Richard Staszkiewicz
Miłosz Łopatto

1. Temat projektu
2. Cel projektu
3. Czym jest RAG?
    1. Jak działa RAG?
4. Jak można ewaluować działanie systemu RAG?
    1. Wybrana metoda ewaluacji
5. Przykładowe modele embeddingowe
    1. Ostatecznie wybrany model - **Silver Retriever Base (v1.1)**
6. Wyniki eksperymentów
7. Wnioski
8. Aplikacja
    1. Technologia
9. Źródła

Dokumentacja końcowa do projektu z przedmiotu NLP

Temat projektu

> ⓘ **Temat projektu**
>
> **RAG (Retrieval Augmented Generation) oraz wpływ różnych embeddingów na działanie tej metody.**

Projekt ten bada wpływ różnych embeddingów na działanie metody Retrieval Augmented Generation (RAG).

Cel projektu

Początkowo chcieliśmy porównać kilka bardziej znanych modeli generujących embeddingi. Jednakże po odkryciu benchmarku MTEB (Massive Text Embedding Benchmark) [9] stwierdziliśmy, że lepszym pomysłem byłoby wybranie modelu, który nie został jeszcze uwzględniony w tym benchmarku. Dzięki temu zamiast liczyć coś co zostało już wcześniej policzone w jakimś stopniu przyczynimy się do rozwoju tego

otwarto-źródłowego projektu, ponieważ uzyskane przez nas wyniki wgramy do wspomnianej wyżej tabeli wyników.

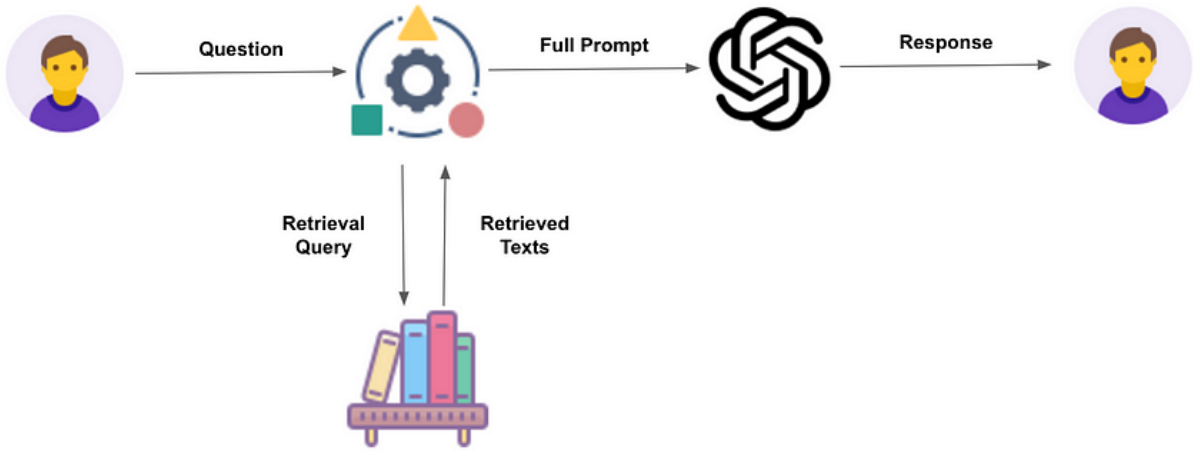
Zakres projektu:

1. Analiza podejścia Retrieval Augmented Generation.
2. Zdefiniowanie metod ewaluacji skuteczności działania metody RAG.
3. Zdefiniowanie testowanych embeddingów.
4. Przeprowadzenie testów.
5. Analiza wyników i wnioski.
6. Utworzenie przykładowej aplikacji prezentującej działanie testowanych systemów.

Czym jest RAG?

Jest to skrót od "Retrieval-Augmented Generation", czyli w tłumaczeniu na polski "Generacja z użyciem mechanizmu odzyskiwania". RAG jest często wykorzystywany w zadaniach związanych z przetwarzaniem języka naturalnego, takich jak systemy odpowiadające na pytania, generatory tekstu czy też w systemach dialogowych. Wdrażanie takiego systemu może przyczynić się do lepszej skuteczności i dostarczenia bardziej trafnych oraz aktualnych odpowiedzi w kontekście danego zadania. [7, 8]

Jak działa RAG?



RAG łączy procesy odzyskiwania informacji (Retrieval) i generacji tekstu (Generation). Działanie RAG można podzielić na kilka kluczowych etapów:

0. **Utworzenie bazy wiedzy**:
    - Podstawą jest utworzenie bazy wiedzy do przeszukiwania w kontekście zapytań użytkownika. W naszym projekcie jej realizacja będzie miała postać

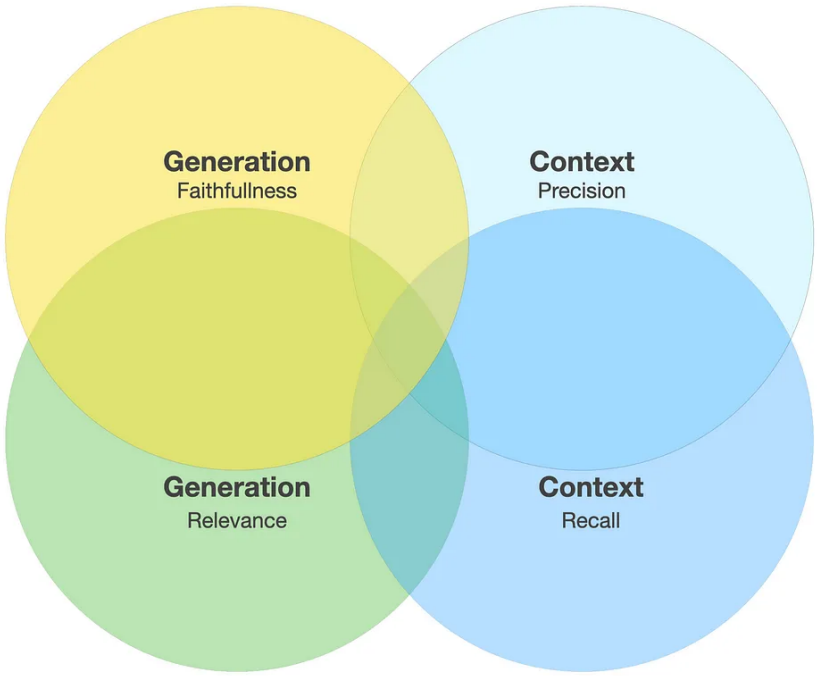
bazy wektorowej zawierającej embeddingi pochodzące z dokumentów PDF zapewnionych przez użytkownika.

1. **Zapytanie Użytkownika**:
   - Zapytanie użytkownika jest przekazywane do systemu RAG. Zapytanie to może być pytaniem, prośbą o wygenerowanie tekstu lub jakimkolwiek innym promptem.
2. **Odzyskiwanie Informacji (Retrieval)**:
   - Zapytanie użytkownika zostaje zembeddowane do bazy wektorowej. W tym kroku należy pamiętać aby wykorzystać ten sam model, którego wcześniej użyliśmy do utworzenia wektorowej bazy danych.
   - System pobiera *k* najbardziej podobnych elementów z bazy wektorowej. Do znalezienia najbardziej podobnych elementów pod względem semantycznym wykorzystywana jest najczęściej odległość cosinusowa (cosine similarity).
3. **Generacja Tekstu (Generation)**:
   - Pobrane *k* najbardziej podobnych elementów jest łączone z zapytaniem użytkownika.
   - Tak przygotowane zapytanie użytkownika wraz z pobranym kontekstem jest wykorzystywane przez model generacji tekstu, aby stworzyć nową, spójną odpowiedź.
   - Ta odpowiedź jest następnie przekazywana użytkownikowi jako końcowy produkt działania systemu RAG.

Jak można ewaluować działanie systemu RAG?

Metodę RAG można ewaluować względem następujących aspektów:

- ewaluacja odzyskiwania informacji (tylko Retrieval)
  - Context Precision (dokładność kontekstu)
  - Context Recall (pełność kontekstu)
- ewaluacja generacji tekstu (tylko Generation)
  - Generation Faithfullness (wierność generacji)
  - Generation Relevance (znaczenie generacji)



Chcąc ewaluować RAG całościowo, możemy patrzeć na średnią harmoniczną powyższych czterech metryk. W taki sposób działa metoda **ragas score** [4]. W tym projekcie skupimy się natomiast przede wszystkim na ewaluacji odzyskiwania informacji (Retrieval). Oczywiście zarówno aspekt Retrieval jak i Generation są niezwykle istotne, natomiast w naszym przypadku, tj. w przypadku badania wpływu różnych embeddingów, szczególnie interesującym będzie właśnie aspekt Retrieval.

Zarówno Generation Faithfullness (czyli mierzenie halucynacji) jak i Generation Relevance (czyli mierzenie jak dobrze wygenerowana odpowiedź odnosi się do zadanego pytania) zależą przede wszystkim od modelu wykorzystanego do wygenerowania ostatecznej odpowiedzi. Natomiast model tworzący embeddingi będzie miał bezpośredni wpływ na aspekt odzyskiwania informacji (Retrieval). W związku z powyższym wybierając metodę ewaluacji skupiliśmy się tylko i wyłącznie na aspekcie Retrieval.

Wybrana metoda ewaluacji

Dokonamy badania miarą główną n-DCG@10 z miarami pomocniczymi MRR@k oraz MAP@k na zadaniu Retrieval projektu MTEB [2] na w sumie 15 benchmarkach po raz pierwszy skompilowanych na projekcie BEIR [3]. Do ewaluacji wykorzystywane jest narzędzie mteb, do podglądu rezultatów oraz oceny manualnej używany jest system RAG z wektorową bazą danych Chroma zbudowany w technologii LangChain z interfejsem w Streamlit.

Przykładowe modele embeddingowe

- **BERT (Bidirectional Encoder Representations from Transformers)** [10]:
  - BERT jest jednym z najbardziej znanych modeli do generowania embeddingów. Jest on szczególnie efektywny w rozumieniu kontekstu, co może być kluczowe w metodzie RAG.
- **GPT-2/GPT-3 (Generative Pretrained Transformer 2/3)** [11]:
  - Te modele, znane z generowania spójnego i kontekstowego tekstu, mogą być użyteczne do badania, jak generatywne embeddingi wpływają na proces generacji w RAG.
- **RoBERTa (A Robustly Optimized BERT Pretraining Approach)** [12]:
  - RoBERTa jest wariantem BERT, który został zoptymalizowany pod kątem większej dokładności. Może to zapewnić interesujące porównanie z tradycyjnym BERT-em.
- **DistilBERT (Distilled Version of BERT)** [13]:
  - Jest to uproszczona i bardziej efektywna wersja BERT-a pod względem obliczeniowym, co może być istotne w zastosowaniach, gdzie szybkość jest kluczowa.
- **T5 (Text-To-Text Transfer Transformer)** [14]:
  - T5 jest wszechstronnym modelem, który traktuje każde zadanie NLP jako zadanie konwersji tekstu na tekst, co może być interesujące w kontekście RAG.
- **ALBERT (A Lite BERT)** [15]:
  - ALBERT to uproszczona wersja BERT-a, która została opracowana przez Google w celu radzenia sobie z problemami wynikającymi z dużych rozmiarów modelu. Używa ona technik redukcji parametrów, co może być korzystne w kontekście efektywności obliczeniowej i skalowalności systemu RAG.

Wyżej wymienione przykładowe modele w większości zostały już uwzględnione w benchmarku MTEB. W związku z tym na platformie HuggingFace postaramy się znaleźć mniej znany model, który jeszcze nie został uwzględniony w tym benchmarku, a który naszym zdaniem ma szanse uzyskać sensowne wyniki.

Co ciekawe w tabeli wyników MTEB w kategorii Retrieval wyróżnione są trzy główne podkategorie językowe - angielski, chiński oraz polski. W związku z tym ~~być może~~ ostatecznie zdecydowaliśmy skupić się na podkategorii dedykowanej dla języka polskiego.

Ostatecznie wybrany model - **Silver Retriever Base (v1.1)**

https://huggingface.co/ipipan/silver-retriever-base-v1.1

Opis modelu Silver Retriever

Jest to model wytworzony przez Piotra Rybaka oraz Macieja Ogrodniczuka z Polskiej Akademii Nauk.

Model Silver Retriever bazuje na modelu HerBERT-base i został dostrojony (ang. fine-tuned) na zestawach danych PolQA i MAUPQA.

Zbiory danych PolQA i MAUPQA

- PolQA
- MAUPQA

Dlaczego wybraliśmy akurat ten model?

Model Silver Retriever w wersji v1.1 spełnił wszystkie trzy kryteria które przede wszystkim braliśmy pod uwagę:

1. Model nie został jeszcze uwzględniony w benchmarku MTEB w kategorii Retrieval dla języka polskiego.
2. Model nie jest zbyt duży, ponieważ jako dwójka studentów nie dysponujemy ogromnymi zasobami mocy obliczeniowej.
3. Model ma szanse zrobić dobry wynik na tle pozostałych modeli.

Co ciekawe model ten w wersji v1 (czyli `silver-retriever-base-v1`) został przetestowany w benchmarku pod nazwą `herbert-base-retrieval-v2`. W związki z tym szczególnie interesujące dla nas będzie porównanie wyników tego modelu dla różnych jego wersji (tzn. v1 oraz v1.1).

Jakie inne modele były brane pod uwagę?

Pod uwagę brane były modele na HuggingFace spełniające następujące wymagania:

1. Obsługuje język Polski. Badanie dot. języka narodowego, będącego obecnie głównym zapotrzebowaniem komercyjnym.
2. Jest modelem kompatybilnym ze strukturą SentenceTransformers. Jest ona niezbędna do celu replikacji badania.
3. Nie znajduje się obecnie w badaniu MTEB.
4. Jest relatywnie mały. Ze względu na ograniczone zasoby, nie posiadamy sił obliczeniowych wymagannych przez większe modele.

Spośród 41 modeli najlepiej powyższe wymagania spełnił Silver Retriever.

Testowe zbiory danych

Name	Hub URL	Description	Type	Category	#Languages	Train #Samples
ArguAna-PL	BeIR-PL/arguana-pl	NFCorpus: A Full-Text Learning to Rank Dataset for Medical Information Retrieval	Retrieval	p2p	1	0
DBPedia-PL	BeIR-PL/dbpedia-pl	DBpedia-Entity is a standard test collection for entity search over the DBpedia knowledge base	Retrieval	s2p	1	0
FiQA-PL	BeIR-PL/fiqa-pl	Financial Opinion Mining and Question Answering	Retrieval	s2p	1	0
HotpotQA-PL	BeIR-PL/hotpotqa-pl	HotpotQA is a question answering dataset featuring natural, multi-hop questions, with strong supervision for supporting facts to enable more explainable question answering systems.	Retrieval	s2p	1	0
MSMARCO-PL	BeIR-PL/msmarco-pl	MS MARCO is a collection of datasets focused on deep learning in search. Note that the dev set is used for the leaderboard.	Retrieval	s2p	1	0
NFCorpus-PL	BeIR-PL/nfcorpus-pl	NFCorpus: A Full-Text Learning to Rank Dataset for Medical Information Retrieval	Retrieval	s2p	1	0

Name	Hub URL	Description	Type	Category	#Languages	Train #Samples
NQ-PL	BeIR-PL/nq-pl	Natural Questions: A Benchmark for Question Answering Research	Retrieval	s2p	1	0
Quora-PL	BeIR-PL/quora-pl	QuoraRetrieval is based on questions that are marked as duplicates on the Quora platform. Given a question, find other (duplicate) questions.	Retrieval	s2s	1	0
SCIDOCS-PL	BeIR-PL/scidocs-pl	SciDocs, a new evaluation benchmark consisting of seven document-level tasks ranging from citation prediction, to document classification and recommendation.	Retrieval	s2p	1	0
SciFact-PL	BeIR-PL/scifact-pl	SciFact verifies scientific claims using evidence from the research literature containing scientific paper abstracts.	Retrieval	s2p	1	0
TRECCOVID	BeIR/trec-covid	TRECCOVID is an ad-hoc search challenge based on the CORD-19 dataset containing scientific articles related to the COVID-19 pandemic	Retrieval	s2p	1	0

Środowisko obliczeniowe

Jako środowisko obliczeniowe (początkowo) wykorzystany został Google Colab z kartą graficzną `Nvidia V100`. Rozmiary zadań zdefiniowanych w benchmarku mteb w kategorii Retrieval dla języka polskiego okazały się zaskakująco różnorodne. Na wspomnianym wcześniej sprzęcie niektóre zadania wykonywały się kilka minut, a inne kilka godzin. Jeden z tasków zasługuje jednak na szczególną uwagę - `MSMARCO-PL`. Do ukończenia go musieliśmy ostatecznie wynająć kartę `Nvidia A100`, która potrzebowała około 9 godzin na wykonanie zadania. Całe szczęście udało się przetestować model na wszystkich zbiorach danych.

Wyniki eksperymentów

Dokładne wyniki dla każdego z zadań można znaleźć w folderze `/mteb_benchmark/results/pl/silver-retriever-base-v1.1`. Wyniki dla każdego z "tasków" zawierają się w oddzielnych plikach w formacie `.json`.

Poniższa tabela jest uzupełnieniem aktualnej tabeli benchmarku mteb w kategorii retrieval dla języka polskiego. Tabela ta została uzupełniona o dodatkowy wiersz z danymi dla testowanego przez nas modeli.

Model	Average	ArguAna-PL	DBPedia-PL	FiQA-PL	HotpotQA-PL	MSMARCO-PL	NFCorpus-PL
mmlw-roberta-large	52.71	63.4	40.27	40.89	71.04	36.63	33.94
mmlw-e5-large	52.63	63.25	39.84	39.9	70.94	36.47	34.03
mmlw-e5-base	50.06	58.4	37.19	34.53	66.25	32.54	33.71
mmlw-roberta-base	49.92	59.02	36.22	35.01	66.64	33.05	34.14
multilingual-e5-large	48.98	53.02	35.82	33.0	67.41	33.38	30.24
multilingual-e5-base	44.01	42.81	30.23	25.52	63.52	29.52	25.98
mmlw-e5-small	42.83	54.31	30.28	29.75	57.14	25.94	27.6
multilingual-e5-small	42.43	37.43	29.27	22.03	60.15	26.94	26.48
st-polish-kartonberta-base-alpha-v1	42.19	56.06	27.0	24.73	50.61	43.25	31.15
st-polish-paraphrase-	34.44	51.87	24.59	22.27	32.11	17.91	24.05

Model	Average	ArguAna-PL	DBPedia-PL	FiQA-PL	HotpotQA-PL	MSMARCO-PL	NFCorpus-PL
from-mpnet							
st-polish-paraphrase-from-distilroberta	32.08	49.42	19.82	19.58	23.47	16.51	22.49
paraphrase-multilingual-mpnet-base-v2	29.16	42.62	20.18	14.68	29.36	12.45	18.53
paraphrase-multilingual-MiniLM-L12-v2	26.66	37.83	18.0	12.49	22.76	10.39	17.16
LaBSE	23.36	38.52	16.1	7.63	19.72	7.22	17.45
distiluse-base-multilingual-cased-v2	21.18	36.7	12.36	8.02	20.83	4.57	16.28
herbert-base-retrieval-v2	39.16	41.97	24.07	24.25	43.41	51.56	25.95
silver-retriever-base-v1.1	37.59	41.72	23.69	22.07	38.51	46.32	24.48

Wnioski

Ze względu na pogorszenie wyników pomiędzy modelami herbert-base-retrieval-v2 (silver-retriever-base-v1) a silver-retriever-base-v1.1 doszliśmy do wniosków, że być może dotrenowanie modelu pogorszyło jego działanie w ogólnych przypadkach. Pogorszenie statystyk wykazuje 10/11 benchmarków, więc z dużą dozą pewności można stwierdzić iż do zadań tego typu należy używać wersji v1 zamiast v1.1.

W celu publikacji rezultatów naszego badania, jesteśmy w trakcie nawiązywania kontaktu z oryginalnymi autorami modeli z rodziny silver-retrieval by potwierdzić nasze wnioski i dokonać kontrybucji do projektu MTEB.

Aplikacja

W celu dokonania oglądu własnego działania wybranego modelu i weryfikacji jego możliwości został on podłączony do bazy wektorowej Chroma. Na podstawie tej bazy, utworzono standardowy interfejs, za pomocą którego użytkownik jest w stanie zapewnić źródła bazie wiedzy a następnie ją odpytać za pośrednictwem LLM. Do testowania zostały jako LLM użyte modele GPT4-turbo oraz Falcon-180b udostępniane

komercyjnie odpowiednio przez OpenAI oraz IBM. Dla wygody użytkownika, dodano także możliwość połączenia się z modelami udostępnianymi na platformach WatsonX oraz HuggingFace.

W trakcie testów z pomocą aplikacji doszliśmy do następujących wniosków:

1. Model silver v1.1 jak na swoją wielkość, zachowuje się bardzo efektywnie
2. Znalezione przez model silver v1.1 informacje są w znacznej większości przypadków związane z zadanym pytaniem i prawie zawsze prawidłowo posortowane od najmniej do najbardziej istotnych.
3. Wszelkie braki w odpowiedziach aplikacji zostały ujawnione jako wina interpretujących modeli językowych. O ile komercyjne zastosowania dawały stabilne wyniki, o tyle większość z nich nie jest dostosowanych do języka polskiego.

Technologia

Do obsługi bazy wektorowej wybrano open-source'ową bazę Chroma. Została ona przedłożona nad komercjalne rozwiązania tj. Pinecone czy DBLance głównie dzięki jej łatwej i darmowej budowie w środowisku lokalnym. Posiadając duże wsparcie społeczności, jest ona również tylko niewiele mniej wydajna od rozwiązań komercyjnych a więcej niż wystarczająco do użytkowania związanego z rozpatrywanym projektem.

Implementacją GUI został wybrany Streamlit, będących jedną z prostszych bibliotek do tego stworzonych. O ile nie posiada on znacznych możliwości regulacji, dzięki wysokiej abstrakcyjności obiektów można łatwo stworzyć powtarzalny, a mimo wszystko estetyczny interfejs. Znajduje on obecnie często zastosowanie w betatestowaniu i udostępnianiu usług świadczonych przez modele językowe.

Backend odpowiadający za przetwarzanie informacji oraz współpracę pomiędzy interfejsem, bazą wektorową (wiedzy) oraz LLM-em został opracowany w technologii LangChain. Jest to jedna z wiodących prym bibliotek udostępniających zaawansowane abstrakcje (nazywane łańcuchami) umożliwiające zarządzanie przepływem informacji w aplikacjiach opartych na modelach językowych. Zapewnia ona integrację z wielką liczbą dostawców zarówno modeli, jak i rozwiązań powiązanych. Zastosowanym łańcuchem został standardowy do takich zadań *RetrievalQA*, który kompiluje odpowiedź modelu na podstawie informacji uzyskanych z bazy wiedzy.

Źródła

[1] Andrew Rosenberg and Julia Hirschberg. 2007. Vmeasure: A conditional entropy-based external cluster evaluation measure. pages 410–420.

[2] Muennighoff, Niklas, Nouamane Tazi, Loïc Magne, i Nils Reimers. "MTEB: Massive Text Embedding Benchmark." ArXiv:2210.07316.

[3] Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models

[4] https://blog.langchain.dev/evaluating-rag-pipelines-with-ragas-langsmith/

[5] https://cobusgreyling.medium.com/rag-evaluation-9813a931b3d4

[6] https://www.linkedin.com/pulse/what-retrieval-augmented-generation-grow-right/

[7] Thulke, David, Nico Daheim, Christian Dugast and Hermann Ney. “Efficient Retrieval Augmented Generation from Unstructured Knowledge for Task-Oriented Dialog.” ArXiv abs/2102.04643 (2021): n. pag.

[8] Lewis, Patrick, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler et al. "Retrieval-augmented generation for knowledge-intensive nlp tasks." Advances in Neural Information Processing Systems 33 (2020): 9459-9474.

[9] "BEIR-PL: Zero Shot Information Retrieval." ArXiv:2305.19840.

[10] Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." ArXiv:1810.04805.

[11] Radford, Alec, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. "Language Models are Unsupervised Multitask Learners." OpenAI.

[12] Liu, Yinhan, Danqi Chen, et al. "RoBERTa: A Robustly Optimized BERT Pretraining Approach." ArXiv:1907.11692.

[13] Sanh, Victor, Lysandre Debut, Julien Chaumond, and Thomas Wolf. "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter." ArXiv:1910.01108.

[14] Raffel, Colin, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer." Google, Mountain View, CA. ArXiv:1910.10683.

[15] Lan, Zhenzhong, Piyush Sharma, et al. "ALBERT: A Lite BERT for Self-supervised Learning of Language Representations." Google AI. ArXiv:1909.11942.

[16] Rybak, Piotr and Maciej Ogrodniczuk. "SilverRetriever: Advancing Neural Passage Retrieval for Polish Question Answering." ArXiv:2309.08469.