Politechnika Łódzka

Wydział Elektrotechniki, Elektroniki, Informatyki i Automatyki

Instytut Informatyki Stosowanej

**PRACA DYPLOMOWA MAGISTERSKA**

Ekstrakcja informacji z utworu literackiego

Extracting essential data from a literary work

Paweł Wolski

234014

Dr hab. Szymon Grabowski, prof. PŁ

Łódź, luty 2020

*Składam serdeczne podziękowania dr hab. Szymonowi Grabowskiemu za poświęcony czas i pomoc w przygotowaniu niniejszej pracy*

Streszczenie:

Słowa kluczowe:

Summary:

Keywords:

Spis treści

[1. Wstęp 8](#_Toc70265507)

[2. Cel i zakres pracy 8](#_Toc70265508)

[3. Algorytmy przetwarzające tekst 8](#_Toc70265509)

[3.1. Historia NLP 8](#_Toc70265510)

[3.2. Podstawowe operacje 8](#_Toc70265511)

[3.3. Inne podobne rozwiązania 8](#_Toc70265512)

[4. Opis środowiska i aplikacji 8](#_Toc70265513)

[4.1. Język Python 8](#_Toc70265514)

[4.2. Framework spacy 8](#_Toc70265515)

[4.3. Framework pyqt 8](#_Toc70265516)

[5. Opis metryk 8](#_Toc70265517)

[5.1. Opis stylów autorskich 8](#_Toc70265518)

[5.2. Porównanie zebranych danych 8](#_Toc70265519)

[5.3. Wnioski wyciągnięte z książek 8](#_Toc70265520)

[6. Projekt aplikacji do ekstrakcji informacji z dzieła literackiego 8](#_Toc70265521)

[6.1. Założenia aplikacji 8](#_Toc70265522)

[6.2. Wymagania aplikacji 8](#_Toc70265523)

[6.3. Obsługa aplikacji 8](#_Toc70265524)

[6.4. Zarys implementacji 8](#_Toc70265525)

[6.5. Opis aplikacji 8](#_Toc70265526)

[6.6. Testy aplikacji 9](#_Toc70265527)

[7. Wnioski 9](#_Toc70265528)

[7.1. Wnioski badawcze 9](#_Toc70265529)

[7.2. Drogi rozwoju aplikacji 9](#_Toc70265530)

[Literatura 10](#_Toc70265531)

[1. Lech Pijanowski, „Przewodnik gier", Wydawnictwo Iskry, 1978 10](#_Toc70265532)

[2. Rob Miles, „Microsoft XNA Game Studio 4.0”, Wydawnictwo Helion, 2012 10](#_Toc70265533)

[3. Steven John Metsker, „C#. Wzorce projektowe”, Wydawnictwo Helion, 2005 10](#_Toc70265534)

[4. Stanisław Wrycza, Bartosz Marcinkowski, Krzysztof Wyrzykowski, „Język UML 2.0 w modelowaniu systemów informatycznych”, Wydawnictwo Helion, 2006 10](#_Toc70265535)

[5. Strony internetowe: 10](#_Toc70265536)

[6. Gry karciane: 10](#_Toc70265537)

[Spis ilustracji 11](#_Toc70265538)

[**OŚWIADCZENIE** 15](#_Toc70265539)

# Wstęp

Rozwój technologiczny sprawił, że w dzisiejszych czasach komputer jest potrzebny w każdej dziedzinie życia. Jest odpowiedzialny za wiele aspektów w funkcjonowaniu każdego człowieka, od pracy zawodowej, edukacje poprzez spędzanie czasu wolnego jak i również rozwój osobisty. Każda z wymienionych dziedzin jest rozważana oraz niesie ze sobą wiele dróg rozwoju. Ciągnie to za sobą rozwój wielu platform, aby jak najbardziej ułatwić życie człowieka, stawiając przed nim coraz nowsze wyzwania.

Według tytułu pracy, o którym mowa wyżej jedną z dziedzin rozwoju osobistego człowieka jak i spędzania wolnego czasu jest czytanie książek. Rozwija to umiejętności czytania ze zrozumieniem jak i pobudza wyobraźnie. Z biegiem czasu przybywa coraz więcej książek i autorów. W dzisiejszych czasach bardzo też popularne stało się pisanie książek przez zwykłych ludzi, dlatego ilość dzieł znacząco wzrosła. Człowiek z bardzo dużą ilością informacji do przetworzenia na co dzień nie jest w stanie przeczytać każdej książki. Ponadto taka analiza jej fragment po fragmencie zajęłaby dużo czasu i wielu ekspertów nie byłoby w stanie poradzić sobie z każdym tytułem na bieżąco.

W pracy podjęto próbę analizy i stworzenia systemu, który by automatycznie przetwarzał informacje znajdujące się w dziełach literackich, dając użytkownikowi końcowe informacje do przeanalizowania i wyciągnięcia wniosków opisujących daną książkę, ale także styl autora jaki został w niej użyty. Jej celem jest stworzenie aplikacji wyciągającej najważniejsze informacji z danego dzieła literackiego, statystycznych wartości gramatycznych, podzielonych także na lokalne części w przypadku książki, w której występują rozdziały. Końcowy użytkownik zamiast czytać książkę miałby wynik jej analizy przed sobą.

# Cel i zakres pracy

Celem pracy jest stworzenie aplikacji na komputery, która będzie wyciągała najważniejsze informacje zawarte w dziele literackim, mogące stanowić metryki do opisu stylu dla danego autora. Zakres pracy obejmuje:

* Zaczerpnięcie informacji z książek o tematyce tworzenia aplikacji w języku Python, aplikacji okienkowych oraz pokrycia testem kodu
* Przegląd literatury o tematyce przetwarzania maszynowego języka naturalnego oraz dostarczonych bibliotek do języka Python
* Wybór metody tworzenia aplikacji, z przygotowaniem logiki
* Implementacja danego rozwiązania, obiektów metod i odpowiedniego interfejsu użytkownika
* Zadbanie o odpowiednią jakość kodu i pokrycie go testem
* Analizę zebranych danych, mogących określić styl danych autorów, dla których były prowadzone badania

# Algorytmy przetwarzające tekst

Wiele nowoczesnych algorytmów skupia się na przetwarzaniu tekstu. Dana struktura mże mieć różne właściwości albo zapisy i jest wielokrotnie przetwarzana przez każdego programistę w trakcie jego pracy. Brakuje uniwersalnej definicji NLP (Natural Language Processing). Definicja słownikowa definiuje to jako analizę tekstu przez komputer i następnie generowanie go w języku naturalnym, zrozumiałym dla człowieka. Mimo iż definicja jest poprawna, nie zawsze oddaje całość obrazu algorytmów zajmujących się przetwarzaniem tekstu.

Pierwszym aspektem jest przetwarzanie tekstu, gdy odbiorcą danych w postaci wejściowej języka naturalnego jest komputer. Drugim natomiast aspektem jest generowanie języka, o ile komputer jest w stanie prezentować dane cyfrowe w postaci komunikatu przedstawionego w języku naturalnym. Mimo iż mają one wiele wspólnego w zakresie metodologii i stosowanych technik przetwarzania, jest to bardziej adekwatna, rozwinięta i złożona domena, podczas gdy generowanie dodatkowo potrzebuje celu określania strategii generacji języka.

Można wyróżnić podstawowe domeny Natural Language Processing na podstawie podejścia zastosowanego do analizy danych: Symboliczne NLP oraz Statystyczne NLP. Pierwsze podejście opiera się na wykorzystaniu określonych algorytmów i reguł w oparciu o analizę zjawisk językowych oraz schematy reprezentacji wiedzy. Był to sposób używany we wczesnych wersjach NLP do kategoryzowania tekstu, rozwiązywania niejasności oraz ekstrakcji informacji. Drugie podejście opiera się na Statystycznym NLP czyli wykorzystaniu technik matematycznych do trenowania modelu z dużą liczną tekstów oraz dużymi etykietami zwanymi także korpusami. Zjawiska językowe są modelowane także na podstawie rzeczywistych przykładów Ma to podejście statystyczne w takich obszarach, jak analizowanie i rozpoznawanie mowy, tagowanie części mowy i wyszukiwanie i tłumaczenie danego tekstu maszynowo.

Inny podział zależy od technik, które stosuje się do analizy danych. Dane techniki są powiązane z działami językoznawstwa. Jedną z nich jest analiza morfologiczna, która w celu zbadania danych morfemów zawartych w słowach używa fragmentów o uniwersalnym znaczeniu. Drugie podejście oparte na analizie leksykalnej polega na interpretacji znaczenia słowa i części mowy. Kolejnym podejściem zajmującym się wykrywaniem struktury gramatycznej zdania oraz określenie zależności między słowami jest analiza składniowa. Następnym przykładem jest analiza semantyczna, która ma na celu wyczucie ujednoznacznia, gdy sprawdzane słowo ma wiele znaczeń. Dokonuje się tego poprzez badanie relacji między znaczeniami słowa w zdaniu. Analiza pragmatyczna jest stosowana do wyszukiwania informacji z otaczającego kontekstu. Należy wtedy przeanalizować tekst, aby wywnioskować dodatkowe znaczenie za nim. Czasami jest niemożliwe, aby określić, do kogo odnoszą się zaimki osobowe, bez wykorzystania specjalnej metody. Analiza fonologiczna jest używana tylko w obszarze rozpoznawania mowy NLP. Dotyczy ze zmianą fal dźwiękowych w tekstach reprezentujących słowa.

Ponieważ NLP ma na celu osiągnięcie ludzkiego zachowania można je uznać za jedną z kategorii wykorzystania Sztucznej Inteligencji. We wczesnej fazie NLP było określane jako naturalne zrozumienie języka, jednak fakt, że obecne systemy można sparafrazować za pomocą wprowadzanego tekstu i odpowiedzi na pytanie na jego temat, ale można także wyciągnąć informacje oraz wnikliwe wnioski z niego, co oznacza, że ludzki poziom poznania nadal pozostaje poza zasięgiem NLP, co może być w przyszłości jego ostatecznym celem, do którego będzie dążył.

## Historia NLP

Idea komputerowego przetwarzania języka pojawiła się już we wczesnych epokach informatyki. W latach 50. słynny komputerowy naukowiec Alan Turing, uważany za twórcę sztucznej inteligencji, próbował odpowiedzieć w swojej pracy na pytanie „Can Machines think?”. Dokładnie nie udało mu się określić jaki jest akt myślenia przez maszynę, dlatego zaproponował odpowiednie rozwiązanie problemu, który mógłby być analogiczny do pierwotnego. Zasugerował, że sztuczna inteligencja była by w stanie oszukać ludzkiego sędziego, aby uwierzył, iż maszyna może być człowiekiem. Orzeczenie musiało być wydane na podstawie nawiązanej między nimi rozmowy. Dlatego podstawowym wymaganiem jakie musiałby spełniać w takiej sytuacji komputer była zdolność do interpretowania języka w sposób naturalny.

Pierwsze próby wykorzystania komputera do badań nad analizą języka zostały podjęte w latach 50. Następstwa drugiej wojny światowej, po której w historii nastąpiła zimna wojna, zwiększyły zapotrzebowanie na rosyjskich i niemieckich tłumaczy, którzy pomogli by w tłumaczeniu istotnych dla brytyjskich i amerykańskich naukowców dokumentów oraz rządów. Badania prowadzone w tym okresie miały na celu automatyzację pracy tłumaczy i skoncentrowanie się na tłumaczeniu maszynowym NLP. Od czasu osiągnięcia zbieżnej technologii z zainteresowaniem rządu USA, badania były w dużej mierze dotowane przez pieniądze państwowe.

Najwcześniejszym podejściem do tłumaczenia maszynowego tekstów było użycie automatycznego słownika, aby przetłumaczyć każe słowo z osobna i podporządkować je zgodnie z zasadami języka docelowego. Wielokrotne próby informatyków zakończyły się niepowodzeniem. Napotkali problemy ze złożonością leksykalną i gramatyczną oraz wieloznacznością pewnych słów. Nawe dwujęzyczni naukowcy mieli trudności z zastosowaniem swojej wiedzy w dziedzinie programowania. Pomoc lingwistów była niezbędna do pomyślnego przeprowadzenia badań. Pomimo obiecujących postępów w rozwoju językoznawstwa teoretycznego i algorytmów przetwarzania języka, naukowcom nie udało się osiągnąć zadowalających efektów, ponieważ zasady językowe, które okazały się znacznie bardziej złożone i skomplikowane niż przypuszczano, a ówczesne komputery nie były zbyt mocne, aby poradzić sobie z tłumaczeniem w rozsądny sposób, a czas na rozważenie przydatności takich metod był krótki. Yehoshua Bar-Hillel, pracownik MIT w dziedzinie tłumaczenia maszynowego twierdził ze w dającej się przewidzieć przyszłości, pełne zautomatyzowanie tłumaczenia w wysokiej jakości nie będzie możliwe, ponieważ komputerom nadal brakuje ludzkich umiejętności, takich jak między innymi zrozumienie kontekstu i analiza znaczenia, aby prawidłowo wybrać tłumaczenie danego słowa, jeśli ma więcej niż jedno znaczenie.

ALPAC (Komitet Doradczy ds. Automatycznego Przetwarzania Języka) wydał w 1966 roku raport, w którym wskazywał na problemy jakościowe i szybkościowe oraz kosztowe MT. Ta opinia polegała na tym, iż po 8 latach pracy nad projektem na Uniwersytecie Georgetown wytworzone dokumenty wejściowe wymagały zaksięgowania. Ich korekta nie trwała dłużej i kosztowała więcej niż tradycyjne tłumaczenie tych samych dokumentów. Stwierdzono wtedy, iż bardzo mało prawdopodobne jest uzyskanie szybkiego tłumaczenie maszynowego. Dany raport doprowadził do odcięcia funduszy od rządu USA i wywarło to bardzo negatywny wpływ na badania związane z NLP do tłumaczenia maszynowego. Jednakże pomimo problemów z dotacjami, badania nad NLP nie zostały całkowicie porzucone. Kilka prototypów systemów dotyczących przetwarzania języka naturalnego zostało rozwiniętych w następnym dziesięcioleciu. W 1966 roku niemiecki amerykański naukowiec i pracownik MIT Joseph Weizenbaum stworzył program symulujący psychoanalitykę o nazwie ELIZA. Dla zadanych słów kluczowych w zdaniach wejściowych, a następnie zadawał danemu pacjentowi pytania na ich temat i zachęcił ich do dalszego wyjaśnienia tematu. Mimo prostoty pomysłu wystarczyło zachować u danego pacjenta poczucie, iż mówca jest słyszalny i rozumiany.



Zdjęcie przestawiające Josehpa przy programie eliza

W 1970 roku twórca programu SHRDLU Terry Winograd rozwinął swój system na potrzeby pracy doktorskiej. Był w stanie przetwarzać polecenia użytkownika w języku naturalnym do manipulacji wirtualnymi blokami i odpowiedzi na ich pytania. W latach 80. NLP zaczęło wychodzić z 14 letniego „bezruchu”. Wzrost mocy obliczeniowej komputerów który wystąpił w danym czasie pozwalał naukowcom spróbować nowych środków takich jak uczenie maszynowe. Wczesne próby wykorzystania algorytmów z tej dziedziny w postaci drzew decyzyjnych generowały złożone reguły, które znacznie różniły się od dotychczasowych. Zmieniło się to gdy podejście językowe zaczęło tracić na znaczeniu na rzecz statystyk. Były to obiecujące wyniki które dawały zastosowanie w modelach statystycznych podejmowania probabilistycznych decyzji na podstawie przypisywanych wag dla danych wejściowych tekstu. Takie modele uzyskały zastosowanie w dziedzinie algorytmów uczenia maszynowego na dużych tekstach z adnotacjami zwanymi też „korpusami”. Modele mogły specjalizować się, aby lepiej pasować do jednej kategorii. Ta metoda była korzystna, dostarczała mniej szerszych zasad zamiast wielu szczegółów i była lepsza w obsłudze nieznanych lub błędnych danych wejściowych.

Popularność statystycznego NLP rosła przez lata 90. równoważnie z rosnącą liczbą zasobów tekstowych, które zaczęły być publikowane w Internecie. Dostarczało to cennych danych do modelu statystycznego jako zbioru treningowego. W ciągu całej dekady zaczęły pojawiać się chatboty. W roku 1997 projekt Jabberwacky został ukończony i finalnie uruchomiony w Internecie. Jego wynik był pewnego rodzaju chatbotem, który miał na celu symulowanie języka naturalnego w humorystyczny dla użytkownik sposób. Potrafił nauczyć się nowych słów, żartów lub innych cech językowych, na podstawie wcześniej prowadzonych rozmów, które były przechowywane przez program.

W XXI wieku NLP rozwija się dalej dzięki dostępności tanich i wydajnych komputerów, popularyzacją na duży popyt oraz zastosowanie modelu big data. Modele rekurencyjnych sieci neuronowych wprowadzone w latach 90. Przez kolejne lata okazały się bardzo przydane w przetwarzaniu głosu i symulowaniu jego. W 2011 roku firma Apple przedstawiła pierwsze urządzenie oparte na sztucznej inteligencji Siri jako asystent głosowy udostępnione klientom kupującym urządzenia z tej marki.

Dziś NLP jest częścią życia każdego człowieka. Tłumaczenie maszynowe jest wbudowane często jako funkcja wyszukiwarek internetowych lub mediów społecznościowych. Strony internetowe posiadają też modele wirtualnych asystentów w postaci chatbotów, które często udzielają odpowiedzi na problemy lub często zadawane pytania.

## Podstawowe operacje

Pierwszym krokiem w kierunku przetwarzania tekstu do języka naturalnego jest segmentacja. Dany tekst jest podzielony na pojedyncze, możliwe do zidentyfikowania jednostki językowe, które stanowią część danych. Taka operacja nazywana jest tokenizacją. Zdecydowanie większość tokenów jest podzielona na wyrazy, uzyskane poprzez podzielenie tekstu spacjami lub odrzucenie danych znaków interpunkcyjnych. Istnieją jednak ważne reguły, które należy przestrzegać. W języku angielskim nieformalne negacje są pisane wspólnie z pomocniczymi. Niektóre metodologie tokenizacji mogą traktować dane słowo jako oddzielne dwa tokeny, które zatwierdzają różne funkcjonalności. Przykładowo zwrot „nie musiał” zostanie podzielone na „musiał” jako słowo pomocnicze, wskazujący czas oraz negację „nie” jako przysłówek wskazujący na zaprzeczenie. Punkt podziału wtedy nie jest ani interpunkcją ani białą spacją. Istnieją różne wyjątki dotyczące identyfikatorów i nazw własnych. Zadanie tokenizacji jest bardzo zróżnicowane w zależności od stosowanego języka i analizowanego tekstu. W języku włoskim dany przedimek określony może być powiązane z apostrofem ze słowem, które poprzedza w danej chwili. Język niemiecki charakteryzuje się długimi słowami, które należy oddzielić w osobnej analizie. W języku japońskim bądź chińskim nie używa się spacji pomiędzy słowami i należy wtedy zastosować inne zasady segmentacji specyficzne dla danego języka.

Drugim zadaniem jakie musi spełniać segmentacja jest podzielenie tekstu na zdania. Znalezienie ich utrudniają problemy podobne w przypadku tokenizacji. W zależności od podejścia, cytaty w zdaniach i po obu stronach można traktować jako oddzielne zdania.

Kolejnym krokiem jest normalizacja. Podczas przetwarzania danych tokenów, powinny być one traktowane w ten sam sposób pod względem semantycznym, mimo iż reprezentujące ich znaki są różne. Jednym z takich przykładów są wielkie litery na początku zdań. Do podstawowych czynność jakie należy w tym celu wykonać jest składanie i wyjmowanie potencjalnej interpunkcji. Analizowane słowa mogą mieć różne zakończenia mimo tych samych znaczeń. Występuje to wielokrotnie przez zastosowanie liczby mnogiej rzeczowników bądź różnych odmian czasowników. Występuje tutaj operacja nazywana procesem heurystycznym. Nie ma on uniwersalnej definicji, ale bazuje na procesie usuwania afiksów i tworzenia bazy danej ze słowa zwanego rdzeniem. Stemming określa użycie różnych końcówek dla danego słowa. Najlepszym algorytmem rdzeniowym był algorytm Martina, zastosowany dla tekstów w języku angielskim. Został on opracowany przez Martina Portera w 1979 roku. Dzięki prostej implementacji opiera się on na stopniowym usuwaniu i zastępowaniu niektórych liter na końcu słowa. W pięciu krokach ostatnie litery są usuwane i zastępowane w celi pozbycia się zakłóceń i uzyskaniu odpowiedniej formy bazowej. W zależności od ułożenia samogłosek lub spółgłosek oraz długości słowa, dane kroki są wykonywane lub pomijane. Niestety niektóre zmiany wprowadzone przez algorytm mogą spowodować zapadnięcie się słowa w wspólną formę bazową, ale o odmiennym znaczeniu. Czasami lepiej zachować tylko podstawową formę odmiennego słowa i ograniczyć redukcję, aby zachować daną część mowy. Innym razem lepiej pobierać informacje obojętne na daną część wypowiedzi i jednakowo traktować całość.

Bardziej złożony i drobiazgowy proces, którego cel jest taki sam jak Steemingu to lematyzacja. W odróżnieniu do wprowadzania słowa, otrzymany wynik jest zawsze prawdziwy. Posługuje się ona głębokim słownictwem i morfologiczną analizą, zwykle z wykorzystaniem danej bazy leksykalnej. Aby uzyskać lepsza wydajność, stosuje się dodatkowe informacje zawarte w kontekście wypowiedzi takie jak jej część albo analizowane słowo. Zasadniczą wadą lemmantyzatorów jest wolniejsze działanie i nieobsługiwanie nowych oraz niestandardowych i wymyślonych słów, które nie występują w bazie danych używanych przez algorytm.

Następnym krokiem jest znakowanie części mowy zwane też także jako znaczniki POS. Jest to proces przypisywania do każdego tokena etykiety, która odpowiada danej części mowy. Liczba przypisywanych znaczników może być też większa niż liczba części mowy występująca w analizowanym tekście i różnić się od znaczników POS. W zależności od szczegółowości podkategorii części mowy są przypisywane przez system. Znaczniki mogą się różnić podstawową i przeszłą formą czasowników lub nawet używać odpowiedniego znacznika dla bezokoliczników. Pomimo braku uniwersalnego zestawu znaczników istnieje wiele bardzo popularnych, które są powszechnie używane. Jeśli chodzi o użyty w aplikacji język angielski jest on zdefiniowany jako zestaw tagów Brown opracowany w 1979 roku na Uniwersytecie Brown zawierający 87 znaczników. Najczęściej używany jest zestaw Penn Treebank, który został uproszczony do 48 znaczników i opracowany na Uniwersytecie Pensylwanii w latach 90. Oba zestawy były częścią projektów opisywania dużej ilości kolekcji „korpusów”. Najważniejszą częścią tagowania opisanego na początku rozdziału jest ujednoznacznie w przypadku słów posiadających wiele znaczeń. Można wyróżnić dwa podejścia oparte na regułach oraz stochastyczne używane go tagowania POS. Były wykorzystane predefiniowane reguły do analizy morfologii analizowanego słowa oraz słów sąsiednich, aby określić jego przynależność do danej kategorii. Jeden z najstarszych oraz bardzo popularnych sposobów tagowania POS i opartym na jego regułach jest tagger firmy Brill. Został przeszkolony na „korpusach”, których tokeny zostały już ręcznie oznaczone i określają zasady przypisywania do danej części mowy. Podejście stochastyczne określa znacznik na podstawie jego częstotliwości oraz prawdopodobieństwa. Najbardziej podstawową metodą jest przypisywanie słowom części mowy, która jest niejednoznaczna. Inna metoda jest oparta na n-gramowym podejściu. Kiedy pojawia się słowo które może być niejednoznacze, określany jest przypisany do niego znacznik, wybierając najbardziej prawdopodobny po wprowadzeniu n poprzedzających znaczników na podstawie sekwencji n kolejnych w danych uczących. Najbardziej dokładny wynik uzyskuje się stosując model Markowa, który uwzględnia zarówno prawdopodobieństwo kolejności znaczników oraz prawdopodobieństwo przypisania jednego z możliwych znaczników do słowa.

Następnym krokiem NLP jest określenie struktury zdań i zależności między słowami. Proces wprowadzenia danej struktury składniowej dla sekwencji słów nazywana jest analizowaniem. Istnieją trzy różne typy analizowania: Shallow parsing, analiza okręgów wyborczych oraz analiza zależności. Pierwszy typ to technika grupowania tokenów w frazy, które mają tą samą funkcję co część mowy ich słowa głównego. Każda fraza składa się z głównego słowa i innych słów, które go opisują. Następnym typem, czyli analizą okręgów wyborczych jest podejściem modelowania hierarchicznej struktury zdania. Parser przetwarza zdanie za pomocą gramatyki formalnej, czyli zbioru reguł i definiuje możliwe konstrukcje w języku. Przedstawia wyniki w postaci struktury drzewa zwanego parse tree. Ostatnim typem, czyli analizą zależności jest identyfikowanie i nazywanie relacji w zdaniu. Każde słowo ma przypisywane inne słowo, z zdania które modyfikuje, zwanego też jego głową. Parser dostaje adnotacje do każdej takiej relacji ze znacznikiem zależności, który opisuje jej charakter.

Ostatnim krokiem jest analiza nastrojów. Jest to proces określenia opinii na dany temat w tekście. Identyfikuje wartości sentymentu, które można przeprowadzić na różnych poziomach oceniając uczucia wyrażane przez poszczególne słowa, zdania, fragmenty lub cały dokument. Wyróżniane są dwa rodzaje systemów analizy nastrojów: oparty na regułach i automatyczny.

Pierwszy opiera się w dużej mierze leksykonach, czyli biblioteki słów ręcznie przypisanych z wartością sentymentu. Oznaczone są punktacją za każdą biegunowość, czyli określony atrybut, czy dan termin ma opinię pozytywną czy negatywną. Drugie systemy wykorzystują podejście języka maszynowego które wymaga przeszkolenia, danego modelu klasyfikatora na dużej liczbie wystąpień oznakowanego dokumentu. Jego zaletą jest przewidywanie z większym prawdopodobieństwem uczuć związanych z nowymi wyrażeniami bądź slangiem.

## Podobne rozwiązania

Rozwijanie wyciągania przydatnych informacji z dużej ilości tekstu jest powszechnie używane. Jednym z przykładów może być Amazon Textract, która jest usługą uczenia maszynowego, wyodrębniająca tekst, pismo odręczne i dane skanowanych dokumentów. Obecnie wiele firm ręcznie wyodrębnia dane z formularzy i tabel. Firma stara się mieć nacisk na zeskanowane dokumenty. Może to służyć szybkiemu zautomatyzowaniu przetwarzania dokumentów i podejmować działanie na uzyskanych informacjach. Dane są bardzo szybko wyodrębniane i przetwarzane.

Użycie NLP jest wykorzystane w tłumaczeniu dużych tekstów w Internecie. Bardzo szybka analiza znaczeniowa jest też ważna w przypadku właściwego przetłumaczeniu teksu na danym portalu internetowym, aby nie wprowadzić użytkownika w błąd. Jest wykorzystane także w celu asystentów cyfrowych dedykowanych dla potencjalnych marek. Doradzają oni i odpowiadają na pytania danego użytkownika w danej dziedzinie. Przykładem jest firma Apple, która stworzyła Siri jako asystenta dla każdego konsumenta z systemem dostarczonym przez tą firmę.

Twiggle wdraża jedne z najbardziej zaawansowanych rozwiązań w dziedzinie nauczania maszynowego, sztucznej inteligencji oraz Natural Language Processing. Stworzona jest, aby wspierać eCommerce, które jest coraz bardziej odpowiednie oraz angażujące dla klientów. Firma skupia się, aby dotrzeć do nich i do sedna interakcji z nimi. Oferuje rozwiązania dla technologii wyszukiwania opartych na interakcjach między ludzkich oraz dokładnego zrozumienia struktury językowej. Boty mogą naśladować zachowanie prawdziwych sprzedawców. Zrozumieć czego potrzebują klienci bez względu na to o co pytają.

Firma TaskUs dostarcza rozwiązania związane z obsługą klienta nowej generacji. Opiera się na zrozumieniu języka naturalnego i dostarcza narzędzie do jego przetwarzania. Oferta firmy wykorzystuje zautomatyzowane procesy, aby dostarczyć rozwiązania klientom szybciej niż kiedykolwiek wcześniej. TaskUs wyróżnia się także możliwością szybkiego skalowania.

Wykorzystanie NLP stosuje się także w dziale zarządzania zasobami ludzkimi (HR), który jest niekiedy żmudnym, powolnym i często niedokładnym zadaniem. Oferuje ono zestaw narzędzi do analizy nastrojów ludzkich, identyfikacji potencjału i improwizacji kompetencji. Może być wykorzystane w przetwarzaniu listów motywacyjnych, danych osobowych oraz CV. Minimalizuje też ludzkie uprzedzenia w podejmowaniu decyzji.

Kolejną dziedziną wykorzystania przetwarzania języka naturalnego jest system opieki zdrowotnej. Aplikacja MedInReal usprawnia trudności związane z poświęceniem odpowiedniej ilości czasu danemu pacjentowi. Asystenci cyfrowi przejmują papierkowe obowiązki, z którymi do czynienia mają przeważnie lekarze. Ułatwiają też komunikację na temat ich leczenia i starają się zrozumieć ich potrzeby.

Analiza dużych zbiorów danych (Big Data) opiera się na wykorzystaniu NLP poprzez wyszukiwanie danych fraz i informacji w dużej kolekcji zbiorów tekstowych. Szybszy i dokładniejszy dostęp do informacji przyśpiesza wszystkie inne procesy za pomocą wyszukiwania, używając zapytań własnymi słowami, dzięki temu użytkownicy nie są już ograniczeni wyborem lub znajomością „właściwych” słów kluczowych. Analiza tekstów jest bardziej skoncentrowana na przeszukiwaniu go.

# Opis środowiska i aplikacji

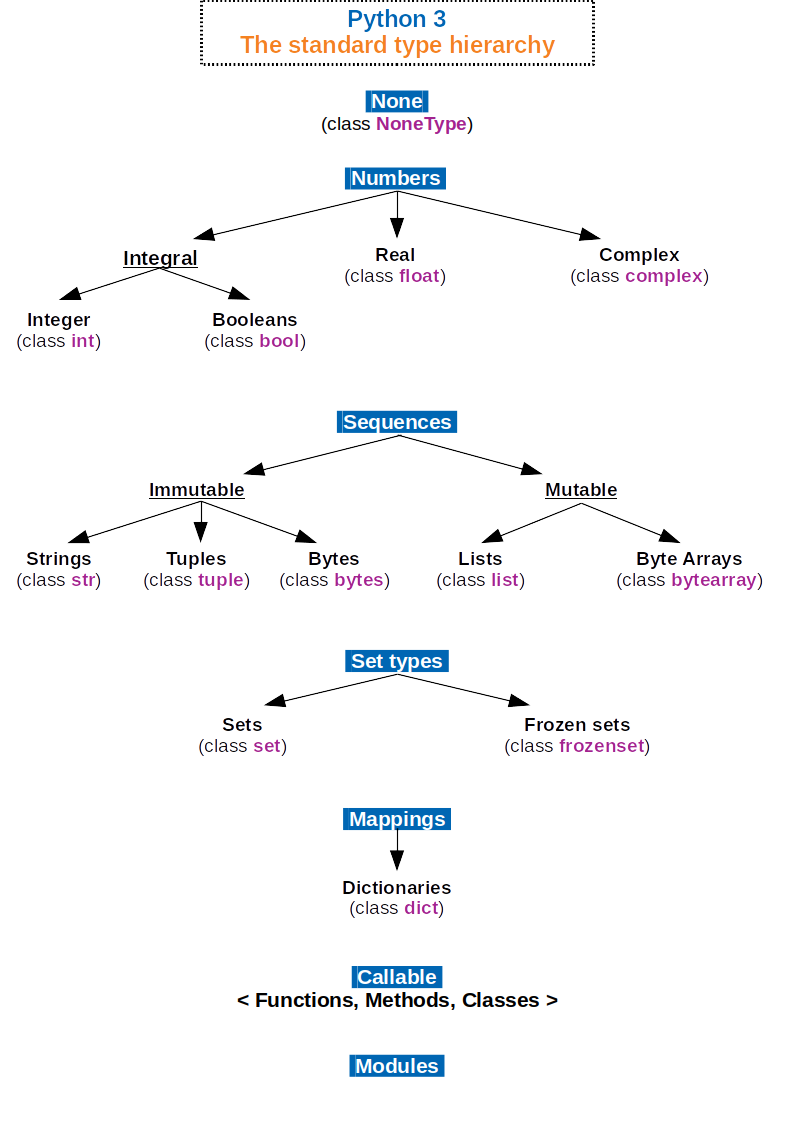
Do stworzenia aplikacji Machine Book Extract, opisanej w dalszej części pracy, został użyty język Python w wersji 3.7.9 oraz dwa frameworki: spacy 3.0.5 i pyqt w wersji 5.0. Całość kodu była przetrzymywana na zdalnym, prywatnym repozytorium. Dany język oraz frameworki zostały wykorzystane z powodu swojego przeznaczenia związanego z przetwarzaniem języka naturalnego oraz interfejsem użytkownika. Poniższa aplikacja jest otwarta na rozszerzenia oraz nowe funkcjonalności. Jest zamknięta na modyfikacje. W celu ergonomii ekranu danego użytkownika użyto środowiska aplikacji okienkowych. Program został pokryty testami użytkownika oraz wykorzystany do wstępnej analizy danych.

## Język Python

Python to wysokiego poziomu język o wysoce rozbudowanym pakiecie bibliotek standardowych. Jego przewodnią ideą jest klarowność oraz czytelność kodu. Cechuje się zwięzłością i przejrzystością. Pyton wspiera także różne paradygmaty programowania. Między innymi imperatywny i obiektowy. W mniejszym stopniu funkcyjny. Posiada pełen system typowania, który jest także dynamiczny i automatyczne zarządzanie pamięcią. Jest podobny do Perl i Ruby pod tym względem. Jest bardzo często używany jako język skryptowy i jest dostępny na wielu systemach operacyjnych.

Język Python powstał w latach 90. jako następca ABC – języka stworzonego w Centrum Matematyki i Informatyki w Amsterdamie. Twórcą Python’a jest Guido van Rossum. Nazwa języka, mimo iż często mylona nie pochodzi od nazwy zwierzęcia, ale od serialu komediowego, który był emitowany w latach 70. na stacji BBC o nazwie „Latający Cyrk Monthy Pythona”. Projektant bowiem był wielkim fanem tego serialu. Wersja 1.2 była ostatnią wydaną w CWI. Następnie praca nad nim była kontynuowana w Wirginii, aż do przeniesienia się twórcy do BeOpen i założenia zespołu BeOpen PythonLabs. Tam został tworzony i rozwijany do dziś Python w wersji 2.0.

Filozofia Pythona opiera się na kilku paradygmatach. Podobnie do języka C++ nie wymusza jednego stylu w jakim dany użytkownik może programować. Możliwe w tym języku jest także programowanie obiektowe, strukturalne i funkcyjne. Typy są sprawdzane dynamicznie, a za zarządzanie pamięcią odpowiada garbage colleciton. Hierarchia typów w języku Python została przedstawiona na Rys. 4.1.



Python ponadto operuje na systemie typów, w których wszystko jest obiektem. Typy wbudowane nie są właściwie klasami, a dana klasa może dziedziczyć z dowolnego typu. Można zatem dziedziczyć z napisów czy słowników. Ponadto możliwe jest dziedziczenie wielokrotne. Jest to podobne jak w języku C++. Typy można odczytywać i porównywać. Atrybuty obiektu mogą być też pobierane. Każda klasa może być też opatrzona dynamiczną dokumentacją w kodzie źródłowym. Dany język posiada także bardzo dobrze rozwiniętą bibliotekę standardową przeznaczoną do wykonywania wielu zadań. Metody danej biblioteki można uzupełnić modułami napisanymi w C lub Pythonie Jest ona też szczególnie dostosowana do tworzenia systemów sieciowych, jako iż obsługuje dużą ilość protokołów. Dostarczone moduły są też w stanie tworzyć systemy obsługujące interfejsy użytkownika, obróbki wyrażeń regularnych oraz skierowane też na serwery WWW. Bibliotek może być też stosowana na wielu platformach, dzięki czemu duże aplikacje mogą być uruchamiane na dużej ilości dedykowanych systemów operacyjnych.

Interpreter Pythona posiada też tryb interaktywny. Mona tam wprowadzać wyrażenia z terminala i otrzymywać wyniki od razu. Może to ułatwić naukę programowania we wczesnym jej etapie. Standardowy moduł nie jest jednak zbyt wygodny i braku wielu w nim funkcji. Przeznaczoną biblioteką do testów jest unittest, który zostanie szerzej opisany w dalszych rozdziałach.

## Framework spaCy

Framework spaCy jest używany przede wszystkim w wykorzystaniu NLP do celów programistycznych. Wspierany jest przez ponad 64 języki. Wykorzystuje wielozadaniowe uczenie się z wstępnie wytrenowanymi transformacjami oraz asynchroniczność w przetwarzaniu słów i zdań. Tokenizacja w nim jest mocno motywowana językowo. Stworzone zostały specjalne komponenty do rozpoznawania jednostek słów, zależności, oznaczania części mowy, analizy morfologicznej, łączenia jednostek i lematyzacji. Jest łatwo rozszerzalny dzięki atrybutom i niestandardowym komponentom.

## Framework pyqt

# Opis metryk

## Opis stylów autorskich

## Metryki trudności czytanego tekstu

Indeks czytelności Flescha

## Porównanie zebranych danych

Po procesie tworzenia aplikacji przystąpiono do wyciagnięcia informacji na podstawie kilku popularnych lektur znajdujących się na stronie: <https://www.gutenberg.org/>. Poniżej zostało przedstawione porównanie informacji wyciągniętych z programu, wraz z wnioskami jakie za sobą niosą wraz z informacjami zawartymi w streszczeniach szczegółowych. Podział analizy został przedstawiony następująco, na początku zostaną omówione ogólne informacje jakie niesie za sobą dany utwór literacki, a następnie przedstawione zostaną statystyki rozdziałowe wraz z zawartymi w nich informacjami.

Jako pierwszą książką poddaną analizie została książka pod tytułem „Opowieść wigilijna” Karol Dickensa. Książka opowiada o samotnym bohaterze imieniem Scrooge, który ze swojej natury jest skąpcem i pragnie jak to każdego roku spędzać święta Bożego Narodzenia samemu, gdyż uważa to za święto ludzi, którzy nie potrafią niczego osiągnąć w życiu. Tego rodzaju święta nigdy nie traktował zbyt poważnie.

Analiza książki zawiera podstawowe wartości definiujące długość książki w znakach i słowach, ilość zdań, oraz ich statystyki w podobnych kryteriach. Ponadto mamy też wartości definiujące ilości dialogów oraz statystyki z nimi związane. Do dyspozycji jest również czas zawarty w książce oraz informacje o częstości występowania bohaterów w tekście. Dane podstawowe zostały zebrane w tabelce Tab. 5.3.1.

|  |  |
| --- | --- |
| *Nazwa* | *Wartość* |
| Długość książki w znakach | 164530 |
| Długość książki w słowach | 26550 |

*Tab. 5.3.1 Podstawowe statystyki wyciągnięte z książki „Opowieść wigilijna” Karol Dickensa*

Z podstawowych informacji zawartych w książce można zauważyć, że książka ma około 27 tys. Słów i ponad 160 tys. znaków. Można wywnioskować, że czas jej przeczytania to od kilku do kilkunastu godzin. Książka zawiera 96 stron. Przy czytaniu 10 stron na godzinę, według kalkulatora szybkości czytania zawartego na stronie <https://www.omnicalculator.com/everyday-life/reading-time> można oszacować że czas przeczytania „Opowieści wigilijnej” to 9 godzin i 35 minut. Następnym ważnym aspektem jest statystyka dialogowa. Ich wartości zostały przedstawione w tabelce Tab. 5.3.2.

|  |  |
| --- | --- |
| *Nazwa* | *Wartość* |
| Liczba dialogów | 682 |
| Średnia liczba słów w dialogach | 17 |
| Średnia liczba znaków w dialogach | 71 |
| Liczba słów w długich dialogach | 148 |
| Liczba słów w krótkich dialogach | 534 |

*Tab. 5.3.2 Podstawowe statystyki wyciągnięte z książki „Opowieść wigilijna” Karol Dickensa*

Ogólna ilość dialogów w książce to koło 700. Średnia liczba słów w dialogu to 17, a znaków to 71. Tekst jest w miarę dynamiczny, dzięki czemu szybko się go czyta. Często występują też dialogi między różnymi osobami np. głównym bohaterem i duchami zawarte w tekście. W tekście dominują krótkie dialogi, zatem można wywnioskować, że jest tam zawarta szybka wymiana zdań między bohaterami. Ogólny wykres ilości dialogów w książce został zawarty na wykresie Wyk. 5.3.1.

*Wyk. 5.3.1 Statystyka dialogowa dla książki „Opowieść wigilijna” Karol Dickensa*

Ostatnim ważnym aspektem jest stosunek czasów zawartych w książce. Definiuje on czas akcji zawarty w książce. Ponadto można wyciągnąć wnioski w jakim stosunku tekst był opisowy i jak zmieniał się czas akcji. Ogólna statystyka podziału czasów została przedstawiona w tabelce Tab. 5.3.3 i wykresie *Wyk. 5.3.2*. Pokazuje ona m. in dominację czasu przeszłego nad teraźniejszym. Oznacza to, że w książce mamy wiele razy odniesienie do retrospekcji lub wydarzeń występujących przed rozpoczęciem głównego wątku akcji. Jest to zgodne, gdyż w książce wiele razy jest odwołanie do osób bądź miejsc występujących w przeszłości.

|  |  |
| --- | --- |
| *Nazwa* | *Wartość* |
| Liczba czasowników | 3485 |
| Liczba czasowników w czasie teraźniejszym | 1115 |
| Liczba czasowników w czasie przeszłym | 2370 |

*Tab. 5.3.3 Statystyki czasowe wyciągnięte z książki „Opowieść wigilijna” Karol Dickensa*

*Wyk. 5.3.2 Statystyki czasów w wykresie kołowym dla książki „Opowieść wigilijna” Karol Dickensa*

Ponadto program opisany w następnych rozdziałach jest w stanie wyciągnąć bohaterów z danej książki oraz częstotliwość ich występowania w książce w stosunku do ilości słów. Wynik kilkuprocentowy może świadczyć o głównych bohaterach tekstu. Program jest w stanie wyciągnąć dowolną ilość głównych bohaterów w zależności od algorytmu użytego w środku. Dany użytkownik musi sam ocenić poprawność wyciągniętych informacji. Natomiast w wielu przypadkach wśród ilości propozycji znajdują się główni bohaterowie książki. W poniższej tabelce zostali przedstawieni najważniejsi bohaterowie Wyciągnięci bohaterowie po odrzuceniu tzw. szumów zostali przedstawieni w tabelce Tab. 5.3.4 i są to główni bohaterowie w książce.

|  |  |
| --- | --- |
| *Nazwa* | *Częstotliwość występowania w książce (%)* |
| Scrooge | 1.2 |
| Marley | 0.23 |
| Fezziwig | 0.73 |

*Tab. 5.3.3 Główni bohaterowie książki „Opowieść wigilijna” Karol Dickensa*

Prócz statystyk globalnych dla danej książki można wyciągnąć statystyki lokalne, jeżeli dana książka dzieli się na rozdziały. Rozważana książka dzieli się na pięć rozdziałów, które zostaną przeanalizowane według wyciągniętych informacji oraz informacji zawartych w streszczeniu szczegółowym. Informacje zawarte o statystykach rozdziałowych zostały zawarte w tabeli Tab. 5.3.5.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Ilość słów** | **Ilość znaków** | **Ilość słów w zdaniu** | **Ilość znaków w zdaniu** | **Ilość zdań** |
| **1** | 5906,00 | 36340,00 | 13,69 | 73,64 | 446,00 |
| **2** | 5697,00 | 34239,00 | 13,83 | 72,18 | 421,00 |
| **3** | 7593,00 | 46364,00 | 17,26 | 93,51 | 447,00 |
| **4** | 4912,00 | 28919,00 | 11,96 | 61,40 | 435,00 |
| **5** | 2206,00 | 13120,00 | 9,49 | 47,03 | 248,00 |
|  |  |  |  |  |  |
|  | **Ilość Dialogów** | **Średnia ilość słów w dialogach** | **Średnia ilość znaków w dialogach** | **Ilość długich dialogów** | **Ilość krótkich dialogów** |
| **1** | 137,00 | 18,83 | 93,10 | 19,00 | 118,00 |
| **2** | 120,00 | 16,72 | 80,16 | 27,00 | 93,00 |
| **3** | 110,00 | 14,65 | 69,75 | 22,00 | 88,00 |
| **4** | 134,00 | 21,78 | 103,97 | 29,00 | 105,00 |
| **5** | 66,00 | 19,05 | 89,26 | 17,00 | 49,00 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  | **Procent długich dialogów** | **Procent krótkich dialogów** | **Ilość przymiotników** | **Procent przymiotników** | **Średnia długość przymiotników** |
| **1** | 13,87 | 86,13 | 384,00 | 6,50 | 10,95 |
| **2** | 22,50 | 77,50 | 387,00 | 6,79 | 11,04 |
| **3** | 20,00 | 80,00 | 560,00 | 7,38 | 15,97 |
| **4** | 21,64 | 78,36 | 310,00 | 6,31 | 8,84 |
| **5** | 25,76 | 74,24 | 112,00 | 5,08 | 3,19 |
|  |  |  |  |  |  |
|  | **Czasowniki w czasie przeszłym** | **Czasowniki w czasie teraźniejszym** | **Czasowniki ogólnie** | **Procent czasu przeszłego do teraźniejszego** | **Procent czasu teraźniejszego do przeszłego** |
| **1** | 444,00 | 324,00 | 768,00 | 42,19 | 57,81 |
| **2** | 455,00 | 268,00 | 723,00 | 37,07 | 62,93 |
| **3** | 598,00 | 346,00 | 944,00 | 36,65 | 63,35 |
| **4** | 416,00 | 293,00 | 709,00 | 41,33 | 58,67 |
| **5** | 158,00 | 134,00 | 292,00 | 45,89 | 54,11 |

*Tab. 5.3.5 Statystyki lokalne dla rozdziałów wyciągnięte z książki „Opowieść wigilijna” Karol Dickensa. Kolory w kolumnach odpowiadają kolejno: zielony – największa wartość, czerwony – najgorsza wartość*

Ilość słów i znaków rośnie i jest kulminacyjna w trzecim rozdziale. Najprawdopodobniej jest to najważniejszy rozdział, gdzie występuje kulminacja akcji w książce. Co za tym idzie, można rozważyć, iż Charles Dickens lub budować napięcie w swoich książkach. Ponadto ilość słów i znaków w zdaniach najgorzej wypada w ostatnich rozdziałach. Może to być spowodowane tym, iż jest tam epilog książki lub napięcie i akcja opada już. Co więcej najwięcej dialogów występuje w czwartym rozdziale. Jest tam przedstawiona rozmowa Scrooge’a razem z Duchem Przyszłych Świąt. Kupcy rozmawiają tam bardzo dużo o pieniądzach, które liczą się dla głównego bohatera przede wszystkim. Najkrótsze dialogi występują na początku książki, gdzie następuje opis głównego bohatera, jego byłego wspólnika Jakuba Marleya oraz krótkie rozmowy razem z innymi bohaterami. W książce dominuje także czas teraźniejszy nad przeszłym. W pierwszym rozdziale znajduje się opis i wstęp do książki i różnice czasowe są tam nieznaczne. Natomiast w drugim i trzeci rozdziale, gdzie rozpoczyna się akcja utworu i rozmowa z duchami widać przewagę czasu teraźniejszego nad przeszłym.

Ostatnim ważnym czynnikiem jest trudność czytanej książki. W poprzednim rozdziale zostały opisane trzy zmienne definiujące trudność czytanego tekstu. Dla „Opowieści wigilijnej” indeks czytelności Flescha ma następującą wartość:

Co oznacza, że tekst jest łatwy do przeczytania dla uczniów w wieku 13-15 lat. ,,Opowieść wigilijna” znajduje się w liście lektur dla klasy VII, więc wartość wskaźnika jest zgodna z założeniami. Dana książka dla prostej miary Gobbledygooka (SMOG) otrzymuje wartość:

Dana wartość oznacza, że tekst można przeczytać po około 5 latach nauki w szkole. Lektura jest dostępna dla klasy siódmej, czyli trzynastego roku życia. Więc jest to dolna granica pierwszego wskaźnika, co za tym idzie wartość prawdziwa może leżeć w połowie między tymi dwoma wartościami. Ostatni wskaźnik formuły Gunning fog również określa ilość lat po których tekst będzie zrozumiały, ale bierze pod uwagę inne wartości. Dla opowieści wigilijnej ma on wartość:

Połowa tej wartości jest równa 4.66 i tyle wynosi liczba lat po której przeczytanie danej książki nie powinno sprawić problemu i jest zbliżona do wartości współczynnika SMOG, zatem oba współczynniki są zgodne.

# Projekt aplikacji do ekstrakcji informacji z dzieła literackiego

## Założenia aplikacji

## Wymagania aplikacji

## Obsługa aplikacji

## Zarys implementacji

## Opis aplikacji

## Testy aplikacji

# Wnioski

## Wnioski badawcze

## Drogi rozwoju aplikacji

# Literatura

### Lech Pijanowski, „Przewodnik gier", Wydawnictwo Iskry, 1978

# Spis ilustracji

Załącznik nr 1

do Procedury antyplagiatowej prac dyplomowych

w Politechnice Łódzkiej

Łódź, dn. ......……………….

**OŚWIADCZENIE**

Paweł Wolski

………………………………………………………….

*(Imię i nazwisko studenta)*

Orla 3 m. 10, 90-317 Łódź

………………………………………………………….

*(Adres)*

209509

………………………………………………………….

*(Nr albumu)*

Wydział Elektrotechniki, Elektroniki,

Informatyki i Automatyki

………………………………………………………….

*(Jednostka organizacyjna prowadząca studia)*

Informatyka

………………………………………………………….

*(Kierunek studiów)*

Studia I stopnia inżynierskie stacjonarne

………………………………………………………….

*(Poziom kształcenia i forma studiów)*

Oświadczam, że poinformowano mnie o zasadach dotyczących kontroli oryginalności pracy dyplomowej w Jednolitym Systemie Antyplagiatowym.

.………………………….......

*(Podpis studenta)*

Załącznik nr 2

do Procedury antyplagiatowej prac dyplomowych

w Politechnice Łódzkiej

Łódź, dn. ......……………….

**OŚWIADCZENIE**

**o samodzielności wykonania i oryginalności pracy dyplomowej**

Paweł Wolski

………………………………………………………….

*(Imię i nazwisko studenta)*

Orla 3 m. 10, 90-317 Łódź

………………………………………………………….

*(Adres)*

209509

………………………………………………………….

*(Nr albumu)*

Wydział Elektrotechniki, Elektroniki,

Informatyki i Automatyki

………………………………………………………….

*(Jednostka organizacyjna prowadząca studia)*

Informatyka

………………………………………………………….

*(Kierunek studiów)*

Studia I stopnia inżynierskie stacjonarne

………………………………………………………….

*(Poziom kształcenia i forma studiów)*

Świadomy/a odpowiedzialności karnej za składanie fałszywych zeznań oświadczam, że:  
1. Przedkładana praca dyplomowa **inżynierska** [[1]](#footnote-1)\*) na temat: **Pasjans – implementacja popularnych łamigłówek karcianych z użyciem wtyczek** została wykonana przeze mnie samodzielnie.

2. Wymieniona wyżej praca:

- nie narusza praw autorskich w rozumieniu ustawy z dnia 4 lutego 1994 r. o prawie autorskim i prawach pokrewnych (t.j. Dz. U. z 2018 poz. 1191, z późn. zm.) oraz dóbr osobistych chronionych prawem cywilnym, a także nie zawiera danych i informacji, które uzyskałem/am w sposób niedozwolony,

- nie była wcześniej podstawą żadnej innej urzędowej procedury związanej z nadawaniem dyplomów wyższej uczelni lub tytułów zawodowych.

3. Jednocześnie wyrażam zgodę/nie wyrażam zgody\*\* na wykorzystanie fragmentów mojej pracy dyplomowej w publikacjach naukowych pracowników Politechniki Łódzkiej za zgodą prodziekana właściwego ds. studenckich, na zasadach wynikających z Ustawy z dnia 4 lutego 1994 r. o prawie autorskim i prawach pokrewnych (t.j. Dz. U. z 2018 poz. 1191, z późn. zm.)

.………………………….......

*(Podpis studenta)*

Załącznik nr 3

do Procedury antyplagiatowej prac dyplomowych

w Politechnice Łódzkiej

Łódź, dn. ......……………….

**OŚWIADCZENIE**

**o zgodności wersji elektronicznej pracy dyplomowej  
z przedstawionym wydrukiem komputerowym**

Paweł Wolski

………………………………………………………….

*(Imię i nazwisko studenta)*

Orla 3 m. 10, 90-317 Łódź

………………………………………………………….

*(Adres)*

209509

………………………………………………………….

*(Nr albumu)*

Wydział Elektrotechniki, Elektroniki,

Informatyki i Automatyki

………………………………………………………….

*(Jednostka organizacyjna prowadząca studia)*

Informatyka

………………………………………………………….

*(Kierunek studiów)*

Studia I stopnia inżynierskie stacjonarne

………………………………………………………….

*(Poziom kształcenia i forma studiów)*

Świadomy odpowiedzialności karnej za składanie fałszywych zeznań oświadczam, że przedkładana na nośniku elektronicznym praca dyplomowa **inżynierska** [[2]](#footnote-2)\*) na temat: **Pasjans – implementacja popularnych łamigłówek karcianych z użyciem wtyczek** zawiera te same treści, co oceniany przez promotora i recenzenta wydruk komputerowy.

Jednocześnie oświadczam, że jest mi znany przepis art. 233 § 1 Kodeksu karnego określający odpowiedzialność za składanie fałszywych zeznań.

.………………………….......

*(Podpis studenta)*

1. \*) Wpisać odpowiednio: licencjacka, inżynierska, magisterska.  
   \*\*) niepotrzebne skreślić [↑](#footnote-ref-1)
2. \*) Wpisać odpowiednio: licencjacka, inżynierska, magisterska. [↑](#footnote-ref-2)