytych do tej pory jednostek nie zwiększył się o nowe elementy, model kończy działanie i zwraca aktualny stan jednostek. Jeżeli stan zmienił się, wykonywana jest kolejna iteracja.” (s. 59). Akceptacji lub odrzucenia rozpoznanej w ten sposób jednostki nazywającej dokonuje tzw. funkcja selekcji. W pracy przetestowano 6 różnych funkcji: tylko inside-out, tylko outside-out, suma modeli, część wspólna modeli, selekcja oparta na pewności oraz selekcja oparta na klasyfikatorze.

Opis autorskiego algorytmu jest przeprowadzony na wysokim poziomie abstrakcji i konieczny byłby przykład wyjaśniający działanie modelu na konkretnych danych, przynajmniej w trakcie jednej iteracji. Opisany w pracy algorytm jest dwukierunkowy, tj. rozpoczyna rozpoznawanie od sekwencji najdłuższej (outside-in) dzieląc ją na składniki lub rozpoczyna od rozpoznawania składników, z których buduje sekwencję ogólną (inside-out), co dawałoby szansę na poprawne rozpoznanie terminów o szyku przestawnym, np. *większość absolutna*/*absolutna większość*.

Opisie testów dwukierunkowego algorytmu iteracyjnego jest poświęcony rozdział 7 *Eksperymenty* (s. 72-93). Testy opisane w pracy przeprowadzono na następujących korpusach: GENIA, NNE, PolEval, GermEval. Algorytm iteracyjny opisany w pracy był trenowany osobno dla każdego korpusu na uczącym zbiorze danych, a następnie algorytm operował na testowej części zbioru. W testach użyto wszystkich 6 funkcji wyboru.

Niestety, korpusy na których prowadzono testy zostały opisane bardzo ogólnie. Nie wiemy więc, jakie typy strukturalne jednostek nazywających miał wykrywać testowany algorytm. Nieco więcej o strukturze danych, na których testowano algorytm autorski dowiadujemy się tylko z opisu danych konkursu PolEval. Autor pisze:

„Dla obu części [ucząca i testowa WL] zastosowano ten sam zestaw 14 kategorii jednostek nazewniczych, z czego 6 stanowi kategorie główne a 8 kategorie podrzędne. Kategorie te odnoszą się do prawdziwych i fikcyjnych i fikcyjnych osób, organizacji, obiektów geograficznych i geopolitycznych, struktur zbudowanych przez człowieka, fraz związanych z datą i czasem. Jednostki nazewnicze należące do głównych kategorii mogą występować zarówno w warstwie zewnętrznej jak i w postaci jednostek zagnieżdżonych, natomiast jednostki z kategoriami podrzędnymi tylko w warstwach zagnieżdżonych, jako uszczegółowienie jednostki nadrzędnej." S. 85. Zatem dopiero w końcowej części pracy dowiadujemy się o danych, na których operuje algorytm Autora. Jednak nie jest informacja pełna – brakuje omówienia popartych przykładami struktur występujących w korpusie named entities.

Wyniki testów porównano z opublikowanymi wynikami uzyskanymi przez inne algorytmy na tym samym zbiorze danych. Wynik tego typu testów jest korzystny dla metody autorskiej. Jednak nie można na tym poprzestać. Tabela 7.6 na s. 80 pokazuje, że miary przyjęte dla oceny algorytmu iteracyjnego pokazują skuteczność, która nie przekracza 90%. Można więc przyjąć, że przynajmniej 10% badanego zbioru to jednostki, których algorytm nie zdołał zinterpretować. Należałoby więc przeanalizować zbiór nierozpoznanych jednostek nazywających, a analiza powinna wyjaśnić dlaczego algorytm ich nie rozpoznał. Wystarczyłoby przeanalizować wyniki uzyskane na korpusie PolEval. W podobny sposób należałoby przeanalizować zbiory false positives oraz false negatives. Wtedy dopiero uzyskalibyśmy pełniejszą ocenę algorytmu. Fakt, że analizy trzeba by było przeprowadzić półautomatycznie, nie usprawiedliwia ich braku.

Konkluzja

Brak wspomnianych wyżej analiz oraz notoryczny brak przykładów, a także brak próbek wyników nie pozwalają ocenić algorytmu, który jest oryginalnym osiągnięciem Doktoranta. Nie ma powodu, by wątpić w przedstawione w pracy wyniki testów przeprowadzonych na wielu korpusach, jednak testy te nie pokazują wszystkich właściwości algorytmu. Dlatego uważam, że praca lokuje się na granicy akceptowalności i trzeba dużej tolerancji, by uznać, że praca mgra inż. Sławomira Dadasa spełnia ustawowe i zwyczajowe wymogi stawiane rozprawom doktorskim. Co powiedziawszy, wnoszę o dopuszczenie Doktoranta do dalszych etapów przewodu doktorskiego.

W. Lubaszewski