Politechnika Warszawska

WYDZIAŁ ELEKTRONIKI
I TECHNIK INFORMACYJNYCH

Instytut Informatyki

Sprawozdanie z realizacji projektu

Zaimplementowanie i eksperymentalna ewaluacja dwóch podejść do sumaryzacji tekstu: ekstrakcyjnej i abstrakcyjnej

Tobiasz Kownacki
Numer albumu 331391

WARSZAWA STYCZEŃ 2026

Spis treści

1. Wstęp

Celem projektu jest zbadanie i porównanie dwóch podejść do problemu sumaryzacji tekstu: ekstrakcyjnego oraz abstrakcyjnego. Problem sumaryzacji polega na przetworzeniu dłuższego tekstu w wersję skróconą, która zachowuje najważniejsze informacje i sens oryginału. W podejściu ekstrakcyjnym model wybiera kluczowe zdania bezpośrednio z tekstu źródłowego i zamieszcza je w podsumowaniu. Z kolei podejście abstrakcyjne polega na zrozumieniu treści przez model i wygenerowaniu nowej treści, która podsumowuje tekst wejściowy, często używając innego słownictwa niż w oryginale.

Jako zbiór danych wykorzystano zbiór **EdinburghNLP/xsum**, który zawiera artykuły BBC wraz z ich **jednozdaniowymi** podsumowaniami. Przykładowy wiersz ze zbioru danych:

Dokument: The Bath-born player, 28, has made 36 appearances for the Dragons since joining from Wasps in 2015. He is in his second season and signed a contract extension in December 2016.
Dragons forwards coach Ceri Jones said: "It's a big blow. Eddie has been excellent all year for us, he has really stepped up to the mark and will be a big loss."However, Jones says Jackson's misfortune can be a chance for others to thrive.
"We are very fortunate to have the likes of Ollie Griffiths, Harrison Keddie, James Thomas who can come into the back-row,śaid Jackson. "Harri has shown glimpses of what he can do all season and there's definitely a player there, so this is an opportunity."
Dragons travel to Munster in the Pro12 on Friday.

Streszczenie: *Newport Gwent Dragons number eight Ed Jackson has undergone shoulder surgery and faces a spell on the sidelines.*

Motywacją do realizacji projektu jest rosnąca potrzeba szybkiego wyciągania najważniejszych informacji z długich tekstów. Przykładem tego jest wprowadzenie przez firmę Alphabet funkcji sumaryzacji wiadomości e-mail. Celem jest porównanie starszych rozwiązań, takich jak `encoder-only` BERT, z nowszymi, ale lekkimi modelami `decoder-only` SmolLM3-3B.

W ramach wkładu własnego opracowano środowisko eksperymentalne pozwalające na porównanie dwóch podejść do sumaryzacji tekstu. Najważniejsze elementy to:

- **Implementacja procesu dostrajania modelu:** Zrealizowano procedurę dostrajania modelu `decoder-only` za pomocą techniki LoRA.
- **Prompt Engineering:** Zaprojektowano schemat promptów wykorzystywanych zarówno w procesie treningu, jak i podczas ewaluacji w trybach `zero-shot` oraz `few-shot learning`. Celem szablonów było uzyskanie zwięzłych, edytorskich podsumowań.

- **Procedura wyświetlania metryk:** Przygotowano procedurę do obliczania i wyświetlania metryk takich jak ROUGE oraz BERTScore.
- **Inferencja w trybie wsadowym:** Przygotowano funkcję umożliwiającą wykonywanie inferencji w trybie wsadowym o podanej przez użytkownika wielkości.

2. Przegląd literatury

2.1. Specyfika zbioru XSum

W literaturze przedmiotu problem sumaryzacji rozpatrywany jest w dwóch paradygmatach: ekstrakcyjnym oraz abstrakcyjnym. Kluczowym wkładem w rozwój podejścia abstrakcyjnego była praca [1], wprowadzająca zbiór danych XSum (*Extreme Summarization*). Zbiór ten składa się z artykułów BBC wraz z ich jednozdaniowymi podsumowaniami. Cechą wyróżniającą XSum jest wysoki stopień abstrakcji referencyjnych streszczeń. Autorzy wykazali, że wcześniejsze popularne zbiory danych cechowały się znacznym pokryciem leksykalnym, co w praktyce faworyzowało prostsze metody ekstrakcyjne, podczas gdy XSum wymaga od modelu rzeczywistego generowania sparafrazowanej treści.

2.2. Wykorzystanie modeli typu `encoder-only` BERT

Opracowany na bazie pracy [2] model BERT wykorzystuje architekturę Transformer [3] w wariancie `encoder-only`. W procesie sumaryzacji ekstrakcyjnej służy on do tworzenia embeddingów zdań. Następnie, za pomocą algorytmów klastrowania, wybierane są zdania kluczowe, najlepiej oddające sens tekstu. W projekcie wykorzystano implementację tego mechanizmu dostępną w bibliotece `bert-extractive-summarizer`, którą opracowano w ramach pracy [4].

2.3. Efektywne obliczeniowo dostrajanie modeli

Współczesna sumaryzacja abstrakcyjna bazuje między innymi na modelach *decoder-only*. Jednakże dostrajanie wszystkich parametrów tych sieci wymaga ogromnych zasobów sprzętowych. Rozwiązaniem tego problemu jest technika LoRA (*Low-Rank Adaptation*), opisana w [5]. Pozwala ona na zamrożenie głównych wag modelu i trenowanie wyłącznie niewielkich macierzy adaptacyjnych. Dzięki temu możliwa jest znaczna redukcja wymagań obliczeniowych bez utraty skuteczności modelu.

2.4. Metryki ewaluacji jakości

Ocena jakości streszczeń jest trudnym zagadnieniem. Standardowym rozwiązaniem jest metryka ROUGE [6], która sprawdza pokrycie n-gramów między tekstem a referencją. Jednak autorzy pracy [7] wskazują, że podejście leksykalne słabo radzi sobie z oceną semantyki i parafraz. Zaproponowali oni metrykę BERTScore. Wykorzystuje ona embeddingi oraz podobieństwo cosinusowe embeddingów, co znacznie lepiej sprawdza się przy sumaryzacji abstrakcyjnej (takiej jak w przypadku zbioru XSum).

3. Opis rozwiązania

3.1. Wykorzystane narzędzia i biblioteki

Implementacja rozwiązania została zrealizowana z wykorzystaniem bibliotek udostępnianych przez Hugging Face. Do kluczowych narzędzi wykorzystanych w projekcie należą:

- **Transformers**: Podstawowa biblioteka służąca do obsługi modeli, tokenizacji oraz procesu inferencji.
- **PEFT (Parameter-Efficient Fine-Tuning)**: Biblioteka umożliwiająca efektywne dostrajanie dużych modeli językowych przy użyciu metod takich jak LoRA.
- **BitsAndBytes**: Wykorzystana do 8-bitowej kwantyzacji modelu generatywnego, co pozwoliło na redukcję zapotrzebowania na pamięć VRAM.
- **TRL (Transformer Reinforcement Learning)**: Wykorzystano klasę `SFTTrainer`, która jest zoptymalizowana pod kątem nadzorowanego dostrajania (Supervised Fine-Tuning) modeli językowych.
- **Bert-extractive-summarizer**: Narzędzie implementujące algorytm sumaryzacji ekstrakcyjnej oparty na klastrowaniu embeddingów generowanych przez model BERT.
- **Evaluate, ROUGE_score, BERT_score**: Zestaw bibliotek służących do obliczania metryk jakości generowanych tekstów.
- **Datasets**: Biblioteka do pobierania i zarządzania zbiorem danych XSum.

3.2. Charakterystyka i podział zbioru danych

W projekcie wykorzystano zbiór danych **EdinburghNLP/xsum**. Zbiór ten zawiera artykuły informacyjne BBC oraz ich profesjonalne, jednozdaniowe streszczenia.

W skład pełnego zbioru danych wchodzi:

- Zbiór treningowy: 204 000 próbek
- Zbiór walidacyjny: 11 300 próbek
- Zbiór testowy: 11 300 próbek

Ze względu na ograniczenia sprzętowe, do eksperymentów wybrano losowe podzbiory danych:

- Podzbiór treningowy: 5000 losowych próbek wybranych ze zbioru treningowego.
- Podzbiór walidacyjny: 500 próbek służących do obliczania metryk w trakcie dostrajania, wylosowanych z zbioru walidacyjnego.
- Zbiór testowy: Do ewaluacji różnych podejść wykorzystano wylosowany podzbiór testowy o liczebności 1000 próbek.

3.3. Podejście ekstrakcyjne (BERT)

Jako model bazowy wykorzystano `bert-large-uncased`. Do realizacji części ekstrakcyjnej użyto biblioteki `bert-extractive-summarizer` [4]. Zapewnia ona wygodną warstwę abstrakcji nad modelem BERT. Z perspektywy użytkownika operacja sprowadza się do przekazania tekstu źródłowego do instancji klasy `Summarizer`, która w odpowiedzi zwraca gotowe streszczenie, składające się z wyselekcjonowanych, najbardziej reprezentatywnych zdań. Eksperymenty przeprowadzono w dwóch konfiguracjach:

1. **Standardowa:** Wykorzystująca domyślne osadzenia z przedostatniej warstwy modelu.
2. **Rozszerzona (Hidden Layer Concatenation):** Wykorzystująca konkatenację dwóch ostatnich warstw ukrytych w celu uzyskania bogatszej reprezentacji semantycznej zdań.

3.4. Podejście abstrakcyjne (SmolLM)

Do generowania streszczeń wykorzystano model `HuggingFaceTB/SmolLM3-3B`. Model został załadowany w precyzji `bfloat16` z wykorzystaniem 8-bitowej kwantyzacji.

W ramach dostrajania modelu zastosowano technikę **LoRA** z następującą konfiguracją:

- `rank`: 8
- `alpha`: 16
- `dropout`: 0.05
- `target_modules`: all-linear

3.4.1. Przebieg procesu uczenia

Do realizacji procesu nadzorowanego dostrajania (SFT) wykorzystano bibliotekę `trl`. Jej zastosowanie znacząco uprościło potok przetwarzania danych, redukując wymagania wstępne jedynie do przygotowania zbioru w ustrukturyzowanym formacie typu `prompt-completion`. Istotnym elementem konfiguracji treningu było zastosowanie mechanizmu maskowania instrukcji (`completion-only loss`). Dzięki temu funkcja straty obliczana jest wyłącznie na podstawie sekcji `completion` (generowanego streszczenia), ignorując treść promptu systemowego oraz instrukcji użytkownika.

Model był trenowany przez 1 epokę na wyselekcjonowanym podzbiorze 5000 próbek. Zastosowano optymalizator `paged_adamw_8bit` w celu dalszej optymalizacji zużycia pamięci.

Parametry procesu treningowego:

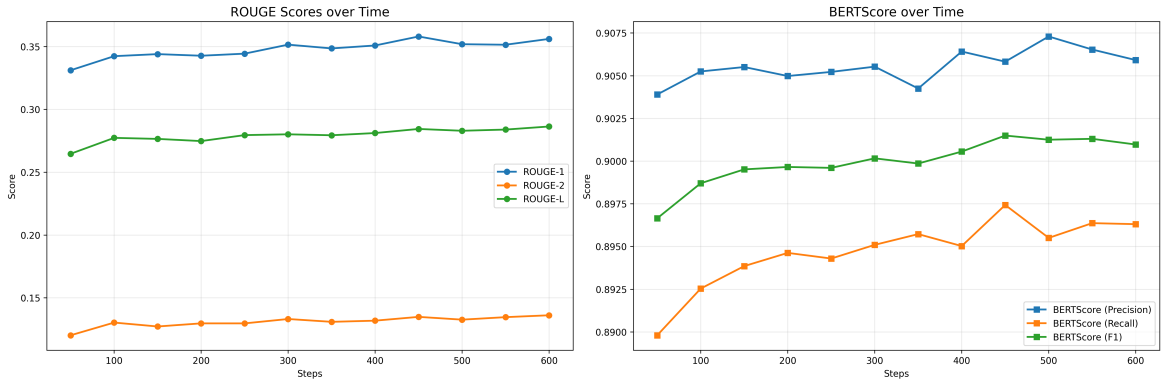
- Learning rate: 10^{-4}
- Batch size: 1
- Gradient accumulation steps: 8

Efektywny rozmiar wsadu wynosił 8, co przy 5000 próbkach przełożyło się na około 625 kroków optymalizacyjnych. W trakcie treningu, co 50 kroków, uruchamiana była proce-

dura ewaluacyjna wykorzystująca zaimplementowany callback `GenerativeMetricsCallback`, który generował próbne streszczenia i obliczał metryki jakości na zbiorze walidacyjnym.

Rysunek 3.1. Wykres funkcji straty na zbiorze treningowym oraz walidacyjnym w trakcie procesu dostrajania modelu SmolLM3-3B.

Wykres 3.1 obrazuje systematyczny spadek wartości funkcji straty, co wskazuje na brak oznak przeuczenia. Mimo odnotowanego chwilowego wzrostu funkcji straty na zbiorze treningowym, w dalszych krokach optymalizacji krzywa powraca do trendu spadkowego. Wartości funkcji straty dla zbioru walidacyjnego pozostają stabilne i nie wykazują istotnych odchyleń.



Rysunek 3.2. Ewolucja wskaźników jakości (ROUGE, BERTScore) dla zbioru walidacyjnego podczas dostrajania modelu SmolLM3-3B.

Wykres 3.2 wykazuje powolny wzrost wartości wszystkich monitorowanych metryk w miarę postępu liczby kroków treningowych. Szczególnie istotny jest stabilny trend wzrostowy metryki BERTScore, co potwierdza, że model z każdym krokiem coraz lepiej odwzorowuje spójność semantyczną referencyjnych streszczeń. Z kolei rosnące wartości

metryki ROUGE wskazują na zwiększającą się zgodność leksykalną generowanych tekstów z referencjami, co świadczy o adaptacji modelu do stylu językowego zbioru XSum.

3.4.2. Szablon konwersacji i Prompt Engineering

W celu zapewnienia jak najlepszych rezultatów, dane wejściowe sformatowano zgodnie z `Chat Template`. Wykorzystano do tego metodę dostępną w tokenizatorze, która strukturyzuje surowy tekst, przypisując fragmentom odpowiednie role: `system`, `user` oraz `assistant`.

Szczególne zastosowanie szablon ten znalazł w metodzie *Few-shot prompting*. W tym scenariuszu kontekst wejściowy został wzbogacony o serię przykładowych par wiadomości pochodzących ze zbioru treningowego. Dla każdego przykładu utworzono sekwencję, w której rola `user` prezentowała treść artykułu, a rola `assistant` dostarczała odpowiadające mu wzorcowe streszczenie. Taka konstrukcja pozwoliła modelowi na poznanie oczekiwanego wzorca odpowiedzi.

Ważnym elementem jest komunikat systemowy, który znajduje się na samym początku każdego prompta. Został on zdefiniowany tak, aby narzucić modelowi rolę profesjonalnego redaktora oraz wymusić pisanie jedynie jednego zdania, tak jak to jest w zbiorze XSum. Treść promptu systemowego brzmi następująco:

"You are a professional BBC news editor. Your task is to write an extremely concise summary of the article provided by the user. The summary must consist of EXACTLY ONE sentence."

3.4.3. Inferencja w trybie wsadowym

W celu optymalizacji czasu generowania streszczeń dla zbioru testowego, zaimplementowano funkcję `generate_summaries` obsługującą przetwarzanie wsadowe. Kluczowe elementy tej implementacji to:

- **Data Collator:** Zastosowano `DataCollatorWithPadding`, który dynamicznie dopełnia sekwencje wewnątrz każdego wsadu do długości najdłuższego elementu.
- **Obsługa wsadu:** Model generuje odpowiedzi dla całej partii danych jednocześnie, co znacząco przyspiesza proces ewaluacji w porównaniu do pętli iterującej po pojedynczych rekordach.

4. Wyniki ewaluacji eksperymentów

4.1. Opis metryk wykorzystanych do oceny jakości

W celu obiektywnej oceny jakości generowanych streszczeń wykorzystano dwie metryki:

- **ROUGE:** Rodzina metryk oparta na leksykalnym pokryciu n-gramów między wygenerowanym tekstem a referencją.
  - — **ROUGE-1:** Mierzy pokrycie pojedynczych słów (unigramów).
  - — **ROUGE-2:** Mierzy pokrycie par słów (bigramów).
  - — **ROUGE-L:** Oparta na najdłuższej wspólnej podsekwencji (LCS).
- **BERTScore:** Metryka wykorzystująca embeddingi z modeli językowych. W przeciwieństwie do ROUGE, BERTScore nie wymaga dokładnego dopasowania słów, lecz mierzy podobieństwo semantyczne za pomocą cosinusa kąta między embeddingami. Wyniki prezentowane są za pomocą trzech wskaźników:
  - — **Precision (Precyzja):** Określa, czy to, co wygenerował model, jest zgodne z referencją.
  - — **Recall (Czułość):** Określa, ile istotnych informacji z tekstu wzorcowego udało się zawrzeć w wygenerowanym streszczeniu.
  - — **F1:** Średnia harmoniczna precyzji i czułości, stanowiąca ogólną, uśrednioną ocenę jakości.

4.2. Ewaluacja podejścia ekstrakcyjnego

W ramach podejścia ekstrakcyjnego przetestowano model `bert-large-uncased` w dwóch konfiguracjach: standardowej oraz rozszerzonej. Wyniki przedstawiono w Tabeli 4.1.

Tabela 4.1. Porównanie wyników modeli ekstrakcyjnych na zbiorze testowym.

Metryka	BERT (Standard)	BERT (Hidden Concat)
ROUGE-1	0.1694	0.1691
ROUGE-2	0.0196	0.0208
ROUGE-L	0.1229	0.1233
BERTScore (Precision)	0.8506	0.8514
BERTScore (Recall)	0.8594	0.8591
BERTScore (F1)	0.8548	0.8551

4.2.1. Omówienie wyników ekstrakcyjnych

Modele ekstrakcyjne osiągnęły relatywnie niskie wyniki metryk ROUGE. Jest to rezultat zgodny z oczekiwaniami dla zbioru XSum, który charakteryzuje się wysoce abstrakcyjnymi,

jednozdaniowymi streszczeniami. Metody ekstrakcyjne nie są w stanie odwzorować syntetycznego stylu referencji BBC.

Wprowadzenie konkatenacji warstw ukrytych nie przyniosło istotnych korzyści. Odnotowane różnice w metrykach są śladowe i niespójne.

4.3. Ewaluacja podejścia abstrakcyjnego

Dla modelu `decoder-only` SmolLM3-3B przeprowadzono ewaluację w trzech scenariuszach: *Zero-shot, Five-shot* oraz po pełnym dostrojeniu metodą LoRA. Zestawienie wyników prezentuje Tabela 4.2.

Tabela 4.2. Ewolucja jakości modelu SmolLM3-3B w kolejnych eksperymentach.

Metryka	**Zero-shot**	**Five-shot**	**Fine-tuned (LoRA)**
ROUGE-1	0.2359	0.2917	**0.3615**
ROUGE-2	0.0559	0.0854	**0.1416**
ROUGE-L	0.1654	0.2158	**0.2891**
BERTScore (Precision)	0.8552	0.8820	**0.9088**
BERTScore (Recall)	0.8870	0.8904	**0.8980**
BERTScore (F1)	0.8707	0.8861	**0.9032**

4.3.1. Omówienie wyników abstrakcyjnych

Analiza wyników wskazuje na znaczącą przewagę podejścia abstrakcyjnego nad ekstrakcyjnym. Nawet podstawowy model w trybie **Zero-shot** osiągnął wynik ROUGE-1 na poziomie 0.2359, deklasując model BERT.

Zastosowanie techniki **Few-shot prompting** pozwoliło modelowi lepiej zrozumieć format wyjściowy, co przełożyło się na wzrost wszystkich metryk, w tym skok ROUGE-1 do 0.2917.

Najlepsze rezultaty uzyskano po procesie **Fine-tuningu**. Model osiągnął ROUGE-1 na poziomie 0.3615 oraz BERTScore F1 przekraczający 0.90.

5. Podsumowanie

Przeprowadzony eksperyment, porównujący ekstrakcyjny model BERT oraz generatywny model SmolLM3-3B na zbiorze XSum, wykazał jednoznaczną przewagę podejścia abstrakcyjnego. Metody ekstrakcyjne okazały się nieskuteczne ze względu na specyfikę zbioru, który wymaga parafrazowania, a nie tylko selekcji zdań.

Tabela 5.1. Porównanie wyników modelu SmolLM3-3B z modelami SOTA (LLaMA-3, Gemma, FLAN-T5) na zbiorze XSum. Wyniki dla modeli referencyjnych pochodzą z pracy [8].

Model	**Parametry**	**ROUGE-1**	**ROUGE-2**	**ROUGE-L**
SmolLM3-3B	3B	0.36	0.14	0.29
LLaMA-3-8B	8B	0.37	0.15	0.29
Gemma-7B	7B	**0.39**	**0.18**	**0.32**
FLAN-T5	11B	0.35	0.13	0.27
BART	0.14B	0.27	0.07	0.21

W celu lepszej interpretacji rezultatów, zestawiono wyniki dostrojonego modelu SmolLM3-3B z wynikami innych również dostrojonych modeli językowych, które zostały przedstawione w ramach pracy [8].

Model SmolLM3-3B osiągnął wynik ROUGE-1 na poziomie 0.36, co plasuje go niemal na równi z modelem LLaMA-3-8B, który osiągnął wynik 0.37. Jest to rezultat bardzo dobry, biorąc pod uwagę, że LLaMA-3 posiada blisko trzykrotnie więcej parametrów. Co więcej, w metryce ROUGE-L oba modele osiągnęły identyczny wynik.

Uwaga do ewaluacji: W zestawieniu pominięto metrykę BERTScore, ponieważ w pracy referencyjnej [8] zastosowano jej wariant znormalizowany, co uniemożliwia bezpośrednie porównanie z wynikami uzyskanymi w tym projekcie.

Propozycje dalszych prac:

- **Analiza strategii dekodowania:** Zbadanie wpływu parametrów sterujących stochastycznością procesu generacji (takich jak temperatura, *Top-k* czy *Top-p* sampling) na wartości metryk ewaluacyjnych.
- **Zwiększenie danych treningowych:** Wykorzystanie pełnego zbioru XSum (ok. 200 tys. próbek) zamiast obecnego podzbioru (5 tys.) w celu poprawy generalizacji.
- **Optymalizacja hiperparametrów:** Przeprowadzenie badań dla parametrów LoRA (r, α).
- **Większe modele:** Przetestowanie większych modeli w celu uzyskania lepszych zdolności wnioskowania.
- **Inne zbiory danych:** Przeprowadzić badania na innych zbiorach danych, które bardziej faworyzują sumaryzację ekstrakcyjną.

Bibliografia

[1] S. Narayan, S. B. Cohen i M. Lapata, „Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization", *arXiv preprint arXiv:1808.08745*, 2018.

[2] J. Devlin, M.-W. Chang, K. Lee i K. Toutanova, „Bert: Pre-training of deep bidirectional transformers for language understanding", w *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, 2019.

[3] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser i I. Polosukhin, „Attention is all you need", *Advances in neural information processing systems*, t. 30, 2017.

[4] D. Miller, „Leveraging BERT for extractive text summarization on lectures", *arXiv preprint arXiv:1906.04165*, 2019.

[5] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen i in., „Lora: Low-rank adaptation of large language models.", *ICLR*, t. 1, nr. 2, s. 3, 2022.

[6] C.-Y. Lin, „Rouge: A package for automatic evaluation of summaries", w *Text summarization branches out*, 2004, s. 74–81.

[7] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger i Y. Artzi, „Bertscore: Evaluating text generation with bert", *arXiv preprint arXiv:1904.09675*, 2019.

[8] T. Rehman, S. Ghosh, K. Das, S. Bhattacharjee, D. K. Sanyal i S. Chattopadhyay, „Evaluating LLMs and Pre-trained Models for Text Summarization Across Diverse Datasets", *arXiv preprint arXiv:2502.19339*, 2025.