Generative Expression Constrained Knowledge-based decoding for Open data

Lucas Lageweg[1] & **Benno Kruit**[2]

[1]CCN Information Dialogue – Team Zoeken & Vinden & Universiteit van Amsterdam

[2]Vrije Universiteit Amsterdam

Introduction

G enerative

E xpression

C onstrained

K nowledge-based decoding

for **O** pen data

Introduction

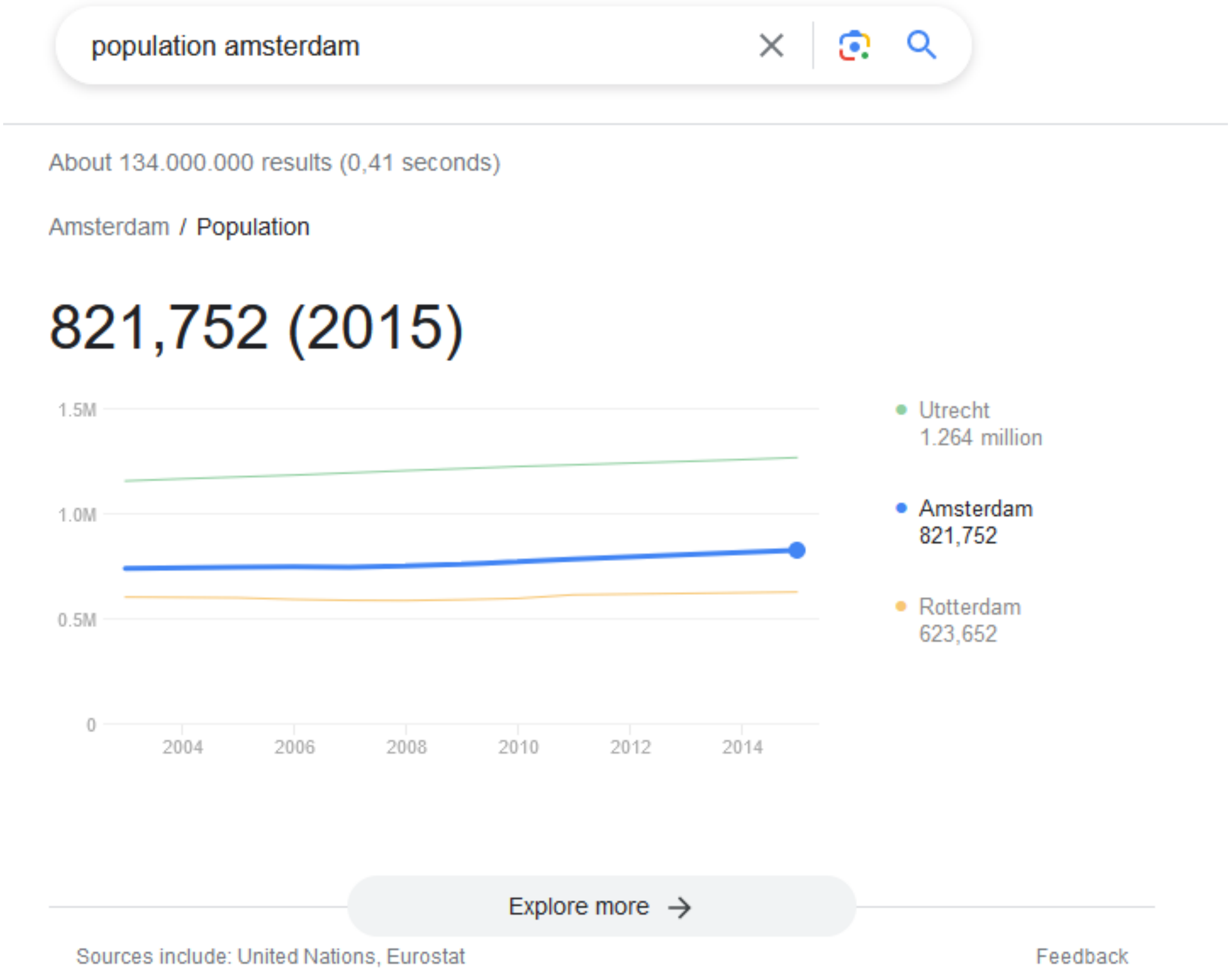
- Statistics Netherlands (CBS)
- Goal CCN Information Dialogue:

  To help users find the desired answer to their questions more quickly.
- Knowledge-base question answering (KBQA)
  - Input: question
  - Output: single table cell

Introduction

- Main challenges:
  - Generating good answers
    - Returning single cells from tables
  - Non-hallucinating
  - Scalability
  - Answer justification

Introduction



Data: StatLine

- Focus: Dutch ‘key figures’ tables

StatLine

Energy consumption private dwellings; type of dwelling and regions

Changed on: 19 October 2022

Topic

Housing characteristics	Regions	Periods	Average consumption of natural gas	Average consumption of electricity
			m3	kWh
Total dwellings	The Netherlands	2018*	1,270	2,790
		2019*	1,180	2,730
		2020*	1,120	2,760
		2021*	1,280	2,810
	Amsterdam	2018*	870	2,090
		2019*	800	2,050
		2020*	770	2,090
		2021*	880	2,130

Source: CBS

Data: StatLine

- Focus: Dutch ‘key figures’ tables

StatLine

Energy consumption private dwellings; type of dwelling and regions

Changed on: 19 October 2022

Measures

Dimensions

Observations

Housing characteristics	Regions	Periods	Average consumption of natural gas (m3)	Average consumption of electricity (kWh)
Total dwellings	The Netherlands	2018*	1,270	2,790
		2019*	1,180	2,730
		2020*	1,120	2,760
		2021*	1,280	2,810
	Amsterdam	2018*	870	2,090
		2019*	800	2,050
		2020*	770	2,090
		2021*	880	2,130

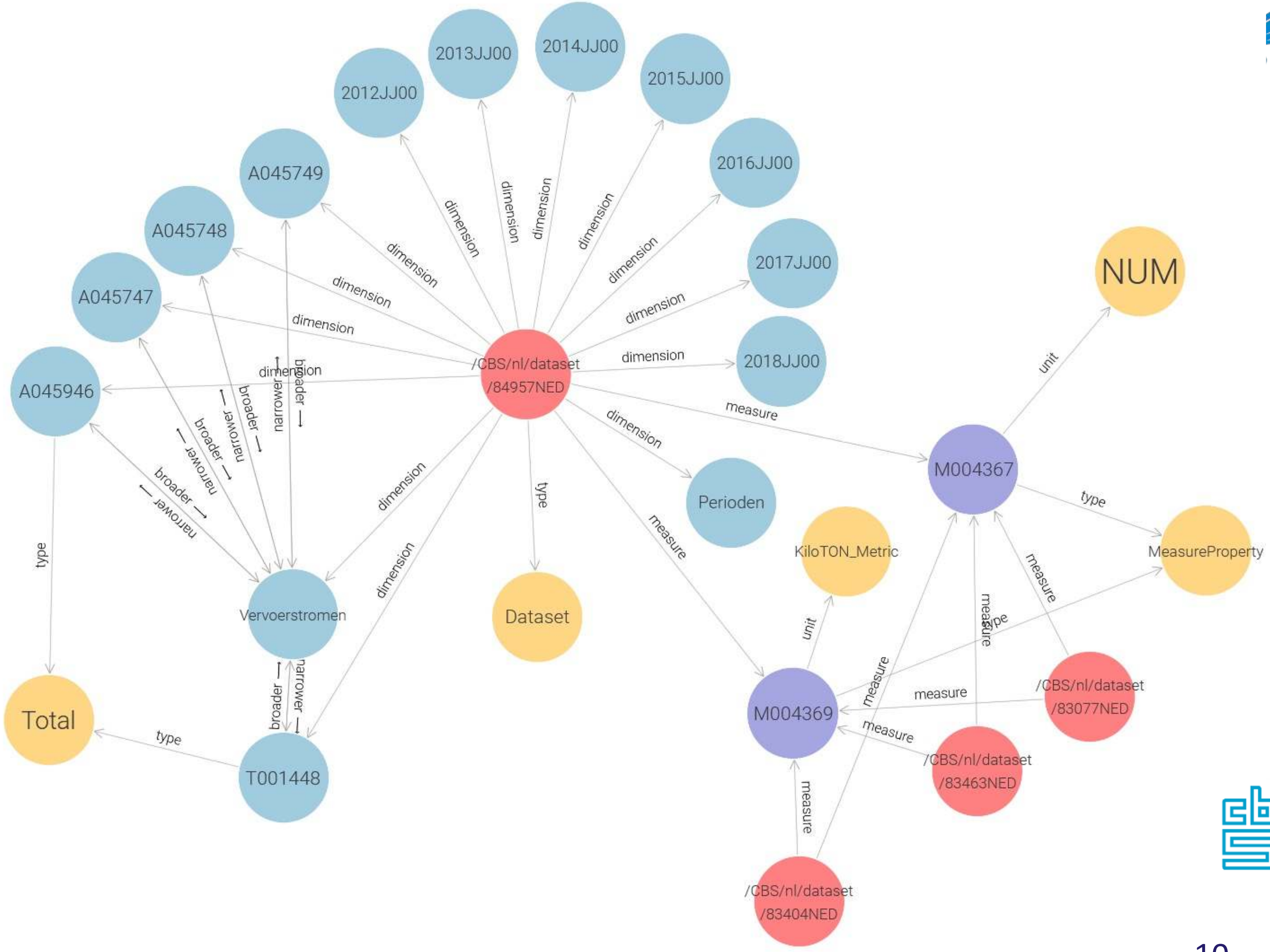
Source: CBS

Data: Open Data Version 4.01 (API)

```
{
  "@odata.context": "https://odata4.cbs.nl/CBS/81528NED/$metadata#Observations",
  "value": [
      {
          "Id": 0,
          "Measure": "M000219",                     MSR: average gas consumption
          "ValueAttribute": "None",
          "Value": 1850.0,
          "StringValue": null,
          “HousingCharacteristics": "T001100",      DIM: total dwellings
          “Regions": "NL01",                        DIM: The Netherlands
          “Periods": "2010JJ00“                     DIM: period
      },
    ]
}
```

Data: Knowledge Graph

- Identifier
- Title
- Description
- PrefLabel
- AltLabel



Method

- S-expressions as **intermediate query representation**

Question:

"How many tourists went abroad by train?"

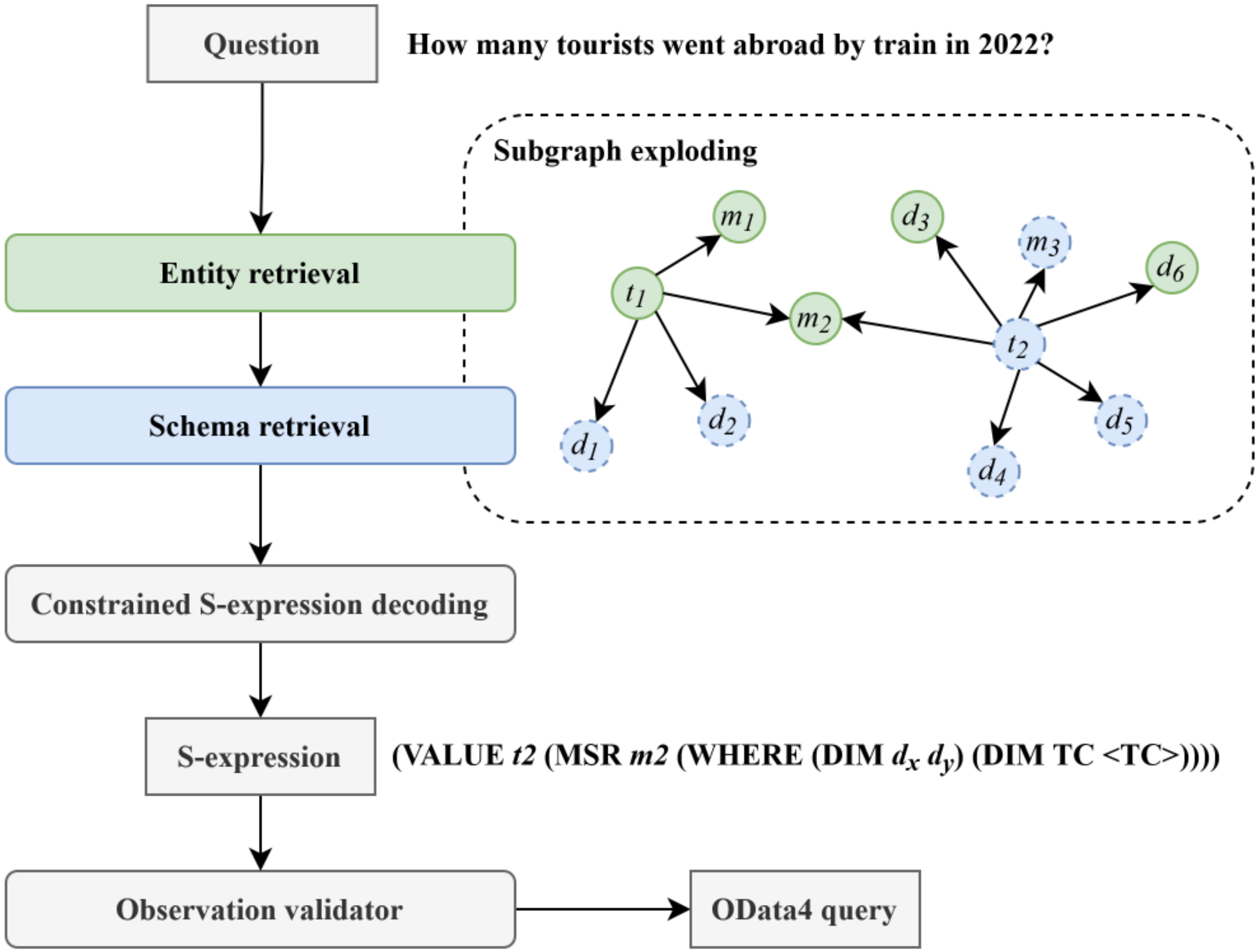
Total number of tourists

Method of transport: train

Output model:

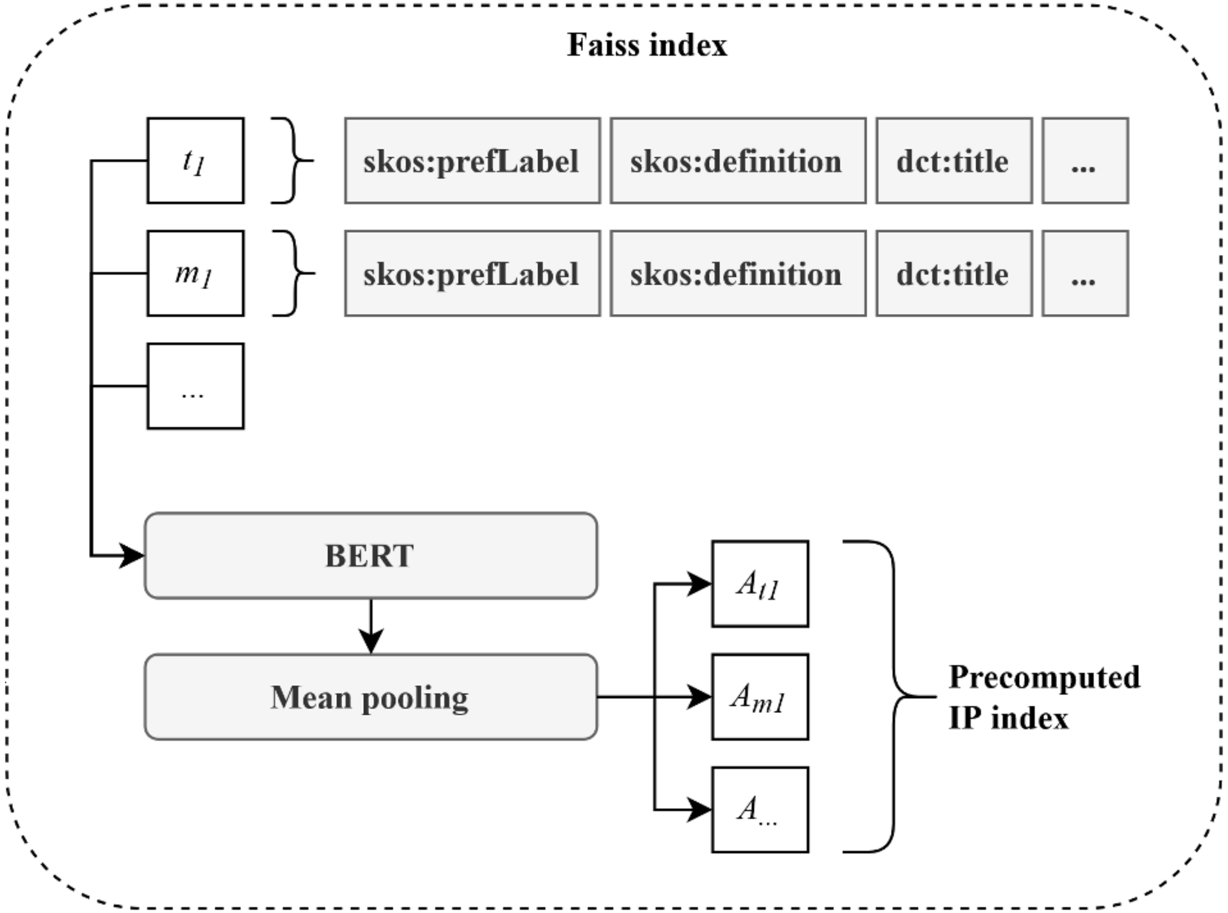
(VALUE (85302NED (MSR M001957 (WHERE (DIM Vakantiekenmerken A046401)))))

Method: entity retrieval



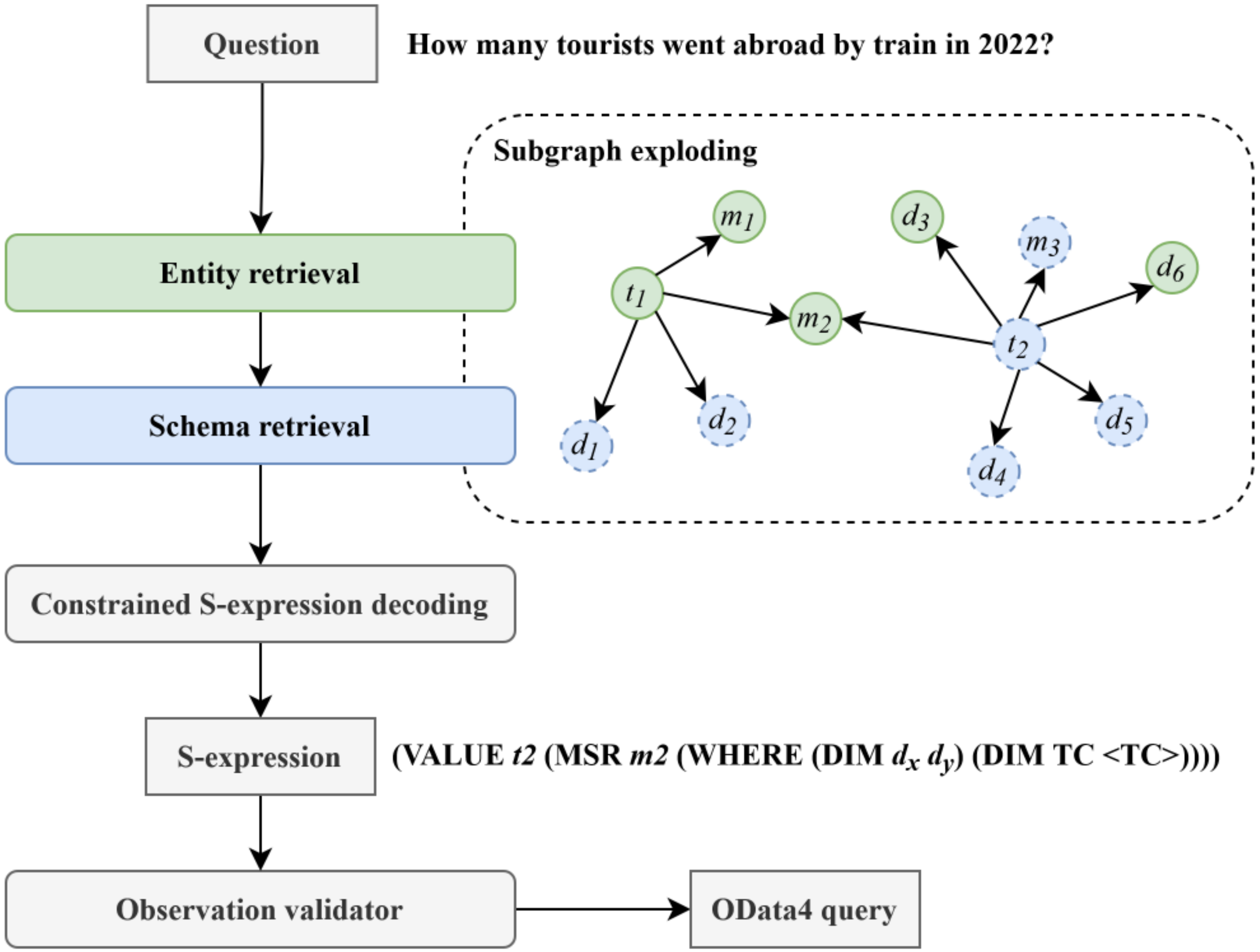
- Sparse retrieval
  - BM25+
  - Elasticsearch
- Dense retrieval
  - Sent. transformer
  - Embedding index

Method: entity retrieval



- Sparse retrieval
  - BM25+
  - Elasticsearch
- **Dense retrieval**
  - **Sent. transformer**
  - **Embedding index**

Method: schema retrieval



- Candidate nodes
- Subgraph exploding
    - GraphDB
    - SPARQL queries

Method: S-expression decoding

- Greedy baseline
  - BM25+ entity retrieval
  - Best scoring nodes → S-expression

Method: S-expression decoding

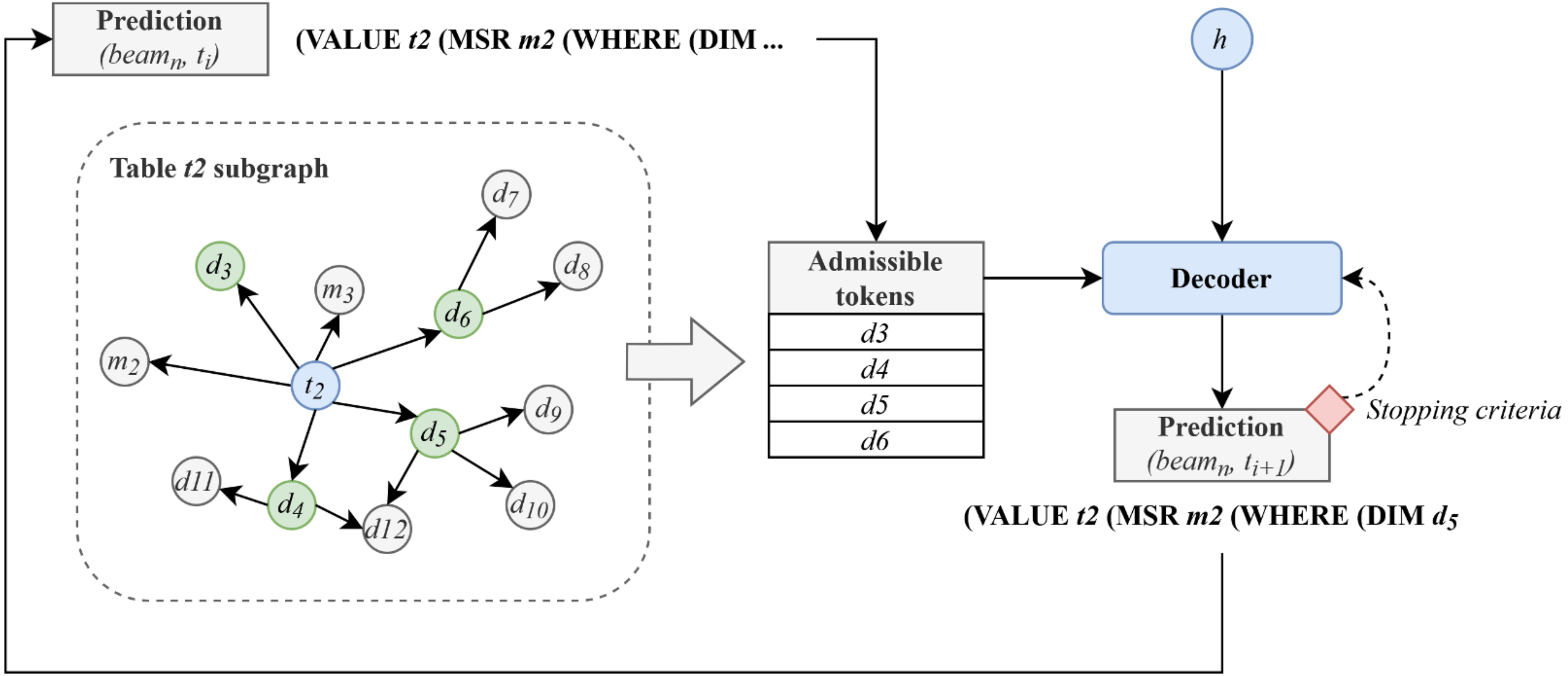
- Encoder-decoder LLM
  1. Add KG identifiers to LLM token vocabulary (~ 25k)
     - Excluding time and geo dimensions
  2. Make fixed embeddings for entity identifiers

- Query-time decoder pipeline
  1. Dense entity retrieval (embedding index)
  2. Prompt generation:

```
[CLS]Uitgaven zorg overheid 2020
[SEP]37789ksz|MSR|D000233;D000286;D000295_2;D003284;M006400;D000194;D000288;D007076_2;D000235;D001690|DIM|Perioden
[SEP]85542NED|MSR|D000881_1;D000883_4;D000881_7;D000881_10;D000898_3;D003333;D000898_10;D000883_2;D000883_1;D000883_11|DIM|MW00000
```

  3. Constrained inference

Method: constrained inference



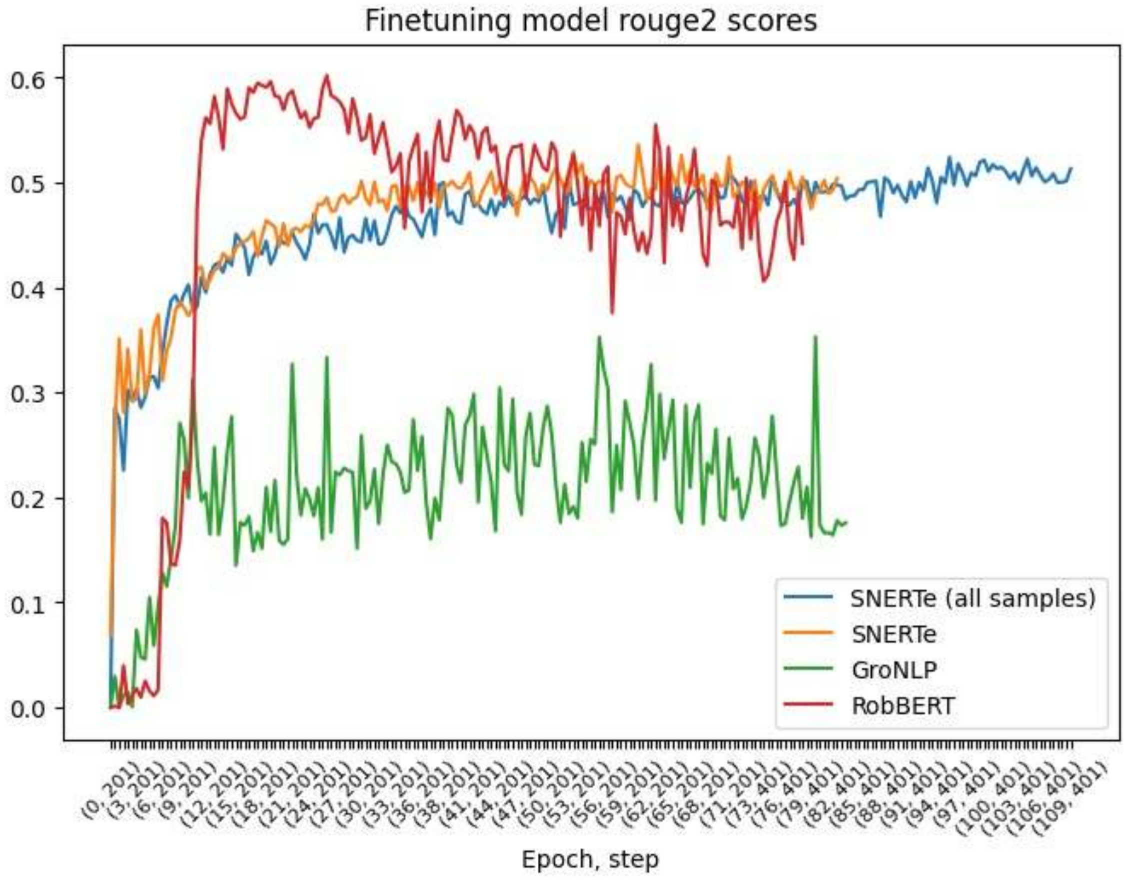
Method: PLMs

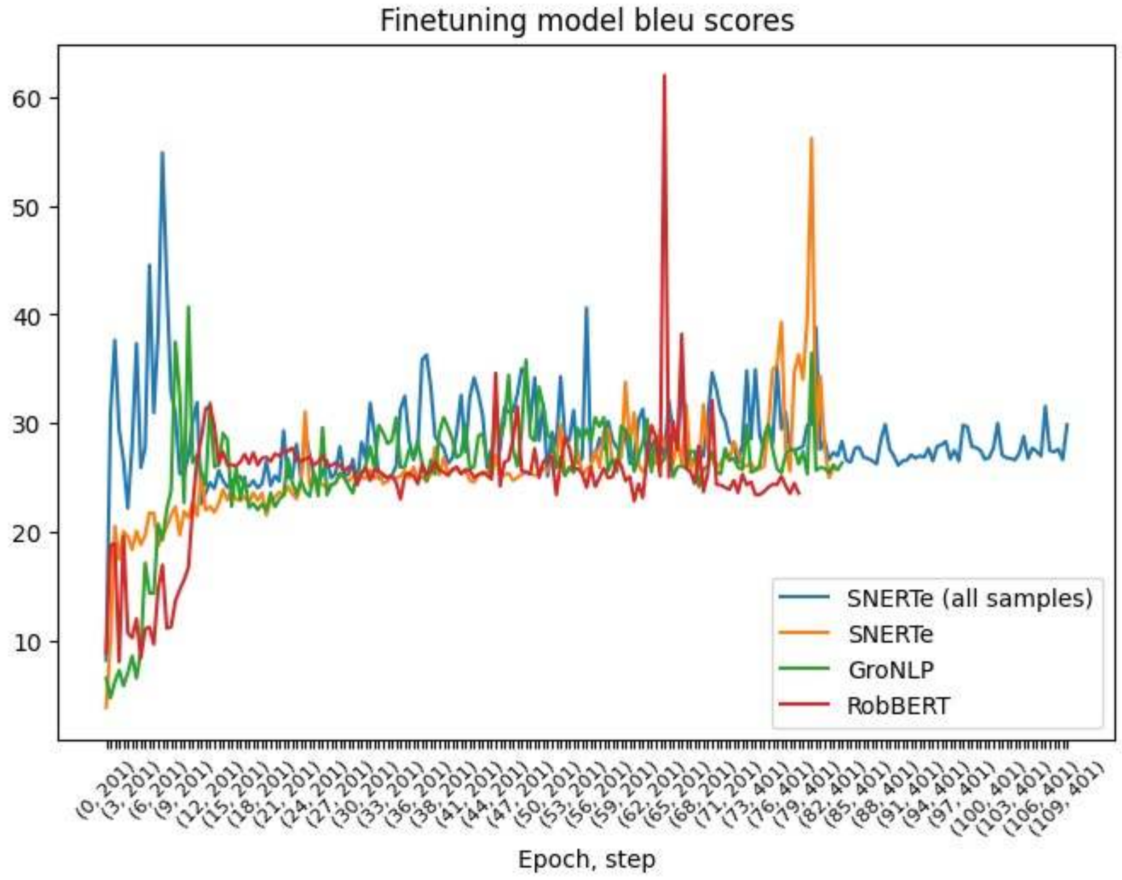
PLMs used for comparison:

1. RobBERT
   - Dutch RoBERTa
2. SNERT
   - CBS domain finetuned RobBERT (MLM)
3. BERTje GroNLP
   - Also used as sentence transformer

Results

- ~1.200 training samples question-expression pairs





Results: Entity retrieval

- Metrics: exact match of target entities
- Similar results between sparse and dense retrieval

		BM25+	Faiss
TABLE	Acc.	**0.530**	0.496
	P	0.114	**0.184**
	MRR	**0.262**	0.239
MSR	Acc.	**0.448**	0.435
	P	0.018	**0.074**
DIM	P	0.023	**0.074**
	R	**0.592**	0.555
	F1	0.040	**0.054**

Table 4: Evaluation results for entity recognition performance of BM25+ and dense vector search using Faiss.

Results: Generated S-expressions

- Evaluation on “key figure” tables dataset

Model	ER	ROUGE-2	BLEU	RS	Table EM	MSR EM	DIM F1
Baseline	BM25+	**0.437**	**62.198**	**0.378**	0.347	**0.198**	**0.621**
	Faiss	0.374	53.025	0.349	**0.396**	0.158	0.496
GroNLP	Faiss	0.294	48.039	0.107	0.181	0.029	0.455
RobBERT	Faiss	0.377	55.042	0.110	0.267	0.038	0.555
SNERTe	Faiss	0.193	40.278	0.031	0.200	0.048	0.214
SNERTe (all samples)	Faiss	0.318	46.182	0.167	0.188	0.100	0.398

Results: generalization evaluation

Key figures:

Model	ER	RS	Table EM	MSR EM	DIM F1
Baseline	BM25+	**0.378**	0.347	**0.198**	**0.621**
	Faiss	0.349	**0.396**	0.158	0.496
GroNLP	Faiss	0.107	0.181	0.029	0.455
RobBERT	Faiss	0.110	0.267	0.038	0.555
SNERTe	Faiss	0.031	0.200	0.048	0.214

All samples:

Model	ER	RS	Table EM	MSR EM	DIM F1
Baseline	BM25+	**0.357**	**0.409**	**0.278**	**0.564**
	Faiss	0.182	0.178	0.105	0.358
GroNLP	Faiss	0.081	0.126	0.039	0.176
RobBERT	Faiss	0.066	0.114	0.027	0.223
SNERTe	Faiss	0.055	0.076	0.013	0.170

Results: conclusions

- BM25+ & baseline better approach for current format
- Dense retrieval performance dropped with more tables
- Fixed embedder did not help learn code representations with the number of training samples available

Results: conclusions

- Recap main challenges:
  - Generating good answers ❌
  - Non-hallucinating ✅
  - Scalability ❌
  - Answer justification ✅

“Future” work

- Entity retrieval reranking by combining sparse & dense search ✅
- Research on effects of increasing training data ✅
- Investigate use of CBS domain-based LLM fine-tuning ✅
- More complex S-expressions 🗓️
- Determine and return when no answer is possible 🗓️