DNS2Vec: learning representations from DNS data

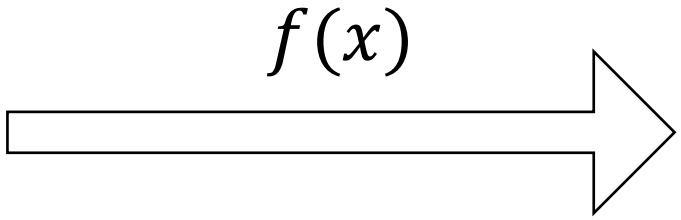
Thymen Wabeke

13 October 2023 | CENTR Tech/R&D @ Paris

Machine learning and representations

- ML methods extract rules from data that can transform an input to an output
- ML methods require a representation of the input, which is an informative description of the concept in the context of the task

Input (x)
example.nl
securepayment.nl
securepay.nl
moresecurepay.nl



Output (y)
Not reported
Reported
Reported
…

How to come to an input representation?

Photos by Lucas Calloch and Tyler Nix on Unsplash

Project goal

How to generate meaningful representations of DNS concepts from query data using representation learning techniques?

Can we use the learned representations in downstream machine learning tasks, for example, clustering and classification?

Word2vec: learn word representations (1/2)

- Developed for text, but also applied to songs (Spotify) and hotels (Airbnb)
- Goal: similar words have similar representations

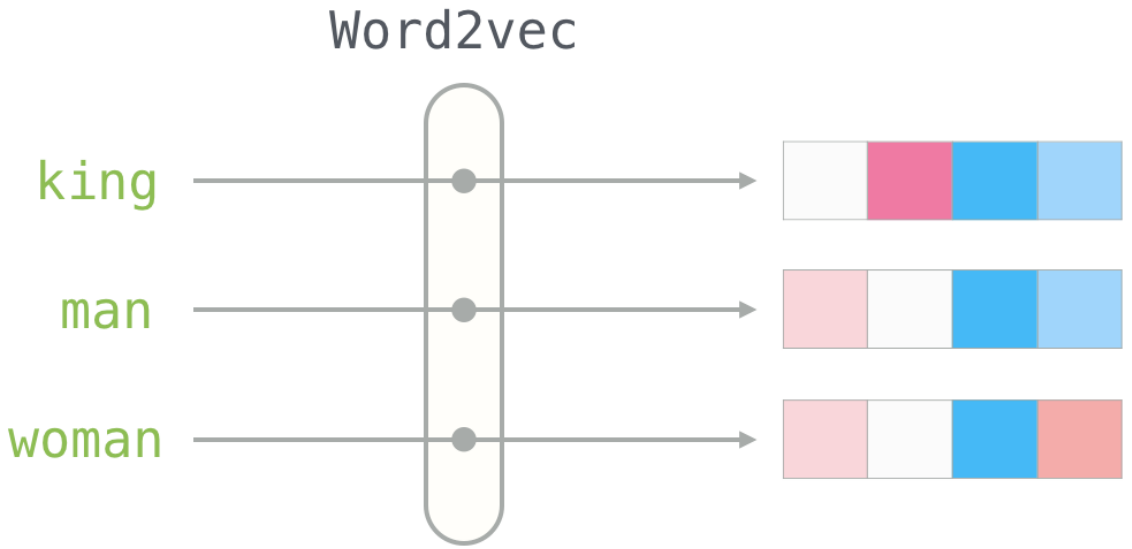
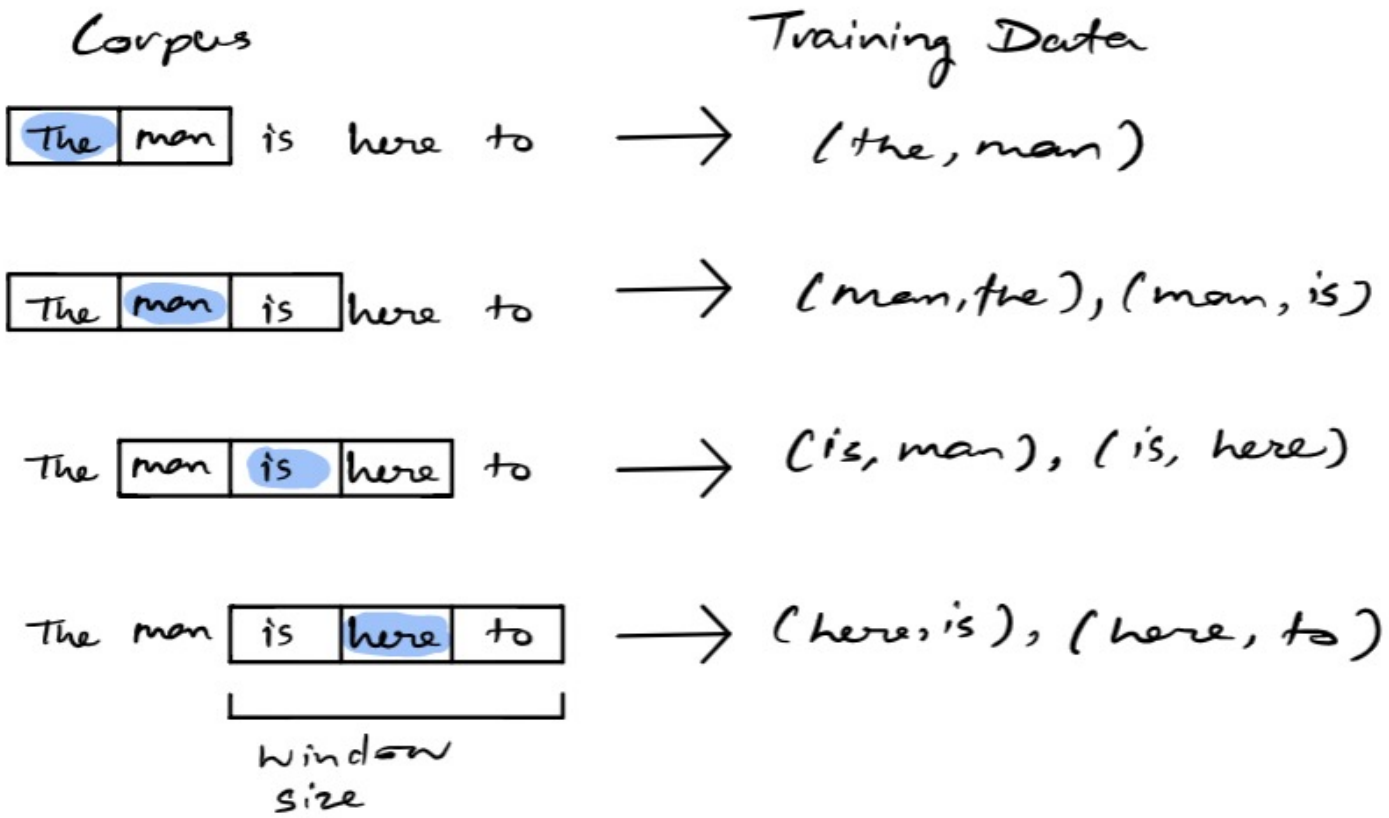


Photo by Jay Alammar on https://jalammar.github.io/illustrated-word2vec/

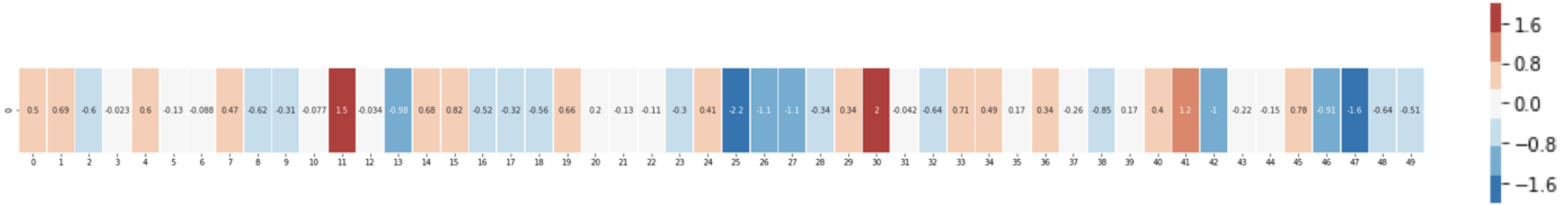
Word2vec: learn word representations (2/2)

- Assumption: similar words appear in similar contexts
- Task: train a model that predicts the surrounding words of a target word



Example of word representations

“King”: `[ 0.50451 , 0.68607 , -0.59517 , … ,-1.6106 , -0.64426 , -0.51042 ]`



♛ - ♂ + ♀ = ♛

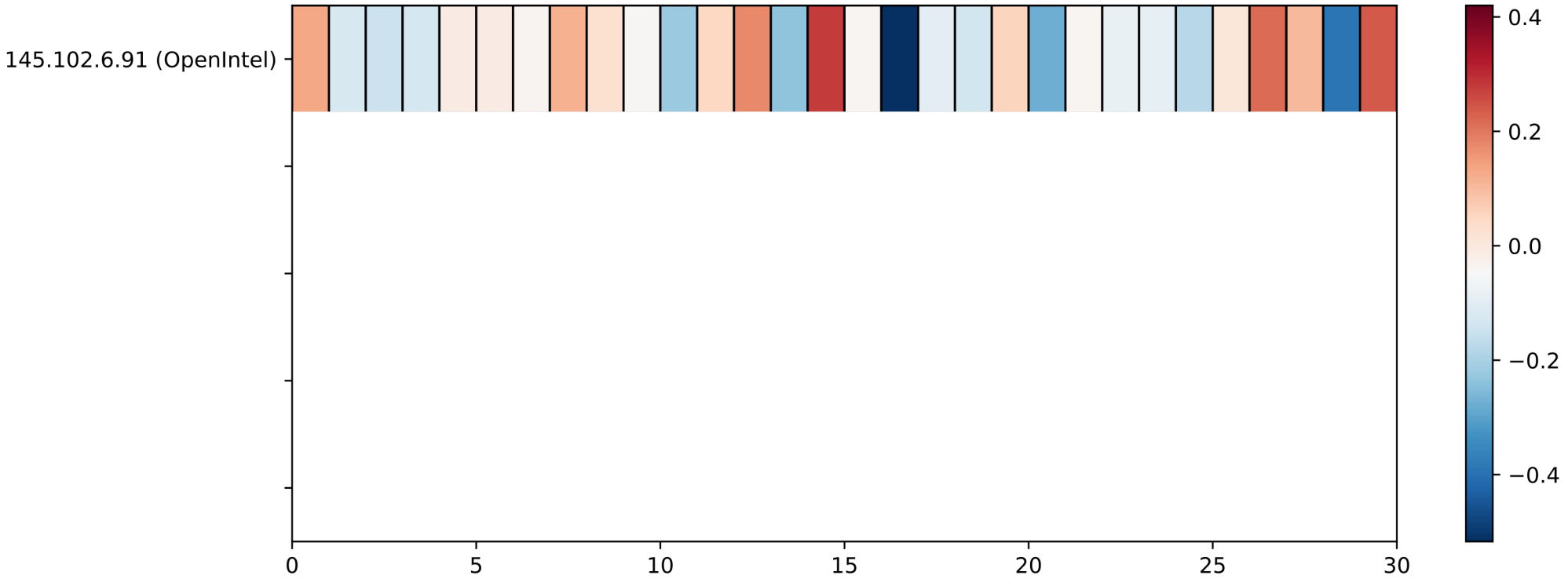
Photo by Jay Alammar on https://jalammar.github.io/illustrated-word2vec/

Mapping DNS data to textual data

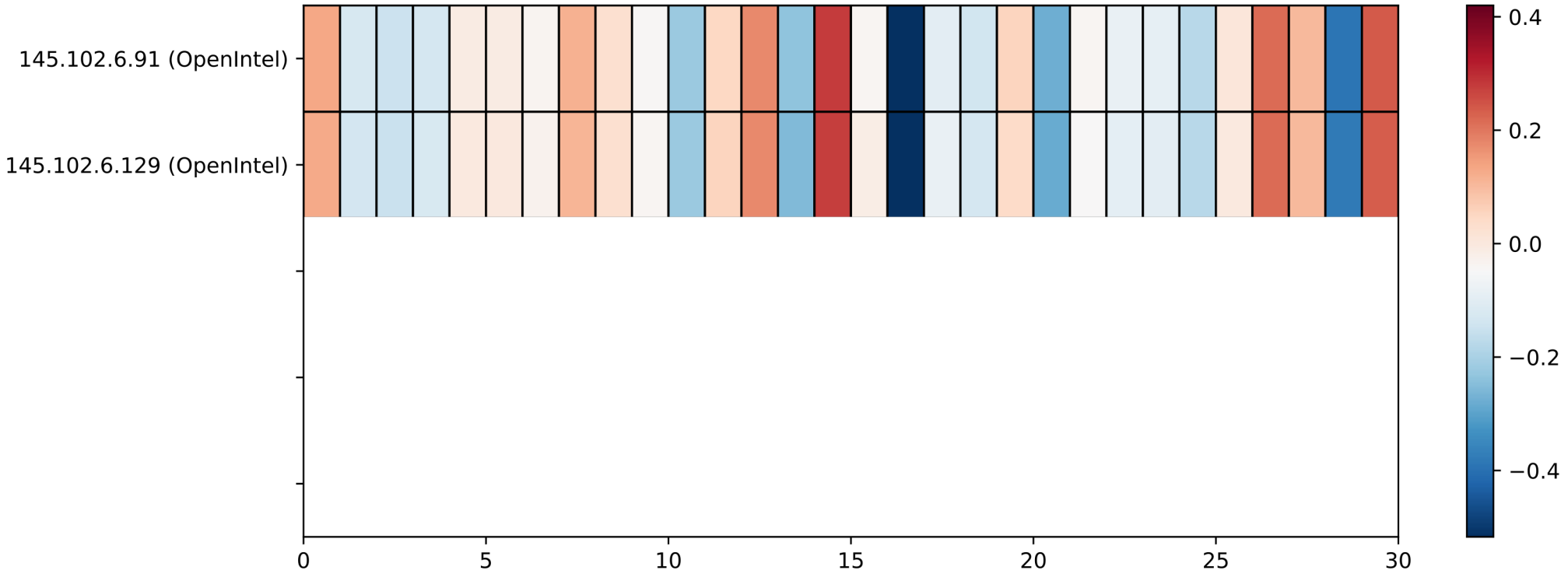
- Assumption: similar resolvers query similar domain names
- Task: train a model that predicts the "surrounding" resolvers of a target resolver

Document	Words
example.nl	['103.22.200.80', '146.190.240.16', '157.90.17.161, …]
sidn.nl	['23.132.96.222', '23.132.96.222', '2400:cb00:386:1024:0:0:ac46:6dad', …]
internet.nl	['145.102.6.55', '146.70.88.198', '157.245.154.205', …]

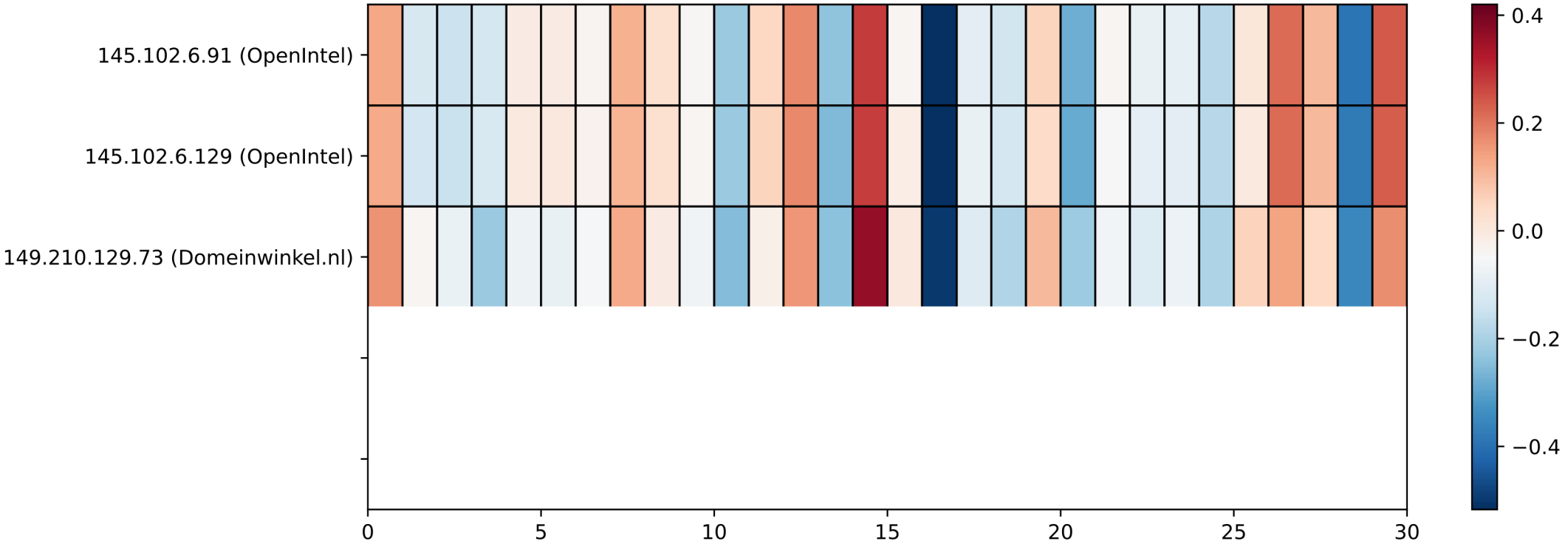
Resolver representations (1/4)



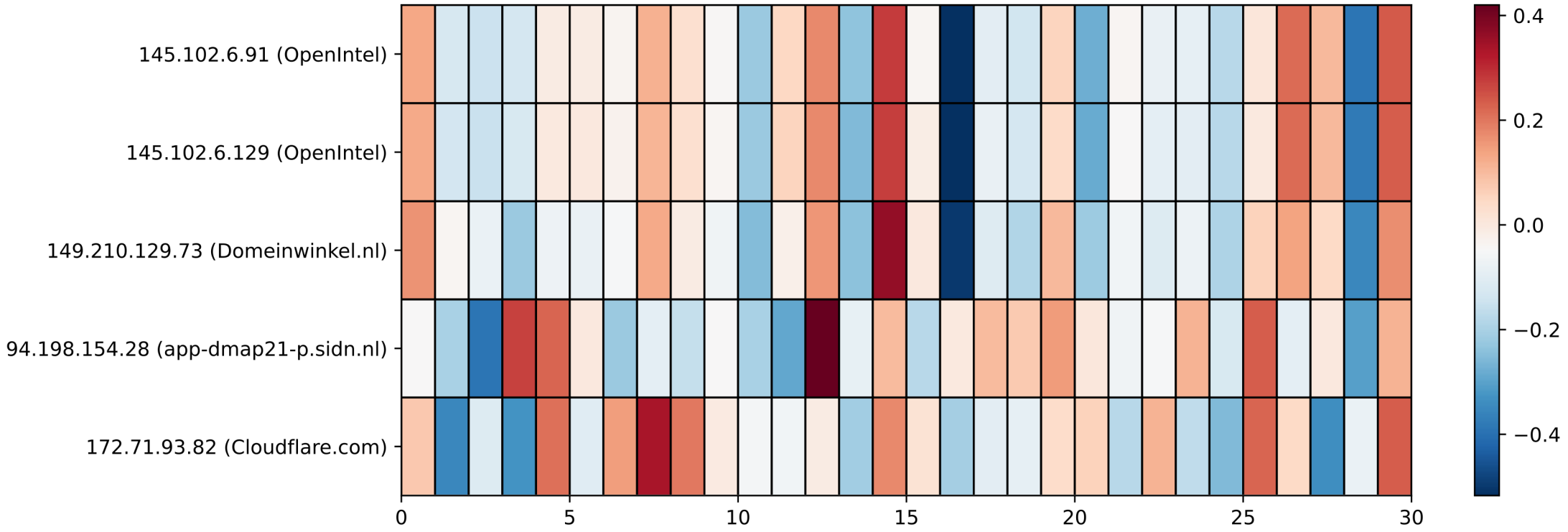
Resolver representations (2/4)



Resolver representations (3/4)



Resolver representations (4/4)



Project goal

How to generate meaningful representations of DNS concepts from query data using representation learning techniques?

Can we use the learned representations in downstream machine learning tasks, for example, clustering and classification?

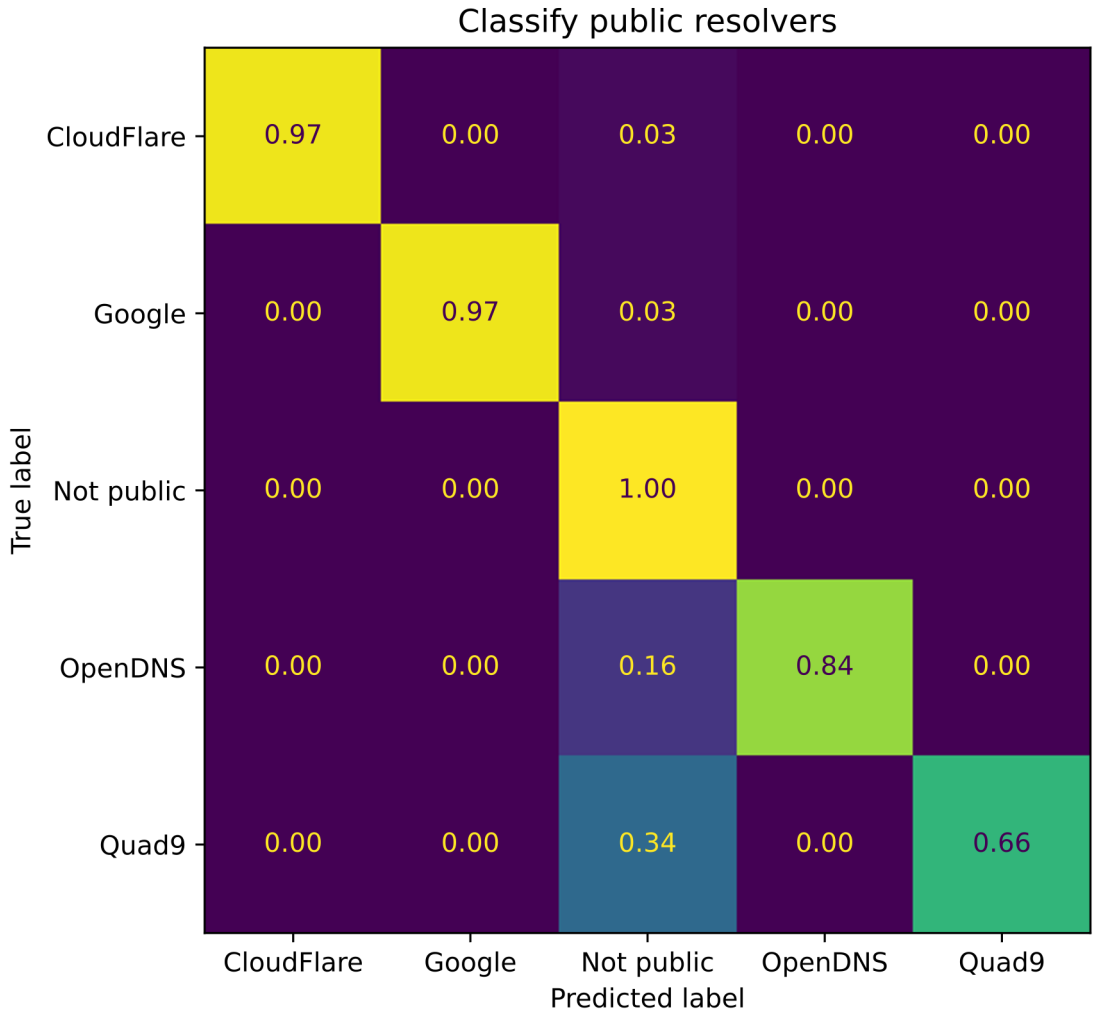
SIDN LABS

Use case: Find more ISP resolvers

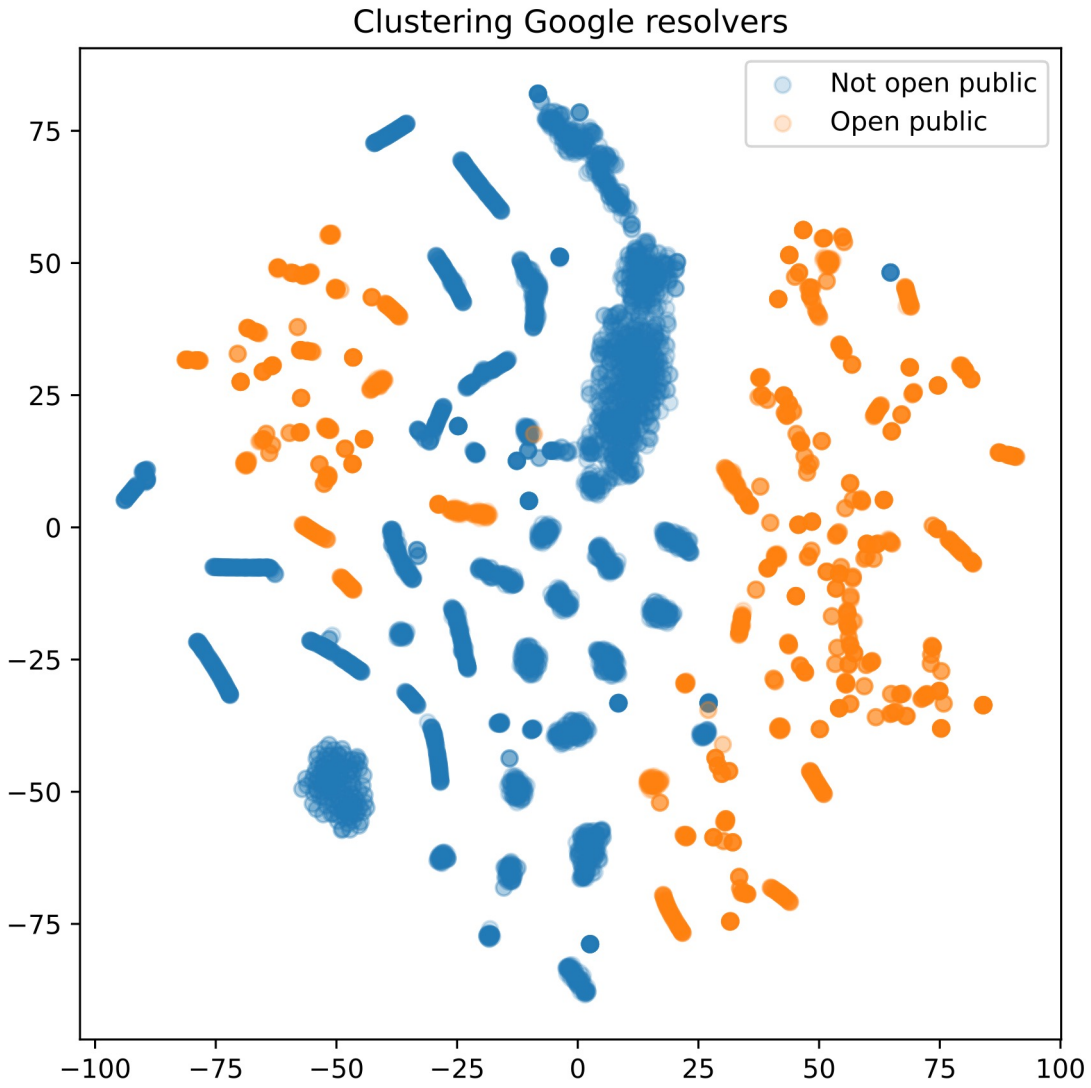
```
>>> model.wv.most_similar(positive=['2a02:a47f:e000:117:0:0:0:216',
                                    '2a02:a47f:e000:115:0:0:0:136',
                                    '2a02:a47f:e000:109:0:0:0:108'])


[('2a02:a47f:e000:115:0:0:0:184', 0.985),
 ('2a02:a47f:e000:115:0:0:0:188', 0.985),
 ('2a02:a47f:e000:117:0:0:0:196', 0.980),
 ('2a02:a47f:e000:115:0:0:0:132', 0.978),
 ('2a02:a47f:e000:117:0:0:0:200', 0.971)]
```

Use case: classify public open resolvers



Use case: cluster Google's resolvers



Project goal

How to generate meaning[illegible]NS concepts from query data using represe[illegible]ues?

Can we use the learned re[illegible]tream machine learning tasks, for example, clustering and classification?

But… What is the next step?

- Can we extend the representations with additional metadata?
- Can we learn representations for domain names?
- Can we gain new insights with the learned representations?
- Can we use the representations to improve our ML projects?
- Can we inspire others to explore the possibility of representation learning?

SIDN LABS

Are there any questions?

Follow us

SIDN.nl

@SIDN

SIDN

Thank you for your attention!