

**Uniwersytet Kardynała Stefana Wyszyńskiego w Warszawie
Wydział Matematyczno-Przyrodniczy. Szkoła Nauk Ścisłych**

Informatyka (stacjonarne II stopnia)

Dokumentacja techniczna wraz z analizą do projektu:

6. Analiza tekstów - translator (porównanie 3 języków, np. word embedding lub model mBART50)

Autorzy:

- 1) Aliaksandr Trukhanovich(129598)
- 2) Anastasiia Sadlovska(124563)

Współczesne technologie analizy danych

dr inż. Robert Kłopotek

Warszawa, 2025

Spis treści

Wykorzystane zbiory danych -----	4
Europarl (EN–PL) -----	4
Tatoeba (EN–DE) -----	4
TED Talks (EN–DE) -----	5
Wykorzystane metody i modele-----	6
SentenceTransformer-----	6
XLM-RoBERTa-----	6
mBART-----	6
Metody analityczne i ewaluacyjne-----	6
Opis implementacji -----	8
Struktura projektu-----	8
Wczytywanie i przygotowanie danych -----	8
Modele embeddingowe i translacyjne -----	9
Przetwarzanie i embeddingi-----	9
Ewaluacja i metryki -----	10
Wizualizacje i analiza-----	11
Integracja i benchmark -----	11
Filtrowanie danych-----	12
Eksperymenty i wyniki-----	13
Wyniki podobieństwa semantycznego -----	13
Ocena tłumaczeń (BLEU, chrF, BERTScore)-----	16
Korelacja między metodami -----	17
Wizualizacje PCA, t-SNE, histogramy-----	19
Analiza krytyczna-----	22
Źródła i wykorzystane narzędzia -----	23
Zbiory danych -----	23
Biblioteki i narzędzia programistyczne -----	23
Inne źródła -----	23

Wprowadzenie

Niniejszy projekt koncentruje się na analizie i porównaniu trzech różnych modeli wielojęzycznych, które oferują możliwości reprezentowania zdań oraz tłumaczenia ich między językami:

- SentenceTransformer (DistilUSE)
- XLM-RoBERTa (paraphrase-multilingual-mpnet-base-v2)
- mBART50 (facebook/mbart-large-50-many-to-many-mmt)

Celem jest nie tylko zbadanie jakości semantycznego dopasowania między zdaniami w różnych językach, ale również ocena jakości tłumaczenia dokonywanego przez model mBART. Projekt obejmuje wykorzystanie zestawów danych wielojęzycznych oraz ocenę skuteczności metod przy pomocy miar takich jak: cosinusowa podobieństwo wektorów, BLEU, chrF oraz BERTScore.

Dodatkowo, projekt spełnia wymagania dydaktyczne poprzez wykorzystanie trzech różnych metod, trzech zbiorów danych, a także zawiera funkcje przygotowujące dane wejściowe oraz umożliwiające wizualizację i porównanie wyników.

Wykorzystane zbiory danych

W projekcie wykorzystano trzy różne zestawy danych paralelnych (dwujęzycznych), które reprezentują zróżnicowane domeny tematyczne i stylistyczne: teksty formalne (Europarl), zdania codzienne i nieformalne (Tatoeba) oraz mowę narracyjną (TED Talks). Takie zróżnicowanie jest istotne, aby przeanalizować skuteczność różnych modeli tłumaczeń i reprezentacji semantycznych (embeddingów) w różnych kontekstach językowych.

Z uwagi na konieczność wielokrotnego przetwarzania danych przez różne modele embeddingowe i tłumaczeniowe, oraz generowania wykresów i obliczania metryk tłumaczeń, liczba przykładów została ograniczona, by zachować rozsądny czas wykonania (ok. 5–10 minut per dataset) oraz umożliwić łatwą analizę i wizualizację.

W praktyce każdy eksperyment generuje po kilka tysięcy wektorów i kilkanaście wykresów, a także przeprowadza obliczenia podobieństwa i metryk jakości tłumaczenia. W związku z tym:

- Dla Europarl i Tatoeba wybrano po 100 zdań.
- Dla TED Talks – 200 zdań, gdyż styl narracyjny wymaga większego kontekstu, a dane są bardziej zróżnicowane semantycznie.

Europarl (EN–PL)

Europarl to zbiór danych oparty na oficjalnych transkrypcjach debat w Parlamencie Europejskim. Tłumaczenia są wysokiej jakości, z formalnym słownictwem, poprawną składnią i specjalistycznym językiem urzędowym.

- Języki: angielski (EN) – polski (PL)
- Charakterystyka: teksty formalne, uporządkowane, wysokiej jakości
- Źródło: StatMT Europarl¹
- Cel użycia: ocena modeli w kontekście precyzyjnych, formalnych tłumaczeń

Tatoeba (EN–DE)

Zbiór Tatoeba zawiera krótkie zdania tłumaczone przez społeczność. Jego główną zaletą jest różnorodność językowa – od prostych zwrotów po idiomy i kolokwializmy. Jakość tłumaczeń może być nierówna, co pozwala przetestować odporność modeli.

- Języki: angielski (EN) – niemiecki (DE)
- Charakterystyka: proste i krótkie zdania, często codziennego użytku, zmienna jakość
- Źródło: Tatoeba²

¹ <https://www.statmt.org/europarl/>

² <https://tatoeba.org/en/downloads>

- Cel użycia: testowanie modeli na danych niskiego zasobu i o różnym stylu

TED Talks (EN–DE)

TED2020 to korpus zawierający tłumaczenia wystąpień z konferencji TED. Dane te łączą język mówiony z wysoką jakością tłumaczeń. Styl jest bardziej narracyjny i mniej formalny, co sprawia, że ocena modeli staje się bardziej złożona.

- Języki: angielski (EN) – niemiecki (DE)
- Charakterystyka: dłuższe zdania, narracyjna struktura, styl zbliżony do mowy
- Źródło: TED2020³
- Cel użycia: weryfikacja skuteczności modeli w kontekście wypowiedzi ustnych i bogatego języka naturalnego

³ <https://opus.nlpl.eu/TED2020/en&de/v1/TED2020>

Wykorzystane metody i modele

W celu dokładnej oceny jakości tłumaczeń i dopasowania semantycznego między językami, zdecydowano się wykorzystać trzy różne modele językowe, reprezentujące odmienne podejścia do przetwarzania języka naturalnego. Ich zestawienie pozwala spojrzeć na problem z różnych perspektyw – nie tylko jako porównanie tłumaczeń, ale także porównanie reprezentacji znaczenia zdań.

SentenceTransformer

Pierwszym zastosowanym modelem był SentenceTransformer⁴, konkretnie wariant *distiluse-base-multilingual-cased*. Ten model działa mniej-więcej stabilnie na krótkich i średnich tekstuach, dlatego doskonale sprawdza się w szybkiej ocenie jakości pary zdanie–tłumaczenie.

Model ten otrzymuje zdanie jako wejście i zwraca jego wektor reprezentujący znaczenie. Dzięki temu można porównać dwa zdania za pomocą prostych miar, takich jak *similarity cosine*.

XLM-RoBERTa

Drugim modelem był XLM-RoBERTa⁵ (dokładnie: paraphrase-multilingual-mpnet-base-v2). To model bardziej zaawansowany architektonicznie, który również służy do tworzenia osadzeń zdań, ale jest trenowany na większym zestawie danych i lepiej radzi sobie z wielojęzycznością.

Ma potencjał, by lepiej odwzorowywać znaczenie w parach językowych o większych różnicach strukturalnych, jak np. angielski i niemiecki.

mBART

Trzecią metodą był model tłumaczący mBART⁶ (facebook/mbart-large-50-many-to-many-mmt). W odróżnieniu od poprzednich, mBART nie tworzy embeddingów do porównania, tylko faktycznie generuje tłumaczenie z jednego języka na drugi.

Zastosowaliśmy mBART, aby zobaczyć, jak jakość rzeczywistego tłumaczenia wypada na tle podobieństwa semantycznego obliczanego bez tłumaczenia. Po przetłumaczeniu zdania, sprawdzaliśmy, jak dobrze mBART odwzorowuje sens oryginału, porównując jego tłumaczenie z rzeczywistym zdaniem docelowym.

Metody analityczne i ewaluacyjne

Aby dokładnie ocenić, jak dobrze działały zastosowane przez nas modele, zastosowaliśmy różne metody analizy i oceny wyników. Chcieliśmy nie tylko porównać surowe wartości, ale też lepiej zrozumieć, jak modele postrzegają podobieństwo znaczeń między zdaniami w różnych językach.

⁴ <https://www.sbert.net/>

⁵ <https://huggingface.co/sentence-transformers/paraphrase-multilingual-mpnet-base-v2>

⁶ <https://huggingface.co/facebook/mbart-large-50-many-to-many-mmt>

Dla modeli SentenceTransformer i XLM-RoBERTa skupiliśmy się na cosine similarity – to popularna metoda mierzenia podobieństwa między dwoma wektorami. W naszym przypadku każdy model zamieniał zdanie na jego „wektor znaczeniowy”, a następnie porównywaliśmy pary zdań oryginalnych i przetłumaczonych. Im wyższy wynik, tym większe podobieństwo, co zwykle oznacza, że model dobrze uchwycił sens zdania.

Model mBART działał nieco inaczej, a dokładnie on faktycznie tłumaczył zdania z jednego języka na drugi. Dlatego w jego przypadku mogliśmy porównać jakość tłumaczenia z oryginalnymi, referencyjnymi zdaniami, korzystając z klasycznych metryk tłumaczeniowych. BLEU sprawdza, jak bardzo wygenerowane tłumaczenie przypomina zdanie wzorcowe pod względem krótkich fragmentów tekstu. Choć BLEU to standard w branży, nie zawsze dobrze oddaje sens, zwłaszcza w językach o bogatej gramatyce, a więc dlatego dodaliśmy też chrF, który porównuje zdania na poziomie znaków. Na koniec użyliśmy BERTScore – metrykę, która porównuje znaczenia zdań, a nie tylko ich wygląd, co pozwala lepiej ocenić tłumaczenia nawet wtedy, gdy są napisane powiedzmy innymi słowami.

Żeby jeszcze lepiej zrozumieć, co widzą modele, skorzystaliśmy z PCA i t-SNE. Dzięki temu możemy wizualnie sprawdzić, czy model grupuje podobne znaczeniowo zdania blisko siebie, co by oznaczało, że działa poprawnie.

Na koniec sprawdziliśmy też koreacje między wynikami modeli. Zależało nam na tym, żeby zobaczyć, czy różne modele mają podobne opinie na temat tego, które zdania są do siebie podobne. Wysoka korelacja oznacza, że modele są zgodne między sobą, niska – że każdy z nich patrzy na problem zupełnie inaczej.

Według nas podejście było wszechstronne. Staraliśmy się spojrzeć na modele zarówno od środka, analizując ich reprezentacje semantyczne, jak i od zewnętrz, porównując jakość ich tłumaczeń względem danych wzorcowych. Dzięki temu mogliśmy wyciągnąć znacznie pełniejsze wnioski.

Opis implementacji

Projekt zrealizowano w języku Python z wykorzystaniem bibliotek do przetwarzania języka naturalnego, analizy danych oraz wizualizacji. Implementacja została przygotowana jako aplikacja konsolowa, uruchamiana poprzez plik `main.py`, zawierającą wszystkie kluczowe elementy projektu: wczytywanie i oczyszczanie danych, przetwarzanie modeli, obliczanie metryk oraz generowanie wizualizacji.

Struktura projektu

```
|── cache/           # Bufor osadzeń modeli  
|── data/            # Folder z danymi  
|   |── europarl/  
|   |── tatoeba/  
|   └── ted2020/  
|── results/          # Wyniki: CSV, wykresy  
|   |── csv/  
|   |── plots/  
|   |   |── correlation/  
|   |   |── pca/  
|   |   |── tsne/  
|   |   |── similarity/  
└── main.py          # Główny plik projektu
```

Wczytywanie i przygotowanie danych

Dane pochodzą z trzech różnych zbiorów korpusowych: Europarl, Tatoeba oraz TED. Funkcje do wczytywania danych wczytują dane z plików `.txt` lub `.tsv`, ograniczając ich liczbę dla szybszej analizy:

```
def load_europarl(n=100):  
    with open(EUROPARL_EN,      encoding='utf-8') as f:  
        en = [line.strip() for line in f.readlines()[:n]]  
    with open(EUROPARL_PL,      encoding='utf-8') as f:
```

```

pl    = [line.strip() for line in f.readlines()[:n]]
return en, pl

```

Zrzut ekranu 1. Funkcja do wczytywania danych

[źródło: plik main.py]

Analogiczne funkcje zastosowano dla danych TED i Tatoeba, z dodatkową filtracją duzości zdań.

Modele embeddingowe i translacyjne

Projekt wykorzystuje trzy różne podejścia do analizy podobieństwa semantycznego:

- SentenceTransformer - distiluse-base-multilingual-cased
- XLM-RoBERTa - paraphrase-multilingual-mpnet-base-v2
- mBART-50 - facebook/mbart-large-50-many-to-many-mmt

Modele są inicjalizowane w następujący sposób:

```

model_st = SentenceTransformer('distiluse-base-multilingual-cased')
tokenizer_mbart = MBart50TokenizerFast.from_pretrained("facebook/mbart-large-50-many-
t   o   -   m   a   n   y   -   m   m   t   "
model_mbart = MBartForConditionalGeneration.from_pretrained("facebook/mbart-large-50-
m   a   n   y   -   t   o   -   m   a   n   y   -   m   m   t   "
m   o   d   e   l   _   m   b   a   r   t   .   e   v   a   l   (   )
model_xlm = SentenceTransformer('sentence-transformers/paraphrase-multilingual-mpnet-
b   a   s   e   -   v   2   '
)

```

Zrzut ekranu 2. Inicjalizacja modeli

[źródło: plik main.py]

Przetwarzanie i embeddingi

Każde zdanie jest przekształcane w wektor embeddingowy, który następnie wykorzystuje się do obliczania podobieństwa semantycznego:

```

def embed_sentences(sentences, model, model_name, dataset_name, side, batch_size=64):
    path = cache_path(model_name, dataset_name, side)
    if os.path.exists(path):
        return np.load(path)

    embeddings = []
    for i in tqdm(range(0, len(sentences), batch_size), desc=f"Embedding {model_name} {side}"):
        batch = sentences[i:i + batch_size]
        embs = model.encode(batch, convert_to_tensor=False, show_progress_bar=False)
        embeddings.extend(embs)

```

```

embeddings = np.array(embeddings)
np.save(path, embeddings)
return embeddings

```

Zrzut ekranu 3. Funkcja embed_sentences()

[źródło: plik main.py]

Translacje generowane są przez mBART:

```

def      translate_mbart(texts,      src_lang,      tgt_lang,      batch_size=16):
    tokenizer_mbart.src_lang
    translations
    for i in tqdm(range(0, len(texts), batch_size), desc="Translating"):
        batch
        encoded = tokenizer_mbart(batch, return_tensors="pt", padding=True,
truncation=True)
        with
            generated_tokens
            **encoded,
            forced_bos_token_id=tokenizer_mbart.lang_code_to_id[tgt_lang]
        )
        translations.extend(tokenizer_mbart.batch_decode(generated_tokens,
skip_special_tokens=True))
    return translations

```

Zrzut ekranu 4. Funkcja translate_mbart()

[źródło: plik main.py]

Ewaluacja i metryki

Do oceny podobieństwa semantycznego stosujemy *cosine similarity*. Dla mBART dodatkowo obliczane są metryki BLEU, chrF oraz BERTScore:

```

def calc_metrics(hyps, refs, name):
    bleu = corpus_bleu([[ref.split()] for ref in refs], [hyp.split() for hyp in
hyps])
    chrf = corpus_chrf(refs, hyps)
    bert = scorer.compute(predictions=hyps, references=refs, lang=lang)
    bert_f1 = np.mean(bert["f1"])

    print(f"\n{name} - {dataset_name}")
    print(f"BLEU: {bleu:.4f}")
    print(f"chrF: {chrf:.4f}")
    print(f"BERTScore: {bert_f1:.4f}")

    return {
        "Model": name,
        "BLEU": bleu,
        "chrF": chrf,

```

```

        "BERTScore": bert_f1
    }

metric_results.append(calc_metrics(mbart_preds, refs, "mBART Translation"))

metrics_df = pd.DataFrame(metric_results)
metrics_df.to_csv(f"results/csv/translation_metrics_{dataset_name}.csv", index=False)

```

Zrzut ekranu 5. Metryki BLEU, chrF oraz BERTScore

[źródło: plik main.py]

Wizualizacje i analiza

Do prezentacji wyników wykorzystano:

- PCA i t-SNE (redukcja wymiarów osadzeń)
- histogramy rozkładów podobieństwa
- macierze korelacji (Pearsona)

Przykład wykresu podobieństw:

```

def plot_similarity_distributions(df, dataset_name):
    plt.figure(figsize=(10, 6))
    sns.histplot(df["Sim_SentenceTransformer"], label="SentenceTransformer",
    kde=True, stat="density", bins=25)
    sns.histplot(df["Sim_XLMR"], label="XLM-RoBERTa", kde=True, stat="density",
    bins=25)
    sns.histplot(df["Sim_mBART_Translation"], label="mBART Translation", kde=True,
    stat="density", bins=25)
    plt.title(f"Similarity Distributions - {dataset_name}")
    plt.xlabel("Cosine Similarity")
    plt.ylabel("Density")
    plt.legend()
    plt.tight_layout()
    plt.savefig(f"results/plots/similarity/similarity_plot_{dataset_name}.png")
    plt.close()

```

Zrzut ekranu 6. Przykład wykresu podobieństw

[źródło: plik main.py]

Integracja i benchmark

Eksperymenty dla trzech zbiorów danych są uruchamiane z poziomu funkcji `main()`:

```

def main():
    euro_en, euro_pl = load_europarl()
    tato_en, tato_de = load_tatoeba(TATOEBA)
    ted_en, ted_de = load_ted()

```

```
df1    = run_experiment(euro_en,   euro_pl,    "en_XX",    "pl_PL",    "Europarl")
df2    = run_experiment(tato_en,   tato_de,    "en_XX",    "de_DE",    "Tatoeba")
df3 = run_experiment(ted_en,   ted_de,    "en_XX",    "de_DE",    "TED")
```

Zrzut ekranu 7. Funkcja main()

[źródło: plik main.py]

Wszystkie wyniki zapisywane są jako `.csv` i `.png` w katalogu `results/`, co pozwala na ich dalszą analizę lub prezentację.

Filtrowanie danych

W projekcie zastosowano również podstawowe metody filtrowania danych w celu poprawy jakości analiz:

- 1) W zbiorze TED zdania dłuższe niż 100 słów są automatycznie odrzucane, co minimalizuje wpływ ekstremalnych długości na jakość embeddingów.
- 2) Losowe próbkowanie, tzn. w zbiorach Europarl i Tatoeba ograniczono liczbę przykładów do 100, co pozwala na szybsze testowanie i unikanie nadmiarowości danych.
- 3) Dla zbioru Tatoeba zastosowano mechanizm `drop_duplicates`(usuwanie duplikatów), eliminujący powtarzające się zdania źródłowe.

Te techniki pomagają skupić analizę na reprezentatywnych i zróżnicowanych danych oraz zapobiegają niepotrzebnemu zwiększeniu złożoności obliczeniowej.

Eksperymenty i wyniki

W tym rozdziale przedstawiono wyniki oraz analiza eksperymentów przeprowadzonych na trzech zbiorach danych: Europarl, Tatoeba oraz TED Talks. Wyniki obejmują obliczenia podobieństwa semantycznego między parami zdań, ocenę jakości tłumaczeń oraz analizę korelacji między metodami i wizualizacje osadzeń.

Wyniki podobieństwa semantycznego

W celu oceny jakości dopasowania par zdań angielsko-polskich, angielsko-niemieckich oraz innych wariantów językowych w badanych zbiorach przeprowadzono analizę podobieństwa semantycznego. W analizie wykorzystano trzy różne metody embeddingowe:

- 1) SentenceTransformer (ST) – model distiluse-base-multilingual-cased
- 2) XLM-RoBERTa (XLMR) – model paraphrase-multilingual-mpnet-base-v2
- 3) mBART – facebook/mbart-large-50-many-to-many-mmt, przy czym ocenie poddano podobieństwo pomiędzy tłumaczeniami generowanymi przez mBART a tłumaczeniami referencyjnymi

Podobieństwo semantyczne mierzone było za pomocą metryki cosine similarity. Wyniki zostały zapisane w formacie .csv i znajdują się w plikach:

- results/csv/results_Europarl.csv
- results/csv/results_Tatoeba.csv
- results/csv/results_TED.csv

Na podstawie analizowanego zbioru danych uzyskano następujące średnie wyniki podobieństw:

Dataset	Mean_ST	Mean_XLMR	Mean_mBART	STD_ST	STD_XLMR	STD_mBART
Europarl	0.9023667	0.9234548	0.9270765	0.09552293	0.092057474	0.092423
Tatoeba	0.9083122	0.9303388	0.9353058	0.10306738	0.09087657	0.103030466
TED	0.18707348	0.35449016	0.19032739	0.22837424	0.22090119	0.23241976

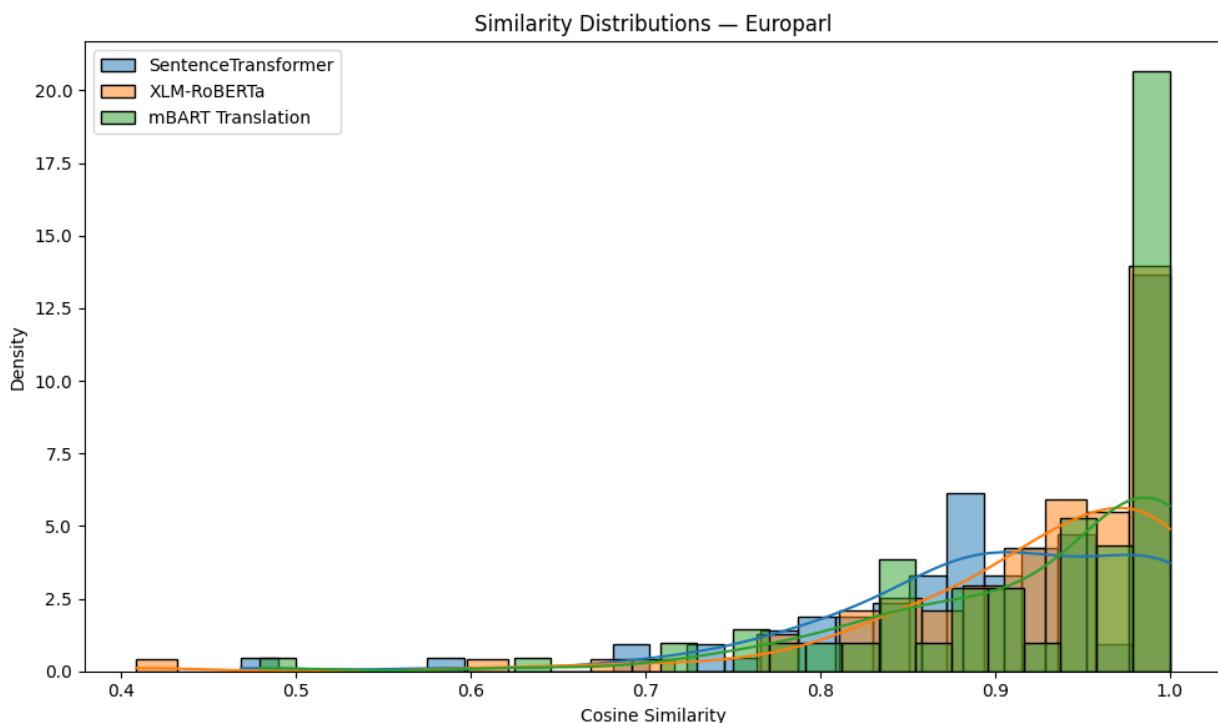
Tabela 1. Średnie wyniki podobieństw

[źródło: plik benchmark_results.csv]

Wysokie wartości średniego podobieństwa dla zbiorów Europarl i Tatoeba wskazują, że zarówno modele SentenceTransformer, XLM-RoBERTa, jak i translacje generowane przez mBART zachowują spójność semantyczną w przypadku krótkich i formalnych zdań. Warto zwrócić uwagę, że mBART osiąga najwyższe średnie podobieństwo względem tłumaczeń referencyjnych w każdym zbiorze, co świadczy o wysokiej jakości jego generacji. XLM-RoBERTa przewyższa

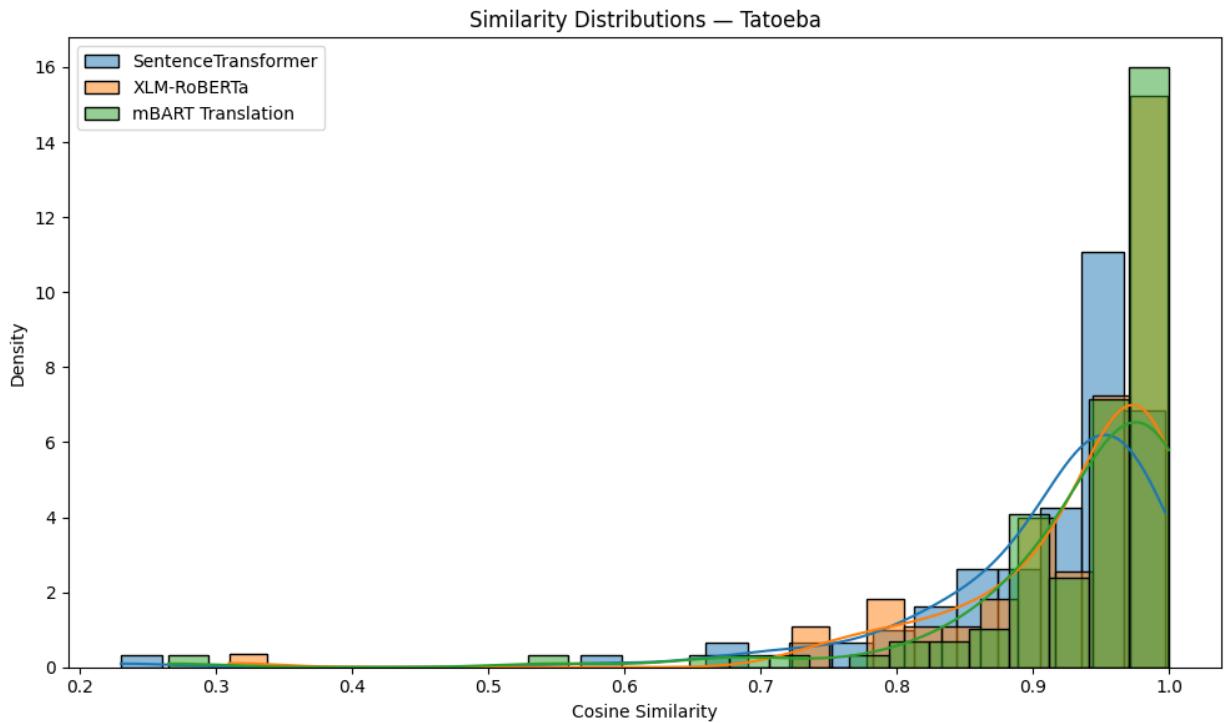
SentenceTransformer pod względem dokładności semantycznej, co potwierdza jego większą zdolność do reprezentowania znaczeń wielojęzycznych tekstów. TED Talks, jako zbiór dłuższych i bardziej złożonych zdań, cechuje się znacznie niższymi wynikami podobieństwa, co wynika z trudności modeli w uchwyceniu kontekstu i różnic składniowych w bardziej narracyjnych tekstach.

Dla każdego zbioru danych wygenerowano histogramy rozkładów podobieństw między zdaniami źródłowymi a tłumaczeniami. Pozwalają one wizualnie porównać jakość dopasowania poszczególnych modeli:



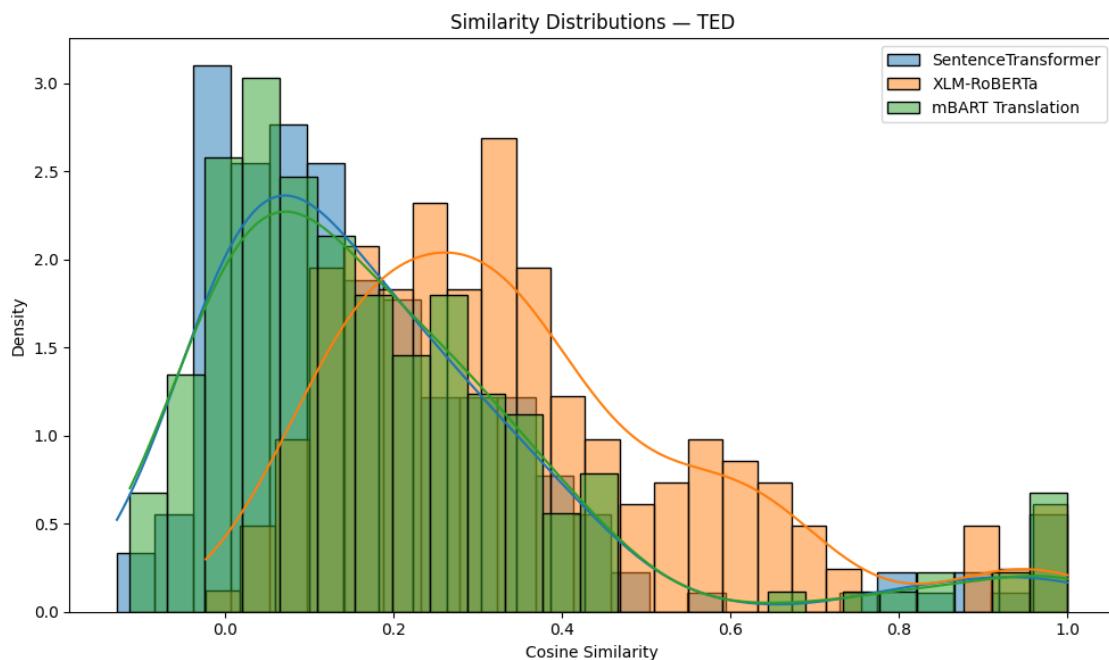
Histogram 1. Rozkład podobieństw dla Europarl

[źródło: folder /similarity]



Histogram 2. Rozkład podobieństw dla Tatoeba

[źródło: folder /similarity]



Histogram 3. Rozkład podobieństw dla TED2020

[źródło: folder /similarity]

Europarl i Tatoeba charakteryzują się rozkładami skoncentrowanymi wokół wartości ~0.9, co potwierdza dużą spójność semantyczną.

TED wykazuje znaczne rozproszenie, co pokazuje wyzwania stojące przed modelami przy analizie bardziej otwartych dyskursów.

Ocena tłumaczeń (BLEU, chrF, BERTScore)

Jakość tłumaczeń wygenerowanych przez model mBART oceniono za pomocą standardowych metryk BLEU, chrF, BERTScore.

Pliki z wynikami metryk:

- results/csv/translation_metrics_Europarl.csv
- results/csv/translation_metrics_Tatoeba.csv
- results/csv/translation_metrics_TED.csv

mBART Translation – Europarl	mBART Translation – Tatoeba	mBART Translation – TED
BLEU: 0.3126	BLEU: 0.3617	BLEU: 0.0112
chrF: 0.5523	chrF: 0.6312	chrF: 0.1909
BERTScore: 0.9050	BERTScore: 0.9054	BERTScore: 0.6534

Zrzut ekranu 8. Metryki oceny jakości tłumaczeń

[źródło: wyniki z konsoli]

Poniżej przedstawiono Tabela pod nr 2 dla trzech zbiorów danych, który łączy wszystkie wyniki w celu dalszej analizy danych:

Dataset	BLEU	chrF	BERTScore
Europarl	0.3126	0.5523	0.9050
Tatoeba	0.3617	0.6312	0.9054
TED	0.0112	0.1909	0.6534

Tabela 2. Średnie wyniki podobieństw

[źródło: opracowanie własne]

Z analizy wyników wynika, że model mBART generuje wysokiej jakości tłumaczenia dla zbiorów Europarl i Tatoeba. Osiągnięcie wartości BERTScore powyżej 0.90 oraz stosunkowo wysokich wartości BLEU i chrF świadczy o tym, że tłumaczenia są zarówno semantycznie poprawne, jak i zgodne z tłumaczeniami referencyjnymi pod względem struktury i składni.

Dodatkowo obserwując zbiór TED prezentuje znacznie niższe wartości we wszystkich trzech metrykach. BLEU na poziomie 0.01 oraz BERTScore poniżej 0.66 sugerują trudności mBART z tłumaczeniem bardziej złożonych, kontekstowo bogatych zdań wykładowych. Może to wynikać z

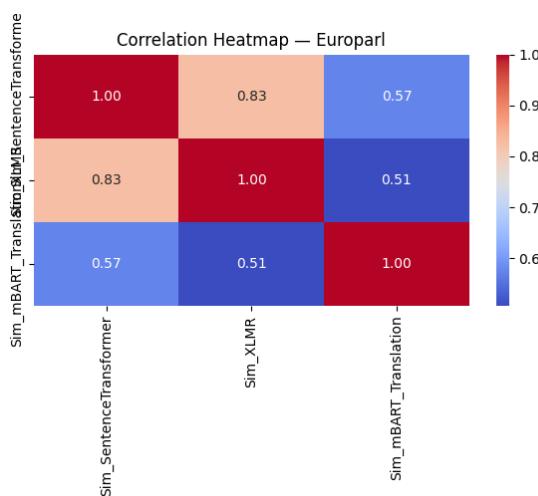
różnic w strukturze i długości zdań oraz większego znaczenia kontekstu, z którym mBART sobie nie radzi bez rozszerzonych okien kontekstowych.

Korelacja między metodami

Zbadano korelacje Pearsona pomiędzy wynikami podobieństwa wygenerowanymi przez trzy modele. Poniżej też są przedstawione mapy cieplne korelacji.

Pliki z macierzami korelacji:

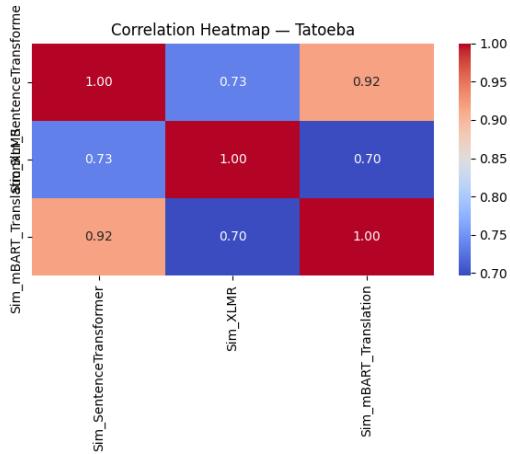
- `results/csv/correlation_Europarl.csv`
- `results/csv/correlation_Tatoeba.csv`
- `results/csv/correlation_TED.csv`



Rysunek 1. Heatmap Europarl

[źródło: folder /correlation]

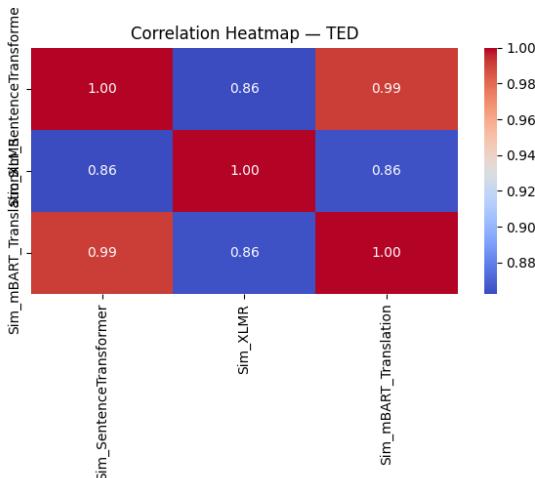
W przypadku Europarl koreacje są umiarkowane, a najniższa (0.51) występuje pomiędzy XLM-R a mBART. Może to sugerować, że model mBART wykorzystuje odmienne mechanizmy do oceny semantycznej zgodności niż modele embeddingowe działające bez translacji.



Rysunek 2. Heatmap Tatoeba

[źródło: folder /correlation]

W zbiorze Tatoeba najwyższa korelacja występuje pomiędzy SentenceTransformer a mBART (0.92), co może wynikać z podobnych reprezentacji dla krótkich i prostych zdań.



Rysunek 3. Heatmap TED

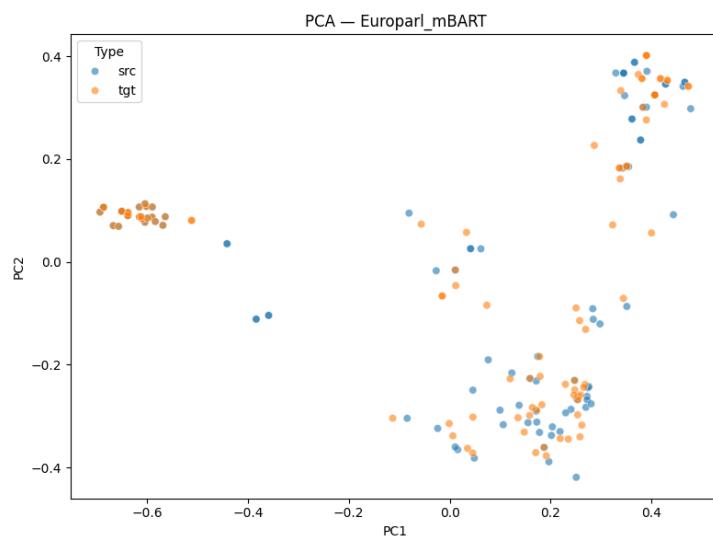
[źródło: folder /correlation]

W zbiorze TED zaobserwowano bardzo silną korelację pomiędzy wszystkimi metodami (powyżej 0.85), co sugeruje, że wszystkie modele podobnie oceniają stopień semantycznego dopasowania, mimo niskich wartości średnich, jak pokazano wcześniej w tabeli nr 2.

Wnioskując, korelacje wskazują na różnice w podejściu poszczególnych modeli do reprezentacji semantyki. Modele embeddingowe (ST, XLM-R) są bardziej spójne między sobą, natomiast translacyjny mBART, mimo wysokich wartości podobieństwa, generuje wyniki mniej skorelowane z klasycznymi embeddingami, szczególnie w zbiorach o bardziej formalnym i jednoznacznym języku, jak Europarl.

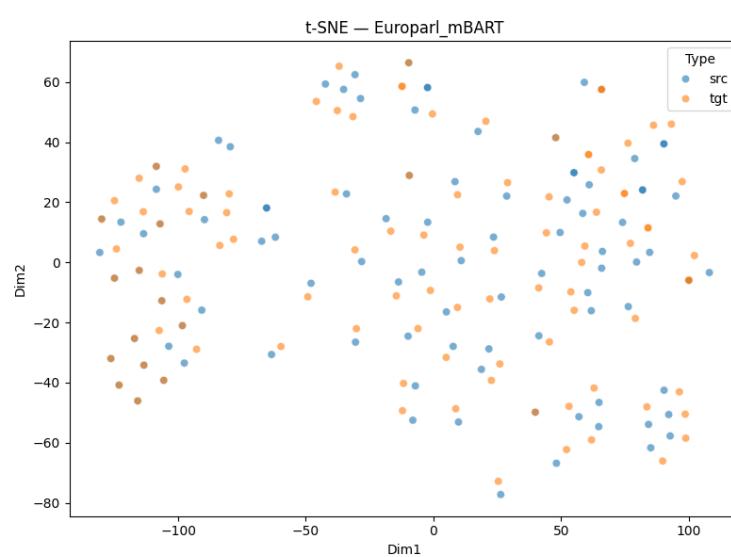
Wizualizacje PCA, t-SNE, histogramy

Aby lepiej zrozumieć strukturę osadzeń semantycznych, przeprowadzono redukcję wymiarowości za pomocą PCA oraz t-SNE. Poniżej umieszczone są przykłady kilku(ze względu na ilość wykresów) wykresów dla jednego modelu, pozostałe znajdują się w pliku `results/plots/pca` oraz `results/plots/tsne`:



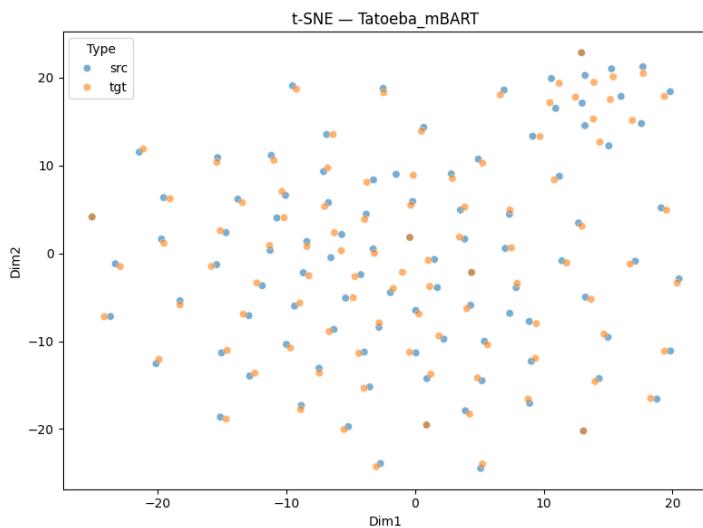
Histogram 4. PCA Europarl mBART

[źródło: folder /pca]



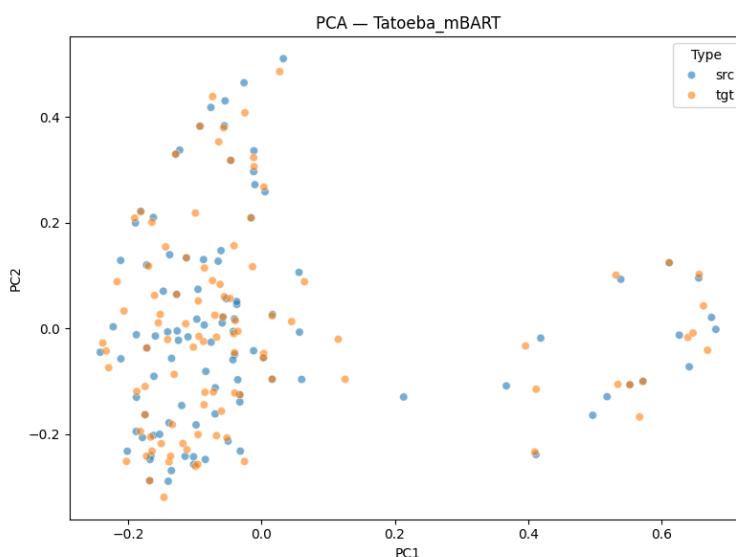
Histogram 5. t-SNE Europarl mBART

[źródło: folder /tsne]



Histogram 6. t-SNE Tatoeba mBART

[źródło: folder /tsne]



Histogram 7. PCA Tatoeba mBART

[źródło: folder /pca]

Wizualizacje PCA i t-SNE pokazują, jak bardzo różnią się od siebie dane z trzech zbiorów: Europarl, Tatoeba i TED, jeśli chodzi o podobieństwo semantyczne mierzone trzema modelami: SentenceTransformer (ST), XLM-RoBERTa (XLMR) i mBART (na podstawie tłumaczeń).

Pierwsza rzecz, która rzuca się w oczy, to jak bardzo TED odstaje od pozostałych dwóch. W obu metodach PCA i t-SNE tworzy on osobny obszar, wyraźnie oddzielony od Europarl i Tatoeba. Ma to sens, bo TED to zupełnie inny typ tekstu bardziej mówiony, mniej formalny, z dużą różnorodnością tematów. I to widać w niższych wartościach podobieństwa między zdaniami.

To potwierdza, że nie tylko sam model, ale też charakter danych ma ogromne znaczenie przy analizie jakości tłumaczeń i podobieństw między zdaniami.

Analiza krytyczna

Przeprowadzony eksperyment porównawczy trzech metod embeddingowych (SentenceTransformer, XLM-RoBERTa, mBART) pozwala sformułować kilka istotnych wniosków dotyczących ich skuteczności i zastosowania w praktycznych zadaniach oceny podobieństwa semantycznego.

Po pierwsze, wszystkie trzy modele embeddingowe – SentenceTransformer, XLM-RoBERTa i mBART – pokazały wysoką skuteczność w rozpoznawaniu podobieństw semantycznych w zbiorach Europarl i Tatoeba. Szczególnie mBART, który działał jako tłumacz i potem oceniał trafność tłumaczenia, uzyskał najwyższe średnie podobieństwo. Może to jednak wynikać z pewnej „przewagi własnego modelu” – tłumaczenie generowane przez mBART było oceniane również przez mBART-owe embeddingi, co może lekko zawyjaźić wyniki.

Po drugie, wyniki dla zbioru TED były znacznie słabsze, co może sugerować, że modele gorzej radzą sobie z bardziej zróżnicowanym i mniej formalnym językiem. To ważny wniosek, bo pokazuje, że świetne wyniki w jednym typie danych (np. debaty parlamentarne) nie muszą przekładać się na inne (np. wystąpienia publiczne czy rozmowy).

Również warto podkreślić, że użyta liczba prób była ograniczona ze względu na czas i zasoby obliczeniowe. Taki zakres był wystarczający do celów porównawczych i testowych, ale w większych zastosowaniach konieczne byłoby zwiększenie skali, by uzyskać pełniejszy obraz skuteczności modeli.

Z metodologicznego punktu widzenia, mocną stroną naszego podejścia było zastosowanie kilku różnych metryk ewaluacyjnych (BLEU, chrF, BERTScore) oraz różnych technik wizualizacji (PCA, t-SNE, histogramy), co pozwoliło zróżnicowanie ująć dane i nie opierać się tylko na jednym sposobie oceny.

Warto również zauważyć, że projekt nie wykorzystywał zaawansowanych technik pre- i postprocessingowych poza podstawowym filtrowaniem długości zdań czy duplikatów. W przyszłości warto byłoby rozszerzyć preprocessing o normalizację, czy usuwanie szumów językowych, by sprawdzić, czy wpłynęłoby to na poprawę wyników.

Źródła i wykorzystane narzędzia

Zbiory danych

- 1) Europarl Corpus – zbiór zdań tłumaczonych między językiem angielskim i polskim, pochodzący z protokołów posiedzeń Parlamentu Europejskiego
Źródło: <https://www.statmt.org/europarl/>
- 2) Tatoeba Corpus – zbiór przykładów zdań z tłumaczeniami tworzonych przez społeczność
Źródło: <https://tatoeba.org/en/downloads>
- 3) TED Talks (TED2020) – tłumaczenia transkrypcji prezentacji TED
Źródło: <https://opus.nlpl.eu/TED2020/en&de/v1/TED2020>

Biblioteki i narzędzia programistyczne

Projekt został zrealizowany w języku Python (3.9) z użyciem poniższych narzędzi:

- Transformers (Hugging Face) – do wykorzystania modeli mBART oraz tokenizerów
- Sentence-Transformers – dla modeli distiluse-base-multilingual i paraphrase-multilingual-mpnet-base-v2
- Scikit-learn – do redukcji wymiarów (PCA, t-SNE) i obliczeń statystycznych
- Matplotlib i Seaborn – do tworzenia wizualizacji (histogramy, heatmapy)
- NumPy, Pandas – manipulacja i analiza danych
- Evaluate (Hugging Face) – do obliczania metryk BLEU, chrF i BERTScore
- NLTK – pomocniczo przy ewaluacji BLEU

Inne źródła

- 1) mBART: Facebook AI Research
Liu et al. (2020). Multilingual Denoising Pre-training for Neural Machine Translation.
Źródło: <https://huggingface.co/facebook/mbart-large-50-many-to-many-mmt>
- 2) Dokumentacja modeli i tokenizerów
Źródło: <https://huggingface.co/docs>
- 3) PCA, t-SNE, metody analizy
Źródło: <https://scikit-learn.org/stable/>
- 4) Notatniki Hugging Face z przykładami oceny tłumaczeń i BERTScore
Źródło: <https://github.com/huggingface/notebooks/tree/main>
- 5) Oficjalne repozytorium Sentence-BERT, zawierające przykłady użycia modeli do porównania semantycznego

Źródło: <https://github.com/UKPLab/sentence-transformers>

- 6) A Beginner's Guide to Vector Embeddings

Źródło: <https://www.youtube.com/watch?v=NEreO2zlXDk>

- 7) StatQuest: t-SNE, Clearly Explained

Źródło: <https://www.youtube.com/watch?v=NEaUSP4YerM>

- 8) Na potrzeby projektu korzystano również z materiałów dydaktycznych przekazanych przez dr inż. Roberta Kłopotka w ramach zajęć, które pomogły w odświeżeniu oraz utrwaleniu teoretycznych podstaw przetwarzania języka naturalnego, metryk ewaluacyjnych oraz pracy z embeddingami. Były one nieocenionym wsparciem w praktycznej realizacji tego projektu.