

Instrukcja systemu CLARIN-PL

Wprowadzenie	2
Logowanie i rejestracja	2
Okno projektów	2
Projekt	3
Repozytorium	3
Sesje	4
Wgrywanie plików	4
Przenoszenie kontenerów między sesjami	5
Menu kontekstowe kontenera	6
Kolejkowanie przetwarzania	8
Uruchamianie narzędzia automatycznego	9
Edytor Audio	9
Optymalny przepływ pracy z edytorem transkrypcji	11
Generowanie finalnego korpusu	11
Szczegółowy opis narzędzi automatycznych	12
Detakcja mowy	12
Diaryzacja	13
Transkrypcje	14
Segmentacja	15
Konwersja zapisu ortograficznego na fonetyczny	16
Rozpoznawanie słów kluczowych	17
Analizy korpusowe	18
Infrastruktura EMU-SDMS	19
Etapy pracy badawczej	19
Załadowanie bazy danych do sesji R	20
Wizualna inspekcja danych	21
Zapytania do bazy danych	21
Ekstrakcja cech audio	24
Wizualna inspekcja danych	24
Dalsze analizy i wnioskowanie statystyczne	28

Wprowadzenie

Osoby zajmujące się naukami humanistycznymi i społecznymi często dysponują dużą ilością nagrań audio zawierających mowę. Przykładami mogą być nagrania radiowe bądź telewizyjne, wywiady, przemowy (parlamentu, publiczne wystąpienia itp.), wykłady, filmy, literatura czytana i inne. Jednak informacje zawarte w plikach dźwiękowych nie są optymalnie przetwarzane oraz w obliczu ich ilości oraz długości, nie wszystkie informacje w nich zawarte mają szansę na wydobywanie. Głównym problemem w przetwarzaniu danych audio jest to iż wymagają znacznie więcej czasu niż tradycyjne dane tekstowe. Dodatkową przeszkodą może być brak wiedzy technicznej autorów zajmujących się badaniami humanistycznymi lub społecznymi na temat tego w jaki sposób przetwarzać większe zbiory nagrań audio. Z tego powodu, wydobywanie informacji z nagrań dźwiękowych bywa często pomijane. Naszym głównym celem było stworzenie darmowego oraz łatwo dostępnego narzędzia dla osób zajmujących się nauką z dziedzin humanistycznych oraz społecznych.

Logowanie i rejestracja

Serwis jest darmowy, lecz z uwagi na zapewnienie prywatności i bezpieczeństwa przechowywanych danych, postanowiliśmy zabezpieczyć dane (oraz wszystkie operacje na nich wykonywane) za pomocą loginu i hasła. Twoje hasło jest szyfrowane i nie używamy ani nie wykorzystujemy Twojego maila w żadnych innych celach aniżeli logowanie do serwisu.

Aby skorzystać z serwisu należy być zalogowanym. Załóż konto poniżej

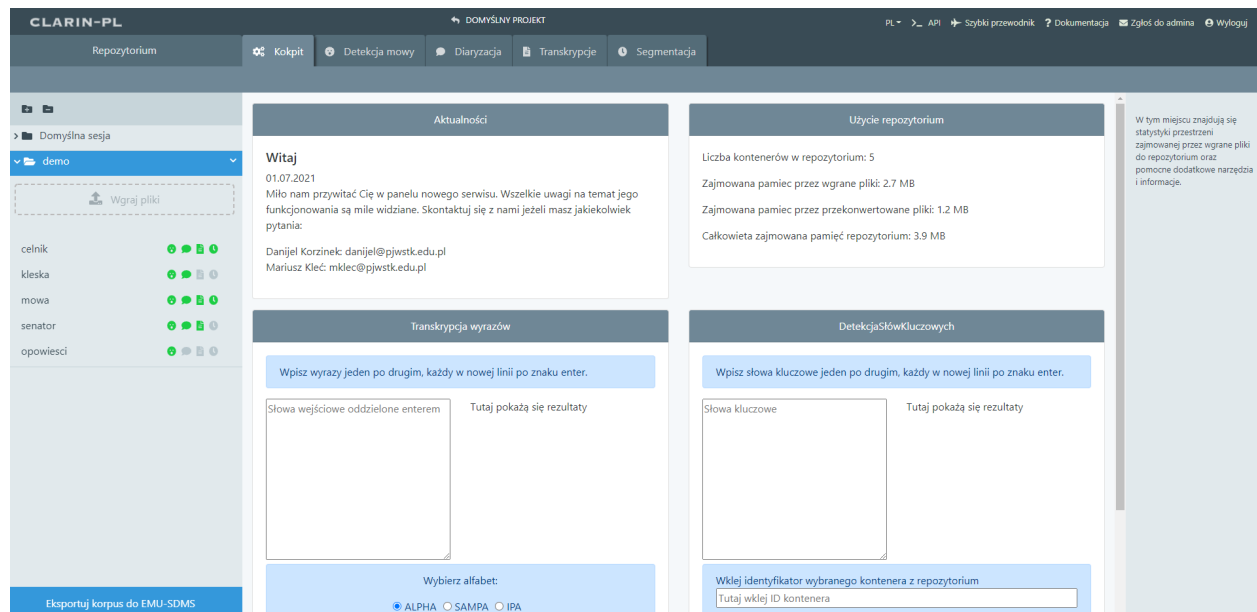
Zarejestruj się	Zaloguj się
Imię	Adres Email
<input type="text" value="Mariusz"/>	<input type="text" value="mklec@pjwtk.edu.pl"/>
Adres Email	Hasło
<input type="text" value="mklec@pjwtk.edu.pl"/>	<input type="password" value="....."/>
Hasło	<input type="password" value="....."/>
<input type="button" value="Zarejestruj się"/>	<input type="button" value="Zaloguj się"/>
	Zapomniałem hasła

Okno projektów

nowy projekt kliknij w przycisk

CLARIN-PL		PL > API ? Dokumentacja ✉ Zgłoś do admina 🌐 Wyloguj		
Nazwa	Identyfikator	Data utworzenia	Akcje	Stwórz projekt
inny	60cb95b79fe12f709830a4b6	June 17th 2021, 8:34:31 pm		W tym miejscu można tworzyć własne projekty które stanowią przestrzeń roboczą dla wgranych plików. UWAGA! w przypadku usunięcia projektu, zostaną usunięte wszystkie wgrane do niego pliki!
DOMYŚLNY PROJEKT	60cb95199fe12f709830a4a5	June 17th 2021, 8:31:53 pm		

Po wyborze projektu ukaze się jego widok podobny do poniższego.



2021

reprezentowanego w postaci nazwy oraz 4 ikonek symbolizujących stan wykonania na danym kontenerze narzędzi automatycznych:



Rozpoznawanie mowy (VAD)



Diaryzacja (DIA)



Rozpoznawanie mowy (REC)



Segmentacja (SEG)


Dzięki temu zawsze wiadomo na których kontenerach zostały uruchomione określone narzędzia, na których ich brakuje a na których wystąpił błąd i należy powtórzyć operację. Po kliknięciu w określoną ikonkę, kontener zostanie dodany do kolejki przetwarzania określonego narzędzia. Kliknięcie w nazwę kontenera umożliwia jej zmianę.

Sesje

Kontenery są zorganizowane ze względu na sesje. Sesje to foldery które można używać do zorganizowania danych.

Nazwę sesji można zmienić klikając na jej istniejącą nazwę.

Zaznaczanie kilku sesji jednocześnie odbywa się poprzez przytrzymanie klawisza Ctrl.


Kliknięcie w ikonę folderu  powoduje rozwinięcie sesji i ukazanie się wszystkich wgranych do niej kontenerów

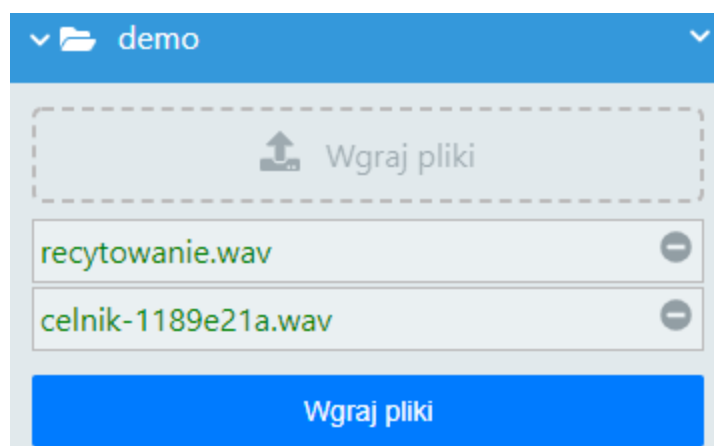
Dodawanie nowej sesji jest możliwe po kliknięciu ikonki 


Sesje możemy usuwać za pomocą ikonki  po uprzednim zaznaczeniu wybranych sesji

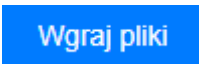
Wgrywanie plików

Pliki wgrywamy porcjami do wybranej sesji. W tym celu należy rozwinąć daną sesję a następnie


kliknąć w ikonkę  **Wgraj pliki**. Po wyborze dowolnych plików audio z dysku twardego, pojawią się one na liście jak na rysunku poniżej

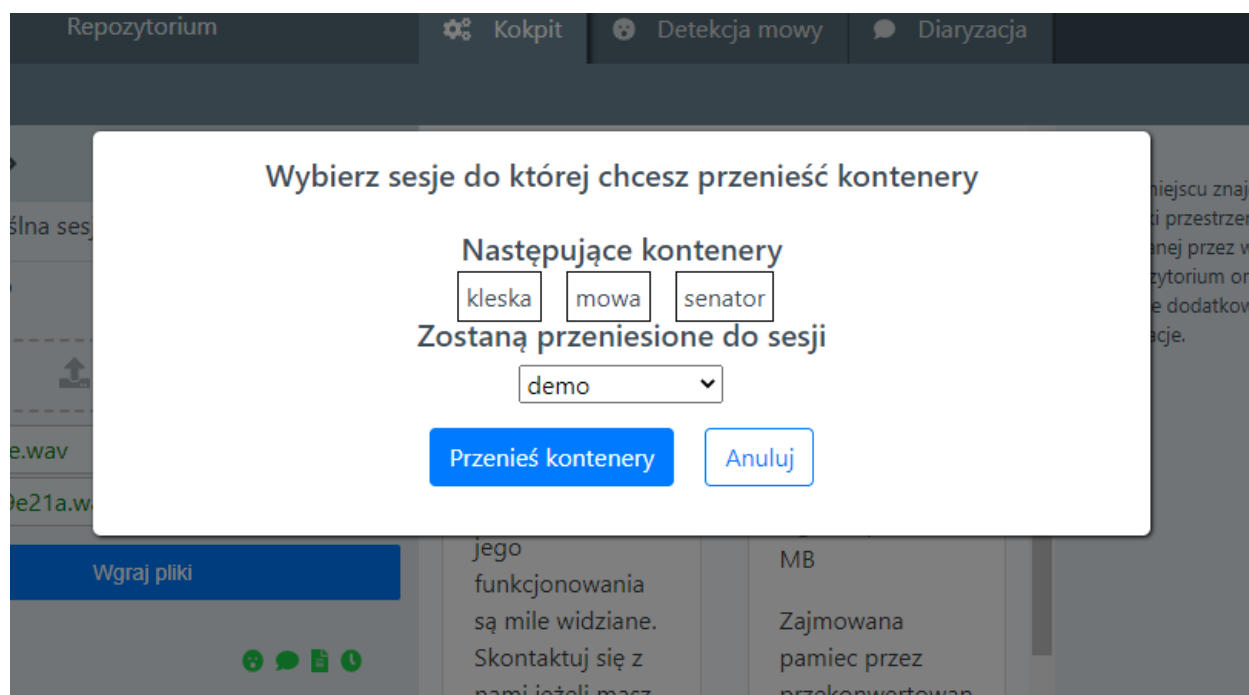


System akceptuje pliki audio w dowolnym formacie. Przy próbie wgrania pliku który nie jest audio, zostanie on oznaczony na wskazanej liście. Użytkownik może zdecydować które z wybranych plików ostatecznie chce wgrać na serwer. Jeżeli zdecyduje się usunąć wybraną pozycję, może to zrobić ikoną .

Po kliknięciu w przycisk  wszystkie zaakceptowane pliki z listy zostaną dodane do wybranej sesji

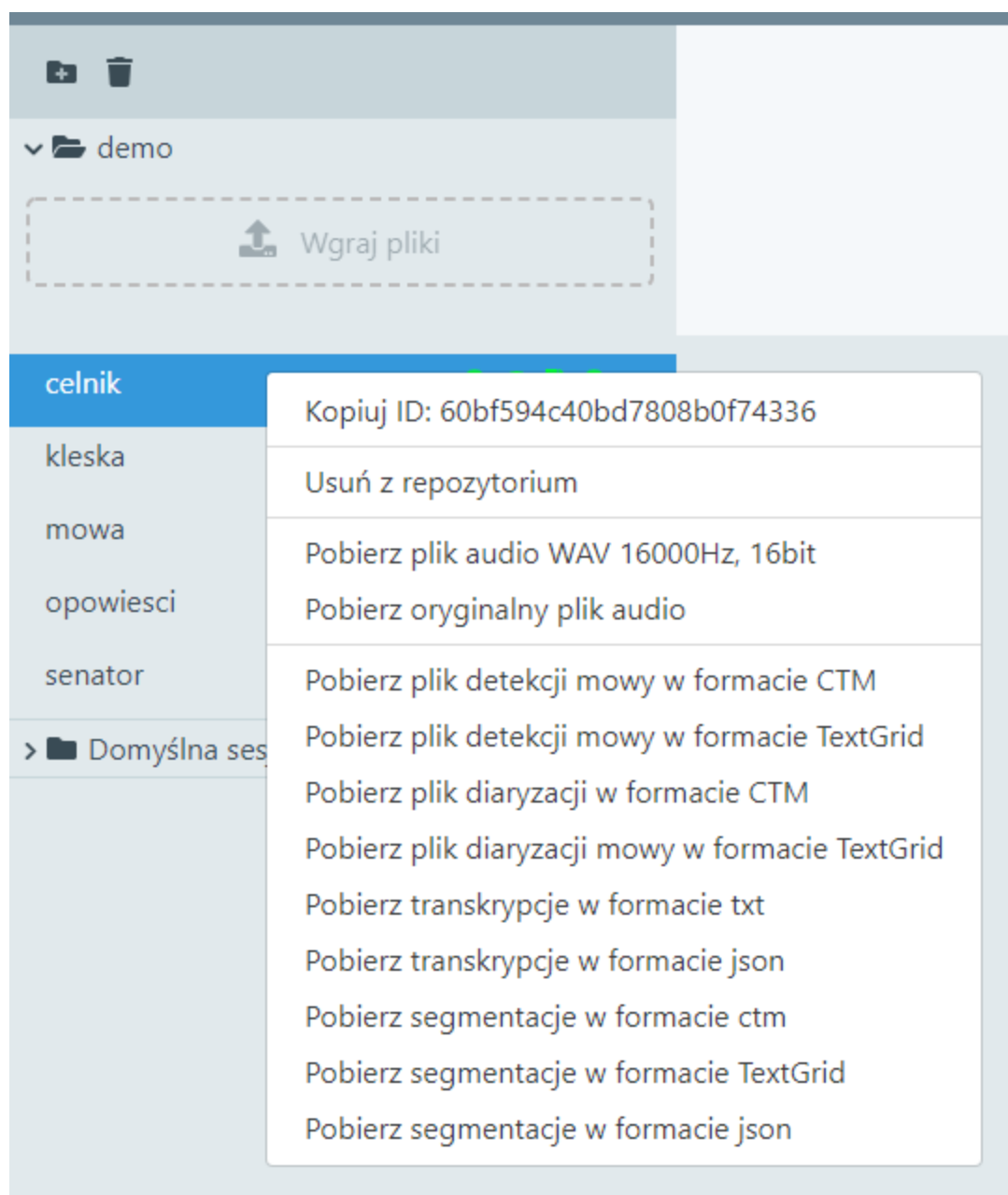
Przenoszenie kontenerów między sesjami

Przytrzymując klawisz Ctrl istnieje możliwość zaznaczenia wielu kontenerów jednocześnie. Po zaznaczeniu przynajmniej jednego, ukaże się ikona  służąca do przenoszenia zaznaczonych kontenerów do wybranej sesji. Po jej kliknięciu ukaże się okno dialogowe z możliwością wskazania do której sesji mają zostać przeniesione wybrane kontenery:



Menu kontekstowe kontenera

Kliknięcie prawym przyciskiem myszy na kontenerze powoduje otwarciem menu kontekstowego w wieloma opcjami, przede wszystkim służącymi do pobrania określonych plików skojarzonych z określonym kontenerem. Mogą to być pliki audio (oryginalny oraz WAV przekonwertowany do 16000 Hz, 16 bit) oraz pliki tekstowe w różnych formatach zawierające wyniki działania narzędzi automatycznych.




Menu kontekstowe zawiera następujące opcje:

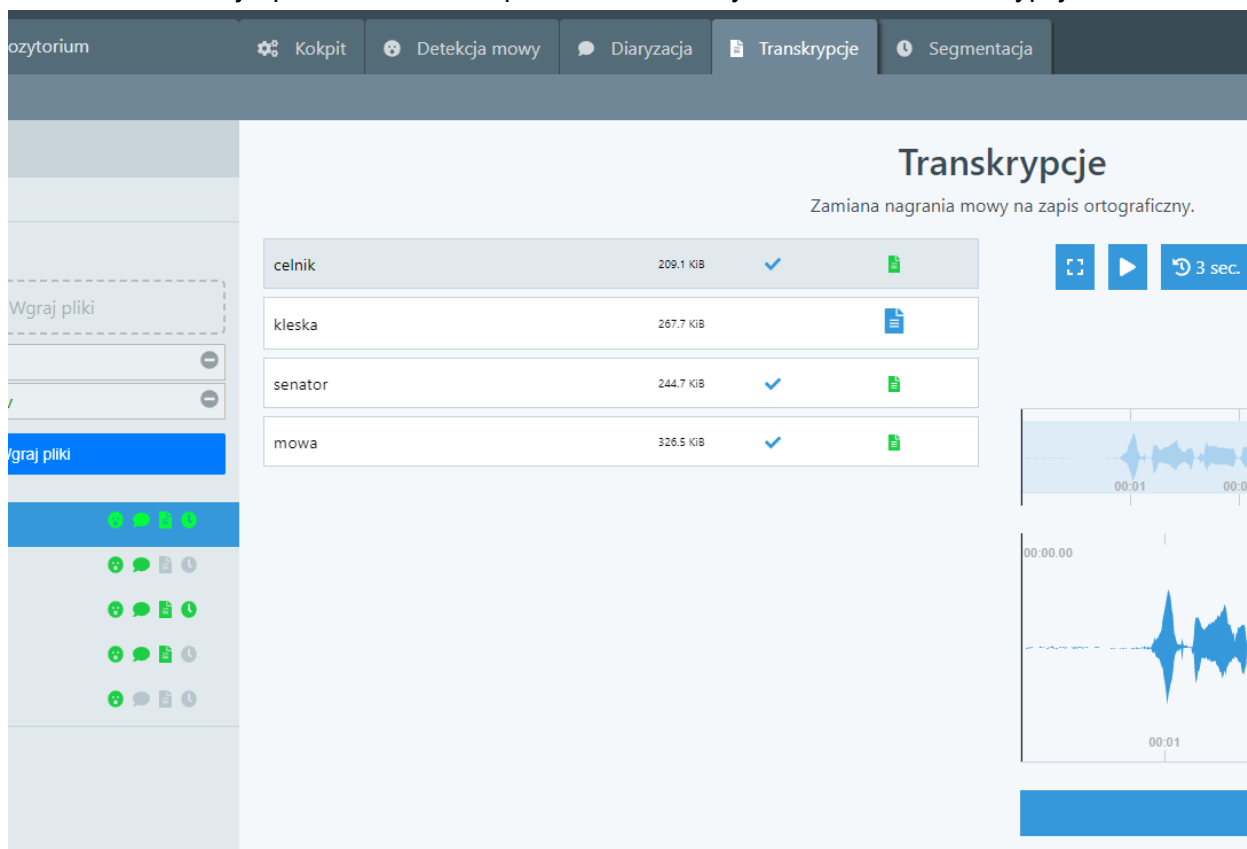
- **Kopiu ID: XXXXXX:** powoduje skopiowanie identyfikatora wybranego kontenera do schowka (np. celem jego użycia w programistycznym wywołaniu API lub w narzędziu do rozpoznawania słów kluczowych).
- **Usuń z repozytorium:** usunięcie danego kontenera z serwera wraz z wszystkimi skojarzonymi z nim plikami.
- **Pobierz plik audio WAV 16000 Hz, 16 bit:** Każdy wgrany plik do repozytorium jest wewnętrznie kopiowany i przechowywany w formacie PCM WAV 16000 Hz, 16 bit. Format ten służy jako wejście do narzędzi automatycznych. Możesz go pobrać wybierając tą opcję.

- **Pobierz oryginalny plik audio:** umożliwia ściągnięcie pliku który został wgrany do repozytorium z komputera użytkownika
- **Pobierz plik detekcji mowy w formacie [CTM || TextGrid]:** Opcje te umożliwiają pobranie wyniku detekcji mowy w formacie CTM lub TextGrid.
- **Pobierz plik diaryzacji w formacie [CTM || TextGrid]:** Opcje te umożliwiają pobranie wyniku diaryzacji w formacie CTM lub TextGrid.
- **Pobierz transkrypcję w formacie [txt || JSON] :** Opcje te umożliwiają pobranie transkrypcji w formacie TXT lub JSON
- **Pobierz segmentacje w formacie [TextGrid || CTM || JSON] :** Opcje te umożliwiają pobranie wyniku segmentacji w formatach CTM, TextGrid lub JSON.

Kolejkowanie przetwarzania

Klikając jedną z ikonek przy kontenerze w repozytorium dodajemy określony plik do kolejki

przetwarzania określonego narzędzia. Dla przykładu, kliknięcie w ikonkę  skutkuje dodaniem kontenera do kolejki przetwarzania rozpoznawania mowy w zakładce “Transkrypcje”:









Podejście takie umożliwia zaplanowanie pracy etapami w taki sposób że w danym czasie użytkownik skupia się jedynie nad wybraną porcją danych. Ponadto ma możliwość uruchomienia danego narzędzia automatycznego do wszystkich plików dodanych do kolejki jednocześnie.





Kliknięcie we wpis w kolejce powoduje załadowanie kontenera do edytora znajdującego się po prawej stronie. Pomocny okaże się skrót klawiaturowy [Alt + n] który powoduje załadowanie do edytora kolejnej pozycji na liście. Dzięki temu istnieje możliwość przetwarzania oraz opisywania poszczególnych kontenerów “od góry do dołu” po kolei.

Na pozycjach w kolejce przetwarzania funkcjonuje również menu kontekstowe. Posiada ono takie same opcje jak w przypadku kontenerów w repozytorium z tą różnicą iż zamiast opcji “usuń z repozytorium” jest opcja “usuń element z listy”

Uruchamianie narzędzia automatycznego

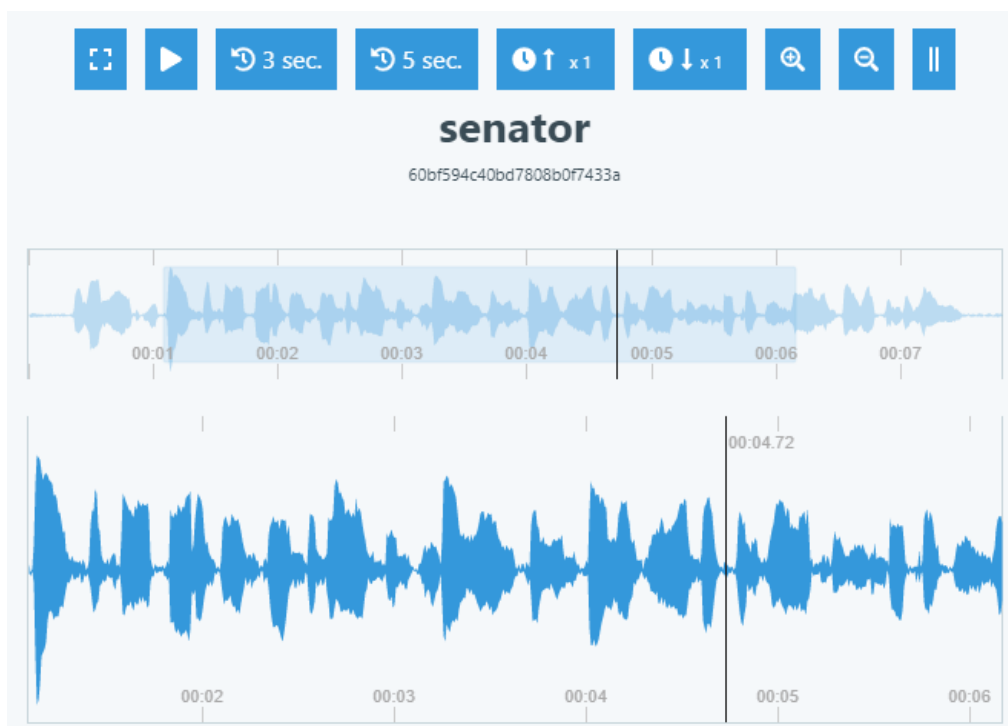
Każdy element na liście przetwarzania posiada ikonę oznaczającą status wykonania danego narzędzia oraz ikonkę służącą do jego uruchamiania:

senator	244.7 KiB	✓	
mowa	326.5 KiB		
kleska	267.7 KiB		
celnik	209.1 KiB		










- ikonka  wskazuje na ukończenie wykonywania narzędzia automatycznego. W tym samym miejscu, może pojawić się również ikona wskazująca iż dane narzędzie jest w trakcie działania  lub wystąpienie ewentualnego błędu .
- Kliknięcie w ikonkę  powoduje uruchomienie danego narzędzia na określonym kontenerze. Jeżeli ikona jest koloru zielonego, wszystkie wcześniejsze rezultaty zostaną nadpisane obecnie wykonującym się procesem (zostanie uruchomiony ponownie).

Edytor Audio

Po kliknięciu w wybraną pozycję w liście przetwarzania, dany kontener zostanie otwarty w wybranej zakładce z możliwością odsłuchiwania oraz korekty wyników narzędzia automatycznego.



Ikony edytora oznaczają odpowiednio:

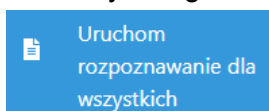
-  Kliknięcie spowoduje rozszerzenie edytora do pełnej szerokości ekranu. Lista przetwarzania znajdzie się pod spodem.
-  Odtwarzanie bądź wstrzymanie odtwarzania. Przydatny skrót klawiaturowy [Alt+I]
-   Podczas odsłuchu pliku audio często istnieje potrzeba powtórzenia odsłuchiwania kilku ostatnich sekund. Przyciski te są dedykowane do odtworzenia odpowiednio 3 lub 5 ostatnich sekund nagrania (względem położenia głowicy odtwarzającej). Przydatne będą skróty [Alt+k] dla ostatnich 3 sekund oraz [Alt+j] dla ostatnich 5 sekund.
-   Za pomocą tych przycisków można przyspieszyć bądź spowolnić prędkość odtwarzania, bez wpływu na wysokość dźwięku. Skrót [Alt+i] przyspiesza nagranie o 50% a skrót [Alt+o] spowalnia o 50%.
-   Powiększanie i oddalanie widoku oscylogramu.
-  wstawia 3 sekundowy region audio z możliwością dostosowania jego granic. Jeżeli regiony nakładają się na siebie, zostaną połączone.

Optimalny przepływ pracy z edytorem transkrypcji

Aby maksymalnie zoptymalizować czas spędzony na transkrybowaniu nagrań, rekomenduje się stosowanie następującego przepływu pracy:

1. Dodanie kilku lub kilkunastu kontenerów do kolejki przetwarzania transkrypcji
2. Kliknięcie w pierwszą pozycję w kolejce celem załadowania kontenera do edytora
3. Kliknięcie w pole tekstowe celem ustawienia kursora do wpisywania transkrypcji.
4. Możemy jednocześnie pisać na klawiaturze, jednocześnie odsłuchiwać fragmenty nagrania używając skrótu Alt+I (play/pause). W przypadku gdy nie zrozumieliśmy danego fragmentu nagrania, można kliknąć Alt+k aby odsłuchać ostatnie 3 sekundy lub Alt+j aby odsłuchać ostatnie 5 sekund nagrania. W tym samym czasie kursor wciąż znajduje się na polu tekstowym i umożliwia nam pisanie transkrypcji jednocześnie odsłuchując plik audio.
5. Po zakończeniu transkrypcji klikamy Alt+m celem zapisania jej na serwerze
6. Klikamy Alt+n aby załadować kolejny plik w kolejce przetwarzania i cały proces rozpoczyna się od początku na kolejnym pliku. Wszystko może odbywać się bez odrywania rąk od klawiatury.

Warto zauważyć iż na początku warto rozważyć uruchomienie narzędzia automatycznego dla wszystkich pozycji w kolejce. W tym celu należy kliknąć przycisk



znajdujący się po prawej stronie ekranu. Następnie należy odczekać aż wszystkie procesy zakończą działania. Może to potrwać nawet kilkanaście minut lub nawet kilka godzin w zależności od liczby plików i ich długości. Zaleca się pozostawienie systemu do czasu zakończenia działania. Po ich zakończeniu można przystąpić do procesu korekty wyników rozpoznawania automatycznego w sposób opisany powyżej.

Generowanie finalnego korpusu

Istnieje możliwość wyeksportowania korpusu w formacie EMU-SDMS. Format ten umożliwia przeprowadzanie dalszych analiz w języku R oraz przeglądanie korpusu w środowisku EMU. Aby kontenery znalazły się w korpusie, muszą być uzupełnione wszystkie poziomy opisy kontenera. Innymi słowy, do korpusu zostaną dodane tylko te kontenery dla których wszystkie

ikonki przy danym kontenerze będą zielone np:

mowa



. Jeżeli zabraknie choć jednego poziomu, kontener nie zostanie włączony do finalnego korpusu.

Aby stworzyć korpus należy kliknąć ikonkę

Eksportuj korpus do EMU-SDMS

oraz poczekać na wygenerowanie pliku. Po wygenerowaniu będzie można zapisać wygenerowany korpus na dysku twardym w postaci pliku ZIP.

Szczegółowy opis narzędzi automatycznych

Niniejszy rozdział zawiera szczegółowy opis działania poszczególnych narzędzi automatycznych zaimplementowanych w serwisie.

Detakcja mowy

Detekcja mowy (Voice Activity Detection: VAD) jest często używane na etapie pre-processingu do wielu narzędzi przetwarzania mowy. Ponieważ dane audio są zwykle nie monogeniczne oraz zawierają zmiksowane fragmenty mowy, muzyki, tła oraz ciszę. Rozróżnienie pomiędzy tymi różnymi typami audio jest niezwykle istotne w skuteczności systemu do transkrypcji. Jego celem jest odizolowanie części zawierających mowę od części zawierających inny typ zdarzeń (cisza, szum, muzyka itp.). Narzędzie to jest kompletnie niezależne od języka oraz domeny wypowiedzi. Niemniej jednak może generować błędy przy bardzo zaszumionych danych. Należy zwrócić uwagę iż narzędzie automatycznego rozpoznawania mowy, wykorzystuje detekcję mowy aby lepiej poradzić sobie z rozpoznawaniem.

Niewielki eksperyment potwierdził wysoki poziom czułości (Recall ~ 99%) oraz średnią precyzję (Precision ~ 58%). Było to jednak zamierzonym celem aby nie utracić żadnych części zawierających mowę, akceptując czasami fragmenty które jej nie zawierają. Jest to spowodowane tym iż pozostałe narzędzia akceptują niewielką ilość zaszumionych danych, ale działają błędnie gdy jakakolwiek część mowy jest pominięta.

Edytor detekcji mowy wizualizuje segmenty w których rzeczywiście występuje mowa. Zaznacza w ten sposób jedynie te części nagrania z mową które okażą się przydatne podczas dalszego przetwarzania, filtrując wszelkiego rodzaju dłuższe fragmenty ciszy, muzyki bądź innych zdarzeń które nie są mową.



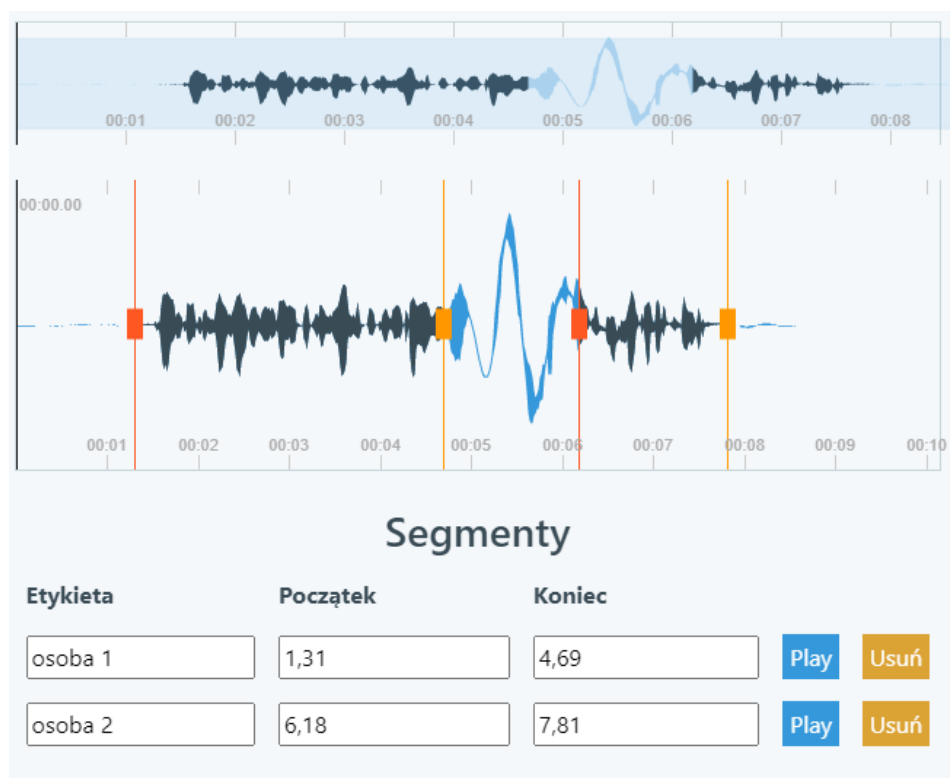
Widok edytora detekcji mowy

Każdy segment posiada swoją graficzną reprezentację na oscylogramie a granice danego segmentu można dowolnie przesuwając. Lista segmentów pojawia się pod oscylogramem wraz z informacją o początku i końcu danego segmentu w sekundach. Pola te można dowolnie edytować. Kliknięcie przycisku **Play** powoduje odtworzenie segmentu. Kliknięcie przycisku **Usuń** spowoduje usunięcie segmentu.

Diaryzacja

Narzędzie to jest używane do segmentacji dużych plików audio na części wypowiedziane przez poszczególne osoby. Istnieje kilka typów strategii segmentacji mówców. Pierwsza to rozpoznawanie momentu zmiany mówcy na innego, druga to dodanie informacji który fragment należy do tego samego mówcy oraz trzecia strategia polega na identyfikacji rozpoznanych fragmentów tak aby wiedzieć kto dokładnie mówi w rozpoznanym segmencie. Nasze narzędzie wspiera jednak drugą strategię w której rozpoznajemy zmiany mówców, wiemy ilu ich jest oraz w jakich momentach nagrania występują. Narzędzie jednak traktuje mówców w sposób anonimowy. Narzędzie to jest użyteczne do adaptacji różnych narzędzi oraz modeli do indywidualnych mówców ale również do innych typów analiz które wymagają segmentacji mówców.

Zasadniczo edytor wygląda identycznie jak w przypadku detekcji mowy, z tą różnicą iż etykiety przybierają unikatowe wartości dla poszczególnych mówców występujących w nagraniu. Jechanie myszką na segment pokazuje jego etykietę w lewym górnym rogu oscylogramu.



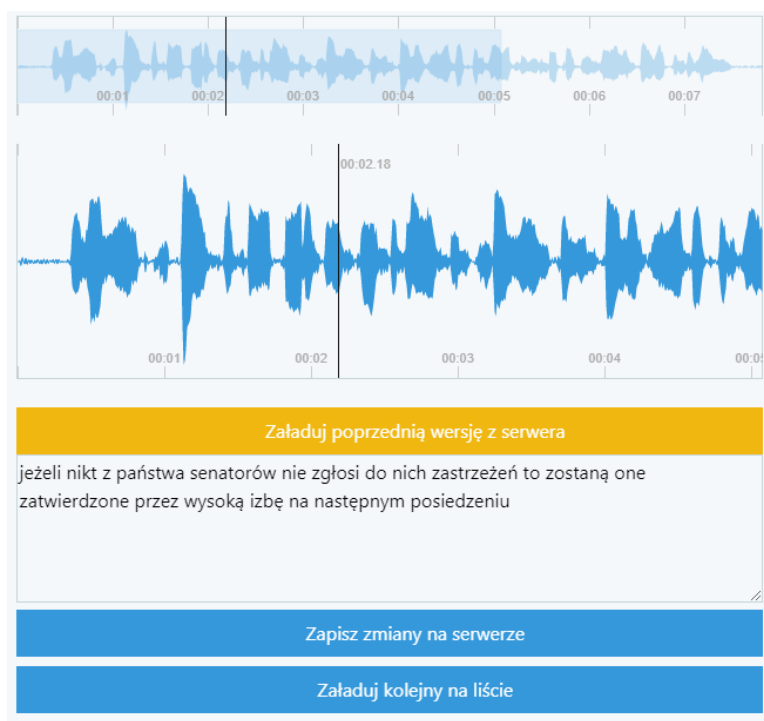
Widok edytora diaryzacji

Transkrypcje

Narzędzie to używa systemu rozpoznawania mowy do wygenerowania najbardziej prawdopodobnej ortograficznej transliteracji nagrań dźwiękowych mowy Polskiej. Na początku sygnał audio jest poddawany ekstrakcji cech w postaci ramek czasowych. Następnie ramki są filtrowane używając modułu VAD. Ramki zawierające tylko mowę są następnie poddawane rozpoznawaniu mówców celem adaptacji modelu akustycznego. Model akustyczny modeluje prawdopodobieństwo występowania słów na podstawie obserwowanych cech akustycznych. Wyjściem z modelu akustycznego są fonemy które muszą zostać zamienione na wyrazy. Robi to moduł konwersji fonemów na grafemy. Ciągi wyrazów wymagają jednak ułożenia w sekwencję odpowiednią dla gramatyki danego języka. Robi to model językowy który oblicza prawdopodobieństwo sekwencji wyrazów. Dekoder wybiera najwyższe prawdopodobieństwo określonej sekwencji i zwraca jako ciąg słów.

Edytor transkrypcji zawiera pole tekstowe pod oscylogramem. Jest ono wypełnione wynikiem automatycznego rozpoznawania mowy. Po dokonaniu ręcznych korekty tekstu, zmiany należy zapisać klikając w przycisk "Zapisz zmiany na serwerze". Istnieje również możliwość załadowania poprzednio zapisanej wersji klikając w przycisk "Załaduj poprzednią wersję z

serwera”. Przycisk “Załaduj kolejny na liście” powoduje załadowanie kolejnego pliku z kolejki do edytora.

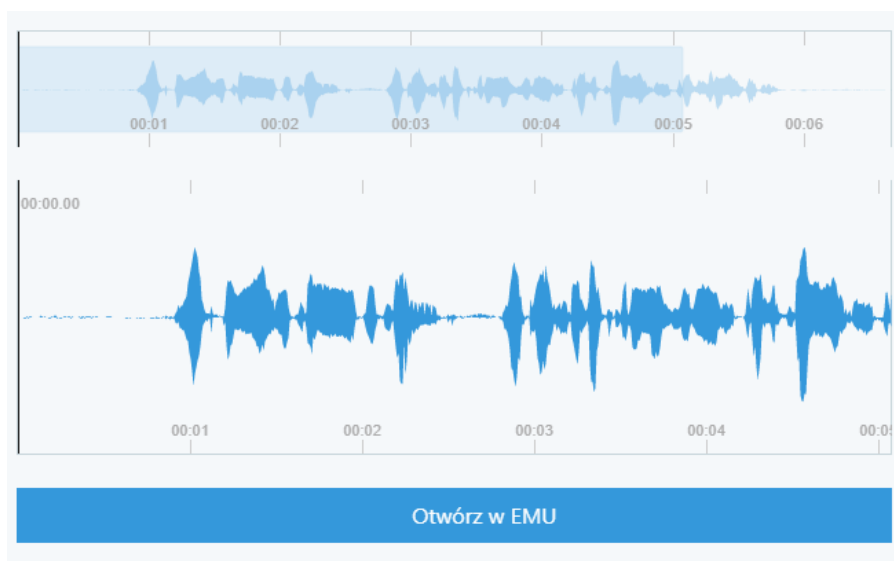


Widok edytora transkrypcji

Segmentacja

Tzw. "Speech alignment" jest jednym z bardziej użytecznych narzędzi. Jest używany do dopasowania sekwencji słów do dostarczonego nagrania audio zawierającego mowę. Wynik narzędzia może być rozumiany jak automatyczne generowanie kodów czasowych gdy znane są zarówno audio jak i jego transkrypcja. Jest to o tyle użyteczne narzędzie ponieważ może być użyte do łatwego przeszukiwania konkretnych zdarzeń w dużych zbiorach nagrań. Umożliwia także obliczanie statystyk odnoszących się do czasu poszczególnych zdarzeń (oraz innych ich charakterystyk).

Segmentacja wykonywana jest automatycznie a jej podgląd można obejrzeć w osobnym edytorze EMU-webApp po kliknięciu przycisku “Otwórz w EMU”.



Widok edytora po ukończeniu segmentacji



Widok rezultatów segmentacji w edytorze EMU-SDMS

Konwersja zapisu ortograficznego na fonetyczny

Narzędzie to pozwala na konwersję każdego tekstu napisanego ortograficznie na jego formę fonetyczną (mówioną). Jest to jeden z podstawowych kroków w każdym procesie przetwarzania danych mowy. Narzędzie akceptuje każdą formę tekstu, jednakże nie wykonuje normalizacji tekstu. Oznacza to iż nie zamienia liczb, dat oraz skrótów w sposób automatyczny. Narzędzie może generować zarówno listy wyrazów uwzględniające różne wymowy (z efektu koartykulacji wynikającego z kontekstu) jak również kanoniczną transkrypcję tekstu.

Narzędzie wykorzystuje wariant alfabetu fonetycznego SAMPA, zmodyfikowanego tak aby zawierał tylko litery alfabetu (bez symboli jak apostrof czy tylda które zostały zastąpione literami i oraz n).

Dostęp do narzędzia został zapewniony z poziomu zakładki “Kokpit” jako iż narzędzie to stanowi dodatek i nie wchodzi w przepływ pracy nad korpusem.

Widok narzędzia do transkrypcji fonetycznej

Rozpoznawanie słów kluczowych

Często dokładna transkrypcja materiału audio nie jest konieczna ponieważ jesteśmy zainteresowani tylko występowaniem pojedynczych słów. Wykrywanie słów kluczowych jest procesem który pobiera plik audio oraz listę słów kluczowych. Następnie generuje listę występowania tych słów w obrębie pliku audio. Należy jednak zwrócić uwagę że model języka ma ograniczoną wielkość słownika, dlatego też niemożliwym jest przewidzieć wszystkich możliwych wyrazów. Z tego powodu, system używa kombinacji słów oraz wyrazów w taki sposób że gdy jest potrzeba znalezienia słowa spoza słownika, używana jest reprezentacja sylabowa danego słowa. Dzięki temu system radzi sobie ze słowami które są spoza słownika, ale jest bardziej podatny na błędy gdy dostarczone są bardzo krótkie słowa kluczowe. W celu przetestowania narzędzia został przygotowany test który pokazał całkowitą precyzję na poziomie ~95% oraz poziom czułości (Recall) dla znanych wyrazów ~82% oraz niski poziom dla wyrazów nieznanach (~20%). Model oparty o sylaby wymaga poprawy w przyszłości aby uniknąć błędów w przypadku wyrazów nieznanach.

Dostęp do narzędzia został zapewniony z poziomu zakładki "Kokpit" jako iż narzędzie to stanowi dodatek i nie wchodzi w przepływ pracy nad korpusem. Aby z niego skorzystać należy kliknąć prawym przyciskiem myszy na wybranym kontenerze w repozytorium i skopiować jego ID. Następnie należy je wkleić w podanym polu w narzędziu.

The screenshot shows a web-based interface for key word detection. At the top, a dark blue header contains the title "Detekcja Słów Kluczowych". Below this, a light blue instruction box says "Wpisz słowa kluczowe jeden po drugim, każdy w nowej linii po znaku enter." (Enter keywords one by one, each on a new line after the enter key). The main area is divided into two columns: the left column is labeled "Słowa kluczowe" (Keywords) and contains a large, empty text area; the right column is labeled "Tutaj pokażą się rezultaty" (Results will appear here). At the bottom, a light blue box contains the instruction "Wklej identyfikator wybranego kontenera z repozytorium" (Paste the identifier of the selected container from the repository), followed by a text input field with the placeholder "Tutaj wklej ID kontenera" (Paste container ID here). Below the input field is a blue button labeled "Uruchom detekcję słów kluczowych" (Run key word detection).

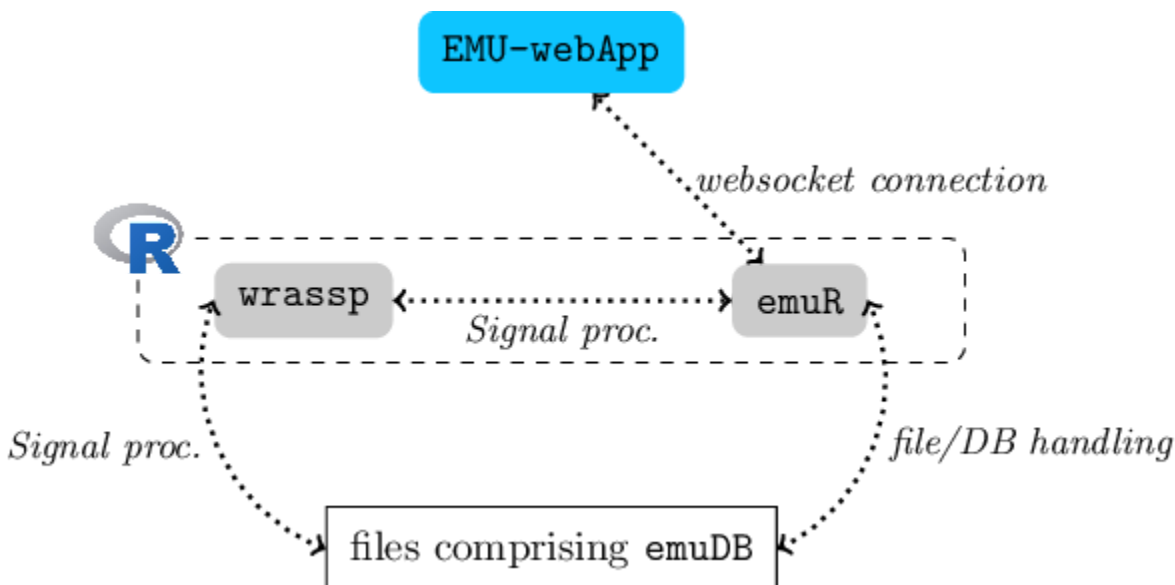
Widok narzędzia detekcji słów kluczowych.

Analizy korpusowe

Analiza korpusowa jest realizowana przede wszystkim w celach badawczych. Jest to złożony proces, wymagający przede wszystkim dobrze opisanej bazy danych oraz narzędzi do jej przetwarzania. Cały proces budowania korpusów po ich końcową analizę oparliśmy o system EMU-SDMS (EMU - Speech Database Management System). Jest on zbiorem narzędzi których celem jest bycie jak najbliższe rozwiązaniom typu "all-in-one" do generowania, manipulacji, zapytań, analizy oraz zarządzania nagraniami audio zawierającymi mowę.

Infrastruktura EMU-SDMS

System jest skoncentrowany wokół języka statystycznego R. System EMU-SDMS zawiera 4 istotne biblioteki: **wrassp**, **emuR**, **emuDB**, **EMU-webApp**. Te 4 komponenty pomagają w przeprowadzaniu badań dotyczących mowy. Dzięki nim możemy np. odpowiedzieć na pytanie badawcze: *biorąc pod uwagę daną bazę danych, czy na wysokość samogłoski a (mierzoną przez jej pierwszy formant) ma wpływ to, czy pojawia się ona w wyrazie czy jako znak funkcyjny?*



Schemat architektury systemu EMU-SDMS

Trzeba zauważyć że pakiet **emuR** jest jedynym komponentem który komunikuje się ze wszystkimi innymi. Z tego powodu jest najistotniejszy. Zajmuje się obsługą plików oraz bazy danych. Używa pakietu **wrassp** do przetwarzania sygnałów oraz dostarcza rezultaty do **EMU-webApp**. Pomimo iż system składa się z 4 komponentów, użytkownik używa przede wszystkim **Emu-webApp** oraz **emuR**.

Etapy pracy badawczej

W przypadku pracy z systemem EMU-SDMS wyglądają one następująco:

1. Załadowanie bazy danych do sesji R (**load_emuDB()**)
2. Anotacja bazy oraz wizualna inspekcja (**serve()**). Operacja ta otwiera **EMU-webApp** w przeglądarce
3. Zapytanie do bazy danych (**query()**). Opcjonalnie przy użyciu funkcji **requery_hier()** lub **requery_seq()**
4. Zbieranie danych śledzących (np. wartości formantów) do wyników zapytania

5. Przygotowanie oraz Wizualna inspekcja danych
6. Przeprowadzenie dalszych analiz oraz przetwarzanie statystyczne

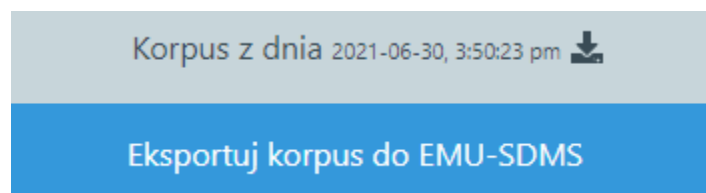
Początkowo użytkownik tworzy referencję do **emuDB** poprzez funkcję **load_emuDB()**. Tak uzyskana referencja do bazy może zostać użyta albo do wyświetlania jej w EMU-webApp (**serve()**) lub do tworzenia zapytań (**query()**). Rezultaty zapytania mogą wtedy być użyte albo do wykonania jednego lub więcej tzw. podzapytań (**requires**) lub do wyekstrahowania wartości cech sygnału które odpowiadają wynikowi zapytania. Finalnie, dane sygnału mogą przejść dalsze przygotowania (np. usuwanie wartości skrajnych tzw. "outliers") oraz sprawdzenie wizualnie przed dalszą analizą. Wynikowy obiekt powyższych funkcji (**data.frame**) może być użyty jako wejście do tysięcy innych funkcji języka R. EMU-SDMS znacząco redukuje liczbę narzędzi z którymi badacze muszą się zmagać oraz pomaga ułatwić odpowiedzi na pytania badawcze. Jedynym wymogiem w użyciu jest podstawowa znajomość platformy R.

Aby lepiej zobrazować zasadę działania systemu, przeprowadzimy wszystkie procesy krok-po-kroku aby odpowiedzieć na pytanie badawcze: *mając dany zbiór danych, wysokość samogłoski a jest zależna od tego czy występuje w wyrazie czy jako znak funkcyjny?*. UWAGA: poniższe opracowanie dotyczy tekstów angielskich dlatego w rzeczywistości dotyczy samogłoski @.

Załadowanie bazy danych do sesji R

Gdy nagrania audio są dostępne należy przekształcić je na format **emuDB**. Istnieje kilka metod które umożliwią nam zaimportowanie kolekcji plików audio wraz z ich transkrypcją w formacie np. TextGrid. Służy do tego jedna z funkcji **convert_TextGridCollection()**. Oczywiście istnieją również inne ułatwiające ten proces. Ich opis jest dostępny w [dokumentacji EMU](#).

Proces ten może być jednak zautomatyzowany za pomocą naszej strony do tworzenia korpusów gdzie w lewym dolnym rogu mamy możliwość wygenerowania korpusu EMU-SDMS z załadowanych oraz opisanych kontenerów w repozytorium danego projektu.



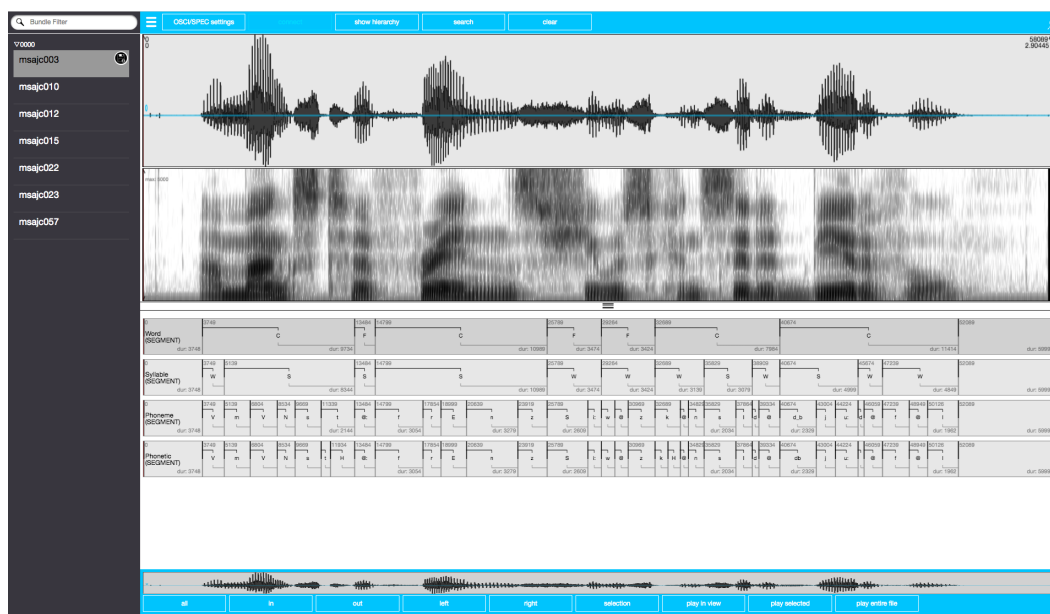
Przyciski do generowania korpusu z repozytorium serwisu

Po utworzeniu korpusu EMU, możemy ściągnąć go na dysk lokalny i dalej pracować na nim w środowisku R.

Ściągnięty korpus można odczytać używając funkcji **load_emuDB()**. Dalej za pomocą funkcji **summary()** możemy podejrzeć strukturę bazy danych.

Wizualna inspekcja danych

Po załadowaniu przychodzi czas na wizualną inspekcję danych w EMU-webApp. Służy to tego funkcja **serve()**. Otwiera ona EMU-webApp w przeglądarce.



Widok stworzonego korpusu w przeglądarce internetowej w EMU-webApp

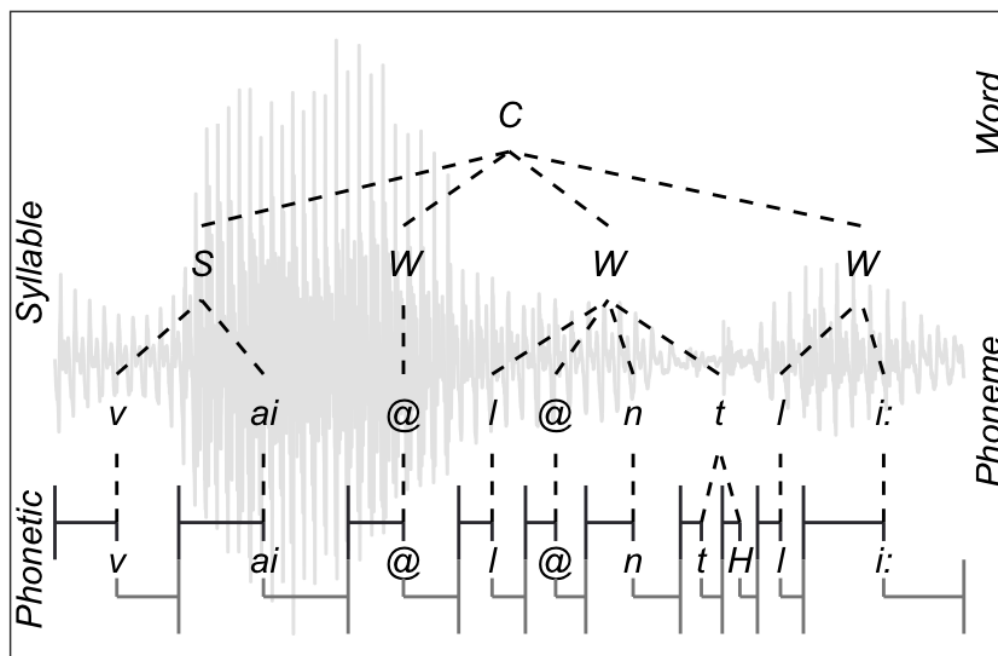
Zapytania do bazy danych

Integralnym krokiem w domyślnym przepływie pracy są zapytania do bazy danych. **emuR** implementuje metodę **query()** do tego celu. Funkcja ta wykonuje wyrażenia EMU Query Language (EQL) oraz wydobywa zanotowane fragmenty z bazy danych które spełniają wymagania zapytania. Możemy wykonywać zapytania osobno dla każdego poziomu np. aby zapytanie zwracało wszystkie fragmenty z poziomu "Syllable", które są równe "S" (silna sylaba). W tym celu wykonaliśmy polecenie:

```
sl_syl = query(emuDBhandle = db_handle, "Syllable == S")
```

Rezultatem takiego zapytania jest obiekt **emuRsegs**, który jest super klasą obiektu **data.frame**. Obiekt ten często odnosi się do listy segmentów które są opisane za pomocą punktu startowego, punktu końcowego, z jakiej sesji oraz paczki pochodzi dany wpis oraz poziom do którego należy.

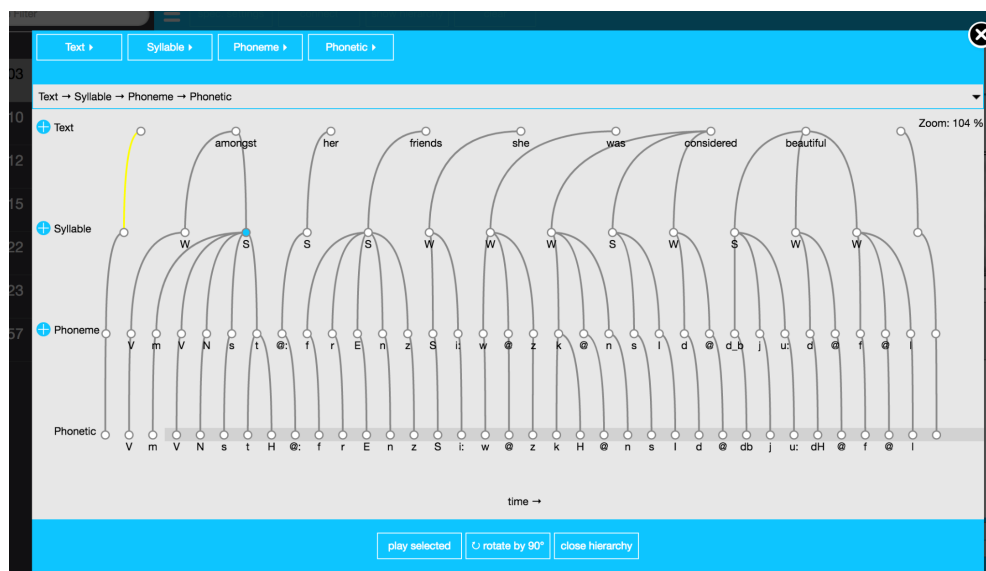
Tego typu zapytania pochodzą z pojedynczego poziomu który pasuje do określonej etykiety. Jednakże EMU-SDMS oferuje mechanizm do wykonywania zapytań wewnątrz poziomów takich jak: *zwróć wszystkie pozycje z poziomu "Phonetic", które zawierają etykietę "n" oraz są częścią zawartości wyrazu*. Aby to było możliwe, EMU-SDMS oferuje dość zaawansowane funkcje modelowania struktury adnotacji. Aby to było jednak możliwe, należy przekształcić płaską strukturę anotacji na hierarchiczną jak zaprezentowano na poniższym rysunku.



Przykład hybrydowej anotacji zawierającej informację czasową (poziom "Phonetic") oraz anotację hierarchiczną (Phoneme, Syllable, Text uwzględniającą ich wewnętrzne powiązania).

Warto zauważyć że tego typu połączenia hierarchiczne są obliczane automatycznie w naszym narzędziu do tworzenia korpusów.

Po utworzeniu struktury można wyświetlić ją w Emu-webApp dzięki komendzie **serve()**. W Emu-webApp możemy zobaczyć utworzoną strukturę klikając przycisk "show hierarchy":



Przykład EMU-webApp wyświetlający automatycznie zbudowaną hierarchię pomiędzy poziomami anotacji.

Hierarchiczna struktura anotacji pozwala na formułowanie zapytań które pomogą odpowiedzieć na bardziej złożone pytania badawcze np. *wysokość samogłoski @ różni się w zależności od tego czy występuje w wyrazie czy jako znak funkcyjny.*

Na początku należy wydobyć wszystkie samogłoski @ z bazy danych z poziomu "Phonetic":

```
sl_vowels = query(db_handle, "Phonetic == @")
```

Potrzebujemy jednak typu wyrazu (zwykły bądź funkcyjny) dla którego zostały znalezione samogłoski @. W tym celu użyjemy funkcjonalności zwanej "requery" z systemu EMU-SDMS. Dzięki temu wydobędziemy typ wyrazu odpowiadający każdej samogłosce @. W skrócie, "requery" porusza się przez hierarchiczną anotację (wertykalnie i horyzontalnie) zaczynając od segmentów które zostały przekazane do funkcji jako parametry.

```
sl_word_type = requery_hier(db_handle, seglist = sl_vowels, level = "Word", calcTimes = FALSE)
```

Należy zwrócić uwagę że liczba wierszy w liście segmentów zwróconych przez funkcję **requery_hier()** jest taka sama jak oryginalna lista samogłosek sl_vowels. Jest to istotne, ponieważ każdy rząd w obu listach segmentów powinien być połączony, pozwalając nam na dopasowanie który segment należy do jakiego typu wyrazu.

Ekstrakcja cech audio

Jeżeli mamy już samogłoski oraz typu wyrazów im odpowiadające (oraz informacje o ich czasie występowania), możemy wyekstrahować cechy sygnału do tych segmentów. Funkcja **get_trackdata()** oblicza wartości formantów w czasie rzeczywistym, używając funkcji estymacji formantów **forest()**, dostarczonej przez pakiet **wrassp**.

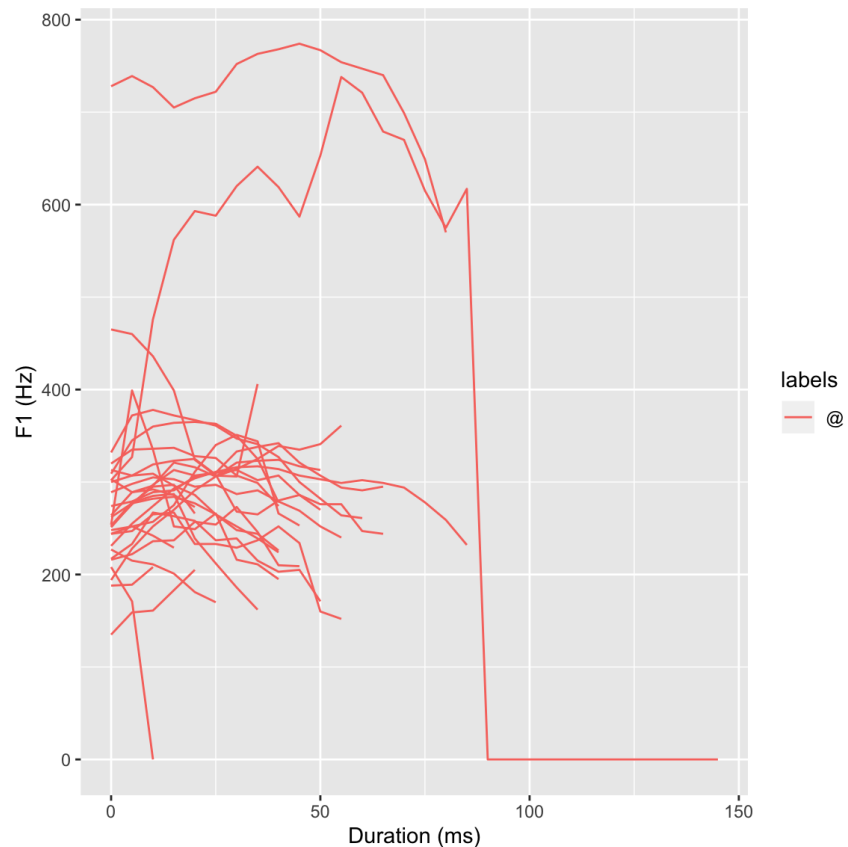
```
td_vowels = get_trackdata(db_handle, seglist = sl_vowels, onTheFlyFunctionName = "forest",  
resultType = "tibble", verbose = F)
```

funkcja **forest()** oblicza 4 wartości formantów. Interesują nas jednak tylko wartości F1 oraz F2.

Wizualna inspekcja danych

Możemy narysować wykres obrazujący trajektorie pierwszego formantu dla wszystkich znalezionych segmentów:

```
# load package  
library(ggplot2)  
  
ggplot(td_vowels) +  
  aes(x = times_rel, y = T1, col = labels, group = sl_rowIdx) +  
  geom_line() +  
  labs(x = "Duration (ms)", y = "F1 (Hz)")
```

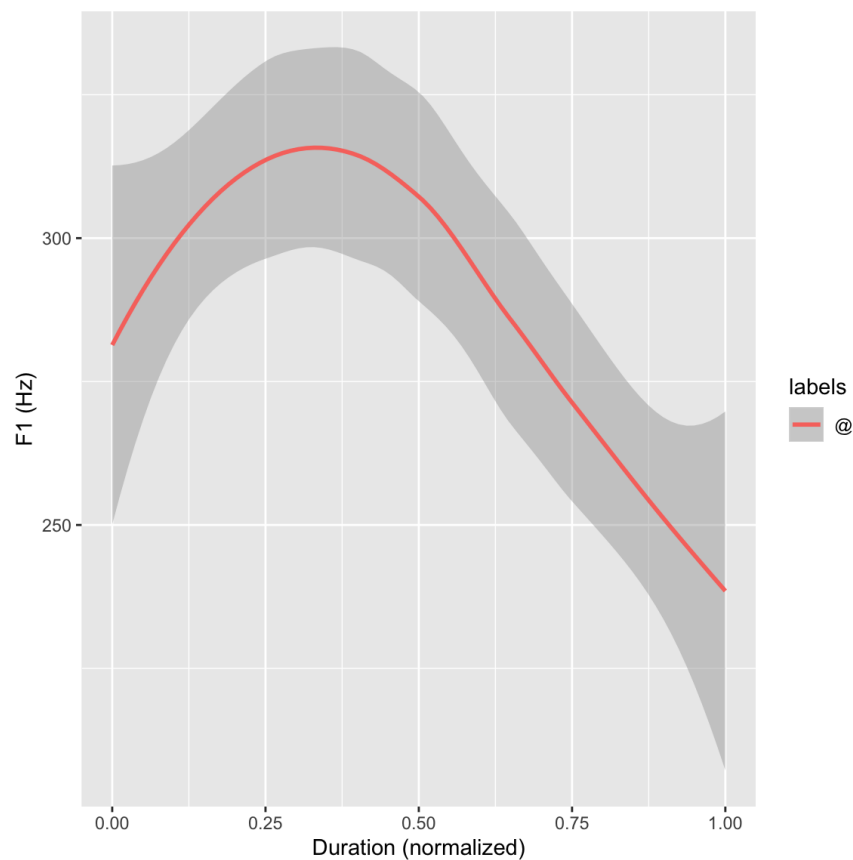



Trajektoria formantów F1 dla wszystkich znalezionych samogłosek @

Jednak aby wyczyścić ten wykres, kolejny zawiera normalizację długości segmentu oraz metodę wygładzającą **geom_smooth()** aby wyświetlić wygładzoną średnią warunkową wszystkich samogłosek @

```
# normalize length of segments
td_vowels_norm = normalize_length(td_vowels)

ggplot(td_vowels_norm) +
  aes(x = times_norm, y = T1, col = labels, group = labels) +
  geom_smooth() +
  labs(x = "Duration (normalized)", y = "F1 (Hz)")
```



Uśredniona i znormalizowana trajektoria formantów F1 dla wszystkich samogłosek @

Powyższe wykresy dają wgląd w trajektorie wszystkich samogłosek @. Do celów eksploracji danych oraz aby uzyskać informację gdzie poszczególne klasy samogłosek leżą w przestrzeni 1 i 2 formantu (które pośrednio dostarczają informacji na temat wysokości samogłoski oraz pozycji języka), możemy użyć ponownie funkcji **ggplot()**. Jednakże tym razem nie chcemy mieć już trajektorii wartości formantów ale zależy nam na dwuwymiarowych danych aby wyświetlić je w przestrzeni F2 x F1. Można to uzyskać np. poprzez ekstrakcję środkowego formantu z każdego segmentu (albo za pomocą parametru **cut** dla funkcji **get_trackdata()** lub używając funkcji **filter()**).

```

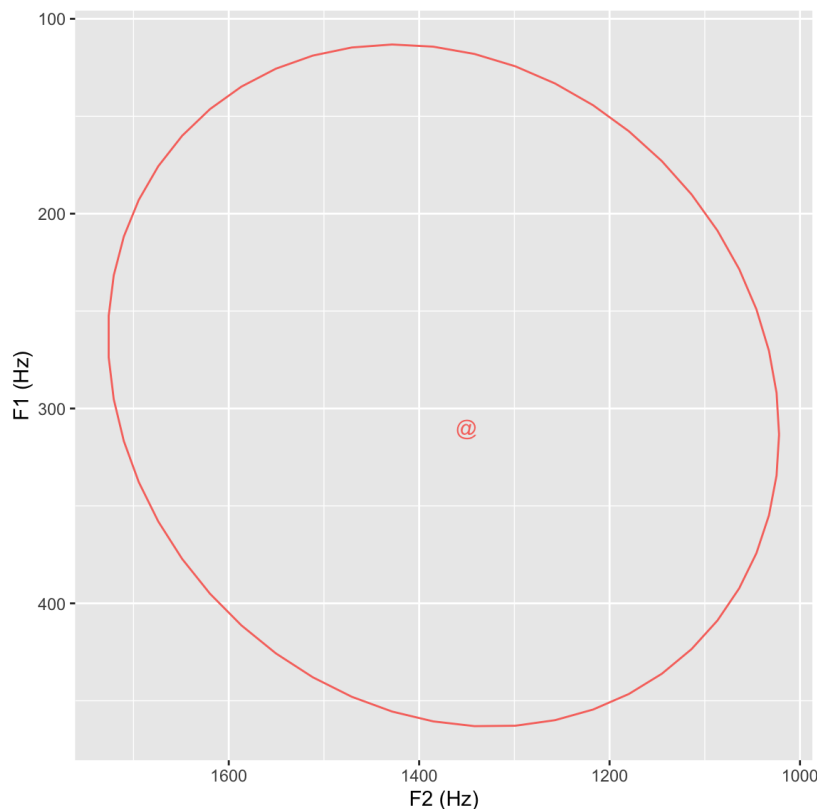
# cut formant trajectories at temporal mid-point
td_vowels_midpoint = td_vowels_norm %>%
  filter(times_norm == 0.5)

# show dimensions of td_vowels_midpoint
dim(td_vowels_midpoint)

# calculate centroid
td_centroids = td_vowels_midpoint %>%
  group_by(labels) %>%
  summarise(T1 = mean(T1), T2 = mean(T2))

# generate plot
ggplot(td_vowels_midpoint, aes(x = T2, y = T1, colour = labels, label = labels)) +
  geom_text(data = td_centroids) +
  stat_ellipse() +
  scale_y_reverse() + scale_x_reverse() +
  labs(x = "F2 (Hz)", y = "F1 (Hz)") +
  theme(legend.position="none")

```



95% elipsa zawierająca centroid dla danych F2 x F1, wyekstrahowanych segmentów samogłosek (z ich środkowych części)

Wykres powyżej pokazuje pierwsze dwa formanty wyekstrahowane z środka każdej samogłoski @. Centroid tych formantów został pokazany na wykresie F2 x F1 oraz ich 95% rozkład.

Powyższe przykłady nie pomagają wprost odpowiedzieć na pytanie zawarte we wstępie, jednak pokazują użyteczność pakietów R jako użyteczne narzędzia do dostarczania wglądu w dane. Aby lepiej zrozumieć język zapytań oraz możliwości generowania wykresów najlepiej przeczytać dokumentację dostępną pod [tym linkiem](#).

Dalsze analizy i wnioskowanie statystyczne

Powyższe analizy dotyczyły jedynie samogłoski @ bez odniesienia do typu wyrazu w którym występowała. Jednakże pytanie badawcze uwzględnia ten kontekst. Możemy więc wyekstrahować 60% formantów z ich trajektorii oraz wyświetlić na wykresie. Jednakże chcemy je pogrupować ze względu na typ wyrazu w którym występują. W tym celu zastąpimy etykiety **td_vowels_mid_sec** etykietami **sl_word_type**:

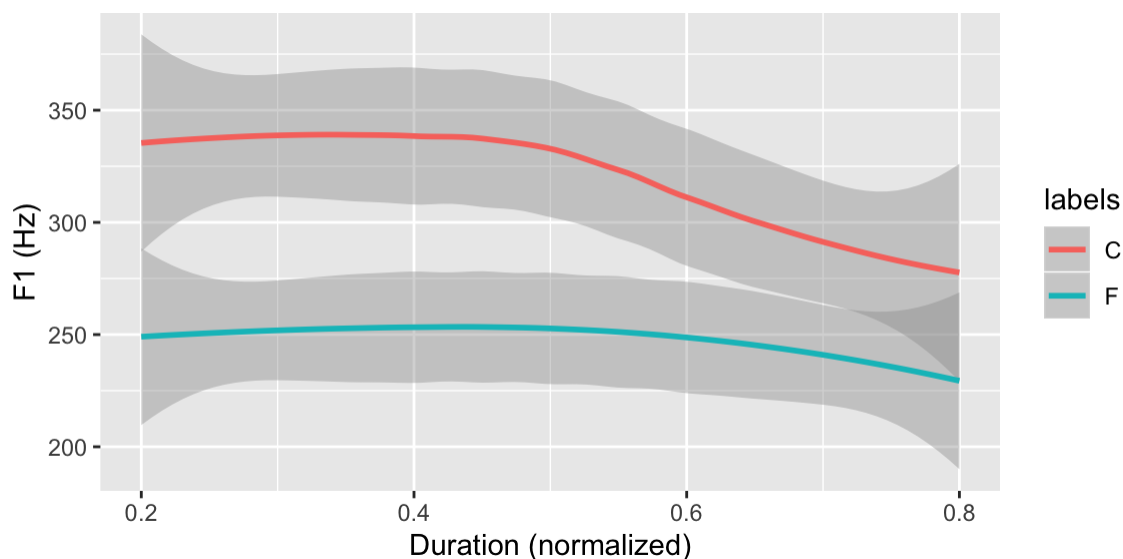
```

# extract central 60% from formant trajectories
td_vowels_mid_sec = td_vowels_norm %>%
  filter(times_norm >= 0.2, times_norm <= 0.8)

# replace labels with those of sl_word_type
td_vowels_mid_sec$labels = sl_word_type$labels[td_vowels_mid_sec$sl_rowIdx]

ggplot(td_vowels_mid_sec) +
  aes(x = times_norm, y = T1, col = labels, group = labels) +
  geom_smooth() +
  labs(x = "Duration (normalized)", y = "F1 (Hz)")

```



Uśrednione kontury formantów F1 wszystkich samogłosek pogrupowanych ze względu na typ wyrazu (funkcyjny F lub zwykły C)

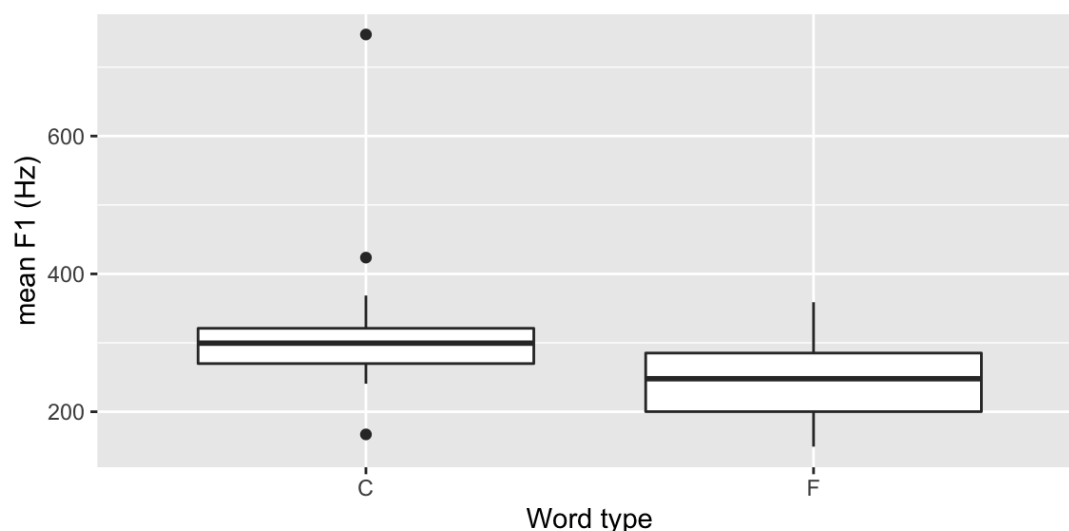
Jak można zauważyć, istnieje różnica między wysokością F1 samogłosek występujących samodzielnie oraz w wyrazie. Aby lepiej to zobrazować możemy wygenerować inny wykres

```

# use group_by + summarise to calculate the means of the 60%
# formant trajectories
td_vowels_mid_sec_mean = td_vowels_mid_sec %>%
  group_by(sl_rowIdx) %>%
  summarise(labels = unique(labels), meanF1 = mean(T1))

# create boxplot using ggplot
ggplot(td_vowels_mid_sec_mean, aes(labels, meanF1)) +
  geom_boxplot() +
  labs(x = "Word type", y = "mean F1 (Hz)")

```



Wykres typu boxplot pokazujący różnicę w rozkładzie F1 w zależności od funkcji samogłoski (jako część wyrazu (C) lub jako znak funkcyjny (F)).

W dalszych krokach można przeprowadzać dalsze wnioskowanie i testowanie hipotez statystycznych. Powyższy opis jest jednak bardzo ogólny i ma na celu jedynie przybliżenie intuicji w jaki sposób działa system EMU-SDMS.

Więcej przykładów oraz wyczerpujący opis użycia EMU-SDMS do rozwiązywania problemów badawczych można znaleźć w [dokumentacji EMU](#).