# Laboratorium 4 Porównanie metryk ewaluacyjnych modeli językowych

## 1. Wprowadzenie

Ewaluacja modeli językowych jest kluczowym elementem oceny ich jakości. W zależności od zadania stosuje się różne metryki, gdyż żadna pojedyncza miara nie jest wystarczająca do pełnej oceny modelu.

Metryki automatyczne — podstawowe:

- BLEU (Bilingual Evaluation Understudy) — mierzy n-gramowe dopasowanie do referencji; stosowany w tłumaczeniu i summaryzacji
- ROUGE (Recall-Oriented Understudy for Gisting Evaluation) — mierzy recall n-gramów; ROUGE-1, ROUGE-2, ROUGE-L
- METEOR — uwzględnia synonimy i stemming, bardziej koreluje z oceną ludzką niż BLEU
- BERTScore — semantyczne podobieństwo z użyciem osadzeń BERT; odporny na parafrazowanie
- Perplexity — miara pewności modelu; im niższa, tym model jest lepiej skalibrowany

Każda metryka ma swoje słabości: BLEU penalizuje kreatywne parafrazowanie, ROUGE faworyzuje długie odpowiedzi, a BERTScore wymaga GPU dla dużych zbiorów. Dlatego w praktyce stosuje się kombinację metryk.

## 2. Przykłady implementacji

### Przykład 1: BLEU i ROUGE

```
# pip install nltk rouge-score

import nltk
from nltk.translate.bleu_score import sentence_bleu, corpus_bleu, SmoothingFunction
from rouge_score import rouge_scorer

nltk.download('punkt', quiet=True)

reference = 'The cat sat on the mat near the window'
hypotheses = [
    'The cat sat on the mat near the window',  # idealne
    'A cat was sitting on the mat by the window',  # parafra
    'The dog played in the garden',  # złe
]

ref_tokens = reference.lower().split()
smooth = SmoothingFunction().method1
```

```
print('=== BLEU ===',)
for hyp in hypotheses:
    hyp_tokens = hyp.lower().split()
    b1 = sentence_bleu([ref_tokens], hyp_tokens, weights=(1,0,0,0), smoothing_function=smooth)
    b4 = sentence_bleu([ref_tokens], hyp_tokens, weights=(.25,.25,.25,.25), smoothing_function=smooth)
    print(f'  BLEU-1={b1:.3f}  BLEU-4={b4:.3f}  |  {hyp[:45]}')

print()
print('=== ROUGE ===')
scorer = rouge_scorer.RougeScorer(['rouge1','rouge2','rougeL'], use_stemmer=True)
for hyp in hypotheses:
    scores = scorer.score(reference, hyp)
    print(f'  R1={scores["rouge1"].fmeasure:.3f} R2={scores["rouge2"].fmeasure:.3f}  RL={scores["rougeL"].fmeasure:.3f}  | {hyp[:40]}')
```

## Przykład 2: BERTScore

```
# pip install bert-score

from bert_score import score as bert_score

references  = ['The cat sat on the mat near the window'] * 3
hypotheses  = [
    'The cat sat on the mat near the window',
    'A cat was sitting on the mat by the window',
    'The dog played in the garden',
]

P, R, F1 = bert_score(hypotheses, references, lang='en', verbose=False)
print('=== BERTScore ===')
for hyp, p, r, f in zip(hypotheses, P, R, F1):
    print(f'  P={p:.3f}  R={r:.3f}  F1={f:.3f}  |  {hyp[:45]}')
```

## Przykład 3: Perplexity modelu językowego

```
import torch
from transformers import AutoModelForCausalLM, AutoTokenizer
import math

def compute_perplexity(text, model_name='gpt2'):
    tokenizer = AutoTokenizer.from_pretrained(model_name)
    model     = AutoModelForCausalLM.from_pretrained(model_name)
    model.eval()
```

```
    inputs   = tokenizer(text, return_tensors='pt')
    with torch.no_grad():
        outputs  = model(**inputs, labels=inputs['input_ids'])
    loss = outputs.loss.item()
    return math.exp(loss)

texts = [
    'The weather is nice today and the sky is clear.',
    'Sky clear nice the today is weather.',     # bezsensowna kolejność
    'Quantum entanglement defies classical physics.',
]

for text in texts:
    ppl = compute_perplexity(text)
    print(f'PPL={ppl:7.1f}  |  {text}')
```

## Przykład 4: Kompleksowe porównanie modeli

```
import pandas as pd
from rouge_score import rouge_scorer
from nltk.translate.bleu_score import sentence_bleu, SmoothingFunction

# Symulacja odpowiedzi różnych modeli na to samo pytanie
reference = ('Climate change is driven by greenhouse gas emissions '
             'from human activities such as burning fossil fuels.')

model_outputs = {
    'GPT-2':    'Climate change results from greenhouse gases produced by humans burning fossil fuels.',
    'BLOOM':    'Global warming is caused by emissions from factories and vehicles using fossil energy.',
    'Baseline': 'Climate change is a global issue affecting weather patterns worldwide.',
}

scorer = rouge_scorer.RougeScorer(['rouge1','rouge2','rougeL'], use_stemmer=True)
smooth = SmoothingFunction().method1
results = []

for model, hyp in model_outputs.items():
    r = scorer.score(reference, hyp)
    b = sentence_bleu([reference.split()], hyp.split(),
                      weights=(.5,.5,0,0), smoothing_function=smooth)
    results.append({'Model':   model,
                    'BLEU-2':  round(b, 3),
                    'ROUGE-1': round(r['rouge1'].fmeasure, 3),
                    'ROUGE-2': round(r['rouge2'].fmeasure, 3),
```

```
                    'ROUGE-L': round(r['rougeL'].fmeasure, 3)})

df = pd.DataFrame(results).set_index('Model')
print(df.to_string())
```

## 3. Zadania do samodzielnego rozwiązania

**Zadanie 4.1: Analiza korelacji metryk z oceną ludzką**

Przygotuj zestaw 10 par (referencja, hipoteza) o zróżnicowanej jakości. Poproś 2–3 osoby o ocenę podobieństwa w skali 1–5 (lub oceń sam z różnych perspektyw). Oblicz BLEU-2, ROUGE-L i BERTScore-F1. Sprawdź korelację Spearmana (scipy.stats.spearmanr) między każdą metryką a oceną ludzką. Która metryka najlepiej koreluje?

**Zadanie 4.2: Porównanie modeli summaryzacji**

Wczytaj 5 artykułów z biblioteki datasets (CNN/DailyMail, split='test[:5]'). Dla każdego artykułu wygeneruj streszczenie za pomocą dwóch modeli: 'facebook/bart-large-cnn' i 'sshleifer/distilbart-cnn-12-6'. Porównaj ROUGE-1, ROUGE-2 i ROUGE-L z referencyjnymi streszczeniami. Który model jest lepszy przy uwzględnieniu rozmiaru i szybkości?

**Zadanie 4.3: Wpływ długości na metryki**

Wygeneruj odpowiedzi o długości 20, 50, 100 i 200 słów na ten sam prompt (np. streszczenie artykułu). Zbadaj, jak długość odpowiedzi wpływa na ROUGE-1, ROUGE-2 i BLEU. Sformułuj hipotezę: czy dłuższa odpowiedź jest zawsze oceniana wyżej? Kiedy metryki mogą być mylące?

**Zadanie 4.4: Implementacja własnej metryki METEOR**

Na podstawie dokumentacji NLTK zaimplementuj obliczanie metryki METEOR (nltk.translate.meteor_score) i porównaj ją z BLEU na 5 przykładach zawierających synonimy (np. 'big' vs 'large', 'happy' vs 'joyful'). Udowodnij, że METEOR lepiej radzi sobie z parafrazami niż BLEU-1.