

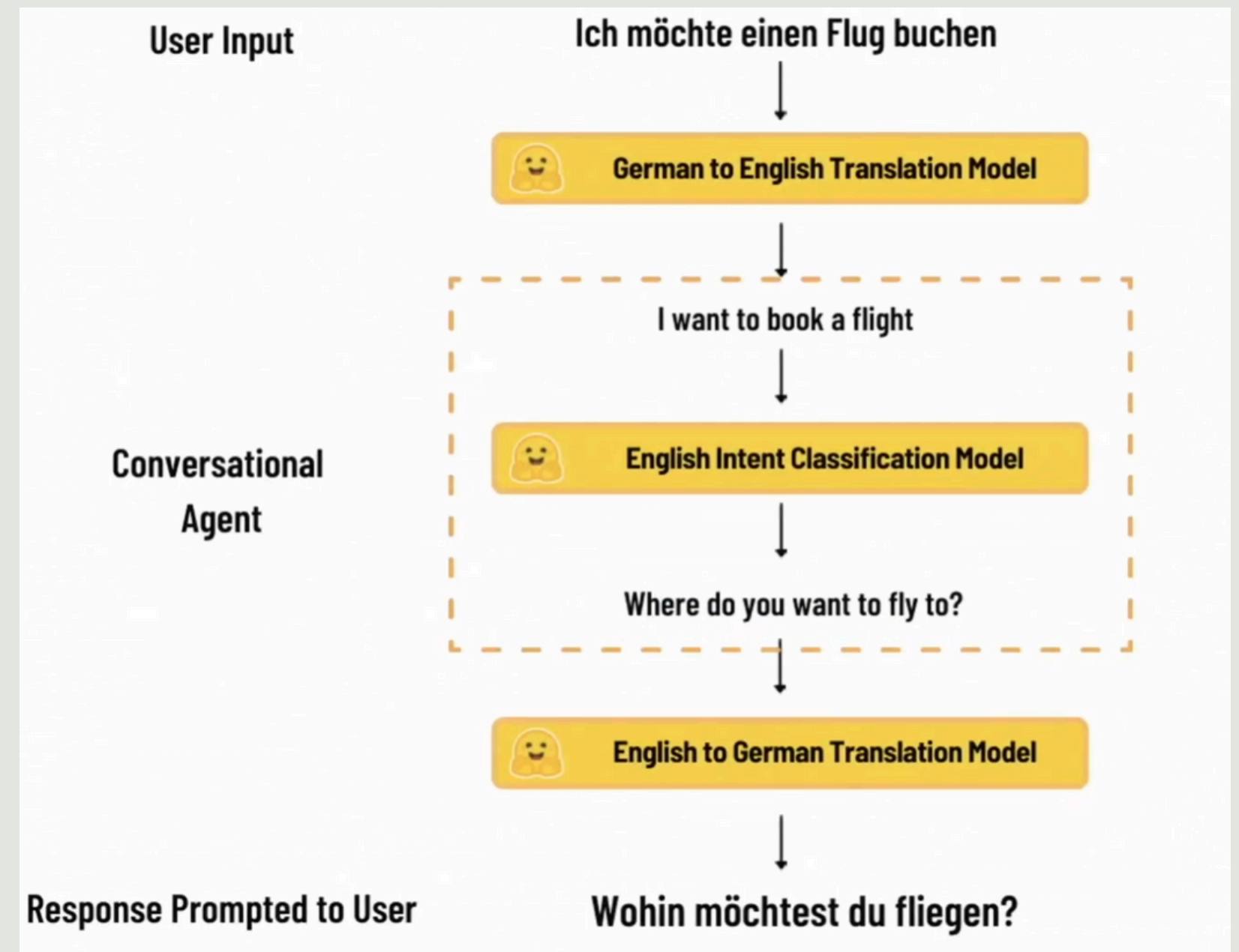
NARZĘDZIE DO TŁUMACZENIA TEKSTU WYKORZYSTUJĄCE TRANSFORMERY 'TRANSLATION' Z PORTALU HUGGING FACE

Przemysław Danak
Bruno Blajda

SPIIS TREŚCI

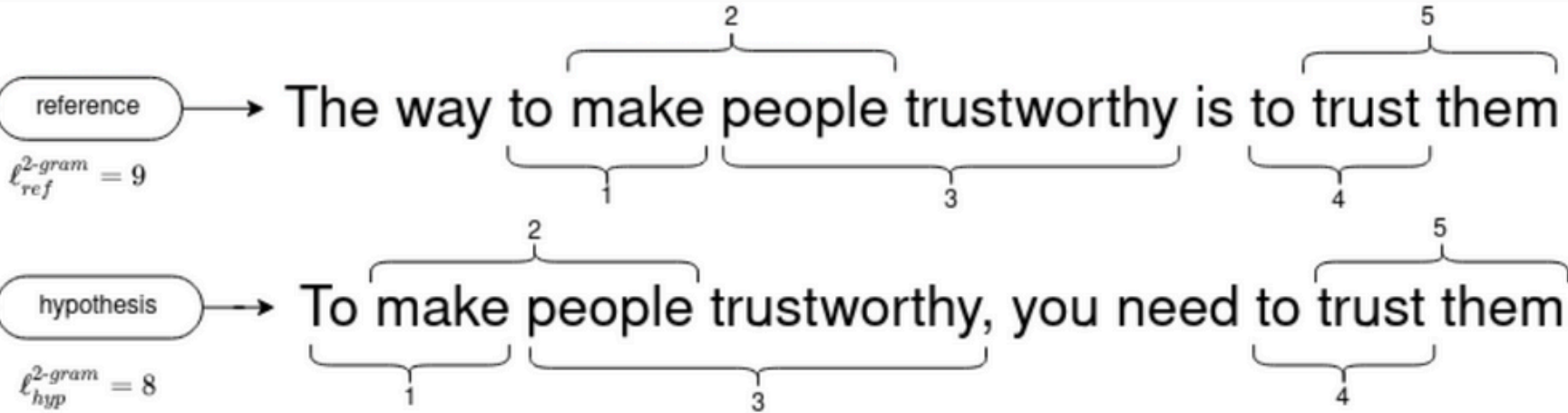
- Problem tłumaczenia języka
- Rodzaje tłumaczenia maszynowego
- Tłumaczenie tekstu z użyciem transformatorów
- T5 (Text-To-Text Transfer Transformer)
- Implementacja
 - Przygotowanie środowiska
 - Pobranie i podział danych
 - Przygotowanie tekstu dla modelu
 - Trenowanie modelu
 - Testowanie modelu
- Bibliografia

Tłumaczenie języków



BLEU score

$$\text{BLEU} = \text{BP} \times \exp \left(\frac{1}{n} \sum_{i=1}^n \log p_i \right)$$



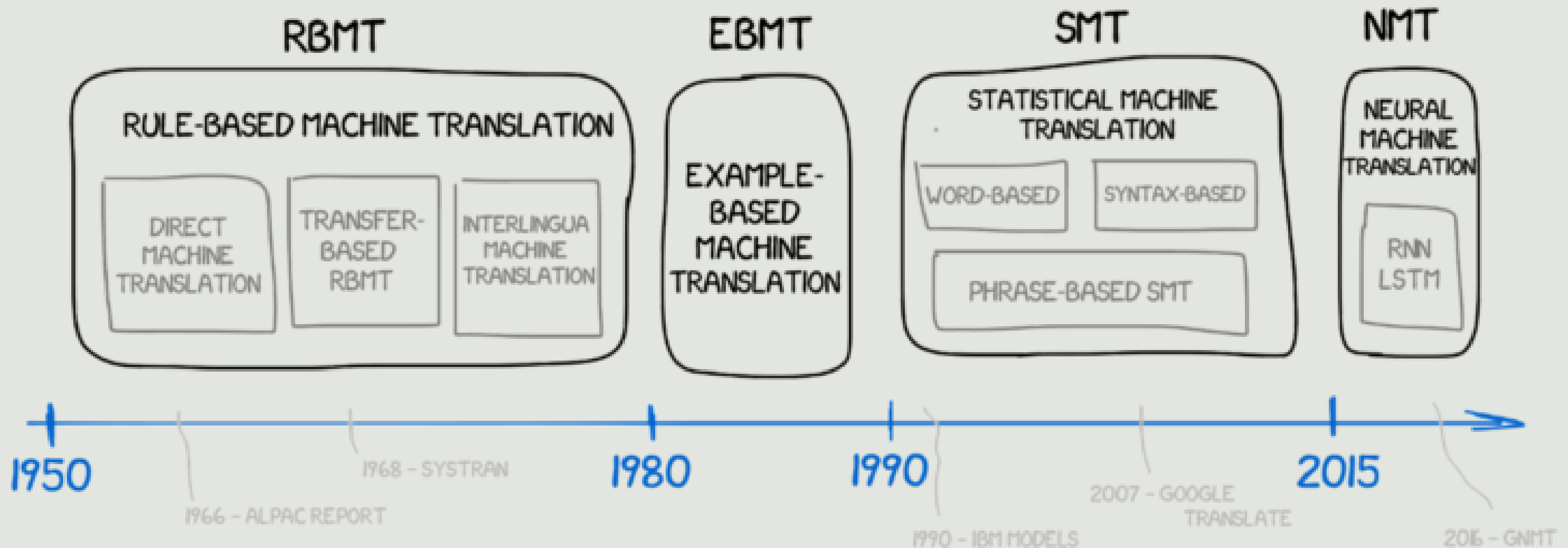
The following table details the precisions for 4 n-grams.

n-gram	1-gram	2-gram	3-gram	4-gram
p_n	$\frac{7}{9}$	$\frac{5}{8}$	$\frac{3}{7}$	$\frac{1}{6}$

BLEU	Interpretacja
< 0.1	Prawie bezużyteczny
0.1-0.19	Trudno zrozumieć sedno
0.2-0.29	Treść jest jasna, ale zawiera znaczące błędy gramatyczne
0.3-0.39	Zrozumiałe dla dobrych tłumaczeń
0.4-0.49	Wysokiej jakości tłumaczenia
0.5-0.59	Bardzo wysokiej jakości, adekwatne i płynne tłumaczenia
≥ 0.6	Jakość często lepsza niż u ludzi

Historia

A BRIEF HISTORY OF MACHINE TRANSLATION

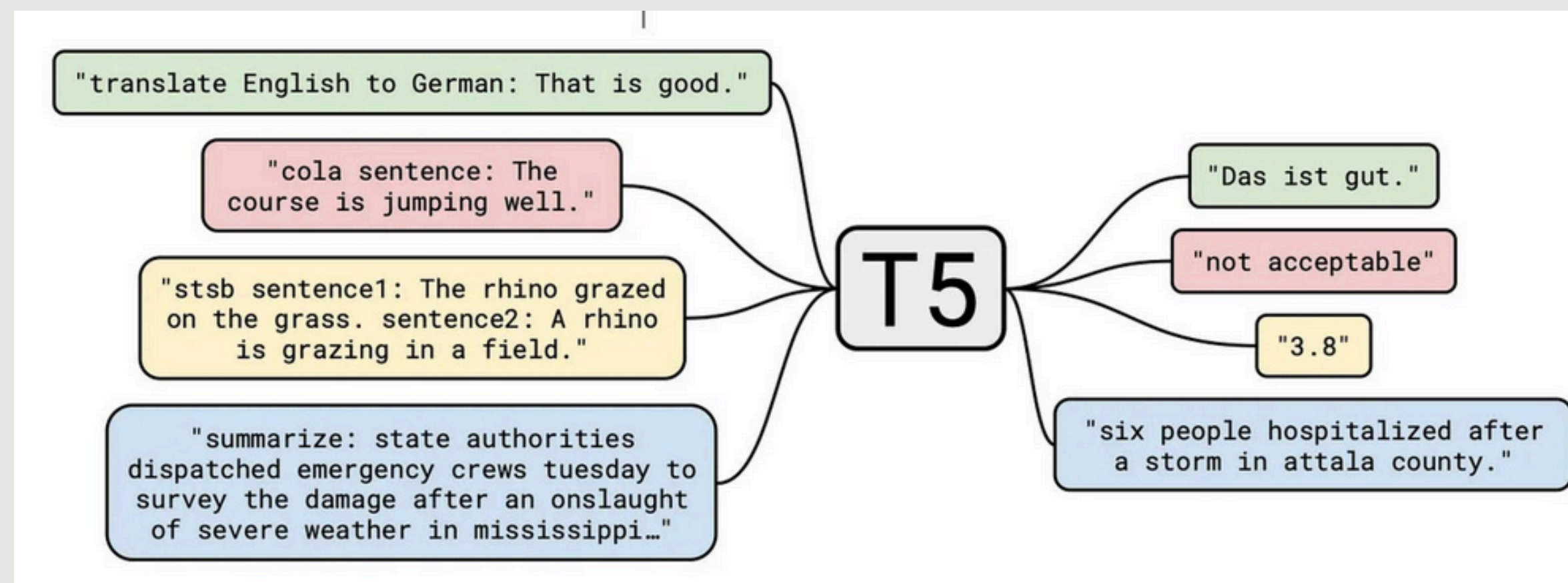


Tłumaczenie tekstu z użyciem transformatorów

- **poprzednie metody miały ograniczenia** w zrozumieniu kontekstu i idiomów
- **obecnie** wykorzystuje się **modele transformatorowe**
- osiągają wyższą precyzję tłumaczenia
- model T5 (Text-To-Text Transfer Transformer)

T5 (Text-To-Text Transfer Transformer)

- **mechanizm uwagi (attention)** – skupianie się na istotnych częściach tekstu
- **analizowanie tekstu w obu kierunkach**



- **wielozadaniowość** – rozwiązuje wiele różnych problemów językowych: tłumaczenia, podsumowywanie tekstu, analiza sentymentu
- **Hugging Face** dostarcza gotowe modele

Przygotowanie środowiska

- **Instalacja potrzebnych bibliotek**

- **Transformers** - umożliwia łatwe korzystanie z najnowszych modeli NLP
- **Datasets** - udostępnia ogromną liczbę zbiorów danych do nauki maszynowej
- **Evaluate** - ocenia modele NLP, udostępnia miarę BLEU (Bilingual Evaluation Understudy) - porównanie tłumaczenia z tłumaczeniem przygotowanym przez ludzi
- **SacreBLEU** - zapewnia standaryzację w ocenie tłumaczeń metryką BLEU

```
pip install transformers datasets evaluate sacrebleu
```


Pobranie i podział danych

- **zbiór danych bilingwalnych** - przykłady zdań w języku źródłowym i ich odpowiedników w języku docelowym - OPUS Books
- **podział danych na dane treningowe i testowe**
 - **dane treningowe (80%)** - używane do nauki modelu. Model analizuje te dane, aby nauczyć się zasad i wzorców tłumaczenia.
 - **dane testowe (20%)** - używane wyłącznie do oceny modelu po treningu.

```
{  
  "id": "126191",  
  "translation": {  
    "en": "Michaud and Grivet prided themselves on their correct attitude.",  
    "fr": "Michaud et Grivet s'applaudirent de leur excellente tenue."  
  }  
}
```

```
from datasets import load_dataset  
  
books = load_dataset("opus_books", "en-fr")  
books = books["train"].train_test_split(test_size=0.2)  
books["train"][0]
```

Przygotowanie tekstu dla modelu

- **pobranie tokenizera** - użycie modelu **google-t5/t5-small** - kompatybilny z modelem transformatorowym T5, szybszy w trenowaniu i generowaniu tłumaczeń w porównaniu z większymi wersjami, mniej wymagający pod względem zasobów sprzętowych

```
from transformers import AutoTokenizer  
  
checkpoint = "google-t5/t5-small"  
tokenizer = AutoTokenizer.from_pretrained(checkpoint)
```

Przygotowanie tekstu dla modelu

- **przygotowanie tekstów wejściowych i wyjściowych** - dodanie prefiksu, który informuje o tłumaczeniu tekstu: *translate English to French: tekst*,
 - definicja funkcji, która za to odpowiada
 - zastosowanie funkcji dla całego zbioru danych (tokenizacja)

```
source_lang = "en"
target_lang = "fr"

prefix = "translate English to French: "

def preprocess_function(examples):
    inputs = [prefix + example[source_lang] for example in examples["translation"]]
    targets = [example[target_lang] for example in examples["translation"]]
    model_inputs = tokenizer(inputs, text_target=targets, max_length=128, truncation=True)
    return model_inputs

tokenized_books = books.map(preprocess_function, batched=True)

print(tokenized_books['train'][0])
```

```
{
  "id": "124887",
  "translation": {
    "en": "He went to find old Michaud, and told him he had just recognized Camille",
    "fr": "Il alla chercher le vieux Michaud et lui dit qu'il venait de reconnaître"
  },
  "input_ids": [ ],
  "attention_mask": [ ],
  "labels": [ ]
}
```

Przygotowanie tekstu dla modelu

- utworzenie obiektu odpowiadającego za dynamiczny padding - zwiększa efektywność, wszystkie zdania w batchu wypełniane są do długości najdłuższego zdania w tej partii, a nie do globalnie ustalonej wartości maksymalnej

```
from transformers import DataCollatorForSeq2Seq  
  
data_collator = DataCollatorForSeq2Seq(tokenizer=tokenizer, model=checkpoint)
```

Definicja funkcji odpowiedzialnej za ocenę modelu

- kluczowa do dostosowania hiperparametrów
- metryka BLEU (Bilingual Evaluation Understudy) - mierzy, jak bardzo wygenerowane tłumaczenia przypominają tłumaczenia referencyjne
- stosowana w treningu

```
import numpy as np
import evaluate

metric = evaluate.load("sacrebleu")

def postprocess_text(preds, labels):
    preds = [pred.strip() for pred in preds]
    labels = [[label.strip()] for label in labels]
    return preds, labels

def compute_metrics(eval_preds):
    preds, labels = eval_preds
    if isinstance(preds, tuple):
        preds = preds[0]
    decoded_preds = tokenizer.batch_decode(preds, skip_special_tokens=True)
    labels = np.where(labels != -100, labels, tokenizer.pad_token_id)
    decoded_labels = tokenizer.batch_decode(labels, skip_special_tokens=True)
    decoded_preds, decoded_labels = postprocess_text(decoded_preds, decoded_labels)
    result = metric.compute(predictions=decoded_preds, references=decoded_labels)
    result = {"bleu": result["score"]}
    prediction_lens = [np.count_nonzero(pred != tokenizer.pad_token_id) for pred in preds]
    result["gen_len"] = np.mean(prediction_lens)
    result = {k: round(v, 4) for k, v in result.items()}
    return result
```

Trenowanie modelu

- dostarczenie modelowi danych treningowych
- określenie hiperparametrów - kluczowe ustawienia, które kontrolują sposób, w jaki model się uczy
- podczas treningu:
 - model generuje tłumaczenia
 - obliczany jest błąd - wynik jest porównywany z rzeczywistym tłumaczeniem referencyjnym
 - po każdej epoce model jest oceniany - pozwala lepiej dostosować model

```
from transformers import AutoModelForSeq2SeqLM, Seq2SeqTrainingArguments, Seq2SeqTrainer

model = AutoModelForSeq2SeqLM.from_pretrained(checkpoint)

small_train_dataset = tokenized_books["train"].select(range(4000))
small_test_dataset = tokenized_books["test"].select(range(1000))

training_args = Seq2SeqTrainingArguments(
    output_dir="translation_model",
    save_strategy="epoch",
    save_steps=500,
    eval_strategy="epoch",
    learning_rate=2e-5,
    per_device_train_batch_size=8,
    per_device_eval_batch_size=8,
    weight_decay=0.01,
    save_total_limit=2,
    num_train_epochs=3,
    predict_with_generate=True,
    fp16=True,
    report_to=["none"]
)

trainer = Seq2SeqTrainer(
    model=model,
    args=training_args,
    train_dataset=small_train_dataset,
    eval_dataset=small_test_dataset,
    tokenizer=tokenizer,
    data_collator=data_collator,
    compute_metrics=compute_metrics,
)

trainer.train()
```

Testowanie modelu

- przygotowanie zdań do tłumaczenia z prefiksem
- tokenizacja tekstu
- generowanie tłumaczenia
- dekodowanie tłumaczenia

Source Sentence	Reference Translation	Model Translation	BLEU Score
translate English to French: The cat is sleeping.	Le chat dort.	Le chat dorme.	35.36
translate English to French: The sky is blue.	Le ciel est bleu.	Le ciel est bleu.	100.00
translate English to French: I like apples.	J'aime les pommes.	Je m'aime les pommes.	39.76
Average BLEU: 0.5837			

```
from transformers import AutoModelForSeq2SeqLM, AutoTokenizer
from evaluate import load
from prettytable import PrettyTable

model = AutoModelForSeq2SeqLM.from_pretrained("translation_model")
tokenizer = AutoTokenizer.from_pretrained("translation_model")

simple_sentences = [
    "translate English to French: The cat is sleeping.",
    "translate English to French: The sky is blue.",
    "translate English to French: I like apples."
]

simple_references = [
    ["Le chat dort."],
    ["Le ciel est bleu."],
    ["J'aime les pommes."],
]

def translate_and_evaluate(sentences, references, model, tokenizer, metric):
    table = PrettyTable()
    table.field_names = ["Source Sentence", "Reference Translation",
                        "Model Translation", "BLEU Score"]

    predictions = []
    scores = []

    for text, reference in zip(sentences, references):
        inputs = tokenizer(text, return_tensors="pt")
        outputs = model.generate(inputs.input_ids, max_new_tokens=50)
        prediction = tokenizer.decode(outputs[0], skip_special_tokens=True)
        predictions.append(prediction)

        individual_score = metric.compute(predictions=[prediction],
                                         references=[reference])["score"]
        scores.append(individual_score / 100)

        table.add_row([text, reference[0], prediction, f"{individual_score:.2f}"])

    print(table)

    average_bleu = sum(scores) / len(scores) if scores else 0.0

    return predictions, average_bleu

metric = load("sacrebleu")

simple_predictions, simple_average_bleu = translate_and_evaluate(simple_sentences,
                                                                simple_references, model, tokenizer, metric)

print(f"Average BLEU: {simple_average_bleu:.4f}")
```


Bibliografia

- Kod opracowano na podstawie: <https://huggingface.co/docs/transformers/en/tasks/translation>
- <https://summalinguae.com/pl/technologie-jezykowe/5-rodzajow-tlumaczenia-maszynowego/>
- <https://gtelocalize.com/types-of-machine-translation/#3-Neural-Machine-Translation-NMT>
- <https://cameronrwolfe.substack.com/p/t5-text-to-text-transformers-part>
- <https://jsonformatter.curiousconcept.com/>
- <https://www.deepl.com>