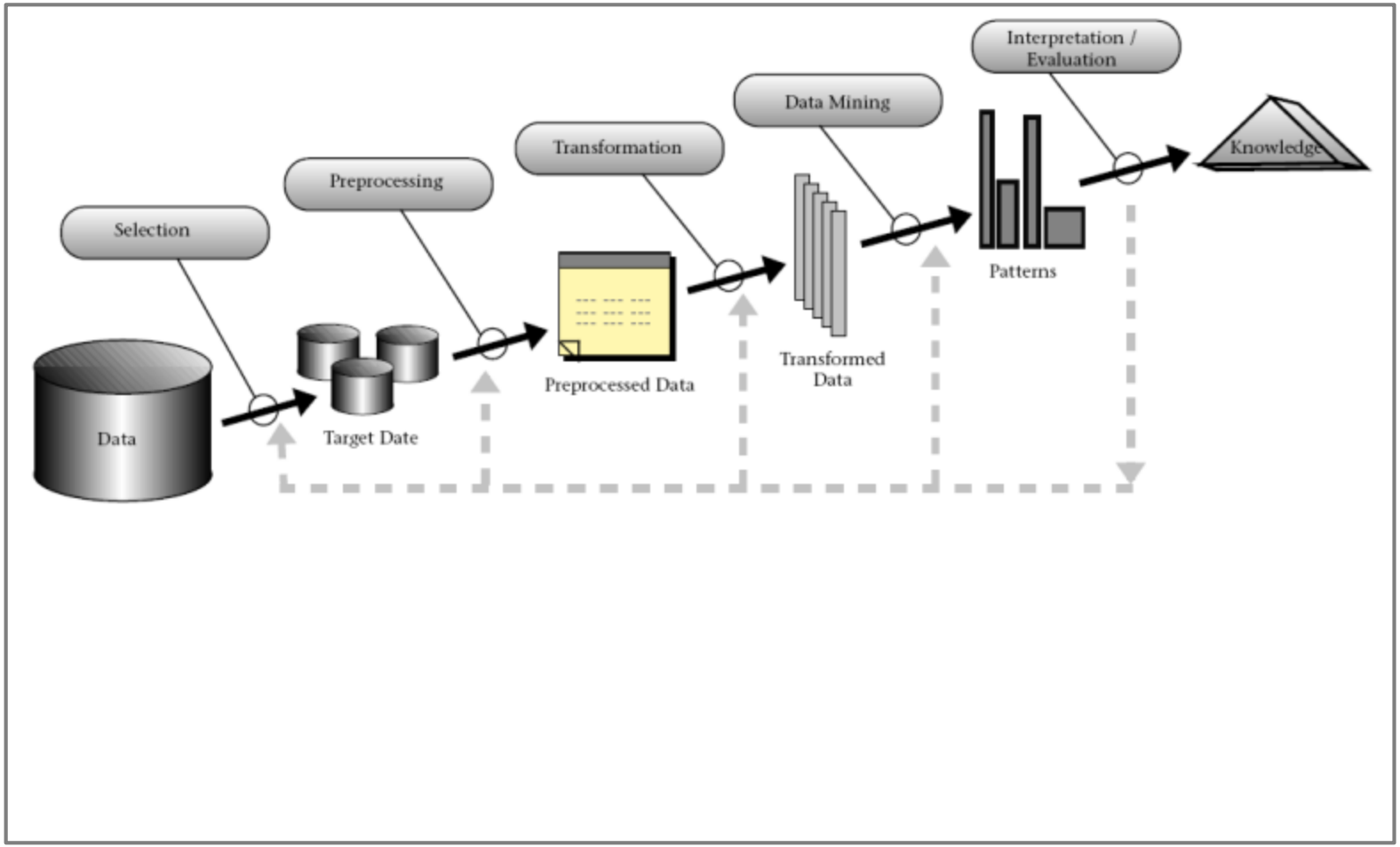
UNIVERSITÄT MANNHEIM

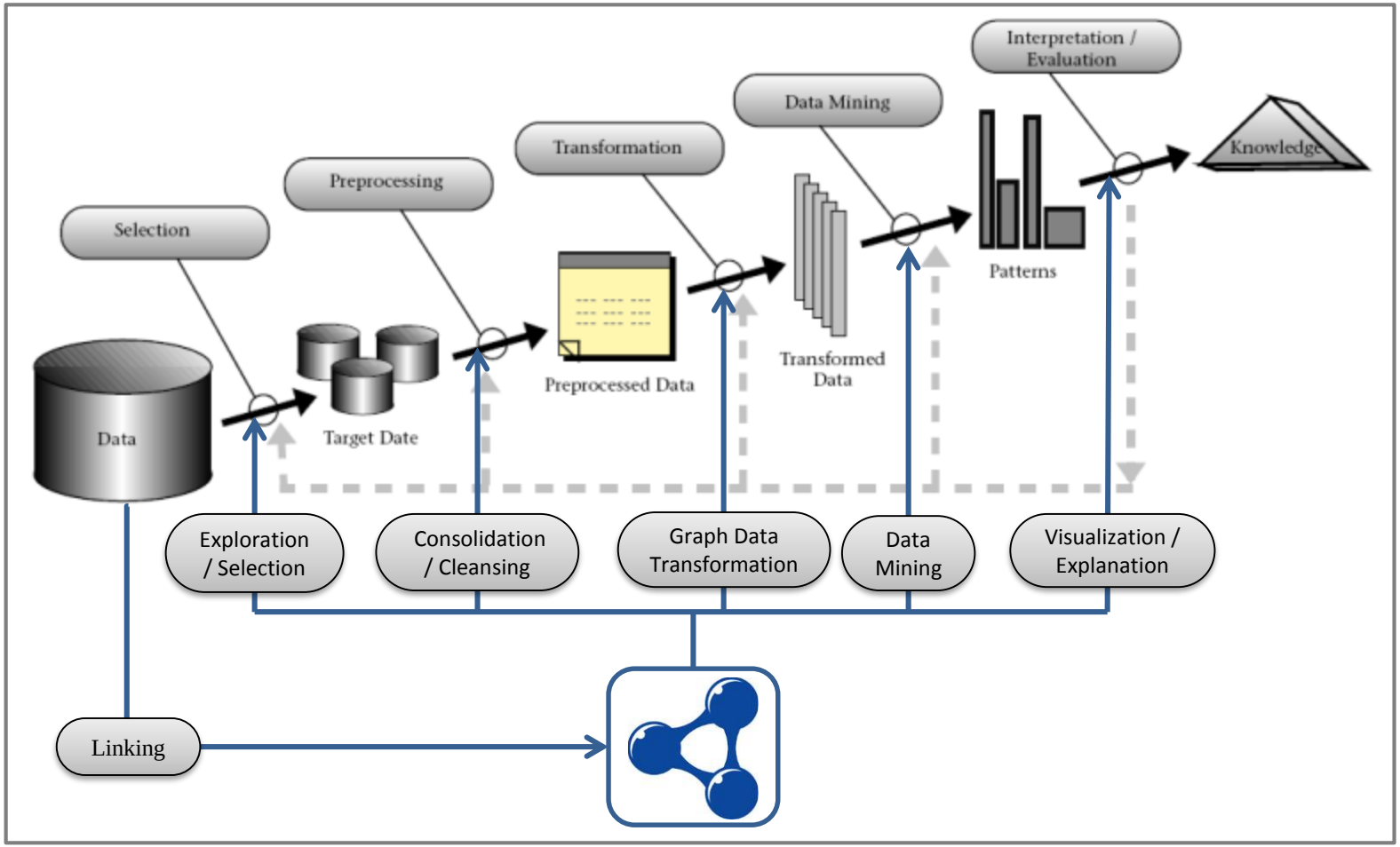
RDF2Vec: RDF Graph Embeddings for Data Mining

Petar Ristoski and Heiko Paulheim

Introduction



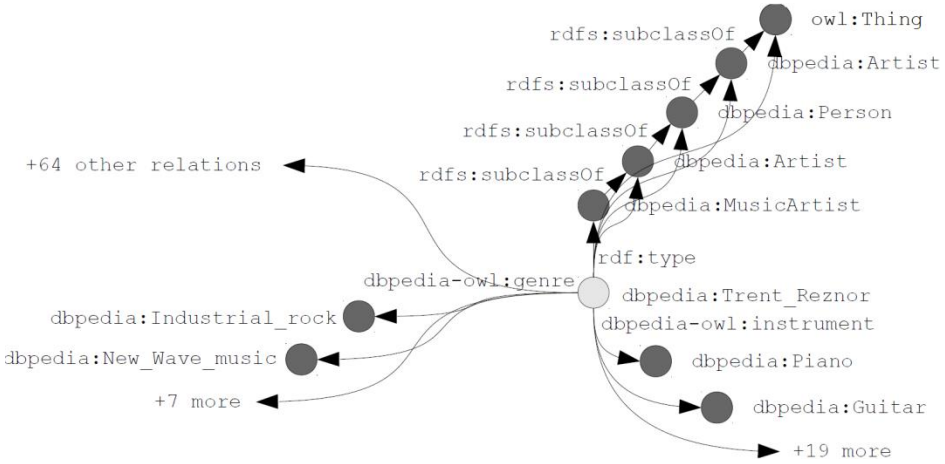
Introduction



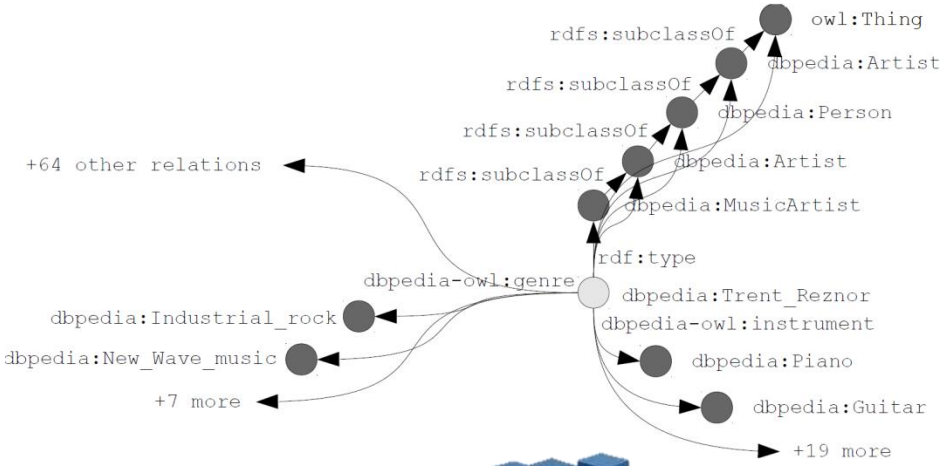
Motivation

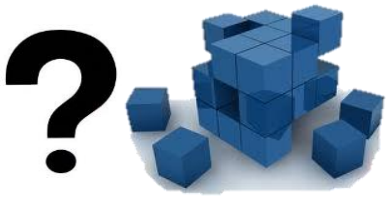
- Standard data mining algorithms require propositional feature vector representation
- Feature space: $V=\{v_1, v_2, \ldots, v_n\}$
- Each instance is represented as an n-dimensional feature vector $(v_1, v_2, \ldots, v_n)$, where for each $1 \leq v_i \leq n$:
  - $v_i \in \{true, false\}$, or $v_i \in \{1,0\}$
  - $v_i \in \mathbb{R}$
  - $v_i \in S$, where S is a finite set of symbols

Motivation

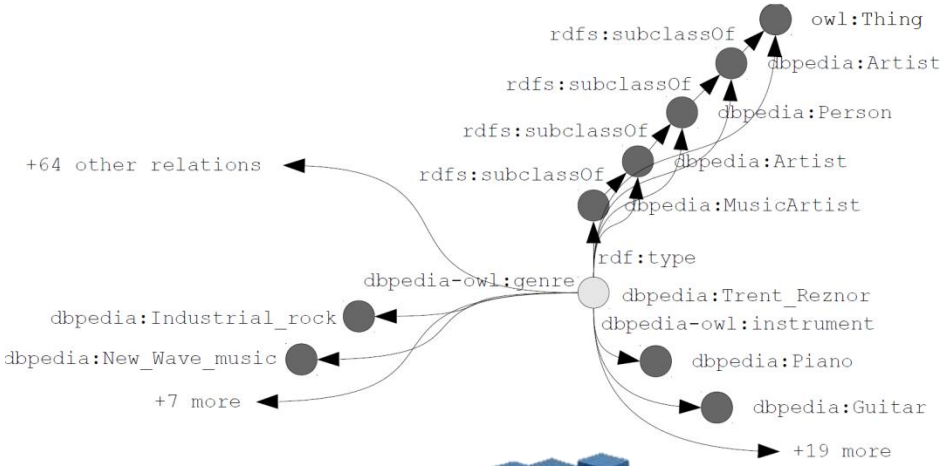

owl:Thing
rdfs:subclassOf
dbpedia:Artist
rdfs:subclassOf
dbpedia:Person
rdfs:subclassOf
dbpedia:Artist
+64 other relations
rdfs:subclassOf
dbpedia:MusicArtist
rdf:type
dbpedia-owl:genre
dbpedia:Trent_Reznor
dbpedia:Industrial_rock
dbpedia-owl:instrument
dbpedia:New_Wave_music
dbpedia:Piano
+7 more
dbpedia:Guitar
+19 more

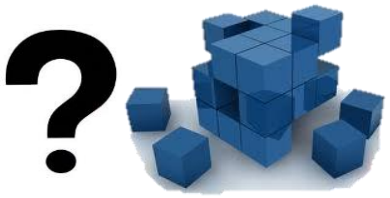
Motivation


owl:Thing
rdfs:subclassOf
dbpedia:Artist
rdfs:subclassOf
dbpedia:Person
rdfs:subclassOf
dbpedia:Artist
+64 other relations
rdfs:subclassOf
dbpedia:MusicArtist
rdf:type
dbpedia-owl:genre
dbpedia:Trent_Reznor
dbpedia:Industrial_rock
dbpedia-owl:instrument
dbpedia:New_Wave_music
dbpedia:Piano
+7 more
dbpedia:Guitar
+19 more



Motivation





Name	Person	Music Artist	Instrument	Genre
Trent Reznor	1	1	1	0
Wolfgang A. Mozart	1	1	1	1
Barack Obama	1	0	0	0

Vision

- Preserve the information given in the original graph
- Unsupervised
  - task and dataset independent
- Compatible with traditional data mining algorithms and tools
- Efficient computation and application
  - Low dimensional representation

RDF2VEC APPROACH

RDF2Vec

- Adaptation of neural language models
  - Word2vec
  - Latent representation of words based on text corpus
- Convert RDF graphs in sequences of entities and relations (sentences)
  - Graph Walks
  - Weisfeiler-Lehman Subtree RDF Graph Kernels
- Train neural language model
  - Each entity and relation is represented as N-dimensional numerical vector
  - Semantically similar entities appear closer in the embedded space
- Use entity vectors in different ML tasks

Word2vec – Neural Language Model

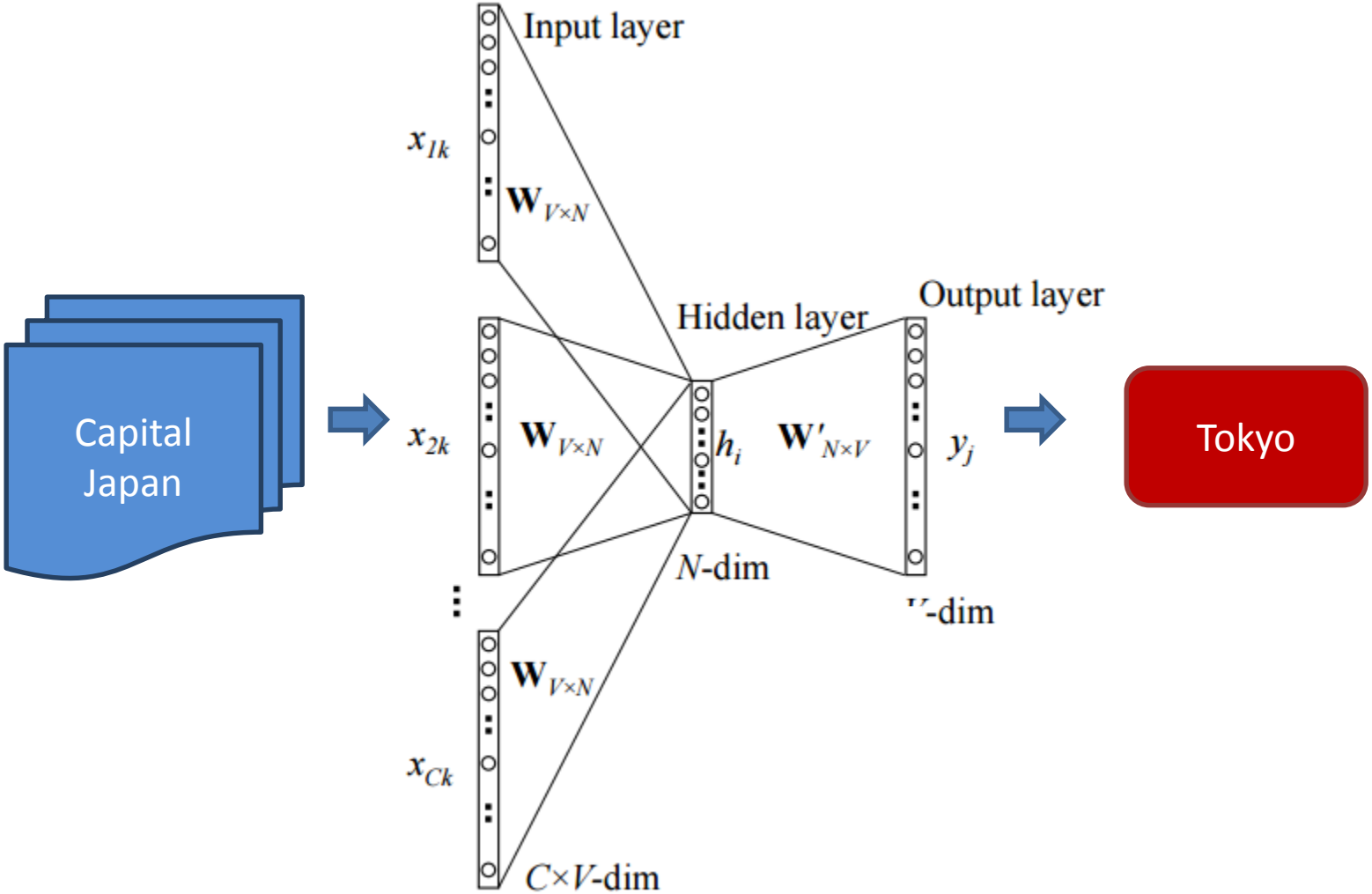
- Two-layer neural net that converts raw text into vectors
  - Each word is represented into a numerical vector

- Continuous Bag-of-Words (CBOW)
  - Predict target words from source context words
  - Tokyo is the capital of Japan

- Skip-gram

[1] Mikolov, Tomas, et al. "Distributed representations of words and phrases and their compositionality." *NIPS,* 2013.
[2] Rong, Xin. "word2vec parameter learning explained." 2014.

CBOW


Capital
Japan
Input layer
x_{1k}
$\mathbf{W}_{V\times N}$
x_{2k}
$\mathbf{W}_{V\times N}$
$\mathbf{W}_{V\times N}$
x_{Ck}
$C\times V$-dim
Hidden layer
h_i
N-dim
Output layer
$\mathbf{W}'_{N\times V}$
y_j
V-dim
Tokyo

Word Embedding

Tokyo = [f_1, f_2, f_3, …, fn]

Japan= [f_1, f_2, f_3, …, fn]

Word Embedding

Tokyo = [f_1, f_2, f_3, …, fn]

Japan= [f_1, f_2, f_3, …, fn]

Word Embedding

Tokyo = [f_1, f_2, f_3, …, fn]

Japan= [f_1, f_2, f_3, …, fn]

Word Embedding

Tokyo = [f_1, f_2, f_3, …, fn]

Japan= [f_1, f_2, f_3, …, fn]

Word Embedding

Tokyo = [f_1, f_2, f_3, …, fn]

Japan= [f_1, f_2, f_3, …, fn]

Word Embedding

Tokyo = [f_1, f_2, f_3, …, fn]

Japan= [f_1, f_2, f_3, …, fn]

v(Japan) - v(Tokyo) + v(Berlin) ≈ v(Germany)

Word2vec – Neural Language Model

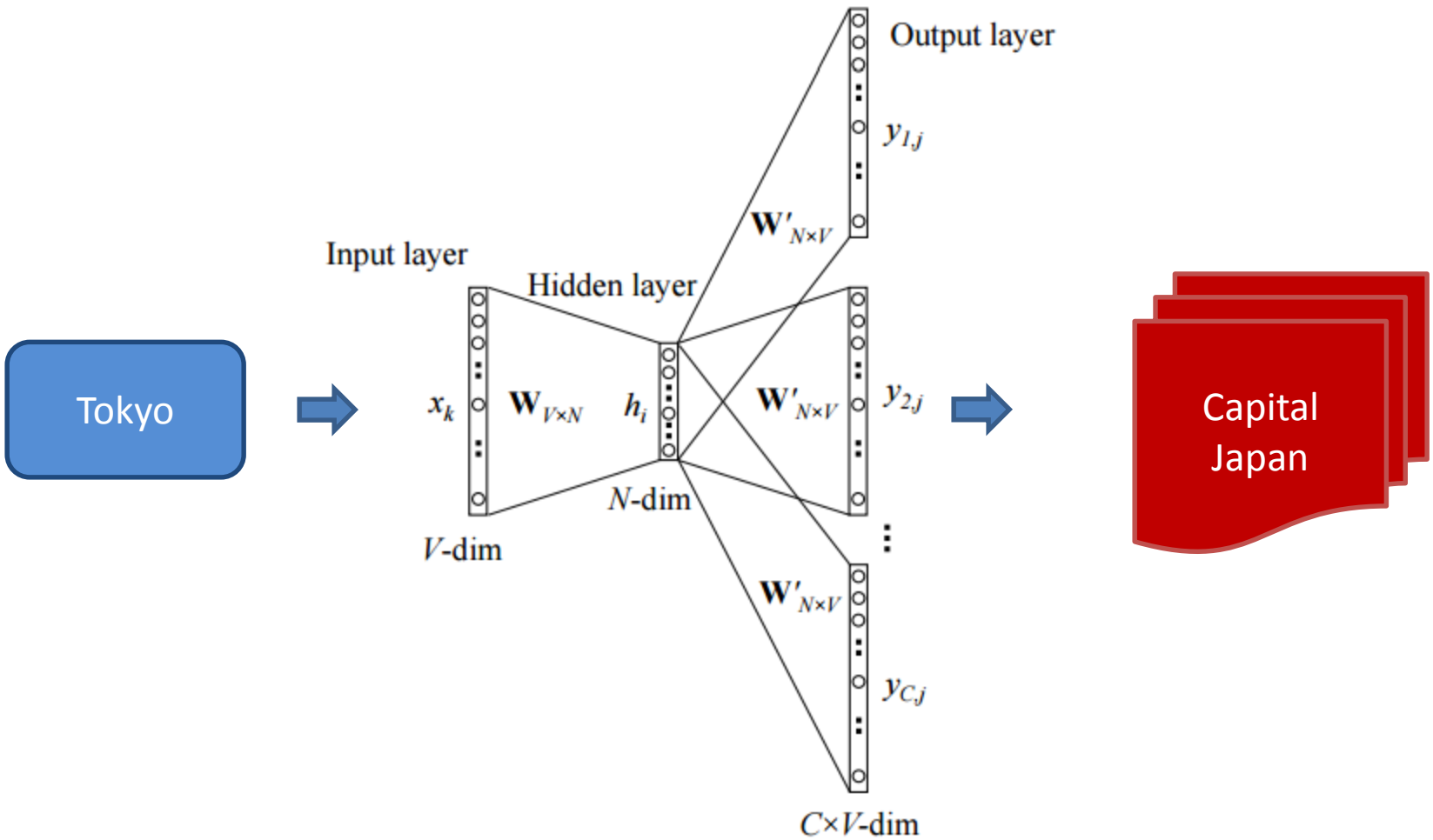
- Two-layer neural net that converts raw text into vectors
  - Each word is represented into a numerical vector

- Continuous Bag-of-Words (CBOW)
  - Predict target words from source context word
  - Tokyo is the capital of Japan

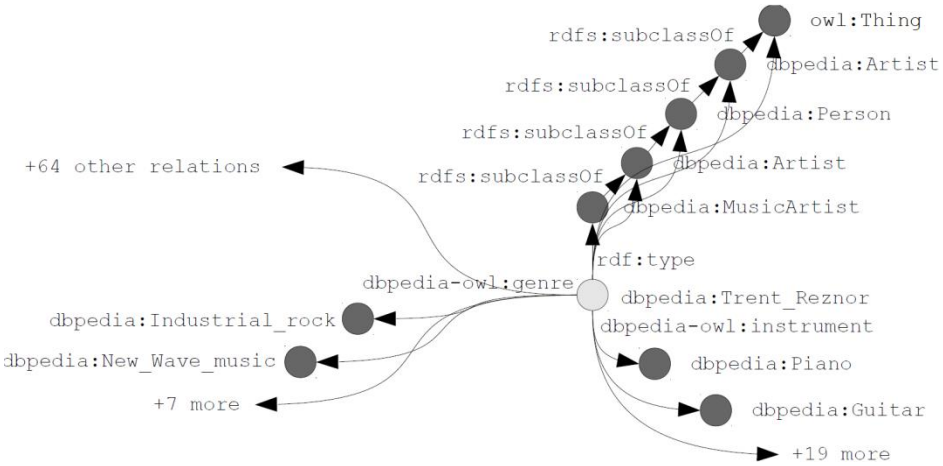
- Skip-gram
  - Predict context words from the target word
  - Tokyo is the capital of Japan

[1] Mikolov, Tomas, et al. "Distributed representations of words and phrases and their compositionality." *NIPS,* 2013.
[2] Rong, Xin. "word2vec parameter learning explained." 2014.

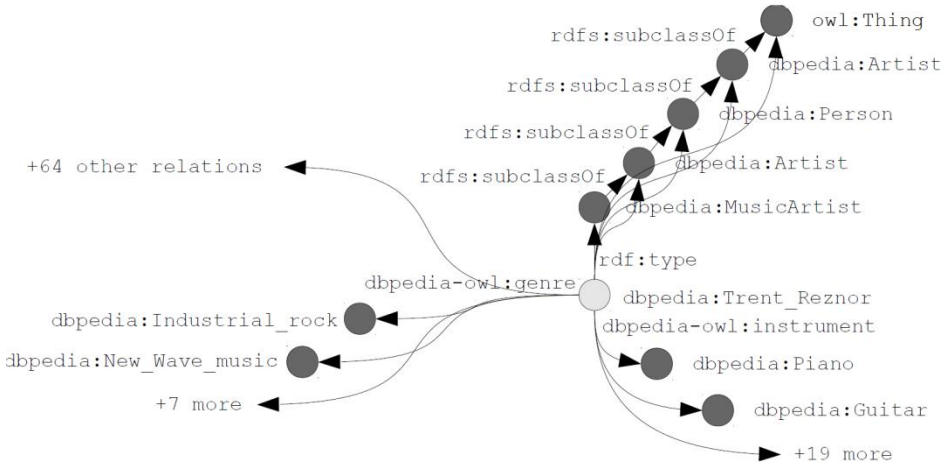
Skip-gram


Tokyo
Input layer
Hidden layer
Output layer
x_k
$\mathbf{W}_{V\times N}$
h_i
$\mathbf{W}'_{N\times V}$
$y_{1,j}$
$y_{2,j}$
$y_{C,j}$
N-dim
V-dim
$C\times V$-dim
Capital
Japan

RDF2vec

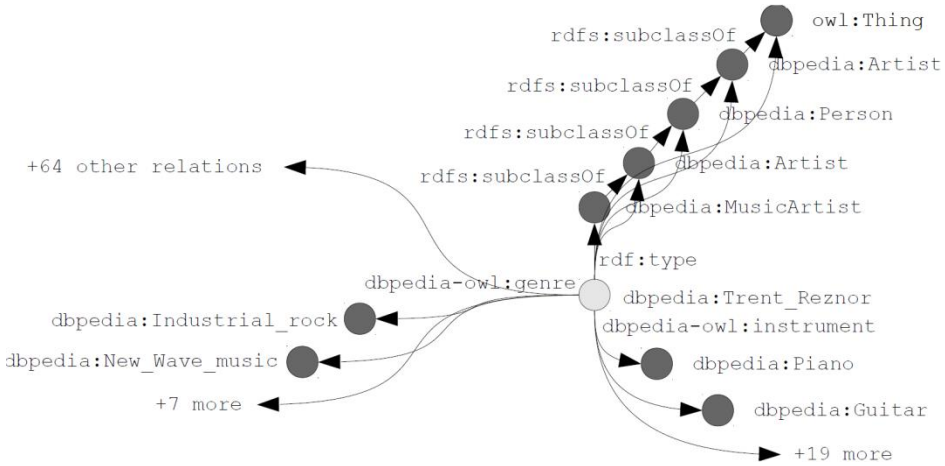

owl:Thing
rdfs:subclassOf
dbpedia:Artist
rdfs:subclassOf
dbpedia:Person
rdfs:subclassOf
+64 other relations
rdfs:subclassOf
dbpedia:Artist
dbpedia:MusicArtist
rdf:type
dbpedia-owl:genre
dbpedia:Trent_Reznor
dbpedia:Industrial_rock
dbpedia-owl:instrument
dbpedia:New_Wave_music
dbpedia:Piano
+7 more
dbpedia:Guitar
+19 more

RDF2vec



- Convert the graph into sequence of tokens (sentences)
  - Graph walks

RDF2vec



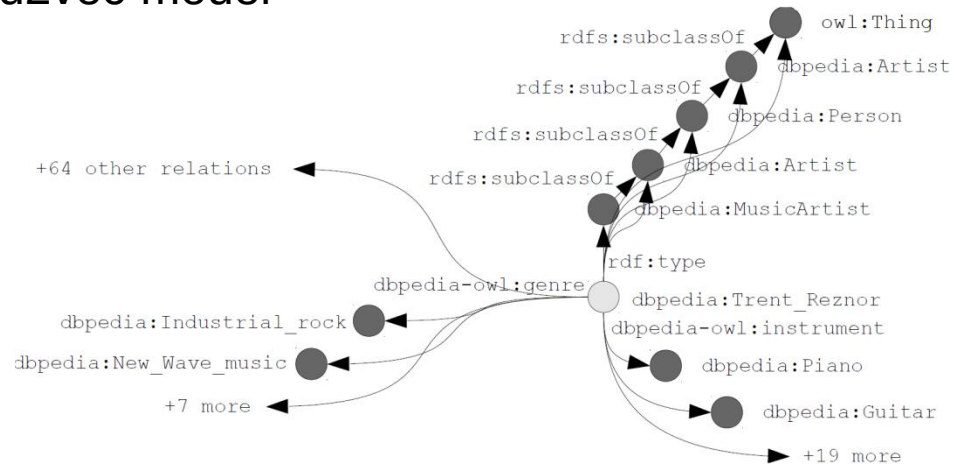
- Convert the graph into sequence of tokens (sentences)
  - Graph walks
  - Weisfeiler-Lehman Subtree RDF Graph Kernels

Graph Walks RDF2vec

- For each entity in the graph:
  - Extract a subgraph with depth *d*
  - Extract walks on the subgraph
  - Build word2vec model

Graph Walks RDF2vec

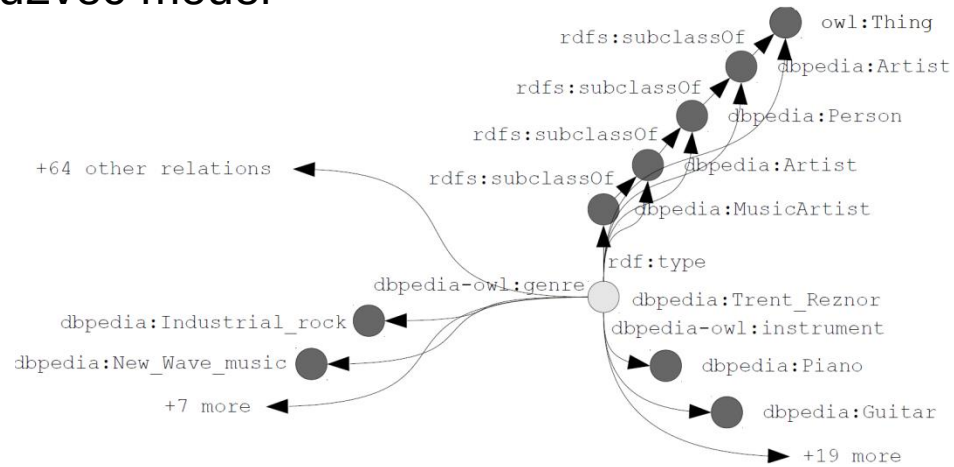
- For each entity in the graph:
  - Extract a subgraph with depth *d*
  - Extract walks on the subgraph
  - Build word2vec model



dbr:Trent_Reznor -> dbo:associatedBand -> dbr:Exotic_Birds -> dbo:bandMember -> dbr:Chris_Vrenna

Graph Walks RDF2vec

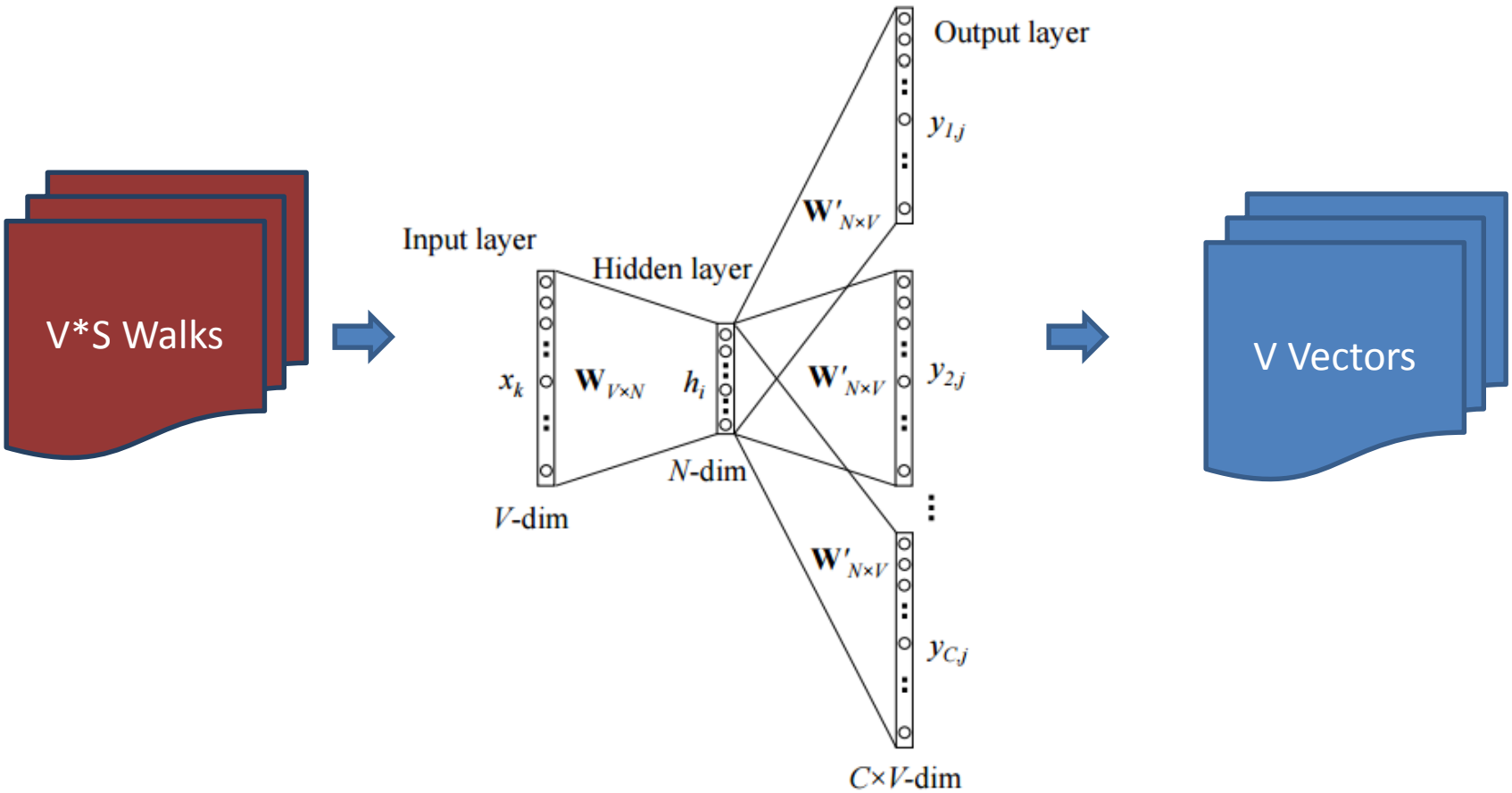
- For each entity in the graph:
  - Extract a subgraph with depth *d*
  - Extract walks on the subgraph
  - Build word2vec model



dbr:Trent_Reznor -> dbo:associatedBand -> dbr:Exotic_Birds -> dbo:bandMember -> dbr:Chris_Vrenna

dbr:Trent_Reznor -> dbo:genre - > dbr:Dark_ambient -> dbo:instrument -> dbr:Field_recording

Random Walks RDF2vec

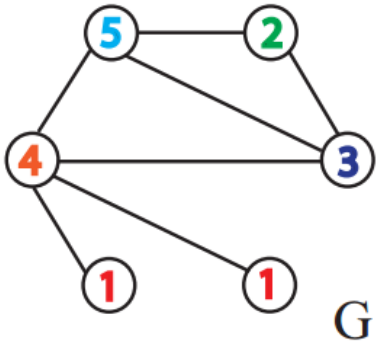


Entity Embedding

dbr:Tokyo = [f_1, f_2, f_3, …, fn]

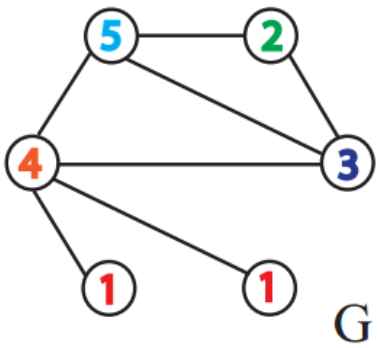
dbr:Japan= [f_1, f_2, f_3, …, fn]

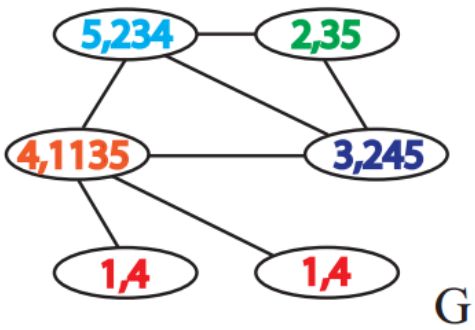
Weisfeiler-Lehman Kernel



Weisfeiler-Lehman Kernel

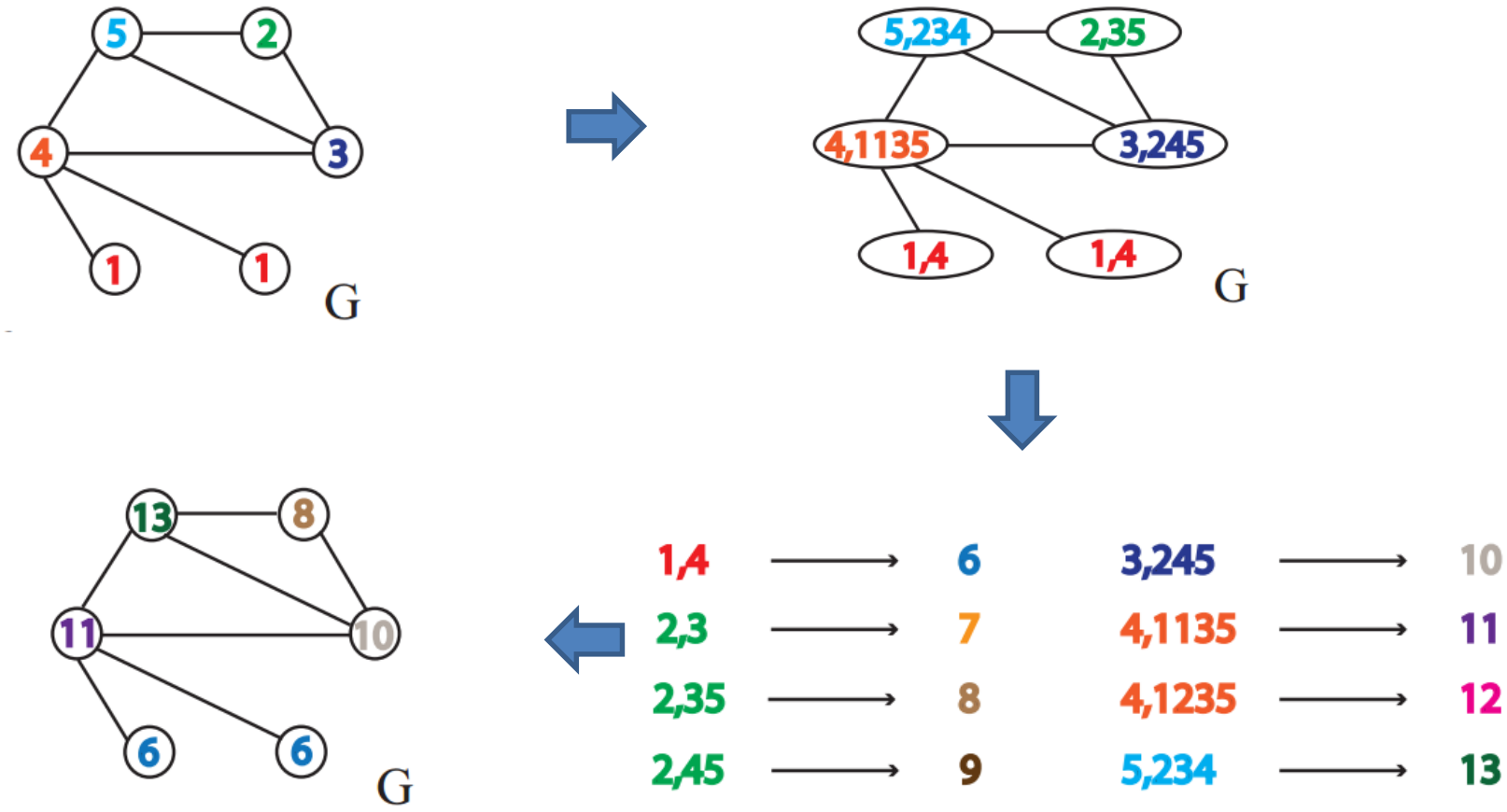
Weisfeiler-Lehman Kernel



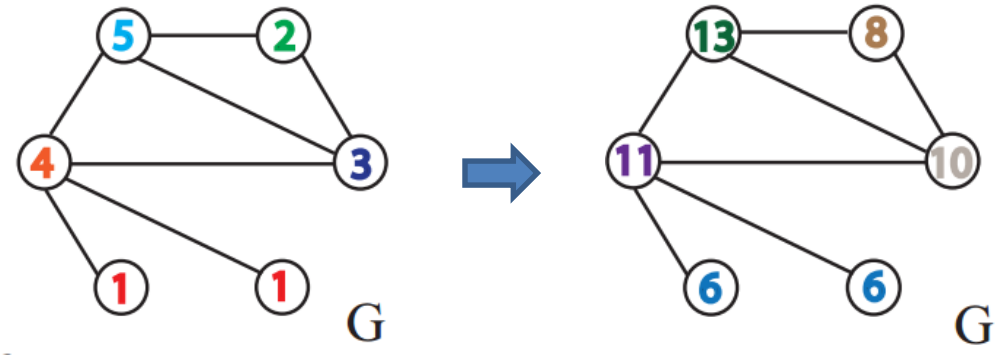


1,4	→	6	3,245	→	10
2,3	→	7	4,1135	→	11
2,35	→	8	4,1235	→	12
2,45	→	9	5,234	→	13

Weisfeiler-Lehman Kernel



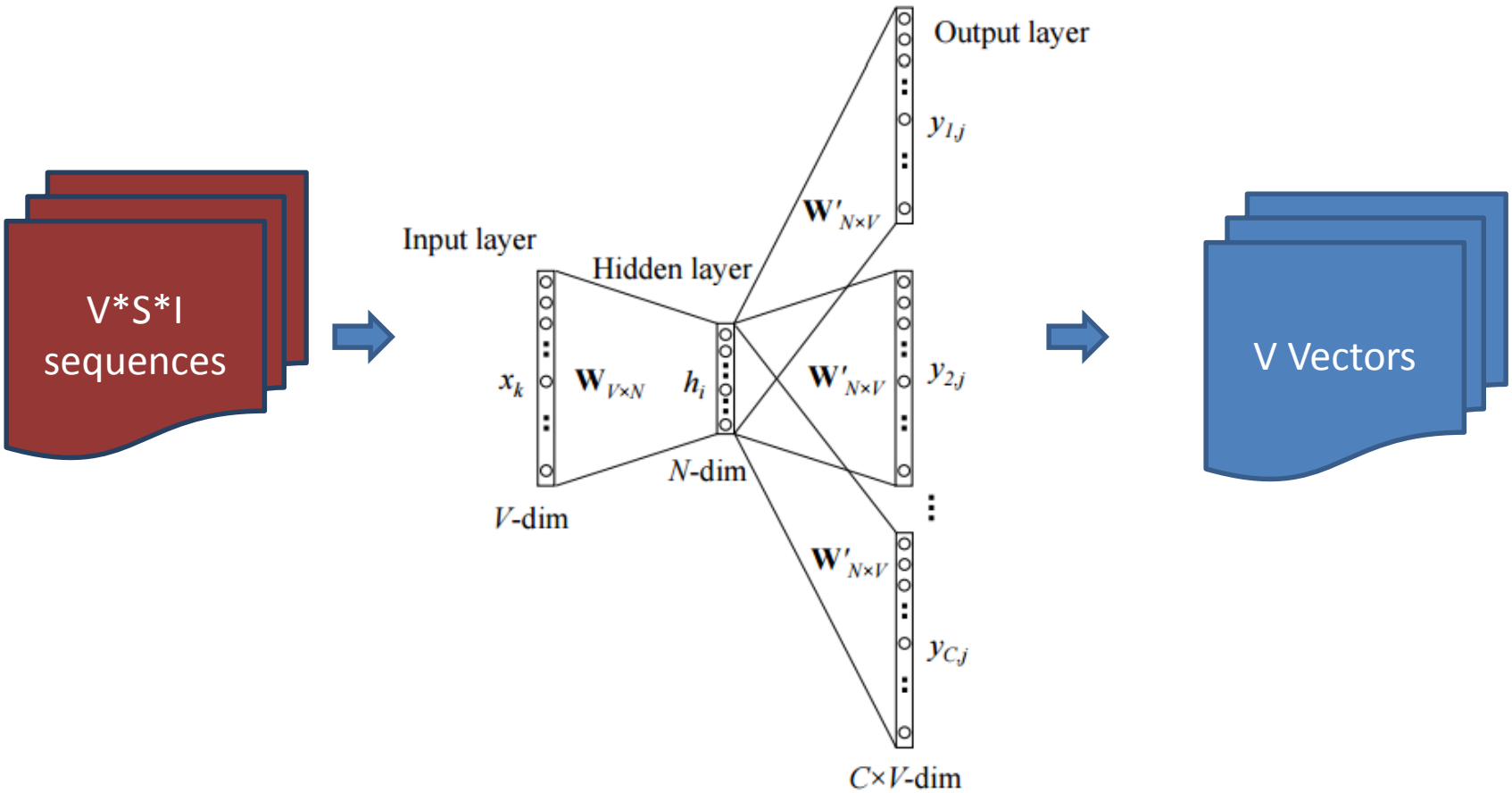
WL Kernel RDF2vec



- Construct sequences using random walks with depth *d* after each iteration for each entity in the graph
- Graph G sequences after 1 iteration:
  - 1->6->11; 1->6->11->13; 1->6->11->10 …
  - 4->11->6; 4->11->13; 4->11->10; 4->11->10->8 …
  - …

de Vries, Gerben KD. "A fast approximation of the Weisfeiler-Lehman graph kernel for RDF data.“ ECML, 2013.

WL Kernels RDF2vec



EVALUATION

Evaluation Setup

- Datasets
  - 3 domain-specific RDF datasets
  - 2 large cross-domain RDF datasets with 5 evaluation datasets
- Tasks
  - Classification: Naive Bayes, k-Nearest Neighbors (k=3), C4.5 decision tree and Support Vector Machines.
  - Regression: Linear Regression, M5Rules, and k-Nearest Neighbors (k=3).
- Baselines
  - Features derived from incoming and outgoing relations and values
  - Features derived from graph substructures: WL and Walk-Count Kernels

Domain Specific RDF Datasets

- Datasets

Dataset	Task	#statements	#instances	#walks	depth	#sequences	WL iter.	WL depth	#sequences
AIFB	C (c=4)	30K	176	all	10	360K	4	2	346K
BGS	C (c=2)	600K	146	all	10	2.4M	4	2	5.3M
MUTAG	C (c=2)	80K	340	all	10	168K	4	2	908K

- Results (accuracy)
  - Best scores per dataset

Dataset	Baseline	Walks2vec	WL2vec (SG 500)
AIFB	92.68	89.55	**93.41**
BGS	91.05	78.10	**96.18**
MUTAG	94.29	82.06	**96.33**

Large Cross-Domain RDF Datasets

- Datasets

Dataset	#instances	depth	#sequences	Vector size	model
DBpedia	5M	4/8	2.5B	200/500	CBOW/SG
Wikidata	17M	4	8.5B	200/500	CBOW/SG

- Evaluation datasets

Dataset	#Instances	ML Task	Original Source
Cities	212	R/C (c=3)	Mercer
Metacritic Albums	1,600	R/C (c=3)	Metacritic
Metacritic Movies	2,000	R/C (c=3)	Metacritic
AAUP	960	R/C(c=3)	JSE
Forbes	1,585	R/C (c=3)	Forbes

Results: classification

- Accuracy Results
  - Best scores only

	Cities	Movies	Albums	AAUP	Forbes
Best Baseline	75.13	79.30	77.94	93.44	76.75
DB2vec CBOW 200 8	77.39	**83.65**	78.44	92.23	88.30
DB2vec CBOW 500 8	76.84	83.25	77.25	90.61	89.86
DB2vec SG 200 8	78.92	83.30	**79.72**	91.04	**90.10**
DB2vec SG 500 8	**89.73**	82.80	78.20	**94.48**	88.53
WD2vec CBOW 200 4	75.56	52.20	51.44	90.18	81.08
WD2vec CBOW 500 4	85.56	51.04	53.28	89.74	80.74
WD2vec SG 200 4	75.48	75.39	64.76	90.50	81.17
WD2vec SG 500 4	83.20	76.30	63.42	90.60	81.17

Results: regression

- RMSE Results
  - Best scores only

	Cities	Movies	Albums	AAUP	Forbes
Best Baseline	17.04	19.19	12.81	6.16	18.32
DB2vec CBOW 200 8	12.55	15.90	11.79	6.47	17.43
DB2vec CBOW 500 8	12.54	15.81	11.30	6.54	17.62
DB2vec SG 200 8	12.85	**15.12**	10.90	6.22	17.85
DB2vec SG 500 8	**10.19**	15.45	**10.89**	6.26	**16.61**
WD2vec CBOW 200 4	17.52	23.39	14.55	6.60	21.77
WD2vec CBOW 500 4	18.33	22.18	14.00	6.08	21.92
WD2vec SG 200 4	18.69	19.10	13.51	6.52	21.59
WD2vec SG 500 4	19.23	19.19	13.23	**6.05**	21.58

Results Summary

- RDF2vec outperform all the baseline approaches
  - Smaller feature vectors - more efficient training than bassline approaches
- WL kernel sequences capture the graph structure better than walks
  - Not efficient on large graphs
  - Large number of sequences produced – not scalable
- Increasing the depth of the paths increases the quality of the embeddings
- The vector dimensionality doesn’t affect the performance
- Skip-Gram models constantly outperforms CBOW models
- DBpedia produces higher quality embeddings than Wikidata

Other Use-Cases

- Recommender systems
- Document modeling
  - Document similarity
  - Entity relatedness
- Alignment of knowledge bases
  - DBpedia and Wikidata
- Knowledge base relation prediction and error detection
- Linking text and semi-structured knowledge to knowledge bases

Conclusion

- RDF2Vec: an approach for learning latent numerical representations of entities in RDF graphs
- Preserves the graph information
- Compatible with all the traditional machine learning algorithms
- More efficient ML models training
- Task and dataset independent approach
- Download the code and the models: http://data.dws.informatik.uni-mannheim.de/rdf2vec/