Realizacja prostego silnika reguł walidacyjnych przy użyciu technik NLP

Mariusz Wójcik

10 czerwca 2019

1 Początki

Jakiś czas temu miałem okazję obejrzeć film, który zrobił na mnie ogromne wrażenie. „Arrival - Nowy Początek" w reżyserii Denisa Villeneuve'a - to obraz niezwykły. Porusza on problematykę szerokopojętej, wielopłaszczyznowej komunikacji (a czasem skutków jej braku) . W niesamowicie sugestywny i obrazowy sposób pokazuje mechanizm kształtowania się podstaw wspólnego języka i nawiązywania kontaktu. Proces stopniowego budowania uwspólnionych modeli pojęciowych prowadzący do porozumiewania się tym samym językiem wydał mi się tak logiczny i uporządkowany, że sprawiał wrażenie niemal algorytmicznego... Pamiętam swoją myśl, że skoro możliwe jest tak precyzyjne określenie reguł stojących u podstaw nawiązania skutecznej komunikacji, to droga do zbudowania inteligentnych maszyn porozumiewających się z nami „ludzkim" językiem wydaje się już bardzo krótka.

Do niedawna wydawało mi się, że porozumiewanie się językiem naturalnym jest domeną przynależną wyłącznie człowiekowi. Miałem poczucie, że prace nad komputerowym przetwarzaniem języka naturalnego mają wymiar wyłącznie akademicki. Okazało się jednak, że dynamiczny rozwój algorytmów sztucznej inteligencji i przetwarzania maszynowego dotknął również tej dziedziny. Gdzieś na styku matematyki, informatyki i lingwistyki wykształciła się dziedzina, która funkcjonuje jako NLP (ang. natural language processing).

Techniki NLP koncentrują się na analizie, przekształcaniu i generowaniu języka naturalnego. Dzięki nim komputery nabywają umiejętności nie tylko analizy tekstu, ale również nauki i wyciągania wniosków. Dają one możliwość analizy nie tylko składni zdań, ale również doszukiwania się ich znaczeń i ukrytych pomiędzy słowami intencji. Czasami uświadamiam sobie że to wszystko razem brzmi jak czysta fantastyka. Bo jak niby sens, znaczenie i intencje można przeliczyć na liczby i twardo zakotwiczyć w dziedzinie algebry liniowej ?

Tajemnicy tej uchyla jedna z najciekawszych książek, jaką miałem przyjemność ostatnio czytać, mianowicie „Natural Language Processing in Action". Jest to bardzo przystępnie napisany przewodnik, dzięki któremu łatwiej oswoić się z podstawowymi prawami rządzącymi światem NLP. Pozycja nie traktuje o rzeczach najłatwiejszych, a mimo to czyta się ją z dużą przyjemnością.

Z teorią często jest tak, że w którymś momencie chciałoby się ją zobaczyć w praktyce. Z tej potrzeby zrodził się pomysł na aplikację, którą możnaby zrealizować przy użyciu technik i algorytmów NLP.
Przyszedł mi do głowy generator kodu aplikacji, który byłby w stanie przekształcić tekst napisany językiem zbliżonym do naturalnego bezpośrednio do kodu wykonywalnego. Oczywiście zakładam że tego typu rozwiązanie miałoby zastosowanie do jakiegoś ściśle określonego aspektu działającej aplikacji, np. walidacji dokumentu, czy sprawdzania reguł poprawności modelu dziedziny.
I tak właśnie powstał mój miniprojekt, którego celem jest zobaczenie o co tak naprawdę chodzi z tym NLP . :) . Zapraszam do zapoznania się z założeniami i otrzymanymi wynikami.

Mam świadomość, że jeśli chodzi o NLP, jestem na początku drogi. Nie mogę powiedzieć nawet tego, że udało mi się zrobić jeden krok, ale wiem jedno... podróż zapowiada się naprawdę imponująco...

2 Założenia i sposób realizacji

3 Abstrakcyjny model reguły

Kluczowym elementem całego rozwiązania jest model, w oparciu o który narzędzia dostępne w bibliotece OpenNLP będą dokonywały analizy reguł. By przygotować taki model konieczne jest dostarczenie dobrej jakości próbki uczącej, która weźmie udział w procesie jego trenowania.

Przygotowanie próbki rozpocznę od opracowania schematu reguły. Oczekuję, że system prawidłowo rozpozna poszczególne składowe każdej reguły, która będzie z nim zgodna.

Na początek wypiszę sobie kilka przykładowych reguł walidacyjnych.

1. Jeśli wiek_pacjenta jest większy od 18 wtedy zgłoś błąd „Pacjent jest osobą dorosłą.”, w przeciwnym wypadku wyświetl komunikat „Pacjent został zakwalifikowany do leczenia pediatrycznego.”.
2. Jeśli data_kwalifikacji jest jest mniejsza od ’01-01-2019’ wtedy zgłoś wyjątek „Data sprzed roku 2019.”, w przeciwnym wypadku sprawdź regułę RS-001.
3. Gdy saldo_rachunku jest większe od 100 oraz saldo_rachunku jest mniejsze niż 1000 wtedy wyświetl komunikat „Saldo rachunku jest prawidłowe.”, w przeciwnym razie zgłoś błąd „Nieprawidłowe saldo rachunku”.
4. Jeśli data_teraz jest niewiększa niż data_ważności wyświetl komunikat „Wniosek jest aktualny.” w przeciwnym wypadku zgłaszaj błąd „Wniosek utracił ważność”.

Przyjmuję uproszczenie, że każda rozpoznawana reguła składała się będzie z trzech wyróżnialnych bloków:

$$\underbrace{WARUNKI}\,\underbrace{AKCJA\_TAK}\,\underbrace{(AKCJA\_NIE)?}$$

Poszczególne bloki oddzielone będą od siebie słowami kluczowymi oznaczającymi rozpoczęcie i zakończenie bloku.

W celu ich wyróżnienia wprowadzam następujące oznaczenia:

1. SK_SW - Start sekcji warunku
2. SK_KW - Koniec sekcji warunku
3. SK_SAN - Start sekcji akcji wykonywanej przy niespełnionym warunku

Schemat reguły przyjmuje następującą postać:

$$\underbrace{SK\_SW}\,\underbrace{WARUNEK}\,\underbrace{SK\_KW}\,\underbrace{AKCJA\_TAK}\,(\underbrace{SK\_SAN}\,\underbrace{AKCJA\_NIE})?$$

Powyższy schemat można odnieść do przykładowej reguły:

Jeśli
SK_SW

data jest jest mniejsza od '01-01-2019' lub data jest większa niż '01-06-2019'
WARUNEK

wtedy
SK_KW

zgłoś wyjątek „Data spoza dopuszczonego przedziału."
AKCJA_TAK

w przeciwnym wypadku
SK_SAN

sprawdź regułę RS-001.
AKCJA_NIE