# Raport

## z realizacji informatycznego projektu badawczo-rozwojowego

### Informacje ogólne

| | |
|---|---|
| Uczelnia | Uniwersytet im. Adama Mickiewicza |
| Kierunek studiów | Informatyka |
| Specjalność | Sztuczna Inteligencja |
| Tytuł projektu | TermHarbour |
| Imiona i nazwiska osób realizujących projekt | Paweł Lewicki<br>Maurycy Oprus<br>Sebastian Piotrowski |
| Opiekun projektu | dr. Rafał Jaworski |
| Termin rozpoczęcia realizacji projektu | 01.10.2024 |
| Termin zakończenia realizacji projektu | 10.06.2024 |
| Słowa kluczowe | Ekstrakcja, tłumaczenie automatyczne, glosariusz, kategorie, terminy specjalistyczne |

# I. Informacje dla interesariuszy projektu

## 1. Streszczenie projektu

W miarę rozwoju technologii, potrzeba precyzyjnego i efektywnego zarządzania specjalistyczną terminologią rośnie. Tłumacze mają do czynienia z szeroką gamą słownictwa z różnych kategorii, które często nie jest im dobrze znane. W trakcie pracy wspomagają się słownikami terminologii z danej branży, jednak mimo to nie ma gwarancji, że każdy termin zostanie przetłumaczony poprawnie. Brak jednolitego glosariusza prowadzi do nieporozumień, błędów w komunikacji i opóźnień w procesach biznesowych. Organizacje z różnych branż borykają się z trudnościami w zrozumieniu i tłumaczeniu specjalistycznych terminów w językach, które nie są dla nich ojczyste. W celu uproszczenia pracy tłumacza oraz zwiększenia jakości tłumaczenia konieczne jest zastosowanie systemowego rozwiązania, które umożliwi lepsze zarządzanie terminologią.
Proponowane rozwiązanie

Nasz projekt ma na celu stworzenie zaawansowanego systemu ekstrakcji i budowania glosariuszy terminologii specjalistycznej. System wykorzystuje techniki przetwarzania języka naturalnego (NLP) i uczenia maszynowego do automatycznego identyfikowania, tłumaczenia i kategoryzowania terminów z różnych dziedzin. Wykorzystujemy narzędzia takie jak spaCy oraz NLTK do przetwarzania tekstu oraz inne algorytmy do dopasowywania zdań i ekstrakcji terminów, m. in. simalign.

Cały projekt został przygotowany w formie aplikacji webowej zbudowanej przy użyciu Flask i SQLite, która pozwala na ekstrakcję i tłumaczenie terminów specjalistycznych. Aplikacja zawiera funkcjonalności takie jak przesyłanie plików, analiza tekstu oraz zarządzanie glosariuszem poprzez dodawanie, edycję i usuwanie rekordów.

Nasze rozwiązanie wyróżnia się na tle innych dzięki integracji wielu technik NLP oraz spełnienia specyficznych potrzeb użytkowników. System nie tylko tłumaczy terminy, ale także je kategoryzuje, co zwiększa jego użyteczność w praktycznych zastosowaniach.

## 2. Zakres projektu

**Przypadki użycia (use cases):**

1. **Ekstrakcja terminów specjalistycznych**:
    - **Opis:** Użytkownik może przesłać dokument tekstowy, z którego system automatycznie wyodrębni terminy specjalistyczne.
    - **Aktorzy:** Użytkownik, System
    - **Scenariusz:**
        1. Użytkownik przesyła dokument tekstowy.
        2. System analizuje dokument, identyfikuje terminy specjalistyczne i generuje listę tych terminów.
        3. Użytkownik otrzymuje listę terminów z ich tłumaczeniami i kategoriami.
2. **Sprawdzenie tłumaczenia terminów**:
    - **Opis:** System umożliwia weryfikację, czy tłumaczenie zawiera wszystkie niezbędne wyraz z terminologii specjalistycznej.

- **Aktorzy:** Użytkownik, System
- **Scenariusz:**
    1. Użytkownik wprowadza tekst do tłumaczenia.
    2. Użytkownik dostarcza tłumaczenie na wybrany język.
    3. System weryfikuje, czy tłumaczenie zawiera te same termy co tekst oryginalny
    4. Użytkownik otrzymuje informację zwrotną, czy wszystkie terminy zostały przetłumaczone poprawnie

3. **Kategoryzacja terminów**:
    - **Opis:** System kategoryzuje terminy według ich dziedziny.
    - **Aktorzy:** Użytkownik, System
    - **Scenariusz:**
        1. Użytkownik przesyła dokument do systemu oraz określa jego domenę.
        2. System automatycznie kategoryzuje wyekstrahowane terminy.
        3. Użytkownik może przeglądać terminy według kategorii w wybranej domenie.
4. **Zarządzanie glosariuszem**:
    - **Opis:** Użytkownik może dodawać, edytować i usuwać terminy w glosariuszu.
    - **Aktorzy:** Użytkownik, System
    - **Scenariusz:**
        1. Użytkownik dodaje nowy termin do glosariusza.
        2. System zapisuje termin w bazie danych.
        3. Użytkownik może edytować lub usuwać istniejące terminy.
5. **Przesłanie wyekstrahowanych terminów do słownika**
    - **Opis:** Użytkownik może przesłać do słownika terminy znalezione w przesłanych przez niego dokumentach
    - **Aktorzy:** Użytkownik, System
    - **Scenariusz:**
        1. Użytkownik przesyła dokument do systemu oraz określa jego domenę.
        2. System automatycznie kategoryzuje wyekstrahowane terminy.
        3. Użytkownik podejmuje decyzję o przesłaniu terminów do słownika
        4. System przesyła terminy i udostępnia je w słowniku

**3. Ocena jakości**

W celu zapewnienia wysokiej jakości opracowanego rozwiązania, zespół przeprowadził szereg testów funkcjonalnych, aby upewnić się, że wszystkie komponenty systemu działają zgodnie z założeniami. Testy te obejmowały:

1. **Testy jednostkowe (Unit Testing)**: Każda funkcjonalność, w tym ekstrakcja terminów, tłumaczenie oraz kategoryzacja, została przetestowana indywidualnie. Testy jednostkowe zostały wykonane przy użyciu narzędzi takich jak pytest, aby upewnić się, że poszczególne moduły działają poprawnie.
2. **Testy integracyjne (Integration Testing)**: Przeprowadzono testy, aby sprawdzić, czy różne moduły systemu współpracują ze sobą bez problemów. Szczególną uwagę zwrócono na integrację modułów NLP, bazy danych i interfejsu użytkownika.
3. **Testy systemowe (System Testing)**: Cały system został przetestowany w warunkach

zbliżonych do rzeczywistych, aby upewnić się, że spełnia wymagania funkcjonalne i niefunkcjonalne. Testy te obejmowały analizę wydajności, skalowalności i niezawodności systemu.

Dzięki przeprowadzeniu gruntownych testów zespół zapewnił wysoką jakość opracowanego rozwiązania. Raporty z testów dostępne w repozytorium github (pod adresem https://github.com/pllewy/TermHarbour/blob/master/documentation/Raport%20z%20test%C3%B3w%20funkcjonalnych.pdf) potwierdzają, że system działa zgodnie z założeniami.

# II. Informacje o przebiegu i rezultacie projektu

## 1. Realizacja wstępnej specyfikacji wymagań

### 1.1 Podsumowanie wstępnej specyfikacji wymagań

Celem projektu było stworzenie oprogramowania wspierającego tłumaczy poprzez generowanie słowników terminologicznych na podstawie specjalistycznych tekstów, z możliwością przeglądania i edycji przez tłumacza. W fazie planowania braliśmy pod uwagę cele biznesowe, potrzeby klientów, funkcje systemu oraz jego ograniczenia.

### 1.2 Realizacja planowanych Funkcjonalności

Projekt został zrealizowany zgodnie z założeniami przedstawionymi w specyfikacji. Wszystkie główne funkcjonalności (FE-1 do FE-6) zostały zaimplementowane:

- FE-1: System skutecznie tworzy słowniki terminologiczne z wprowadzonej bazy tekstów.
- FE-2: Słowniki są kategoryzowane zgodnie z tematyką.
- FE-3: System sprawdza odpowiedniki terminów w tłumaczeniach.
- FE-4: Tłumacze mogą przeglądać i ręcznie modyfikować słowniki.
- FE-5: Słowniki są podzielone na kategorie tematyczne.
- FE-6: System zwraca odpowiednie tagi dokumentu.

### 1.3 Zgodność z Celami Biznesowymi

Cele biznesowe zostały w pełni zrealizowane:

- BO-1: System przedstawia tłumaczowi słowniki terminologii specjalistycznej na podstawie dziedziny.
- BO-2: Weryfikacja liczby i poprawności tłumaczeń terminów została zaimplementowana.
- BO-3: Możliwość ręcznego dostosowania zawartości słownika została udostępniona.

### 1.4. Spełnienie Kryteriów Sukcesu

System spełnia wszystkie kryteria sukcesu:

- SC-1: Poprawnie utworzono słowniki terminów specjalistycznych.
- SC-2: Słowniki zostały poprawnie podzielone na kategorie tematyczne.
- SC-3: System wskazuje potencjalne błędy w tłumaczeniu.

- SC-4: Interfejs użytkownika jest intuicyjny i prosty w obsłudze.

### 1.5 Ograniczenia

W trakcie realizacji projektu pojawiły się pewne ograniczenia, które były opisane w początkowych założeniach:

- LI-1: Obsługa trzech języków (angielski, polski, hiszpański) została zapewniona, jednak tekst angielski jest zawsze wymagany - system nie obsługuje analizy tekstów w języku hiszpańskim i polskim bez podania dokumentu w języku angielskim.
- LI-2: Ograniczone zasoby sprzętowe wpływają na szybkość przetwarzania tekstów.

## 2. Realizacja harmonogramu prac

| Termin | Punkt kontrolny | Produkt | Priorytet |
|---|---|---|---|
| 31.10.2023 | Wstępna specyfikacja wymagań | Dokument wymagań projektowych | 1 |
| 12.11.2023 | Przygotowanie prezentacji koncepcji projektu | Prezentacja multimedialna i schemat wystąpienia | 2 |
| 14.11.2023 | Prezentacja koncepcji projektu | Wystąpienie | 1 |
| 31.11.2023 | Architektura i harmonogram prac | Rozdział dokumentu założeń projektowych | 1 |
| 30.12.2023 | KPI (kryteria sukcesu) i analiza ryzyka | Rozdział dokumentu założeń projektowych | 1 |
| 19.01.2024 | Przygotowanie posteru | Poster i schemat wystąpienia | 2 |
| 23.01.2024 | Sesja posterowa | Wystąpienie | 1 |
| 31.01.2024 | Badania oraz prace nad poszczególnymi modułami projektu | Koncepcja rozwiązań i algorytmów | 1 |
| 31.03.2024 | Lokalne repozytorium projektu zgodne z wytycznymi | Repozytorium lokalne | 1 |
| 20.04.2024 | Pierwszy commit na repozytorium produkcyjnym | Repozytorium GitHub | 1 |
| 17.05.2024 | Pierwszy kamień milowy - stworzenie kompletnych, osobnych modułów projektu | Repozytorium GitHub | |
| 27.05.2024 | Drugi kamień milowy - połączenie istniejących modułów projektów | Repozytorium GitHub | |

| | | | |
|---|---|---|---|
| 08.06.2024 | Testy integracyjne | Repozytorium GitHub | |
| 10.06.2024 | Ostatni commit na repozytorium produkcyjnym | Repozytorium GitHub | 1 |
| 10.06.2024 | Raport z wykonania projektu | Raport PDF | 1 |

**Podział prac:**

- Paweł Lewicki:
    - moduł porównywania zdań i analiz tłumaczeń
    - wystawienie aplikacji webowej
    - zarządzanie projektem
- Sebastian Piotrowski:
    - ekstrakcja terminów specjalistycznych
    - preprocessing i postprocessing tekstu
- Maurycy Oprus:
    - tworzenie i testowanie modeli do kategoryzacji
    - obsługa bazy danych do przechowywania słowników specjalistycznych

Praca nad dokumentacją wykonywana była na wspólnych spotkaniach zespołu, na których wyznaczane były dalsze zadania dla poszczególnych członków zespołu.

## 3. Spełnienie kryteriów sukcesu

### 3.1 Ocena kryteriów sukcesu:

- Efektywność ekstrakcji termów: wysoka skuteczność w identyfikacji oraz ekstrakcji terminów specjalistycznych z tekstu, minimalizacja liczby błędów w procesie ekstrakcji - kryterium spełnione
- Prawidłowość znajdowania synonimów oraz tłumaczeń terminów specjalistycznych - kryterium częściowo spełnione
- Poprawność tagowania terminów: przypisywanie kategorii tematycznych i tagów do wyekstrahowanych termów - kryterium spełnione
- Integracja z narzędziami wsparcia: biblioteka spacy (ekstrakcja), opus (korpus), iate (terminologia) - kryterium częściowo spełnione (brak integracji z IATE)
- Skuteczność w proponowaniu sugestii tłumaczeń oraz tagów pracy - kryterium spełnione
- Wizualizacja potencjalnych błędów w tłumaczeniu przez tłumacza - kryterium spełnione
- Zadowolenie użytkownika: przejrzysty interfejs - kryterium spełnione
- Odpowiednia złożoność obliczeniowa: szybki czas działania programu - kryterium częściowo spełnione (ze względu na ograniczone zasoby)

## III. Opis uzyskanego wyniku badawczego

W ramach projektu zespół podjął się rozwiązania problemu ekstrakcji i analizy terminologii specjalistycznej w tekstach. W ramach wcześniejszych prac przygotowane zostały narzędzia

umożliwiające analizę gramatyczną tekstów, przygotowanie pojedynczych słów do dalszej analizy, oraz modele języka mogące wesprzeć zespół przez wskazanie odległości między poszczególnymi wyrazami wskazując na ich podobieństwo. Przy wykorzystaniu między innymi tych narzędzi zespół przygotował szereg algorytmów ekstrakcji pierwotnych i złożonych termów, które następnie są identyfikowane w dwóch językach podanych przez użytkownika i łączone w celu podwyższenia skuteczności tworzonych słowników.

Algorytmy te pozwoliły zespołowi na przeprowadzenie następujących eksperymentów:

I. Analiza skuteczności metod biblioteki simalign porównywania tekstów wielojęzycznych w przypadku zastosowania jej na całym tekście, wyekstrahowanych termach i pojedynczych zdaniach/paragrafach tekstów z języka naturalnego
II. Porównanie modeli kategoryzacyjnych i dobór odpowiedniego do rozwiązywanego problemu
III. Analiza i porównanie skuteczności metod ekstrakcji i dobór odpowiednich do rozwiązywanego problemu. Rozwiązanie problemu pre i post processingu w celu zwiększenia jakości otrzymywanych rezultatów

Przy czym powyższe eksperymenty przebiegły w następujący sposób:

a) Analiza skuteczności metod biblioteki simalign porównywania tekstów wielojęzycznych w przypadku zastosowania jej na całym tekście, wyekstrahowanych termach i pojedynczych zdaniach/paragrafach tekstów z języka naturalnego

W ramach eksperymentu porównane zostały trzy podejścia do problemu powiązania terminologii specjalistycznej dwóch języków. Danymi wejściowymi do eksperymentu był tekst oryginalny, tekst przetłumaczony (sprawdzony systemem eksperckim), a także lista znalezionych terminów specjalistycznych z języka źródłowego i docelowego.

Pierwszym wykorzystanym podejściem do rozwiązania problemu była metoda analizy list terminologii specjalistycznych dwóch języków. Głównym założeniem metody była potencjalna niższa złożoność obliczeniowa oraz pamięciowa, a uzyskiwane wyniki miały być równie wysokie. Niestety jakość uzyskanych wyników była niska, dlatego konieczne było uzupełnienie problemu o większą liczbę danych.

Kolejnym podejściem było wykorzystanie całości tekstu źródłowego do stworzenia sieci połączeń między tekstami w różnych wersjach językowych, a następnie wyszukanie tych połączeń, których składowe znajdują się w dostępnych listach terminologii specjalistycznej. Zgodnie z oczekiwaniami wyniki wynikające z zastosowania tego podejścia były wyższe niż po zastosowaniu pierwszej metody, jednak liczba wykonywanych operacji przewyższyła dostępne zasoby sprzętowe dla dużych dokumentów.

Z tego powodu opracowane zostało trzecie podejście wykorzystujące naturalny podział tekstu na paragrafy i inne naturalnie występujące przerwy w tekście. Przy tekstach źródłowych pochodzących głównie z artykułów naukowych takie podejście daje możliwość ograniczenia zasobów pamięciowych przy jednoczesnym zachowaniu wysokiej jakości zawartości (treść tłumaczona rzadko trafia do różnych zdań i na pewno nie trafia do różnych paragrafów). Wykorzystując to podejście do analizy

trafił wyłączony fragment tekstu źródłowego, a następnie postępowano zgodnie z algorytmem wykorzystanym w poprzednim paragrafie. To podejście pozwoliło na ograniczenie złożoności obliczeniowej algorytmu simalign działającego ze złożonością O(n^2), natomiast spowodowało, że model wykorzystywany do ekstrakcji definiowany był wielokrotnie, co ostatecznie doprowadziło do spowolnienia programu.

Dlatego ostateczną formą eksperymentu było zastosowanie algorytmu simalign na małej strukturze tekstu, a następnie porównania uzyskanych wyników z listą otrzymaną przez analizę całego dokumentu. To podejście pozwoliło na optymalizację złożoności obliczeniowej obydwu wykorzystanych algorytmów, przy jednoczesnej poprawie uzyskiwanych rezultatów.

b) Porównanie modeli kategoryzacyjnych i dobór odpowiedniego do rozwiązywanego problemu

Początkowa faza testów skupiała się na wykorzystania modelu LDA (Latent Dirichlet Allocation) z biblioteki gensim. Takie rozwiązanie wraz z fine-tuningiem parametrów generowało rozwiązanie, w którym wartości coherence były wysokie (grupa słów pozornie zwarcie należała do wspólnego tematu). Jednak bardziej dogłębna analiza generowanego rozwiązania pozwoliła stwierdzić, że tworzone kategorie były trudne do określenia - pogrupowane terminy specjalistyczne nie tworzyły oczywistych kategorii dla użytkownika, pozwalających skutecznie określić jakie tematy zostały poruszone w dokumencie. Zważając na użyteczność takiego rozwiązania testowano również zbliżone rozwiązania z biblioteki tomotopy (LDA, HDP - Hierarchical Dirichlet Process), z podobnym rezultatem.

Ostatecznie w projekcie wykorzystano model Lbl2Vec, będący rozwinięciem modelu Doc2Vec z biblioteki gensim. Takie podejście pozwoliło predefiniować podkategorie w danej domenie (bardziej ogólnej kategorii), na podstawie słów kluczowych często w niej wykorzystywanych. Do określenia skuteczności ostatecznego modelu klasyfikacyjnego wykorzystano zbiory tekstów z określonymi wcześniej kategoriami. Model poprawnie kategoryzował teksty w granicach 80-90% w zależności od zbioru tekstów, dobranych słów kluczowych oraz liczby kategorii w zbiorze tekstów. Testowano również wykorzystanie modelu Lbl2TransformerVec, który pozwalał osiągać lepsze wyniki dzięki możliwości tworzenia wektorów z dowolnych słów. Eliminowało to problem ograniczonego słownictwa przy doborze słów kluczowych, jednak napotkano problem z integracją tego rozwiązania z projektem końcowym.

Największymi ograniczeniami w generowaniu i testowaniu kolejnych modeli klasyfikacyjnych była dostępność tekstów w danej kategorii oraz konieczność stworzenia klasyfikacji nienadzorowanej. Ze względu na dostępność tekstów klasyfikacja opiera się wyłącznie na tekstach angielskojęzycznych. Jednak opracowane rozwiązania zakładają i umożliwiają dalszy rozwój oraz rozbudowę proponowanego rozwiązania.

c) Analiza i porównanie skuteczności metod ekstrakcji i dobór odpowiednich do rozwiązywanego problemu. Rozwiązanie problemu pre i post processingu w celu zwiększenia jakości otrzymywanych rezultatów

Celem eksperymentu było wybranie najbardziej efektywnej metody ekstrakcji terminologii z tekstów.

Skupiono się na trzech podejściach: Ekstrakcja za pomocą wzorców (Pattern Matching), Named Entity Recognition (NER) oraz Term Frequency-Inverse Document Frequency (TF-IDF).
Pierwsza metoda skupiała się na wykorzystaniu reguł gramatycznych do identyfikacji terminów. Wykorzystano narzędzie spacy Matcher do wykrywania fraz składających się z wcześniej zdefiniowanych wzorców części mowy.

Kolejna metoda - NER została użyta do identyfikacji nazwanych jednostek w tekście takich jak imiona, miejsca, organizacje, wydarzenia itp. W eksperymentach wykorzystano wbudowane w Spacy modele NER, wytrenowane na dużych zbiorach danych.

Ostatnią metodą wykorzystaną w eksperymencie jest metoda TF-IDF. Jest to technika statystyczna używana do oceny znaczenia słów w dokumencie. Wysoka wartość TF-IDF oznacza, że słowo jest ważne w danym dokumencie ale rzadkie w korpusie dokumentów.

Po przetestowaniu powyższych metod, dodany został również pre oraz post-processing tekstu w celu usunięcia niepotrzebnych znaków, czy też słów w tekście. Na końcu nastąpiły modyfikacje wzorców, parametrów powyższych metod w celu podwyższenia jakości ekstrahowanych terminów.

W wyniku eksperymentu metoda Pattern Matching okazała się skuteczna w identyfikacji terminów, zwłaszcza w kontekstach, gdzie terminologia jest dobrze zdefiniowana przez wzorce gramatyczne. Jednakże jej skuteczność zależy od precyzji zdefiniowanych wzorców i może wymagać manualnej regulacji w zależności od specyfiki tekstu. Najlepsze wyniki uzyskała dla języka angielskiego. Metoda NER efektywnie poradziła sobie z identyfikacją specyficznych jednostek, takich jak imiona, miejsca, organizacje czy wydarzenia, jednocześnie osiągając bardzo wysoką skuteczność.

Niestety, metoda TF-IDF nie przyniosła pozytywnych wyników ze względu na zbyt małą liczbę tekstów w korpusie trójjęzycznym. TF-IDF wymaga dużej ilości danych, aby efektywnie ocenić znaczenie słów, a w tym przypadku liczba dokumentów w wersji angielskiej, hiszpańskiej oraz polskiej była niewystarczająca.

Do ostatecznej ekstrakcji terminologii zostały wykorzystane połączone ze sobą metody Pattern Matching oraz Named Entity Recognition.

Powyższe eksperymenty pokazały, że w ramach projektu udało się skutecznie zdefiniować istotny problem badawczy, a dzięki wykorzystaniu zaawansowanych algorytmów sztucznej inteligencji uzyskano produkt, którego zastosowanie może mieć pozytywny wpływ na pracę tłumaczy i który może stać się projektem komercyjnym. Jednocześnie złożoność rozwiązywanego problemu sprawia, że projekt ten może być z powodzeniem rozwijany i uzupełniany o kolejne metody i algorytmy dotyczące analizy tekstu, tworzenia i wykorzystania korpusów języka, czy dużych modeli językowych. Każde z powyższych zostawia duży potencjał do prowadzenia dalszych badań naukowo-rozwojowych, a których wyniki pozwolą na udoskonalenie otrzymanego produktu.