

# Implementacja aplikacji wspierającej analizę głosu w oparciu o repozytorium COVAREP

Hubert Daniszewski (s19110)

Promotor - prof. dr inż. Krzysztof Szklanny

Styczeń 2024

## Spis treści

1 Wykorzystanie Microsoft Copilot w celu liczenia parametrów COVAREP	3
---	---

# 1 Wykorzystanie Microsoft Copilot w celu liczenia parametrów COVAREP

W ramach projektu, dodatkowo została podjęta próba wykorzystania modelu generatywnego Microsoft Copilot w celu zweryfikowania, czy jest on zdolny do wygenerowania kodu pozwalającego na wyliczenie parametrów Peak Slope, Cepstral Peak Prominence oraz Normalized Amplitude Quotient.

Microsoft Copilot jest oparty na 3 różnych modelach sztucznej inteligencji [Microsoft (2023)]:

- ChatGPT-4 - duży model językowy opracowany przez firmę OpenAI, uczony na bardzo dużym korpusie tekstów w celu dostarczania trafnych informacji na różnorodne tematy [Roumeliotis and Tselikas (2023)]
- Prometheus - zastrzeżony przez Microsoft zbiór technik pracy z modelami uczenia maszynowego od OpenAI mających na celu zapewnienie trafnych, aktualnych i ukierunkowanych wyników przy jednoczesnej poprawie bezpieczeństwa [Mehdi (2024)]
- DALL-E 3 - model firmy OpenAI służący do generowania obrazów na podstawie tekstu

Modele ChatGPT-4 oraz DALL-E 3 oparte są o architekturę transformerów, która została opisana w artykule "Attention is all you need" [Vaswani et al. (2017)].

W ramach eksperymentu, Microsoft Copilot został poproszony o wygenerowanie skryptów w języku Octave pozwalających wyliczyć parametry PS, CPP oraz NAQ przy pomocy zapytań w następującym formacie:

```
How to calculate X when analyzing a voice recording in
↪ Octave?
```

Gdzie X jest pełną nazwą jednego z trzech parametrów. Każde zapytanie zostało wykonane jako pierwsze w osobnym czacie w języku angielskim. W przypadku błędów w trakcie uruchamiania skryptu, do modelu zostały wysłane kolejne wiadomości opisujące zaistniały problem do momentu wygenerowania funkcjonalnego kodu. Dodatkowo, po uzyskaniu skryptów zwracających wartości dla zadanych parametrów, została przekazana wiadomość z informacją o tym, że wynik jest różny od wyników dostarczanych przez COVAREP z prośbą o dokonanie analizy skryptów z repozytorium COVAREP, wyjaśnienie z czego wynika różnica w wynikach oraz zmodyfikowanie kodu wygenerowanego przez model w ten sposób, by dane parametry były liczone zgodnie z implementacją z repozytorium COVAREP. Te zapytania zostały napisane zgodnie z następującym formatem:

The output of the script is different from the COVAREP's  
 ↪ implementation of the X calculation algorithm. Please  
 ↪ take a look at the script at: <LINK DO SKRYPTU LICZĄCEGO  
 ↪ PARAMETR W COVAREP> and analyze what may be the cause of  
 ↪ the output differences. If possible, adjust your code so  
 ↪ that it matches COVAREP's calculations.

Wszelkie błędów w trakcie wykonywania skryptów uwzględniających COVAREP również zostały zaadresowane w kolejnych zapytaniach do modelu. Pełna historia czatów jest dostępna w plikach PS\_copilot.txt CPP\_copilot.txt oraz NAQ\_copilot.txt. Do kodu wygenerowanego przez Copilot zostały też ręcznie wprowadzone zmiany dotyczące nazw plików wejściowych, funkcji wyświetlających wyniki w konsoli, itp. w celu zwiększenia komfortu przeglądania wyników. Wartości obliczone przez skrypty przed i po uwzględnieniu repozytorium COVAREP zostały zebrane, porównane z wartościami wyliczonymi przez skrypty COVAREP i zestawione razem w tabeli 1.

	COVAREP	Microsoft Copilot	Microsoft Copilot po uwzględnieniu COVAREP
PS	-0.395743	0.0903	0.0062304
CPP	7.11601	0.78335	6.3649
NAQ	0.0946473	-0.077603	-0.30935

Tabela 1: Porównanie wartości parametrów głosowych w zależności od źródła implementacji funkcji liczących

Ostatecznie, skrypty wygenerowane przez Microsoft Copilot nie liczą parametrów PS, CPP oraz NAQ prawidłowo. Uwzględnienie COVAREP w zapytaniach nie miało istotnego wpływu na jakość wyników. Powody tego mogą być następujące:

- Zbyt duży poziom skomplikowania algorytmów liczących parametry - modele GPT są uczone wszechstronnie. Nawet przy użyciu metod *fine tuningu*, są one narażone na "halucynacje", czyli sytuacje, w których model fabrykuje informacje z ryzykiem wprowadzenia użytkownika w błąd [Azamfirei et al. (2023)]
- Niewystarczająca jakość zapytań - pomimo wykorzystywania dobrych praktyk, takich jak pisanie konkretnych poleceń, rozwijanie skrótów oraz nakierowywanie modelu poprzez "ciąg myśli", nie mamy pewności, czy model wygeneruje odpowiedzi zgodne z prawdą. Błędy mogą wynikać dodatkowo m.in. ze względu na brak możliwości modelu do zweryfikowania przekazywanych informacji, przetrenowania lub stroniczości w danych uczących [Taulli (2024)].

## Literatura

- Razvan Azamfirei, Sapna R. Kudchadkar, and James Fackler. 2023. Large language models and the perils of their hallucinations. *Critical Care* 27, 1 (March 2023), 1–2. <https://doi.org/10.1186/s13054-023-04393-x>
- Yusuf Mehdi. 2024. *Reinventing search with a new AI-powered Microsoft Bing and Edge, your copilot for the web*. Microsoft. <https://blogs.microsoft.com/blog/2023/02/07/reinventing-search-with-a-new-ai-powered-microsoft-bing-and-edge-your-copilot-for-the-web/>
- Microsoft. 2023. How Copilot works, technically speaking. <https://www.microsoft.com/en-us/bing/do-more-with-ai/how-bing-chat-works>
- Konstantinos I. Roumeliotis and Nikolaos D. Tselikas. 2023. ChatGPT and Open-AI Models: A Preliminary Review. *Future Internet* 15, 6 (May 2023), 192. <https://doi.org/10.3390/fi15060192>
- Tom Taulli. 2024. *Ai-assisted programming: Better planning, coding, testing, and deployment*. O'Reilly Media, Inc. 51–56 pages.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. <https://doi.org/10.48550/ARXIV.1706.03762>