

BlockingPy: przybliżone najbliższe sąsiedztwo do blokowania rekordów na potrzeby deduplikacji i integracji danych

lic. Tymoteusz Strojny
dr Maciej Beręsewicz, prof. UEP

Ośrodek Metodologii Badań Ludnościowych
Urząd Statystyczny w Poznaniu

Warszawa, 02.07.2025



Plan Prezentacji

- ▶ Wstęp - Integracja Danych
- ▶ Blokowanie
- ▶ Nowe podejście z ANN
- ▶ Pakiet BlockingPy i Opis Procedury
- ▶ Przykład Użycia
- ▶ Zastosowania
- ▶ Podsumowanie
- ▶ Dodatkowe Informacje

- ▶ Łączenie rekordów opisujących tę samą jednostkę bez wspólnego identyfikatora (ID, PESEL).
- ▶ Potrzeba porównania każdej możliwej pary rekordów.
- ▶ Problem: Liczba porównań jest zbyt duża.
- ▶ Rozwiązanie: **blokowanie** – redukcja liczby porównań poprzez grupowanie.

Tradycyjne metody blokowania

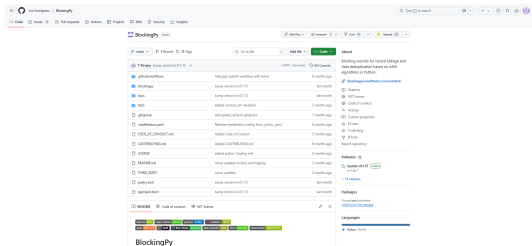
- ▶ **Reguły deterministyczne:**
np. ta sama data urodzenia i pierwsza litera nazwiska.
- ▶ **Algorytmy fonetyczne:** Soundex, Metaphone.
- ▶ **Sorted neighbourhood:** grupy na posortowanych danych.
- ▶ **Ograniczenia:**
 - wrażliwość na błędy, transliterację, braki danych
 - zbyt duże grupy lub brak dopasowań

Algorytmy przybliżonych najbliższych sąsiadów (ANN)

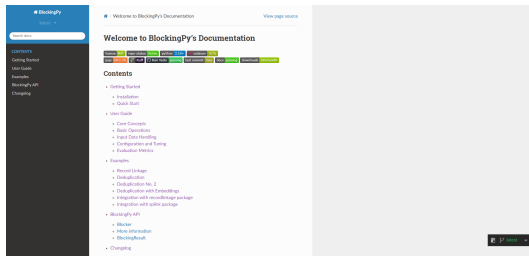
- ▶ Algorytmy znajdujące sąsiadów na podstawie podobieństwa.
- ▶ Metryki (np. Euclidean, Cosine, IP).
- ▶ Działają w sub-kwadratowej złożoności obliczeniowej.

Pakiet BlockingPy

Opracowano pakiet w języku Python na licencji MIT umożliwiający blokowanie za pomocą algorytmów ANN.



Rys.: GitHub



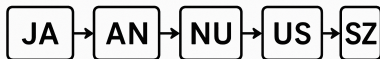
Rys.: Dokumentacja

Opis zaproponowanej procedury

- ▶ Transformacja danych wejściowych
- ▶ Wyszukiwanie najbliższego sąsiada
- ▶ Wykorzystanie grafu
- ▶ Utworzenie bloków
- ▶ Opcjonalnie – konfiguracja parametrów
- ▶ Opcjonalnie – ewaluacja wyników

Transformacja danych wejściowych - n-gram i embedding

JANUSZ



Rys.: Przykład n-gramu

MinishLab/
model2vec



Fast State-of-the-Art Static Embeddings

10
Contributors

78
Used by

8
Discussions

2k
Stars

92
Forks



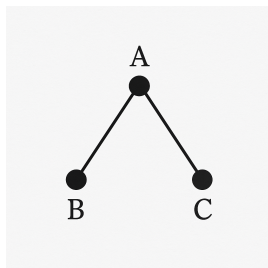
Rys.: Embedding z model2vec

Algorytmy ANN

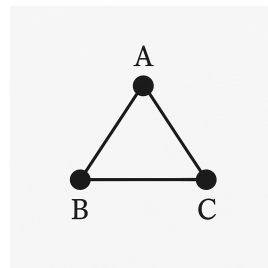
Dostępne algorytmy:

- ▶ **HNSW** (faiss, hnswlib, voyager)
- ▶ **LSH** (faiss, mlpack)
- ▶ **NN-descent** (pynndescent)
- ▶ **Annoy** (Spotify)
- ▶ **k-NN** (faiss, mlpack)

Utworzenie grafu



Rys.: Wyniki z ANN



Rys.: Uzupełnienie brakujących połączeń

Unikalne komponenty grafu $\Rightarrow$ bloki

Przykład kodu

```
from blockingpy import Blocker
import pandas as pd

df = pd.read_csv("...")

df["txt"] = df["imie"] + df["nazwisko"] + df["data_ur"]

blocker = Blocker()

blocks = blocker.block(
    x=df["txt"],
    ann="voyager",
    random_seed=2025
)
```

Informacja zwrotna

```
print(blocks)
# =====
# Blocking based on the voyager method.
# Number of blocks: 2723
# Number of columns created for blocking:
#   1169
# Reduction ratio: 0.999564
# =====
# Distribution of the size of the blocks:
# Block Size | Number of Blocks
#           2 | 926
#           3 | 883
#           4 | 363
#           5 | 211
```

```
print(blocks.result)
```

#	x	y	block	dist
# 0	1	0	0	0.102041
# 1	5974	2	1	0.390295
# 2	7378	3	2	0.425410
# 3	5562	4	3	0.396494
# 4	1389	5	4	0.461184

- ▶ Pełność (Recall)
- ▶ Precyzja (Precision)
- ▶ Wskaźnik fałszywych negatywów (False Negative Rate, FNR)
- ▶ Wskaźnik fałszywych pozytywów (False Positive Rate, FPR)
- ▶ Miara F1 (F1 Score)
- ▶ Macierz pomyłek (Confusion Matrix)

Przykład użycia - Dane

Wygenerowane dane personalne, 10 tyś. rekordów (4 tyś. duplikatów)

#	first_name	second_name	last_name	region	birth_date	personal_id
0	MAJA	OLGA	LEWICKA	ZACHODNIOPOMORSKIE	22/10/1935	DKK423341
1	POLA	LEOKADIA	RUTKOWSKA	ZACHODNIOPOMORSKIE	29/11/1956	LJL907920
2	ALEKSANDRA	RYBAK	ZÓFIA	KUJAWSKO-POMORSKIE	–	DAT77p499
3	ALEKSANDRA	RYBAK	ZÓFIA	KUJAWSKO-POMORSKIE	–	–
4	ZOFIA	ALEKSANDRA	RYBAK	KUJAWSKO-POMORSKIE	24/03/1982	DAT770499
5	LAÓRA	JAGYEŁŁO	JOANNA	WIELKOPOLSKIE	10/11/1984	LNRt57399
6	LAURA	JOANNA	JAGIEŁŁO	WIELKOPOLSKIE	10/11/1984	LNR657399
7	MARIA	KOZA	WIKTÓRIA	DOLNOŚLĄSKIE	–	HEH671979
8	WIKTORIA	MARIA	KOZA	DOLNOŚLĄSKIE	09/09/1982	HEH671989
9	NIKOLA	BRONISŁAWA	WIĘCKOWSKA	ŚLĄSKIE	09/11/1992	JKR103426

Tab.: Przykład danych

Przykład użycia - Instalacja & Import

```
!pip install blockingpy  
  
from blockingpy import Blocker  
import pandas as pd  
  
data = pd.read_csv('dane_10000.csv')
```

Przykład użycia - Łączenie pól

```
data['txt'] = (  
    data['first_name'].fillna('') +  
    data['second_name'].fillna('') +  
    data['last_name'].fillna('') +  
    data['region'].fillna('') +  
    data['birth_date'].fillna('') +  
    data['personal_id'].fillna('')  
)
```


Przykład użycia - Blokowanie

```
control_ann = {  
    "hnsw": {  
        'distance': "cosine",  
        'M': 40,  
        'ef_c': 500,  
        'ef_s': 500  
    }  
}  
  
blocker = Blocker()  
eval_result = blocker.block(  
    x=df_eval['txt'],  
    ann='hnsw',  
    true_blocks=true_blocks_dedup,  
    verbose=1,  
    control_ann=control_ann,  
    random_seed=42)
```

Przykład użycia - Informacja zwrotna

```
print(eval_result)
#=====
# Blocking based on the hnsw method.
# Number of blocks: 2972
# Number of columns created for blocking:
#     1169
# Reduction ratio: 0.999649
# =====
# Distribution of the size of the blocks:
# Block Size | Number of Blocks
#           2 | 1113
#           3 | 997
#           4 | 391
#           5 | 200
#           6 | 88
#           7 | 65
#           8 | 39
```

```
print(eval_result.metrics)
# Evaluation metrics (standard):
# recall : 99.0151
# precision : 29.2353
# fpr : 0.0248
# fnr : 0.9849
# accuracy : 99.9751
# specificity : 99.9752
# f1_score : 45.142
```

- ▶ Oszacowanie liczby osób przekraczających granicę PL-UA po 24.02.2022.
- ▶ Statystyka publiczna, Big Data, Medycyna, E-Commerce itd.
- ▶ Dane tekstowe z błędami, brakami, różnymi transliteracjami

Wersja w języku R

The screenshot shows the GitHub repository page for 'ncn-foreigners / blocking'. The repository is public and has 2 watchers, 0 forks, and 13 stars. The main branch is 'main'. The repository contains several files and folders, including .github, R, data, inst, man, tests, vignettes, .Rbuildignore, .gitignore, and DESCRIPTION. The 'About' section describes the package as an R package for blocking records for record linkage / data deduplication based on approximate nearest neighbours algorithms. It also provides links to the repository's README, activity, custom properties, and a report repository.

File/Folder	Description	Last Commit
.github	upload changes to test-coverage	last month
R	tests and docs	2 weeks ago
data	RLdata500	3 weeks ago
inst	tests and docs	2 weeks ago
man	tests and docs	2 weeks ago
tests	local changes	2 weeks ago
vignettes	changes for submission	2 weeks ago
.Rbuildignore	build ignore	2 months ago
.gitignore	changes to data, vignette etc	last month
DESCRIPTION	changes before minor update	last week

Rys.: Pakiet blocking w języku R

Źródło: <https://github.com/ncn-foreigners/blocking> (M.Beręsewicz, A.Struzik)

Podsumowanie

- ▶ Tradycyjne metody blokowania są niewystarczające.
- ▶ Pakiet BlockingPy wykorzystuje ANN i grafy do skutecznego blokowania.
- ▶ Jest dostępny do użytku poprzez PyPI.
- ▶ Planowane są rozszerzenia funkcjonalności.

- ▶ W ramach działań nad pakietem powstał artykuł szczegółowo opisujący funkcjonalność i architekturę pakietu.

Strojny, T., & Beręsewicz, M. (2025). BlockingPy: approximate nearest neighbours for blocking of records for entity resolution. arXiv preprint arXiv:2504.04266.

- ▶ Prace nad artykułem oraz pakietem zostały sfinansowane przez Narodowe Centrum Nauki w ramach grantu OPUS 20 nr 2020/39/B/HS4/00941.
- ▶ Na podstawie pakietu i artykułu powstała praca licencjacka.

Dziękuję za uwagę

<https://poznan.stat.gov.pl/osrodki/osrodek-metodologii-badan-ludnosciowych/>