



Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie

Wydział Informatyki

PRACA DYPLOMOWA

## **Generacja muzyki przy pomocy dużych modeli językowych**

**Music generation with Large Language Models**

Autor:	<b>Filip Ręka</b>
Kierunek:	<b>Informatyka — Data Science</b>
Opiekun pracy:	<b>dr hab. Maciej Smółka prof. AGH</b>

Kraków, 2024



*Tutaj możesz umieścić treść podziękowań. Tutaj możesz umieścić treść podziękowań. Tutaj możesz umieścić treść podziękowań. Tutaj możesz umieścić treść podziękowań.*



## Streszczenie

Duże modele językowe (ang. *Large Language Models* **LLM**) charakteryzują się zdolnością do generacji języka oraz innych zadań w przetwarzania języka naturalnego, takich jak na przykład klasyfikacja. Zdolność tą nabierają podczas czasochłonnego oraz intensywnego obliczeniowego treningu metodami samo odraz pół-nadzorowanego, podczas którego uczą się one relacji z wielkiej ilości dokumentów tekstowych. LLMy mogą zostać wykorzystane do generacji tekstu, formy generatywnej sztucznej inteligencji, poprzez pobieranie tekstu wejściowego i wielokrotne przewidywanie kolejnego tokenu lub słowa w tekście. Strukturę muzyki można porównać struktury tekstu pisanego, gdzie każda nuta odpowiada literze lub słowu, akordy zdaniom a dłuższe i sekwencję paragrafom. Poniższa praca, zamierza zbadać możliwości generacyjne LLMów wytrenowanych na muzycznych zbiorach danych.



## Abstract

Abstract in English [\[1\]](#) ...

# Spis treści

<b>Lista kodów źródłowych</b>	<b>xiii</b>
<b>1 Wstęp</b>	<b>1</b>
1.1 Cel i zakres pracy . . . . .	1
<b>2 Część literaturowa</b>	<b>3</b>
2.1 Cyfrowa reprezentacja muzyki . . . . .	3
2.1.1 WAV (ang. <i>waveform audio format</i> ) . . . . .	3
2.1.2 MIDI (ang. <i>Musical Instrument Digital Interface</i> ) . . . . .	3
2.1.3 Podobieństwa reprezentacji muzyki oraz tekstu . . . . .	3
2.1.4 Tokenizacja . . . . .	3
2.2 Zbiory danych . . . . .	3
2.2.1 Johann Sebastian Bach Chorales . . . . .	3
2.2.2 The MAESTRO v3.0 . . . . .	3
2.2.3 Million Song Dataset . . . . .	3
2.3 STOA . . . . .	3
2.4 Architektury transformera . . . . .	4
2.4.1 Algorytm uwagi (ang. <i>attention</i> ) . . . . .	4
2.4.2 Warianty mechanizmu uwagi . . . . .	4
2.4.2.1 Self attention . . . . .	4
2.4.2.2 Multi-headed attention . . . . .	4
2.4.2.3 Flash attention . . . . .	4
2.4.3 Budowa transformera . . . . .	4
2.4.4 Modele tranformerowe . . . . .	4
2.4.4.1 <i>Classic</i> transformer . . . . .	4
2.4.4.2 SeqGAN . . . . .	4
2.4.4.3 Mistral . . . . .	4
2.5 Architektura <i>state space</i> . . . . .	4
2.5.1 Mamba . . . . .	4
2.5.2 Tutaj się rozdrobnić trzeba . . . . .	4



<b>3</b>	<b>Część badawcza</b>	<b>7</b>
3.1	Opis <i>pipeline-u</i> . . . . .	7
3.2	Porównanie architektur użytych modeli . . . . .	7
3.3	Prezentacja otrzymanych wyników . . . . .	7
3.4	Porwanie wyników . . . . .	7
<b>4</b>	<b>Zakończenie</b>	<b>9</b>
<b>Dodatek A.</b>	<b>Typowe elementy składowe pracy dyplomowej z informatyki</b>	<b>11</b>
A.1	Tabele . . . . .	11
A.2	Rysunki . . . . .	13
A.2.1	Wewnętrzne . . . . .	13
A.2.2	Zewnętrzne . . . . .	14
A.3	Kody źródłowe . . . . .	14
A.4	Algorytmy . . . . .	16
A.5	Wzory . . . . .	16
A.5.1	Przykłady . . . . .	17
A.6	Twierdzenia i podobne struktury . . . . .	17
	<b>Uwagi Autora</b>	<b>19</b>
	<b>Bibliografia</b>	<b>21</b>

5

Zawartość spisu treści — tytuły rozdziałów oraz ich liczba zależą od tematyki pracy — należy ustalić z opiekunem pracy.

## Spis rysunków

2.1	Schemat transformera. . . . .	5
2.2	Schemat modelu Mamba. . . . .	6
A.1	Prosty rysunek <i>TikZ</i> . . . . .	13
A.2	Bardziej złożony rysunek <i>TikZ</i> . . . . .	13
A.3	Logo Wydziału Informatyki. . . . .	14

## Spis tabel

A.1	Pomiary zużycia energii elektrycznej. . . . .	11
A.2	Tabela, która zawiera dużą liczbę wierszy. . . . .	11
A.3	Tabela zawierająca długi tekst. . . . .	12

# Lista algorytmów

1	Disjoint decomposition. . . . .	16
---	---------------------------------	----



# Lista kodów źródłowych

A.1	Przykładowy kod źródłowy sformatowany za pomocą pakietu 'listings'. . . .	15
A.1.	Przykładowy listing sformatowany za pomocą pakietu 'minted'. . . . .	15



# 1. Wstęp

**Uwaga 1.1.** Tytuł oraz strukturę rozdziału należy ustalić z opiekunem pracy.

Wprowadzenie w tematykę pracy.

## 1.1. Cel i zakres pracy

Streszczenie specyfikacji wymagań Promotora.





## 2. Część literaturowa

### 2.1. Cyfrowa reprezentacja muzyki

#### 2.1.1. WAV (ang. *waveform audio format*)

#### 2.1.2. MIDI (ang. *Musical Instrument Digital Interface*)

#### 2.1.3. Podobieństwa reprezentacji muzyki oraz tekstu

#### 2.1.4. Tokenizacja

[2]

### 2.2. Zbiory danych

#### 2.2.1. Johann Sebastian Bach Chorales

Dataset [3]

#### 2.2.2. The MAESTRO v3.0

Dataset [4]

#### 2.2.3. Million Song Dataset

Dataset i takie cytowanko [5]

### 2.3. STOA

Tutaj nie wiem do końca w jakiej kolejności chciałbym o tym pisać, ponieważ z jednej strony przedstawienie STOA przed czymkolwiek jest ok, ale nie chciałbym pisać o czymś czego jeszcze w pracy nie wprowadziłem.

## **2.4. Architektury transformera**

### **2.4.1. Algorytm uwagi (ang. *attention*)**

### **2.4.2. Warianty mechanizmu uwagi**

#### **2.4.2.1. Self attention**

#### **2.4.2.2. Multi-headed attention**

#### **2.4.2.3. Flash attention**

### **2.4.3. Budowa transformera**

### **2.4.4. Modele tranformerowe**

#### **2.4.4.1. *Classic* transformer**

#### **2.4.4.2. SeqGAN**

#### **2.4.4.3. Mistral**

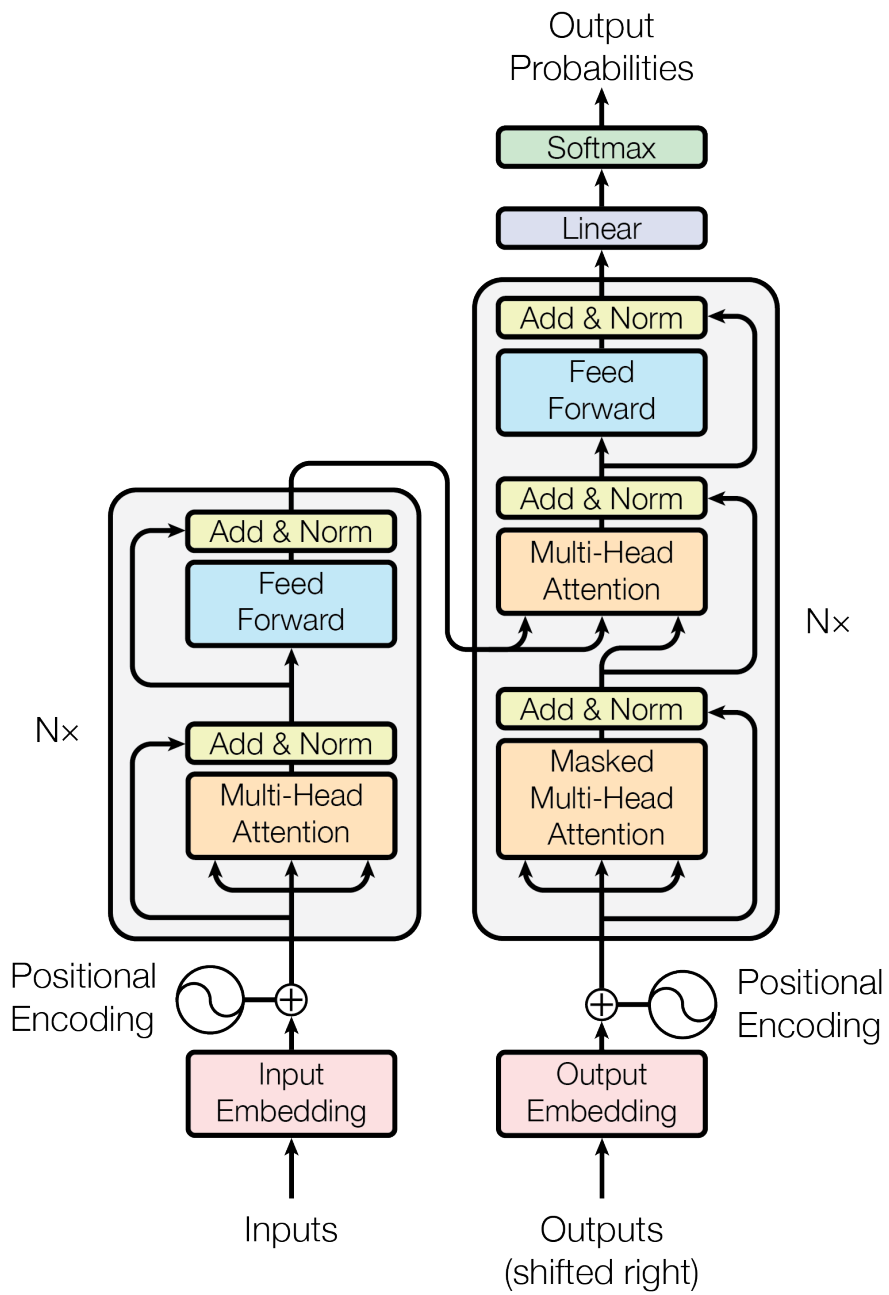
## **2.5. Architektura *state space***

### **2.5.1. Mamba**

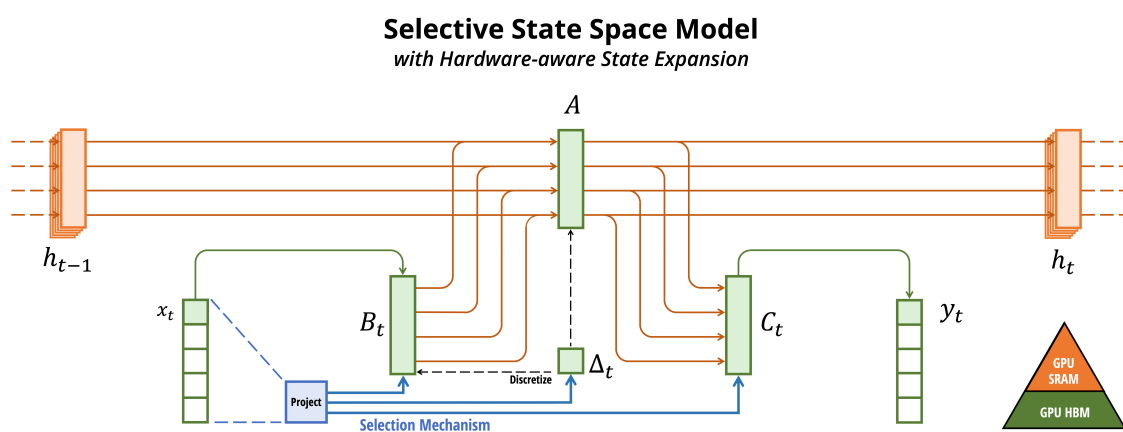
### **2.5.2. Tutaj się rozdrobnić trzeba**

**Uwaga 2.1.** Tytuł oraz strukturę rozdziału należy ustalić z opiekunem pracy.

Aktualny stan wiedzy, na dany temat, na podstawie dostępnej literatury naukowej oraz specjalistycznej.



Rysunek 2.1.: Schemat transformera.



Rysunek 2.2.: Schemat modelu Mamba.

## 3. Część badawcza

**Uwaga 3.1.** Tytuł oraz strukturę rozdziału należy ustalić z opiekunem pracy.

### 3.1. Opis *pipeline-u*

Tutaj zamierzam opisać w jaki sposób modele zostały stworzone, jakie biblioteki zostały użyte, jaki sprzęt został użyty podczas treningu

### 3.2. Porównanie architektur użytych modeli

### 3.3. Prezentacja otrzymanych wyników

### 3.4. Porwanie wyników

...



## 4. Zakończenie

**Uwaga 4.1.** Tytuł oraz strukturę rozdziału należy ustalić z opiekunem pracy.

1. Podsumowanie.
2. Możliwości dalszego rozwoju.
3. Potencjalne obszary zastosowania pracy.





# Dodatek A.

## Typowe elementy składowe pracy dyplomowej z informatyki

### A.1. Tabele

Uwaga A.1.

- Każda tabela powinna być opisana w treści pracy.
- Podpis ma być przed tabelą.

W tabeli [A.1](#) przedstawiono wyniki pomiarów.

Tabela A.1.: Pomiary zużycia energii elektrycznej.

L.p.	Wartość
1	12345,6789
	45,89
2	45,678901

Jeżeli tabela zawiera dużą liczbę wierszy i może nie zmieścić się na stronie — patrz tabela [A.2](#) — skorzystaj z pakietu *longtable* [\[6\]](#).

Tabela A.2.: Tabela, która zawiera dużą liczbę wierszy.

	1	2	3	4	5	6	7	8	
Student 1									

	1	2	3	4	5	6	7	8	
Student 2									
Student 3									
Student 4									
Student 5									
Student 6									
Student 7									
Student 8									
Student 9									

Tabele, w których występuje długi tekst, a co za tym idzie może się on nie zmieścić — musi zostać zawinięty, z pomocą przychodzi środowisko 'tabularx' [7] — patrz tabela A.3.

Tabela A.3.: Tabela zawierająca długi tekst.

Wpis wielokolumnowy!		TRZY	CZTERY
jeden	Szerokość tej kolumny zależy od szerokości tabeli.	trzy	Kolumna czwarta będzie zachowywać się w taki sam sposób jak druga kolumna o tej samej szerokości.

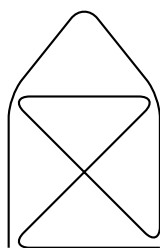
## A.2. Rysunki

### Uwaga A.2.

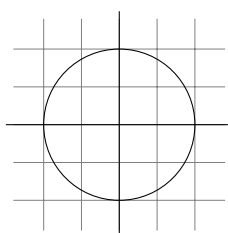
- Rysunki powinny być przerysowane samodzielnie albo używane tylko te, których twórcy zezwolili na ich rozpowszechnianie oraz kopiowanie, czyli np. rysunki objęte licencją Creative Commons.
- Każdy rysunek powinien być opisany w treści pracy.

### A.2.1. Wewnętrzne

Klasa *agh-wi*, automatycznie, dołącza pakiet *TikZ* [8] — dostarcza on komend pozwalających na tworzenie grafik. Przykładowe grafiki pokazano na rysunku A.1 oraz A.2.



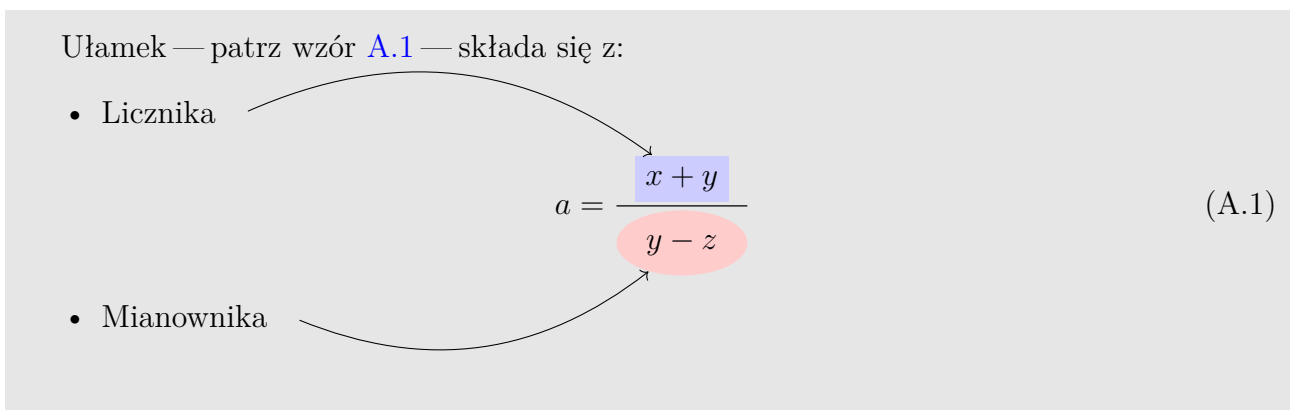
Rysunek A.1.: Prosty rysunek *TikZ*.



Rysunek A.2.: Bardziej złożony rysunek *TikZ*.

Oprócz rysunków eksponowanych możliwe jest tworzenie grafik będących  częścią zdania.

*TikZ* pozwala również na kreślenie po powierzchni strony, np. możemy narysować strzałki pomiędzy elementami strony.



### A.2.2. Zewnętrzne

Oczywiście możliwe jest również dołączanie rysunków zewnętrznych — pakiet *graphicx* [9] pozwala na wstawianie grafik zapisanych w plikach: '.png', '.jpg' oraz '.pdf'. Rysunek A.3 wstawiono przy użyciu tego pakietu.



Rysunek A.3.: Logo Wydziału Informatyki.

## A.3. Kody źródłowe

Najpopularniejszymi pakietami, które umożliwiają składanie kodów źródłowych programów, są:

**listings** [10] — kod źródłowy jest formatowany bezpośrednio przez  $\text{\LaTeX}$ -a — nie jest używany żaden, zewnętrzny, formater kodu.

Kod źródłowy A.1: Przykładowy kod źródłowy sformatowany za pomocą pakietu 'listings'.

```
1 /* Pierwszy program w C++ */
2
3 #include <iostream>
4
5 int main() {
6     std::cout << "Hello World!";
7     return 0;
8 }
```

**minted** [11] — formatuje kod źródłowy przy użyciu biblioteki języka Python o nazwie *Pygments* [12].

Kod źródłowy A.1.: Przykładowy listing sformatowany za pomocą pakietu 'minted'.

```
1 /* Pierwszy program w C++ */
2
3 #include <iostream>
4
5 int main() {
6     std::cout << "Hello World!";
7     return 0;
8 }
```

### Uwaga A.3.

- Podpis ma być przed kodem źródłowym.
- **Proszę używać tylko jednego z tych pakietów**; w przeciwnym razie otrzymasz taki efekt, jak w przykładowej pracy — obydwa listingi mają ten sam numer.

Kod źródłowy w C++ sformatowany przy użyciu pakietu *listings*, pokazano na listingu A.1; sformatowany przy użyciu pakietu *minted*, pokazano na listingu A.1.

## A.4. Algorytmy

Pakiet *algorithm2e* [13] to jeden z kilku, które pozwalają zapisywać algorytmy w formie pseudokodu — patrz algorytm 1.

**Uwaga A.4.** Podpis ma być przed algorytmem.

---

**Algorytm 1:** Disjoint decomposition.

---

```
input : A bitmap  $Im$  of size  $w \times l$ 
output: A partition of the bitmap

1 special treatment of the first line;
2 for  $i \leftarrow 2$  to  $l$  do
3   special treatment of the first element of line  $i$ ;
4   for  $j \leftarrow 2$  to  $w$  do
5      $\text{left} \leftarrow \text{FindCompress}(Im[i, j - 1]);$ 
6      $\text{up} \leftarrow \text{FindCompress}(Im[i - 1,]);$ 
7      $\text{this} \leftarrow \text{FindCompress}(Im[i, j]);$ 
8     if  $\text{left}$  compatible with this then //  $0(\text{left}, \text{this}) == 1$ 
9       if  $\text{left} < \text{this}$  then  $\text{Union}(\text{left}, \text{this});$ 
10      else  $\text{Union}(\text{this}, \text{left});$ 
11    end
12    if  $\text{up}$  compatible with this then //  $0(\text{up}, \text{this}) == 1$ 
13      if  $\text{up} < \text{this}$  then  $\text{Union}(\text{up}, \text{this});$ 
14      // this is put under up to keep tree as flat as possible
15      else  $\text{Union}(\text{this}, \text{up});$ 
16      // this linked to up
17    end
18  end
19 foreach element  $e$  of the line  $i$  do  $\text{FindCompress}(p);$ 
20 end
```

---

## A.5. Wzory

L<sup>A</sup>T<sub>E</sub>X bardzo dobrze sprawdza się w przypadku prac dyplomowych zawierających wzory matematyczne<sup>1</sup>.

---

<sup>1</sup>W przypadku złożonych wzorów warto zastosować pakiet *amsmath* [14].

### A.5.1. Przykłady

Wzór  $E = mc^2$  jest częścią zdania.

$$\left| \sum_{i=1}^n a_i b_i \right| \leq \left( \sum_{i=1}^n a_i^2 \right)^{1/2} \left( \sum_{i=1}^n b_i^2 \right)^{1/2} \quad (\text{A.2})$$

Wartości zmiennej opisano wzorem A.3.

$$x = \begin{cases} y & \text{dla } y > 0 \\ \frac{z}{y} & \text{dla } y \leq 0 \end{cases} \quad (\text{A.3})$$

Wzór A.4 to wzór wielowierszowy.

$$\begin{aligned} 2x^2 + 3(x-1)(x-2) &= 2x^2 + 3(x^2 - 3x + 2) \\ &= 2x^2 + 3x^2 - 9x + 6 \\ &= 5x^2 - 9x + 6 \end{aligned} \quad (\text{A.4})$$

**Uwaga A.5.** Należy używać tylko dwóch rodzajów wzorów:

1. „W linii”.
2. Eksponowane, numerowane.

## A.6. Twierdzenia i podobne struktury

Twierdzenie nr 1 opublikował, w roku 1691, francuski matematyk Michel Rolle.

**Twierdzenie 1 (Rolle’a)** *Jeśli dana funkcja  $f: \mathbb{R} \rightarrow \mathbb{R}$  jest:*

1. ciągła w przedziale  $[a, b]$
2. jest różniczkowalna w przedziale  $(a, b)$
3. na końcach przedziału  $[a, b]$  przyjmuje równe wartości:  $f(a) = f(b)$ ,

*to w przedziale  $(a, b)$  istnieje co najmniej jeden punkt  $c$  taki, że  $f'(c) = 0$ .*

Teraz coś z informatyki ...

**Definicja 1** *Bit to najmniejsza jednostka informacji w komputerze.*

**Definicja 2** *Bajtem nazywamy ciąg ośmiu bitów.*





# Uwagi Autora

- Aktualna wersja klasy jest dostępna pod adresem <https://github.com/polaksta/LaTeX/tree/master/agh-wi><sup>1</sup>.
- Skoro Twoja praca dyplomowa powstała w L<sup>A</sup>T<sub>E</sub>Xu, to zachęcam Cię również do przygotowania prezentacji (na obronę pracy magisterskiej) w tym języku. Najpopularniejszą klasą do tworzenia tego typu dokumentów jest *beamer* [15].
- Pod adresem <https://github.com/polaksta/LaTeX/tree/master/beamerthemeAGH><sup>2</sup> możesz znaleźć, stworzony przeze mnie, nasz uczelniany szablon dla prezentacji L<sup>A</sup>T<sub>E</sub>X Beamer.
- Treść wszystkich rozdziałów tej, przykładowej, pracy dyplomowej znajduje się w jednym pliku — **nie jest to polecane rozwiązanie**. W przypadku pisania własnej pracy warto umieścić zawartość każdego z rozdziałów w osobnych plikach, a następnie dołączać je do dokumentu głównego — patrz opis na stronie <https://www.dickimaw-books.com/latex/thesis/html/include.html>.
- Jeżeli pewne elementy mają być wyróżniane w **jednakowy** **sposób**, to proponuję nie używać bezpośredniego stylowania, tzn.

```
1 \colorbox{red!50}{jednakowy} \colorbox{red!50}{sposób}
```

ale zdefiniować własną komendę stylującą, np. `\alert`,

```
1 \newcommand{\alert}[1]{\colorbox{red!50}{#1}}
```

a następnie użyć jej w dokumencie.

```
1 \alert{jednakowy} \alert{sposób}
```

Dzięki temu, jeżeli będziesz chciał / chciała zmienić sposób stylowania tych elementów, np. niebieskie tło zamiast czerwonego, to wystarczy zmodyfikować, tylko, definicję komendy, zamiast zastępować, w tekście pracy dyplomowej, wybrane (niekoniecznie wszystkie!) wystąpienia tekstu `red`, tekstem `blue`.

<sup>1</sup>W przypadku Overleaf-a jest ona pod adresem <https://www.overleaf.com/read/fnvcvqjyrbyw#5ac622>

<sup>2</sup>W przypadku Overleaf-a jest on pod adresem <https://www.overleaf.com/read/fkjdtbnbrfhj#9c6184>

Stanisław Polak

# Bibliografia

- [1] Ashish Vaswani i in. *Attention Is All You Need*. 2023. arXiv: [1706.03762](https://arxiv.org/abs/1706.03762) [cs.CL].
- [2] Nathan Fradet i in. „MidiTok: A Python package for MIDI file tokenization”. W: *Extended Abstracts for the Late-Breaking Demo Session of the 22nd International Society for Music Information Retrieval Conference*. 2021. URL: <https://archives.ismir.net/ismir2021/latebreaking/000005.pdf>.
- [3] Darrell Conklin. *Bach Chorales*. UCI Machine Learning Repository. DOI: <https://doi.org/10.24432/C5G>
- [4] Curtis Hawthorne i in. „Enabling Factorized Piano Music Modeling and Generation with the MAESTRO Dataset”. W: *International Conference on Learning Representations*. 2019. URL: <https://openreview.net/forum?id=r1lYRjC9F7>.
- [5] Thierry Bertin-Mahieux i in. „The Million Song Dataset”. W: *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011)*. 2011.
- [6] *The longtable package*. URL: <http://mirrors.ctan.org/macros/latex/required/tools/longtable.pdf>.
- [7] *The tabularx package*. URL: <http://mirrors.ctan.org/macros/latex/required/tools/tabularx.pdf>.
- [8] *The TikZ and PGF Packages*. URL: <http://mirrors.ctan.org/graphics/pgf/base/doc/pgfmanual.pdf>.
- [9] *Packages in the ‘graphics’ bundle*. URL: <http://mirrors.ctan.org/macros/latex/required/graphics/grfguide.pdf>.
- [10] *The Listings Package*. URL: <http://mirrors.ctan.org/macros/latex/contrib/listings/listings.pdf>.
- [11] *The minted package: Highlighted source code in L<sup>A</sup>T<sub>E</sub>X*. URL: <http://mirrors.ctan.org/macros/latex/contrib/minted/minted.pdf>.
- [12] *Strona WWW biblioteki „Pygments”*. URL: <https://pygments.org/>.
- [13] *algorithm2e.sty — package for algorithms*. URL: <http://mirrors.ctan.org/macros/latex/contrib/algorithm2e/doc/algorithm2e.pdf>.
- [14] *User’s Guide for the amsmath Package*. URL: [http://mirrors.ctan.org/macros/latex/required/amsmath/amsldoc.pdf](http://mirrors.ctan.org/macros/latex/required/amsmath/amslatex/amsldoc.pdf).

- [15] *The beamer class*. URL: <http://mirrors.ctan.org/macros/latex/contrib/beamer/doc/beameruserguide.pdf>.