

Can you develop an application that not only analyzes, **but also evaluates statements on video recordings?** If you have a knack for **web technologies and artificial intelligence is your ally**, then the task is waiting for you! Your goal is to create a tool that instantly evaluates the quality of speeches in official recordings - whether for media training or quick analysis by experts. A simple interface, AI algorithms and your idea are the key to success

### Task description:

1. The competition must deliver two solution components:

- a. A component installed, working, available to the jury for evaluation at the link indicated by the team
- b. Element of source codes with a description of their installation and documentation at the link indicated by the team

**It is required that at the time the solution is submitted for evaluation, it must be made available to the end user. The working solution must be accessible to the evaluators via a link in the browser (Google Chrome or FireFox)** to be indicated by the implementation team at the time of submission of the solution for evaluation. It is stipulated that during the evaluation of the task, the Competition Jury will focus primarily on the content and functional aspects of the solution. Technical issues of the solution may be verified at the stage of its presentation.

**The availability of the solution through the website according to the above guidelines is an essential element for its evaluation.**

2. **What we provide:**

- a. We provide a collection of 20 videos - instructional materials with intentionally applied language errors. These recordings last up to 30 seconds

The recordings include various scenarios: Speech of 1 person who speaks **in Polish** with possible modifications/language errors or other modifications

- b. Video file format: mp4
- c. Link to the recordings is available at: **DISCORD**

In case of technical problems, please report to the **Mentors from the Ministry of Finance** of this task

- d. Each recording has:
  - a board with the information clause: **“The recording is instructional material (with intentionally applied language errors), prepared for the BreakWordTraps - HackYeah 2024 task.”**
  - a board with the applied modifications (errors) in the given recording
  - transcription/subtitles

3. Applied modifications (errors) :

- a) interludes
- b) speaking too fast
- c) repetitions
- d) changing the topic of speech
- e) too many numbers
- f) too long, difficult words, sentences
- g) jargon
- h) foreign language
- i) pausing too long
- j) speaking louder
- k) speaking too quietly, in a whisper
- l) second plan - another person on the set
- m) turning away, twisting, gesticulating
- n) facial expressions
- o) false words
- p) inconsistent speech with the transcript
- q) noise
- r) use of the passive side, e.g. given, indicated, summarized
- s) accentuation

4. Best practices and additional information:

- During the solution evaluation session (pitching), the Jury will evaluate on the basis of **recordings other** than those provided to the contestants. The recordings will be provided **on a flash drive. Recording time of approximately 1min. The recordings include various modifications and errors indicated in point 3.**
- The team before sending to the Jury must check that everything works, e.g. recording on the organizer's platform, links, instructions, etc.
- The team should prepare a short presentation on the solution.

5. **It is not allowed to disseminate the recordings without the permission of the Ministry of Finance** for purposes other than the implementation of the task at the Hackathon

6. **It is required** that for evaluation in the first stage by the Jury in the submitted solution, the information for each recording provided should include:

- a) counted measure of foginess
- b) the evaluation and assignment of modification (error) from the list in item 3

7. **It is required** that the recognition of modifications is done automatically using artificial intelligence methods

## 8. General solution direction for Teams:

Using machine learning algorithms, deep learning, language models (LLM), image generation, audio files, APIs for open source AI solutions, etc., that is, in general, using artificial intelligence methods:

- assess the intelligibility of a message
- perform emotion analysis based on recordings, e.g. gestures, facial expressions, tone of speech
- calculate a measure of plain language: a measure of foginess e.g. Gunning Fog Index, Flesch Readability Index or similar
- measures of subtitle quality
- find among the shared recordings the moment when the modification occurred and assign a tag
- determine the sentiment of the speech (emotion, tone of speech, hate speech)
- describe a short textual summary of the speech in Polish - what was understood by the recipient (key messages)
- assess the structure of the speech - whether the introduction, development and conclusion were preserved
- assess the target group of the analyzed recording in terms of age group or education
- assess the quality of the recording and speech, such as pauses, “possible errors or linguistic slip-ups”.
- assess whether the speech is made by the same person, one person or more people
- assess whether the statement is in Polish or detect the occurrence of words, phrases in another language
- detect irregularities between the utterance and the transcription under the recording

9. The choice **of artificial intelligence methods**, metrics used, ranges, ranges **of definition of terms**, e.g.: sentiment, quality, linguistic errors other than those indicated in point 3, naming, tagging or other aspects **is at the discretion of the teams**.

10. It is allowed to download other video recordings, generate their own recordings to improve the performance of models or applications with similar scenarios. Code scripts to create or download additional **recordings are part of the project and should be included in the final solution**.

11. **It is unacceptable** to download, process recordings that have a suspected possibility of **“malicious code”**.

12. **It is required** to use tools:

- libraries and source codes of the **Python** language
- development platforms designed to create web applications in the **Python** language such as: Streamlit, Flask, Django or others
- if using vector databases - **Postgres database (>=14)**
- Linux or Windows operating system

13. During the competition, if another programming language is used, the approval of the Jury of the Ministry of Finance is required

14. It is allowed to use other technologies **with Python integration** including:

- cloud technologies
- use of **Open AI, Generative AI (GenAI)**
- use of other AI components (free) through APIs

15. **It is unacceptable** to use libraries, scripts, codes, data sources, external APIs, tools of programs about which there is a suspicion of the possibility of **“malicious code”**.

16. The solution should be predominantly based on free services, components, libraries. If it is necessary to use paid services, the total monthly cost should not exceed 500 PLN or 100 euros.

17. **It is required** to good programming practices is required including:

- structure of folders, files and organization of data
- organization and openness of source code
- use of git
- simplified documentation (descriptions can be in Polish)
- portability of the code

18. **It is required** to create instructions for running the code and the technologies used with the version.

19. **It is required** at the time of evaluation of the competition task that the repository made available for evaluation by the Jury, has the necessary codes, libraries, links necessary to run the application

20. **It is required** that by **10:00 p.m. on the first day** of the Hackathon, the teams provide on the organizer's platform **a brief information** (one page in the form of a PDF / Word document):

- short description of the functionality max 5 sentences
- names of the artificial intelligence components
- names of the technology and the architecture of the solution (simple image)
- links to the solution

21. **It is required** that the descriptions in the web application are in Polish

22. Other required and proposed functionality **of the application**:

- **(required)** loading a video file from disk
- **(required)** comparison of another video file(s)
- **(required)** evaluation of similarity between recordings, text
- **(required)** making a transcription of the text with its display on the timeline

- **(required)** automatic evaluation of linguistic aspects of statements with reference to simple language with calculation of indicators and interactive graphs
- **(required)** identification and assignment of a tag for evaluation of linguistic aspects
- **(required)** identification of the sentiment of an utterance along with interactive graphs
- **(required)** identification of the target group of the utterance
- **(required)** extraction of key tasks, phrases
- **(required)** preparation of 10 questions for utterances
- **(required)** facial detection, assignment of emotions and gestures
- **(required)** statistics or characterization of audio file, text, sound
- **(required)** use of machine (ML) or deep learning (DL) models to solve the problems indicated in the document in the following areas: image recognition (computer vision), analysis of audio files (audio), large language models (large language model) or classification.
- **(option)** preparing a summary
- **(option) marking of unintelligible words**
- **(option)** improving the text
- **(option)** identification of people in the background
- **(option)** proposal of word selection
- **(option)** proposal for translation of text into English
- **(option)** models and visualizations for recognition of speech units (NER - Named Entity Recognition), parts of speech (Polish grammar)
- **(option)** other interesting interactive charts
- **(option) identification of whether the audio was generated by AI**

23. Other functionalities in the operation **of the application** than those indicated above are allowed.

Any additional functionality will be taken into account in the evaluation of the Jury of the Competition.

Czy potrafisz stworzyć aplikację, która nie tylko analizuje, ale też **ocenia wypowiedzi na nagraniach video**? Jeśli masz dryg do **technologii webowych**, a **sztuczna inteligencja to Twój sprzymierzeniec**, to zadanie czeka na Ciebie! Twoim celem jest stworzenie narzędzia, które błyskawicznie oceni jakość występów w oficjalnych nagraniach – czy to w celu szkoleń medialnych, czy szybkiej analizy przez ekspertów. **Prosty interfejs**, algorytmy AI i Twój pomysł to klucz do sukcesu

### Opis zadania:

1. W wyniku konkursu dostarczone muszą zostać dwa elementy rozwiązania:
  - a. Element zainstalowany, działający, dostępny dla jury do oceny pod wskazanym przez zespół linkiem
  - b. Element kodów źródłowych z opisem ich instalacji oraz dokumentacją pod wskazanym przez zespół linkiem

**Wymagane jest, aby w momencie przekazania rozwiązania do oceny było ono udostępnione dla użytkownika końcowego. Działające rozwiązanie musi być dostępne dla oceniających za pomocą linku w przeglądarce (Google Chrome lub FireFox), który zostanie wskazany przez zespół realizujący w momencie przekazywania rozwiązania do oceny. Zastrzega się, że w trakcie oceny zadania, Jury Konkursu skupi się przede wszystkim na aspektach merytorycznych oraz funkcjonalnych rozwiązania. Kwestie techniczne rozwiązania mogą zostać zweryfikowane na etapie jego prezentacji.**

**Dostępność rozwiązania przez stronę www wg. powyższych wytycznych to element niezbędny do jego oceny.**

2. **Co dostarczamy:**
  - a. Udostępniamy zbiór 20 filmów wideo – materiałów instruktażowych z celowo zastosowanymi błędami językowymi. Nagrania te trwają do 30 sekund. Nagrania zawierają różne scenariusze: Wystąpienie 1 osoby, która wypowiada się **w języku polskim** z możliwymi modyfikacjami/błędami językowymi lub innymi modyfikacjami
  - b. Format pliku wideo: mp4
  - c. Link do nagrań jest dostępny na DISCORDzie  
W przypadku problemów technicznych należy zgłosić się do **Mentorów z Ministerstwa Finansów tego zadania**
  - d. Każde nagranie posiada:
    - planszę z klauzulą informacyjną: „**Nagranie jest materiałem instruktażowym (z celowo zastosowanymi błędami językowymi), przygotowanymi na potrzeby zadania BreakWordTraps - HackYeah 2024**”
    - planszę z zastosowanymi modyfikacjami (błędami) w danym nagraniu
    - transkrypcję/napisy

3. **Zastosowane modyfikacje (błędy) :**

- a) przerywniki
- b) za szybkie tempo mówienia
- c) powtórzenia
- d) zmiana tematu wypowiedzi
- e) za dużo liczb
- f) za długie, trudne słowa, zdania
- g) żargon
- h) obcy język
- i) za długa pauza
- j) mówienie głośniej
- k) mówienie za cicho, szeptem
- l) drugi plan – inna osoba na planie
- m) odwracanie się, przekręcenia, gestykulacja
- n) mimika
- o) nieprawdziwe słowa
- p) niezgodna wypowiedź z transkrypcją
- q) szum
- r) używanie strony biernej np. podano, wskazano, podsumowano
- s) akcentowanie

4. **Dobre praktyki oraz informacje dodatkowe:**

- W trakcie trwania sesji oceny rozwiązania (pitching), Jury dokona oceny na podstawie **innych nagrań** niż dostarczone uczestnikom konkursu. Nagrania zostaną dostarczone **na pendrive. Czas nagrania ok. 1min. Nagrania zawierają różne modyfikacje oraz błędy wskazane w punkcie 3.**
- Zespół przed wysłaniem do Jury musi sprawdzić, czy wszystko działa np. zapis na platformie organizatora, linki, instrukcje itp.
- Zespół powinien przygotować krótką prezentację dotyczącą rozwiązania.

5. **Niedozwolone jest upowszechnianie nagrań bez zgody Ministerstwa Finansów** w innych celach niż realizacja zadania na Hackathonie

6. **Wymagane** jest, aby do oceny w pierwszym etapie przez Jury w nadesłanym rozwiązaniu została zawarta informacja dla każdego dostarczonego nagrania:

- a) policzona miara mglistości
- b) ocena oraz przypisanie modyfikacji (błędu) z listy w punkcie 3

7. **Wymagane** jest, aby rozpoznawanie modyfikacji odbywało się w sposób automatyczny z wykorzystaniem metod sztucznej inteligencji
8. **Ogólny kierunek rozwiązania dla Zespołów:**

Wykorzystując algorytmy uczenia maszynowego (machine learning), uczenia głębokiego (deep learning), modeli językowych (LLM), generowania obrazów, plików audio, API do rozwiązań open source w zakresie AI itp., czyli ogólnie wykorzystując metody sztucznej inteligencji:

  - dokonać oceny zrozumiałości przekazu
  - dokonać analizy emocji na podstawie nagrań np. gestykulacja, mimika, ton wypowiedzi
  - wyliczyć miarę prostego języka: miara mglistości np. współczynnik mglistości Gunninga (Gunning Fog Index), Indeks czytelności Flescha lub podobnych
  - miary jakości napisów
  - znaleźć wśród udostępnionych nagrań moment, w którym doszło do modyfikacji oraz przypisać tag
  - określić sentyment wypowiedzi (emocje, ton wypowiedzi, hate speech)
  - opisać krótkie tekstowe podsumowanie wypowiedzi w języku polskim – co zostało rozumiane przez odbiorcę (kluczowe przekazy)
  - ocenić strukturę wypowiedzi - czy był zachowany wstęp, rozwinięcie i zakończenie
  - ocenić grupę docelową analizowanego nagrania ze względu na grupę wiekową lub wykształcenie
  - ocenić jakość nagrania oraz wypowiedzi np. pauzy, , „możliwe błędy lub wpadki językowe”
  - ocenić, czy wypowiada się ta sama osoba, jedna osoba lub jest ich więcej
  - ocenić, czy wypowiedź jest w języku polskim lub wykryć wystąpienie słów, fraz w innym języku
  - wykryć nieprawidłowości między wypowiedzią, a transkrypcją pod nagraniem
9. Wybór **metod sztucznej inteligencji**, stosowanych metryk, przedziałów, zakresów **definicji pojęć** np.: sentymentu, jakości, błędów językowych innych niż wskazanych w punkcie 3, nazewnictwa, tagowania lub innych aspektów **należy do decyzji zespołów**.
10. Dopuszcza się pobranie innych nagrań wideo, wygenerowanie własnych nagrań w celu ulepszenia działania modeli lub aplikacji o podobnych scenariuszach. Skrypty kodu służące do utworzenia lub pobrania dodatkowych nagrań **są częścią projektu i powinny znaleźć się w ostatecznym rozwiązaniu**.
11. **Niedopuszczalne** jest pobranie, przetwarzanie nagrań mających podejrzenie możliwości wystąpienia „**złośliwego kodu**”.
12. **Wymagane jest** wykorzystanie narzędzi:
  - bibliotek oraz kodów źródłowych języka **Python**



- platform programistycznych przeznaczony do tworzenia aplikacji internetowych w języku **Python** np.: Streamlit, Flask, Django lub inne
  - w przypadku korzystania baz wektorowych - baza danych **Postgres (>=14)**
  - system operacyjny Linux lub Windows
13. W trakcie trwania konkursu, w przypadku wykorzystania innego języka programowania konieczna jest zgoda Jury Ministerstwa Finansów
14. Dopuszcza się wykorzystanie innych technologii z **integracją z językiem Python** w tym:
- technologii chmurowych
  - wykorzystanie **Open AI, Generative AI (GenAI)**
  - wykorzystanie innych komponentów AI (darmowych) poprzez API
15. **Niedopuszczalne** jest korzystanie z bibliotek, skryptów, kodów, źródeł danych, zewnętrznych API, narzędzi programów co, do których istnieje podejrzenie możliwości wystąpienia „**złośliwego kodu**”.
16. Rozwiązanie powinno być w przeważającej części oparte o darmowe usługi, komponenty, biblioteki. W przypadku konieczności wykorzystania usług płatnych całkowity miesięczny koszt nie powinien przekroczyć 500 zł lub 100 euro.
17. **Wymagane** jest przestrzeganie dobrych praktyk programistycznych w tym:
- struktura folderów, plików oraz organizacja danych
  - organizacja oraz otwartość kodu źródłowego
  - korzystanie z git’a
  - uproszczona dokumentacja (opisy mogą być w języku polskim)
  - przenośność kodu
18. **Wymagane** jest stworzenie **instrukcji uruchomienia kodu** oraz zastosowanych technologii z podaniem wersji.
19. **Wymagane** jest w momencie oceny zadania konkursowego, aby udostępnione repozytorium do oceny przez Jury, posiadało niezbędne kody, biblioteki, linki niezbędne do uruchomienia aplikacji
20. **Wymagane** jest, aby do godziny **22:00 pierwszego** dnia Hackatonu zespoły dostarczyły na platformie organizatora **skróconą informację** (jedna strona w postaci dokumentu PDF / Word):
- krótki opis funkcjonalności max 5 zdań
  - nazwy komponentów sztucznej inteligencji
  - nazwy technologii oraz architektura rozwiązania (prosty obrazek)
  - linki do rozwiązania

21. **Wymagane jest**, aby opisy w aplikacji webowej były w języku polskim.

22. Inne wymagane oraz proponowane funkcjonalności **aplikacji**:

- **(wymagane)** załadowanie z dysku pliku wideo
- **(wymagane)** porównanie innego pliku/plików wideo
- **(wymagane)** ocena podobieństwa pomiędzy nagraniami, tekstem
- **(wymagane)** dokonanie transkrypcji tekstu wraz z jej wyświetleniem na osi czasu
- **(wymagane)** automatyczna ocena aspektów językowych wypowiedzi w nawiązaniu do prostego języka wraz z wyliczeniem wskaźników oraz interaktywnymi wykresami
- **(wymagane)** identyfikacja oraz przypisanie tagu do oceny aspektów językowych
- **(wymagane)** określenie sentymentu wypowiedzi wraz z interaktywnymi wykresami
- **(wymagane)** określenie grupy docelowej wypowiedzi
- **(wymagane)** ekstrakcja kluczowym zadań, fraz
- **(wymagane)** przygotowanie 10 pytań do wypowiedzi
- **(wymagane)** wykrywanie twarzy, przypisanie emocji oraz gestykulacji
- **(wymagane)** statystyki lub charakterystyka pliku audio, tekstu, dźwięku
- **(wymagane)** wykorzystanie modeli machine (ML) lub deep learning (DL) do rozwiązania wskazywanych w dokumencie problemów w zakresie: rozpoznawania obrazu (*computer vision*), analizy plików dźwiękowych (*audio*), dużych modeli językowych (*large language model*) czy klasyfikacji.
- **(opcja)** przygotowanie podsumowania
- **(opcja) zaznaczenie wyrazów niezrozumiałych**
- **(opcja)** poprawa tekstu
- **(opcja)** identyfikacja osób na drugim planie
- **(opcja)** propozycja doboru słów
- **(opcja)** propozycja tłumaczenia tekstu na język angielski
- **(opcja)** modele oraz wizualizacje do rozpoznawania jednostek mowy (NER - Named Entity Recognition), części mowy (polska gramatyka)
- **(opcja)** inne ciekawe interaktywne wykresy
- **(opcja) identyfikacja czy audio zostało wygenerowane przez AI**

23. Dopuszcza się inne funkcjonalności w działaniu **aplikacji** niż wskazane powyżej.

Każda dodatkowa funkcjonalność będzie brana pod uwagę w ocenie Jury Konkursu.