BERT

Bidirectional Encoder Representations from Transformers

Architektura i zadania pre-treningu: MLM i NSP

- ✦ Wprowadzenie do architektury Transformer
- ✦ Budowa modelu BERT
- ✦ MLM — Masked Language Modeling
- ✦ NSP — Next Sentence Prediction
- ✦ Przykłady i zastosowania praktyczne

1. Czym jest BERT?

BERT (Bidirectional Encoder Representations from Transformers) to model językowy opracowany przez Google AI w 2018 roku. Jest oparty na architekturze Transformer i stanowi przełom w dziedzinie przetwarzania języka naturalnego (NLP). Kluczową cechą BERT jest jego ***dwukierunkowość*** — model analizuje kontekst słowa zarówno z lewej, jak i z prawej strony jednocześnie.

Kluczowa intuicja

Wyobraź sobie zdanie: "Bank był pełny ludzi." Słowo "bank" może oznaczać instytucję finansową lub brzeg rzeki. BERT rozumie właściwe znaczenie, analizując CAŁY kontekst zdania jednocześnie — nie tylko słowa poprzedzające, jak wcześniejsze modele.

1.1 Kontekst historyczny — dlaczego BERT był przełomem?

Przed BERT-em modele językowe przetwarzały tekst jednokierunkowo:

- GPT (OpenAI) — czytał tekst od lewej do prawej
- ELMo — łączył dwa jednokierunkowe modele, ale nie prawdziwie dwukierunkowo

- Word2Vec — reprezentacje słów bez kontekstu zdaniowego

BERT jako pierwszy zastosował prawdziwie dwukierunkowe kodowanie za pomocą mechanizmu self-attention, co pozwoliło na znacznie lepsze rozumienie kontekstu semantycznego.

1.2 Warianty modelu BERT

Model	Warstwy	Głowice	Parametry
BERT-Base	12	12	110 milionów parametrów
BERT-Large	24	16	340 milionów parametrów

2. Architektura BERT

BERT opiera się na architekturze **Transformer Encoder**. Składa się ze stosu warstw encodera, z których każda zawiera dwa kluczowe mechanizmy: **Multi-Head Self-Attention** oraz **Feed-Forward Network**.

2.1 Embeddingi wejściowe

Każdy token wejściowy jest reprezentowany przez sumę trzech rodzajów embeddingów:

Typ embeddingu	Opis
Token Embedding	Reprezentacja wektora dla każdego tokenu z słownika (WordPiece, ~30k tokenów)
Segment Embedding	Informacja o tym, do której sekwencji (A lub B) należy token — kluczowe dla NSP
Position Embedding	Informacja o pozycji tokenu w sekwencji (BERT uczy się tych embeddingów)

2.2 Specjalne tokeny

BERT używa dwóch specjalnych tokenów, które pełnią kluczowe role:

[CLS]
Token klasyfikacyjny — umieszczany na początku każdej sekwencji. Jego reprezentacja z ostatniej warstwy BERT jest używana do zadań klasyfikacyjnych (np. NSP). Skrót od "Classification".

[SEP]
Token separatora — umieszczany na końcu każdego zdania. Oddziela dwie sekwencje w zadaniu NSP. Skrót od "Separator".

2.3 Mechanizm Self-Attention

Self-Attention to serce architektury Transformer. Dla każdego tokenu obliczana jest **waga uwagi** względem każdego innego tokenu w sekwencji. Pozwala to modelowi "skupić się" na najbardziej istotnych słowach przy kodowaniu znaczenia danego tokenu.

Wzór na self-attention:

$$\text{Attention}(Q, K, V) = \text{softmax}(QK^T / \sqrt{d_k}) \cdot V$$

gdzie: Q = Query, K = Key, V = Value, d_k = wymiarowość klucza

3. MLM — Masked Language Modeling

MLM to pierwsze z dwóch zadań pre-treningu BERT. Jego celem jest nauczenie modelu **przewidywania zamaskowanych tokenów** na podstawie otaczającego kontekstu.

> **Analogia do ćwiczeń językowych**
> MLM przypomina ćwiczenia z uzupełnianiem luk: "Kot siedział na ___.". Model musi odgadnąć brakujące słowo, analizując pełny kontekst zdania — zarówno słowa przed luką, jak i po niej.

3.1 Procedura maskowania

W MLM losowo maskowane jest **15% tokenów** z sekwencji wejściowej. Każdy wybrany token jest następnie traktowany według następującego schematu:

Prawdopodobieństwo	Akcja	Opis
80%	**Zamiana na [MASK]**	Token zostaje zastąpiony specjalnym tokenem [MASK]
10%	**Zamiana losowa**	Token zostaje zastąpiony losowym tokenem ze słownika
10%	**Bez zmian**	Token pozostaje niezmieniony (model "nie wie" czy był wybrany)

Dlaczego taka strategia? Gdyby zawsze używać [MASK], model nauczyłby się ignorować tokeny bez [MASK]. Losowość sprawia, że model musi generować użyteczne reprezentacje dla KAŻDEGO tokenu.

3.2 Przykład działania MLM

Zdanie wejściowe (oryginalne):
"Warszawa jest stolicą Polski i największym miastem w kraju."

Zdanie po maskowaniu (wejście do modelu):
"Warszawa jest [MASK] Polski i największym [MASK] w kraju."

Zadanie modelu (przewidywanie):
Pozycja 3: przewidzieć token "stolicą" Pozycja 7: przewidzieć token "miastem"

3.3 Wizualizacja sekwencji tokenów dla MLM

[CLS]	War-	szawa	jest	[MASK]	Pol-	ski	[MASK]	[SEP]
CLS	tok	tok	tok	MASK	tok	tok	MASK	SEP

Tokeny niebieskie = specjalne | Tokeny pomarańczowe = zamaskowane | Białe = normalne

3.4 Po co MLM? — Znaczenie dla reprezentacji

MLM zmusza BERT do budowania głębokiego rozumienia języka, ponieważ:

- Model musi rozumieć semantykę i składnię zdania, a nie tylko statystyczne współwystępowania
- Dwukierunkowość kontekstu — model widzi CAŁE zdanie (poza zamaskowanym tokenem)
- Uczy się reprezentacji kontekstowych — to samo słowo ma różne reprezentacje w różnych kontekstach

Ważna uwaga

MLM powoduje rozbieżność między pre-treningiem a fine-tuningiem — token [MASK] nigdy nie pojawia się w rzeczywistych tekstach. Jest to znana wada BERT, którą rozwiązują późniejsze modele (np. XLNet, RoBERTa).

4. NSP — Next Sentence Prediction

NSP to drugie zadanie pre-treningu BERT. Model uczy się **przewidywać, czy dwa zdania następują po sobie w oryginalnym tekście**. Zadanie to ma na celu nauczenie modelu rozumienia relacji między zdaniami.

> **Dlaczego NSP jest ważne?**
> Wiele zadań NLP wymaga rozumienia relacji między zdaniami: odpowiadanie na pytania (pytanie + kontekst), wnioskowanie (przesłanka + hipoteza), spójność tekstów. MLM sam w sobie nie uczy tych relacji — do tego służy NSP.

4.1 Jak działa NSP?

Podczas pre-treningu BERT otrzymuje pary zdań jako wejście. Połowa par pochodzi z rzeczywistych, kolejnych zdań (etykieta: IsNext), a połowa to losowo dobrane pary (etykieta: NotNext):

Etykieta	Przykład pary zdań	Wynik NSP
IsNext	Zdanie A: "Pies biegał po parku." Zdanie B: "Nagle zatrzymał się przy fontannie."	**IsNext ✓**
NotNext	Zdanie A: "Pies biegał po parku." Zdanie B: "Gospodarka Polski rośnie w szybkim tempie."	**NotNext ✗**

4.2 Format wejścia dla NSP

Sekwencja wejściowa dla NSP ma zawsze następującą strukturę:

[CLS]	Zdanie A	[SEP]	Zdanie B	[SEP]
Segment A	*Segment A*	*Segment B*	*Segment B*	

Token [CLS] pełni kluczową rolę — jego reprezentacja z ostatniej warstwy BERT jest przekazywana do klasyfikatora binarnego (IsNext / NotNext). Jest to prosta warstwa liniowa z funkcją softmax.

4.3 Wizualizacja przepływu danych w NSP

[CLS]	Pies	biegał	parku	[SEP]	Zatrzymał	się	przy	fontannie	[SEP]
Seg CLS	*Seg A*	*Seg A*	*Seg A*	*Seg SEP*	*Seg B*	*Seg B*	*Seg B*	*Seg B*	*Seg SEP*

Niebieski = [CLS]/[SEP] | Jasnoniebieski = Segment A | Zielony = Segment B

4.4 Kontrowersje wokół NSP

Późniejsze badania (RoBERTa, 2019) wykazały, że NSP może być **mniej efektywne** niż pierwotnie sądzono. Model RoBERTa pominął NSP w pre-treningu i osiągnął lepsze wyniki na wielu benchmarkach. Możliwe przyczyny:

- Zadanie IsNext/NotNext jest zbyt łatwe dla modelu
- NSP może wprowadzać mylące sygnały treningowe
- Dłuższe sekwencje bez NSP dają lepsze wyniki

Wniosek

NSP było ważnym krokiem naprzód w 2018 roku i pomogło osiągnąć SOTA na wielu zadaniach wymagających rozumienia relacji zdaniowych (np. Question Answering, NLI). Jednak nowsze badania sugerują, że nie jest niezbędne — co pokazuje, jak dynamicznie rozwija się dziedzina NLP.

5. Połączenie MLM i NSP — Pre-trening BERT

BERT jest trenowany jednocześnie na obu zadaniach. Funkcja straty jest sumą strat obu zadań:

$$L_{całkowita} = L_{MLM} + L_{NSP}$$

Taki wielozadaniowy pre-trening pozwala modelowi nauczyć się zarówno rozumienia słów w kontekście (MLM), jak i relacji między zdaniami (NSP).

5.1 Dane i zasoby treningowe

- BooksCorpus: 800 milionów słów (książki w języku angielskim)
- English Wikipedia: 2,5 miliarda słów (tylko tekst, bez tabel i list)
- Łączny rozmiar danych: ok. 3,3 mld słów
- Czas treningu BERT-Large: 4 dni na 64 TPU v3

5.2 Fine-tuning na zadaniach docelowych

Po pre-treningu BERT może być dostosowany (fine-tuned) do konkretnych zadań NLP przez dodanie prostej warstwy wyjściowej:

Zadanie NLP	Użyte tokeny	Przykład
Klasyfikacja tekstu	Reprezentacja [CLS]	*Analiza sentymentu, wykrywanie spamu*
NER (Named Entity Recognition)	Reprezentacje tokenów	*Rozpoznawanie nazw: osób, miejsc*
Question Answering	[CLS] + tokeny	*SQuAD, TriviaQA*
Natural Language Inference	Reprezentacja [CLS]	*Wnioskowanie: sprzeczność/neutralny/wynika*

6. Podsumowanie — Kluczowe koncepcje

MLM — Masked Language Modeling	NSP — Next Sentence Prediction
✦ Maskowanie 15% tokenów ✦ 80/10/10 strategia masek ✦ Przewidywanie zamaskowanych tokenów ✦ Uczy rozumienia słów w kontekście	✦ Pary zdań IsNext / NotNext (50/50) ✦ Klasyfikacja na podstawie [CLS] ✦ Tokeny segmentu A/B ✦ Uczy relacji między zdaniami

Pytania kontrolne

1. Dlaczego BERT maskuje 15% tokenów, a nie 50%? Co by się stało przy zbyt dużej proporcji masek?

2. Jaką rolę pełni token [CLS] i dlaczego jest umieszczany na początku sekwencji?

3. Wyjaśnij różnicę między embedding pozycyjnym w BERT a w oryginalnym Transformerze.

4. Model RoBERTa pominął NSP. Jakie to ma implikacje dla naszego rozumienia pre-treningu modeli językowych?

5. Podaj przykład zadania NLP, do którego NSP pomaga bardziej niż MLM, i uzasadnij swoją odpowiedź.