Rodzaje • Mechanizmy • Zastosowania

Osadzenia słów i reprezentacje wektorowe

1. Wprowadzenie do reprezentacji słów

Modele przetwarzania języka naturalnego (NLP) operują nie na literach, lecz na liczbach. Aby komputer mógł przetwarzać tekst, każde słowo musi zostać przekształcone w uporządkowaną reprezentację matematyczną. Sposób, w jaki zapisujemy znaczenie słów, ma fundamentalne znaczenie dla jakości niemal każdego zadania NLP — od klasyfikacji tekstu, przez tłumaczenie maszynowe, aż po duże modele językowe (LLM).

W ciągu ostatnich dekad ewoluowaliśmy od prostych, symbolicznych kodowań (one-hot encoding), przez liczenie współwystępowań słów, aż po gęste, uczone reprezentacje zwane osadzeniami (ang. embeddings). Celem tego ćwiczenia jest wyjaśnienie pojęcia osadzania słów, jego genezy, działania oraz znaczenia we współczesnych systemach opartych na transformerach.

1.1. Dlaczego reprezentacja słów jest kluczowa?

- Komputery nie rozumieją języka — wykonują operacje arytmetyczne na wektorach.
- Dobra reprezentacja powinna zachowywać podobieństwo semantyczne: słowa o zbliżonym znaczeniu powinny znajdować się blisko siebie w przestrzeni wektorowej.
- Jakość reprezentacji determinuje, ile „wiedzy” model może wydobyć z danych treningowych.

Uwaga wprowadzająca

> *Klasyczny dylemat NLP: jak zakodować znaczenie słowa tak, aby były widoczne relacje między słowami (synonimia, antonimia, kategorie, powiązania)? Ta sama esencja stała u podstaw rozwoju od one-hot encoding, przez word2vec, aż po kontekstowe osadzenia transformerów.*

2. One-hot encoding — punkt wyjścia i jego ograniczenia

Najprostszym sposobem reprezentacji słowa jest wektor one-hot („jeden aktywny bit”). Jeśli słownik zawiera n słów, każdemu słowu przypisujemy wektor długości n, w którym jedynka znajduje się tylko na pozycji odpowiadającej temu słowu.

```
# Słownik V=5 słów
slowa = ['cat','dog','fish','apple','run']   # indeksy: cat=0, dog=1 ...
n_cat   = [1,0,0,0,0]
n_dog   = [0,1,0,0,0]
# kosinus pomiędzy n_cat a n_dog = 0 (ortogonalne)
```

Problem: wszystkie wektory są wzajemnie ortogonalne (kąt 90°), więc iloczyn skalarny dowolnych dwóch różnych słów wynosi zawsze 0. Model nie ma żadnej informacji o tym, że „cat” i „dog” są zwierzętami, a „apple” nie jest. Wektory zajmują też dużo pamięci (rozmiar słownika) i nie niosą żadnej wiedzy o znaczeniu.

3. Hipoteza dystrybucyjna — fundament nowoczesnych osadzeń

Przełomem była obserwacja znana jako hipoteza dystrybucyjna, sformułowana w XX wieku przez językoznawców, m.in. Johna Firtha:

Hipoteza dystrybucyjna (J.R. Firth, 1957)

> *"You shall know a word by the company it keeps." — Znasz słowo po towarzystwie, w jakim bywa. Znaczenie słowa można wyprowadzić z kontekstu, w którym się pojawia.*

Innymi słowy: słowa, które występują w podobnych kontekstach, mają podobne znaczenie. To założenie pozwala uczyć reprezentacji w sposób czysto statystyczny — bez ręcznej adnotacji znaczeń.

3.1. Matryce współwystępowania (co-occurrence)

Pierwsze podejście praktyczne polegało na zbudowaniu macierzy współwystępowań: dla całego korpusu zliczano, ile razy każde słowo pojawiło się w oknie o zadanej wielkości

wokół każdego innego słowa. Powstałe wektory kolumn opisują „profil kontekstowy” słowa.

- Rozmiar okna (np. ±2 lub ±5 tokenów) wpływa na rodzaj uchwyconej informacji: małe okno → relacje syntaktyczno-semantyczne, duże okno → tematyka/dziedzina.
- Suche liczby mocno dominują słowa częste (typu „i”, „to”), dlatego stosuje się ich transformację PMI (Pointwise Mutual Information), najczęściej dodatnią PPMI (Pointwise Mutual Information mierzy, ile razy częściej dwa słowa występują razem, niż można by oczekiwać, gdyby występowały całkowicie niezależnie).
  - PMI > 0 → słowa współwystępują częściej, niż wynikałoby z przypadku (silna relacja),
  - PMI ≈ 0 → współwystępują zgodnie z przypadkiem,
  - PMI < 0 → współwystępują rzadziej niż by losowo wypadało (co zwykle nie jest użyteczne informacyjnie).
- Wejście stanowi miara siły związku między parą słów, a nie surowe liczebności.

One-hot	Brak wiedzy o znaczeniu; ortogonalne wektory	Punkt wyjścia, teoretycznie
Co-occurrence (PPMI)	Konto profili kontekstowych; rzadkie, duże wektory	Analiza semantyczna korpusów
Osadzanie (word2vec)	Gęsty, krótki wektor uczeniu się z predykcji kontekstu	Nowoczesne modele NLP

4. Word2vec — CBOW i Skip-gram

W 2013 roku grupa Tomasa Mikolova (Google) przedstawiła technikę word2vec, która zastąpiła kosztowne analizy macierzowe szybkim trenowaniem sieci neuronowych. Pomysł: zamiast jawnie budować ogromną macierz, trenujemy prostą sieć neuronową, która przewiduje słowa w kontekście i „przy okazji” odkrywamy gęste wektory słów (osadzenia).

4.1. Architektura CBOW

CBOW (Continuous Bag-of-Words) przewiduje słowo docelowe na podstawie jego otoczenia. Sieć dostaje średnią wektorów słów kontekstu (np. 2 słowa z lewej i 2 z prawej) i próbuje wskazać poprawne słowo środkowe.

```
kontekst: "kot ___ płot"   →   cel: "siedzi"
CBOW:  f( kot, płot ) = p(siedzi)
```

4.2. Architektura Skip-gram

Skip-gram działa odwrotnie: podaje jedno słowo, a uczy się przewidywać słowa z jego kontekstu. Lepiej radzi sobie ze słowami rzadkimi, choć wymaga więcej danych.

```
słowo: "siedzi"   →   predykcja: {kot, płot}
Skip-gram:  f(siedzi) = p(kot), p(płot)
```

4.3. Uczenie efektywne: Negative Sampling i Hierarchical Softmax

Podczas treningu sieć musi wskazać poprawne słowo spośród wszystkich wyrazów słownika: jej ostatnia warstwa zwraca rozkład prawdopodobieństwa na cały słownik przy użyciu funkcji miękkiego maksimum (softmax), która zamienia surowe wyniki na prawdopodobieństwa sumujące się do jednego.

Pełne obliczenie miękkiego maksimum nad całym słownikiem (setki tysięcy klas) byłoby bardzo kosztowne. Stosuje się więc przybliżenia:

- **Negative Sampling (NEG) —** zamiast aktualizować wszystkie wektory, losujemy kilka tzw. negatywnych przykładów (słów spoza kontekstu) i uczymy się odróżniać poprawne konteksty od błędnych.
- **Hierarchical Softmax —** słownik organizowany jest w drzewo binarne (kopiec Huffmana), dzięki czemu koszt spada z O(V) do O(log V).

Wniosek

> *Esenja word2vec nie tkwi w samej architekturze, lecz w sposobie uczenia: sieć, zmuszana do predykcji kontekstu, kompresuje znaczenie do niewielkiej liczby współczynników wektora — stąd powstają gęste osadzenia.*

5. Algebra wektorów — fascynujący efekt uboczny

Osadzenia word2vec wykazują zdumiewającą własność: relacje semantyczne i syntaktyczne wyrażają się jako proste operacje arytmetyczne na wektorach. Klasyczny przykład:

Przykład analogii (king – man + woman ≈ queen)

> *wektor(„król”) − wektor(„mężczyzna”) + wektor(„kobieta”) ≈ wektor(„królowa”)*

Analogicznie: „Madryt” − „Hiszpania” + „Francja” ≈ „Paryż”. Podstawą tych zachowań jest regularność, z jaką słowa pojawiają się w kontekstach. Dzięki temu osadzenia umożliwiają:

- znajdowanie synonimów i powiązanych pojęć (podobieństwo kosinusowe, cos-similarity),
- rozwiązywanie testów analogii (np. benchmark WS353, analogy-reasoning tasks),
- klasteryzację dokumentów i tematów,
- wizualizację przestrzeni semantycznej (np. PCA / t-SNE).

6. GloVe i FastText — rozwinięcia idei

6.1. GloVe (Global Vectors)

GloVe (Pennington i in., 2014, Stanford) łączy podejście lokalne (predykcja kontekstu jak w word2vec) z globalną statystyką korpusu (macierz współwystępowań). Minimalizuje różnicę między iloczynem skalarnym osadzeń a logarytmem liczby współwystąpień. Dzięki temu dobrze oddaje zarówno relacje semantyczne, jak i probabilistyczne własności słów.

6.2. FastText

FastText (Bojanowski, 2016, Facebook) reprezentuje słowo jako sumę osadzeń jego znakowych n-gramów (subwords). Dzięki temu potrafi radzić sobie ze słowami spoza słownika (out-of-vocabulary) i lepiej obsługuje języki fleksyjne (takie jak polski), gdzie istnieje wiele form odmiany jednego słowa.

7. Zalety, ograniczenia i dziedzictwo osadzeń statycznych

Cecha	Osadzenia statyczne	Osadzenia kontekstowe (transformer)
Znaczenie słowa	Jedno znaczenie niezależnie od kontekstu	Znaczenie zależne od otoczenia
Polisemia („bank”)	Słabo — miesza znaczenia	Dobrze — różne znaczenia osobno
Kontekst	Nie uwzględnia zdania	Uwzględnia całe zdanie/zdania
Reprezentatywne modele	word2vec, GloVe, FastText	BERT, ELMo, GPT

Głównym ograniczeniem osadzeń statycznych jest ich „sztywność”: słowo otrzymuje jeden wektor, nawet jeśli ma kilka znaczeń. Te ograniczenia doprowadziły do powstania osadzeń kontekstowych (ELMo, a następnie BERT i inne transformery), które tworzą wektor słowa dopiero w kontekście całego zdania. Mimo to mechanizmy takie jak warstwa Embedding, podobieństwo kosinusowe czy idea algebraicznych relacji przeniknęły do współczesnych architektur i pozostają aktualne.

8. Podsumowanie

- Słowa muszą być reprezentowane liczbowo; dobór reprezentacji decyduje o możliwościach modelu.
- Hipoteza dystrybucyjna uzasadnia uczenie znaczenia z kontekstu.
- Word2vec (CBOW/Skip-gram), GloVe i FastText pozwalają uzyskać gęste osadzenia, w których podobne słowa są blisko siebie.
- Operacje arytmetyczne na wektorach odzwierciedlają relacje semantyczne (analogie).
- Osadzenia statyczne dały początek kontekstowym reprezentacjom, które stanowią dziś fundament LLM.

Myśl przewodnia

> *Zrozumienie osadzeń słów to klucz do zrozumienia, skąd biorą się reprezentacje w dużych modelach językowych — bo transformer „widzi" świat poprzez wektory, których nauczony został na danych.*