# Knowledge Graph Embeddings for Downstream Machine Learning Tasks with RDF2Vec

Terencio AGOZZINO

terencio.agozzino@std.heh.be

MA2 IT Engineering

*Advisor:*
Prof. Dr. Femke ONGENAE

*Supervisors:*
Dr. Ir. Gilles VANDEWIELE
Dr. Ir. Samuel CREMER
Ir. Bram STEENWINCKEL

# IDLab

# IDLab[1]

- Influential research group of imec.
- Focuses on Internet technologies and data science.
- Makes contributions to future global standards (e.g., W3C and 5G).
- Located at the University of Ghent and the University of Antwerp.
- Counts more than 500 collaborators.

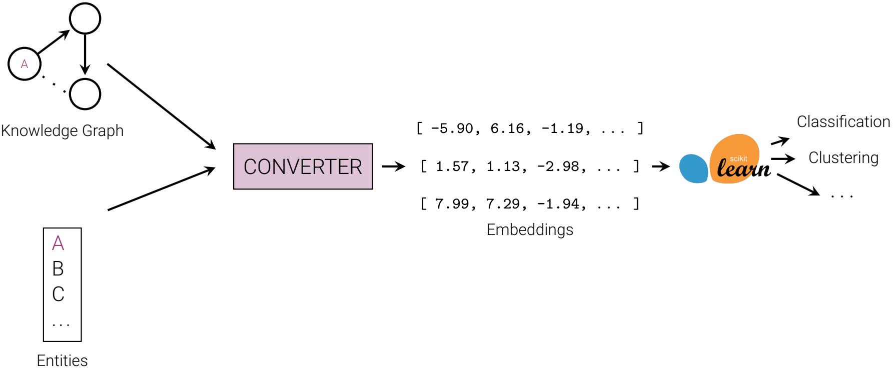
[1] Internet Technology & Data Science Lab.

# Internship

- Associated with the Master’s Thesis over a period of 16 weeks.
- Took place within the Knowledge Management team of the IDLab’s Discover research center cluster which optimizes intelligent agents and their interactions.
- The Knowledge Management team created more than twenty research projects.

# Motivation

# Motivation



A
Knowledge Graph
A
B
C
...
Entities
CONVERTER
[ -5.90, 6.16, -1.19, ... ]
[ 1.57, 1.13, -2.98, ... ]
[ 7.99, 7.29, -1.94, ... ]
Embeddings
scikit learn
Classification
Clustering
...

# Knowledge Graph

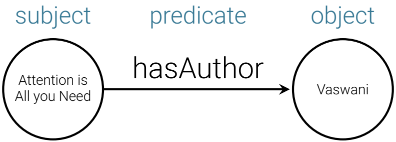


Figure: Triple.

Example of URI[2]: $\overbrace{\texttt{http://dl-learner.org/carcinogenesis}}^{\mathit{URL}\,[3]}\overbrace{\texttt{\#d187}}^{\mathit{URN}\,[4]}$

[2] Uniform Resource Identifier.

[3] Uniform Resource Locator.

[4] Uniform Resource Name.

# Knowledge Graph

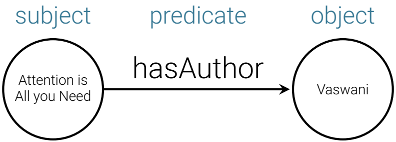


Figure: Triple.

Example of URI[2]: $\overbrace{\texttt{http://dl-learner.org/carcinogenesis}}^{\mathit{URL}^{3}}\overbrace{\texttt{\#d187}}^{\mathit{URN}^{4}}$

[2] Uniform Resource Identifier.

[3] Uniform Resource Locator.

[4] Uniform Resource Name.

# Knowledge Graph

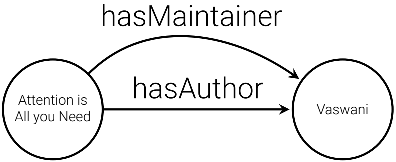


Figure: Knowledge Graph (KG).

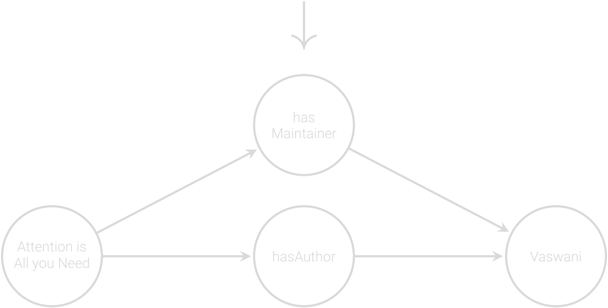


Figure: Oriented Graph.

# Knowledge Graph

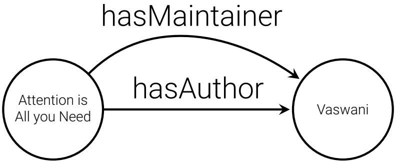


Figure: Knowledge Graph (KG).

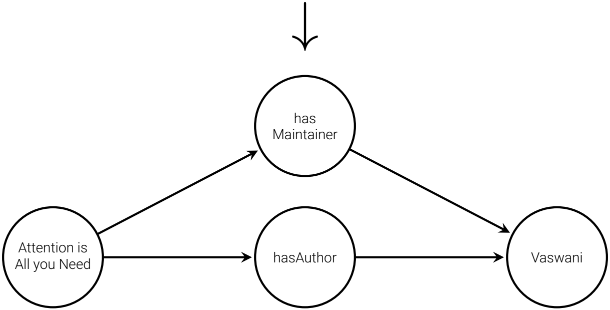
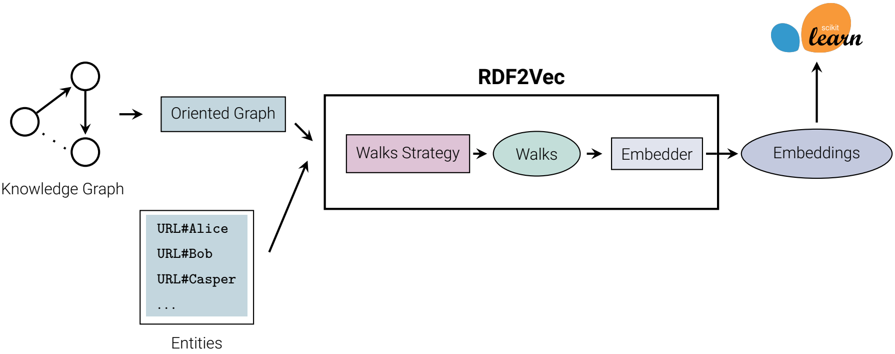


Figure: Oriented Graph.

# RDF2Vec[5]

[5] Resource Description Framework To Vector.

# RDF2Vec



Knowledge Graph

Oriented Graph

URL#Alice
URL#Bob
URL#Casper
...

Entities

RDF2Vec

Walks Strategy

Walks

Embedder

Embeddings

# RDF2Vec

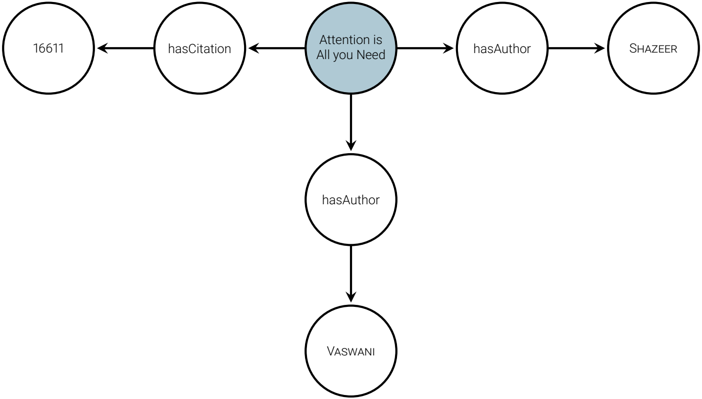


Figure: Walk Extraction for an Oriented Graph (Part I).

Walk Extraction: Attention is All you Need

# RDF2Vec

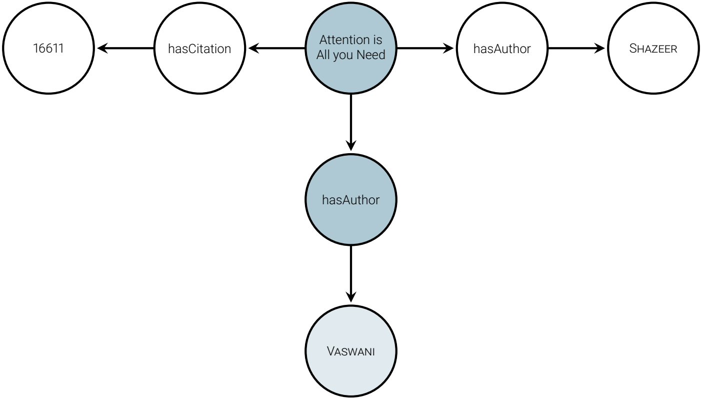


Figure: Walk Extraction for an Oriented Graph (Part II).

Walk Extraction: Attention is All you Need → hasAuthor

# RDF2Vec

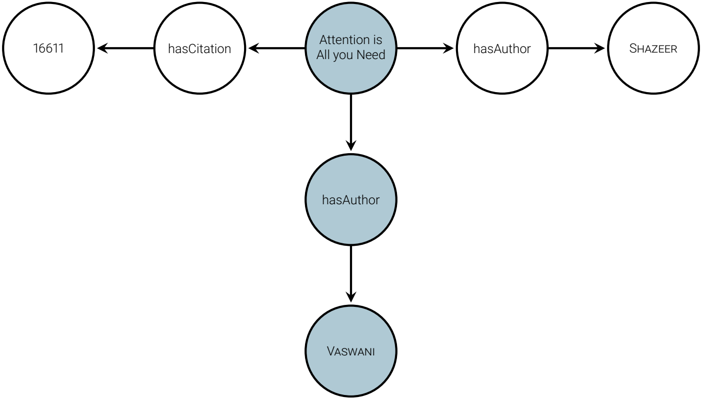


Figure: Walk Extraction for an Oriented Graph (Part III).

Walk Extraction: Attention is All you Need → hasAuthor → VASWANI

# Internship

# Scope of the Internship

Consists in improving `pyRDF2Vec`:

1. Add better support for different KG sizes.
2. Add first support of *literals*.
3. Avoid re-training a model when new nodes[6] are added in the KG.
4. Improve the documentation of the library.
5. Add Continuous Integration / Delivery / Testing.

Where the goals are *ordered according to importance*.

[6] Also called *vertices*.

# Architecture

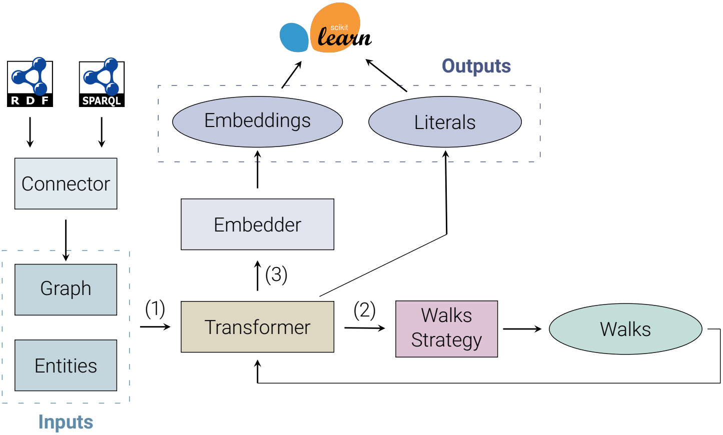


Figure: Workflow of `pyRDF2Vec`.

# Optimization Mechanisms

Consists of reducing the walk extraction time by using:

- a cache memory;
- multiprocessing;
- cache pre-filling;
- an appropriate data structure for the nodes;
- a connection pool;
- ...

# Optimization Mechanism Results

# Benchmarks Setup

| Data Set | Triples | Entities | Relations | Size |
|---|---:|---:|---:|---|
| `MUTAG` | 74,567 | 22,534 | 24 | small |
| `AM` | 5,700,371 | 933,471 | 100 | medium |
| `DBP:Cities` | 651,580,976 | 4,233,000 | 7,992 | large |

Table: Properties of the Data Sets Used for the Benchmarks.

Performed on the IDLab's servers with 4 CPUs and 64 GB RAM.

# Benchmark Results

| Data Set | Speedup | | |
|---|---|---|---|
| | Minimum | Maximum | Average |
| `MUTAG` | 6.20 | 13.28 | 9.74 |
| `AM` | 1.44 | 8.94 | 5.19 |
| `DBP:Cities` | 2.22 | 3.23 | 2.72 |

Table: Speedup Results with Optimization Mechanisms.

- caching and pre-filling benefit small KGs like `MUTAG` the most;
- multiprocessing is more interesting for large KGs like `DBP:Cities`.

# Literals

# Literals

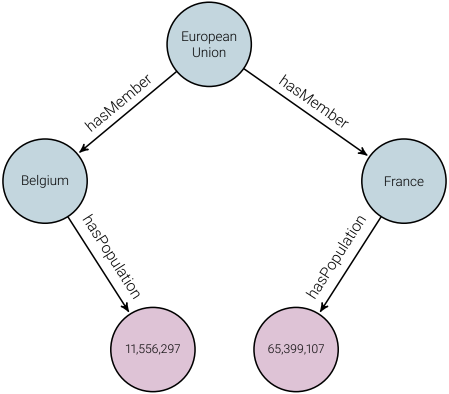


Figure: Oriented Graph with 5 Nodes and 4 Edges.

# Literals

Literals may:

- be specific to some entities;
- be too expensive to extract;
- not be interesting.

→ Let the user specify the literals to extract

For example: [

$$\begin{array}{ll}
[\ \overbrace{\texttt{hasCapital}}^{\texttt{predicate 1}} \rightarrow \overbrace{\texttt{hasLatitude}}^{\texttt{predicate 2}}\ ], & \#\ 50.85;\ 48.85 \\
[\ \texttt{hasCapital} \rightarrow \texttt{hasLongitude}\ ], & \#\ 4.35;\ 2.34 \\
[\ \texttt{hasPopulation}\ ], & \#\ 11{,}55;\ 65{,}39 \\
\ldots & 
\end{array}$$

]

# Literals

Literals may:

- be specific to some entities;
- be too expensive to extract;
- not be interesting.

→ Let the user specify the literals to extract

For example:

```
[
     predicate 1        predicate 2
    ┌─────┴─────┐    ┌──────┴─────┐
  [ hasCapital  →  hasLatitude ],       # 50.85; 48.85
  [ hasCapital  →  hasLongitude ],      # 4.35; 2.34
  [ hasPopulation ],                    # 11,55; 65,39
  ...
]
```

# Online Learning

# Online Learning

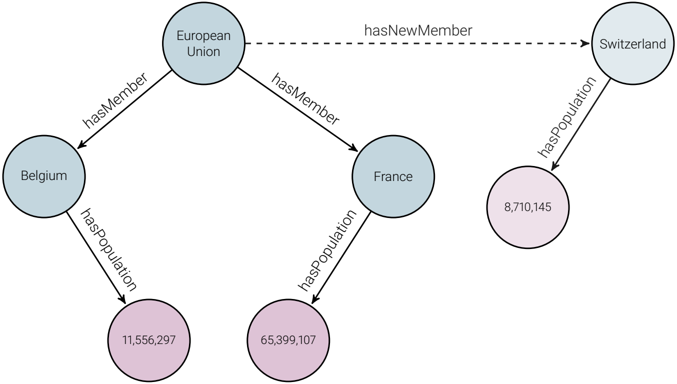


Figure: Oriented Graph With 7 Nodes and 6 Edges.

→ Consists in updating the vocabulary of walks of an initial model.

# Online Learning

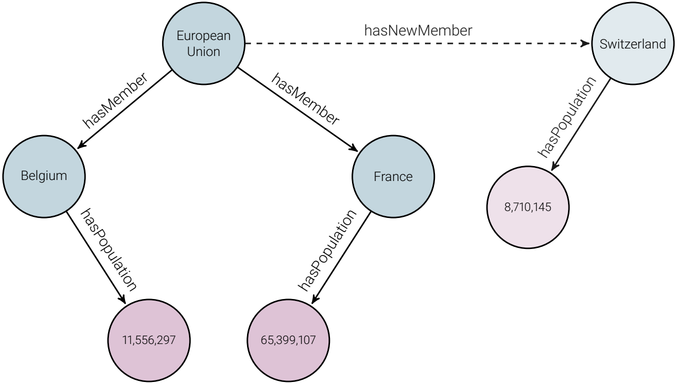


Figure: Oriented Graph With 7 Nodes and 6 Edges.

→ Consists in updating the vocabulary of walks of an initial model.

# Demo Time! 🥁

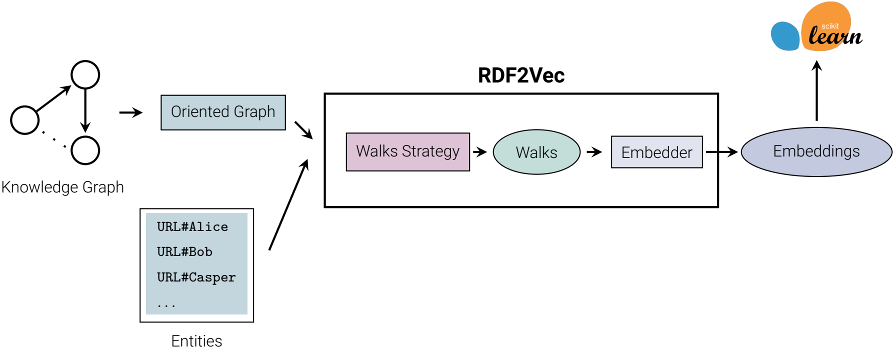
# Demo Time! 🥁

On the menu:

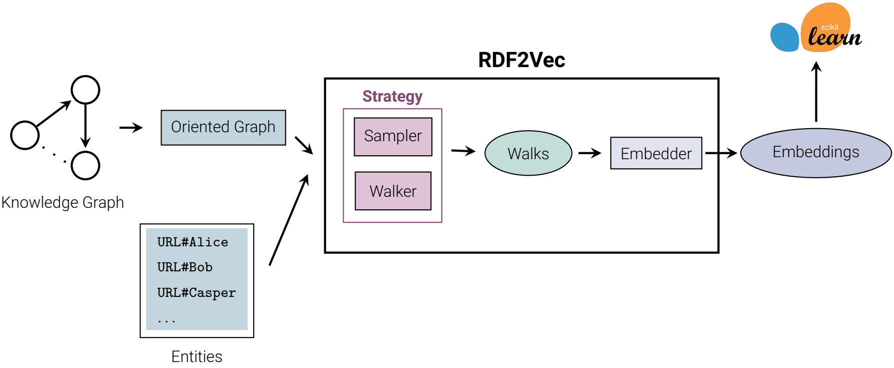
- The `pyRDF2Vec` GitHub repository.
- An example of using literals.
- ~~An example of using online learning~~ (available on GitHub).

# A Trip to Sesame Street: Evaluation of BERT and Other Recent Embedding Techniques Within RDF2Vec

# RDF2Vec


Knowledge Graph
Oriented Graph
URL#Alice
URL#Bob
URL#Casper
...
Entities
RDF2Vec
Walks Strategy
Walks
Embedder
Embeddings
scikit
learn

# RDF2Vec



URL#Alice
URL#Bob
URL#Casper
...

# RDF2Vec

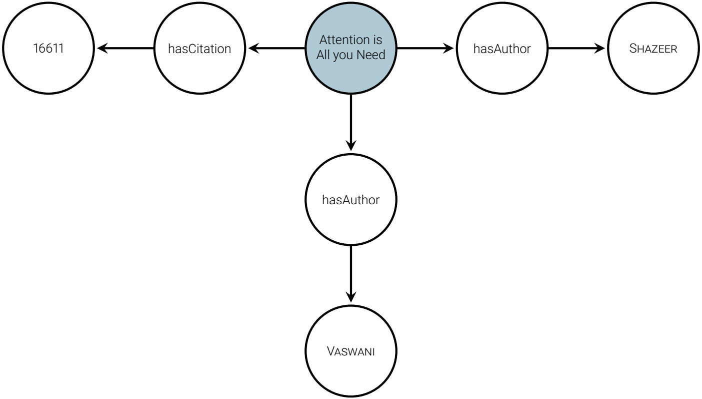


Figure: Oriented Graph.

# RDF2Vec

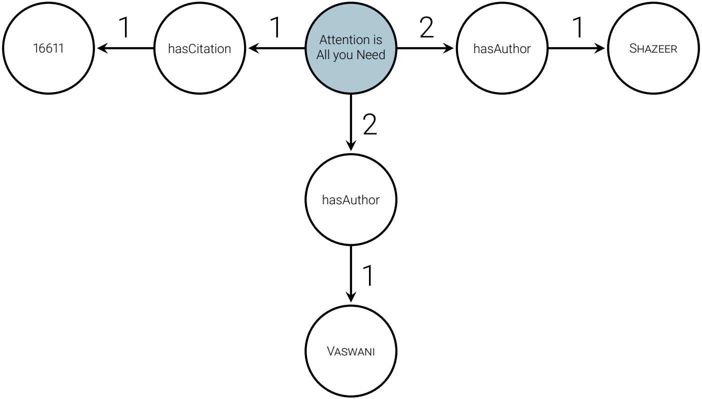


Figure: Walk Extraction for an Oriented Graph (Part I).

Walk Extraction: Attention is All you Need

# RDF2Vec

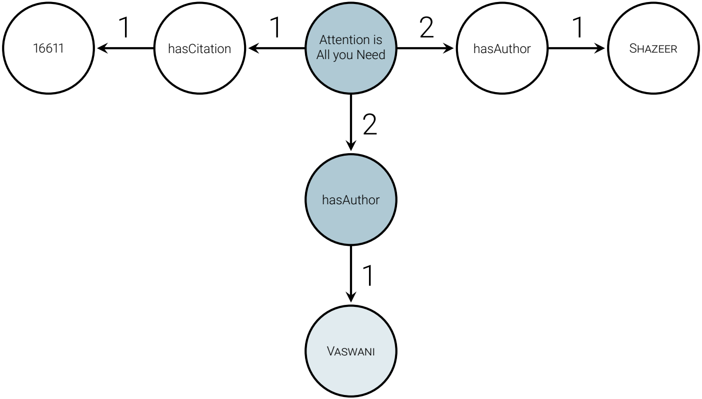


Figure: Walk Extraction for an Oriented Graph (Part II).

Walk Extraction: Attention is All you Need → hasAuthor

# RDF2Vec

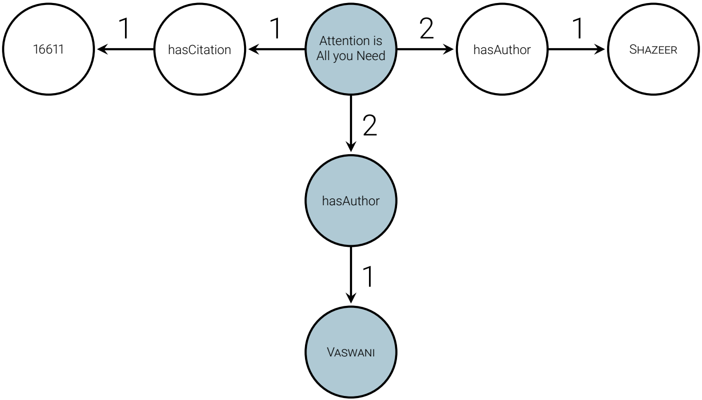
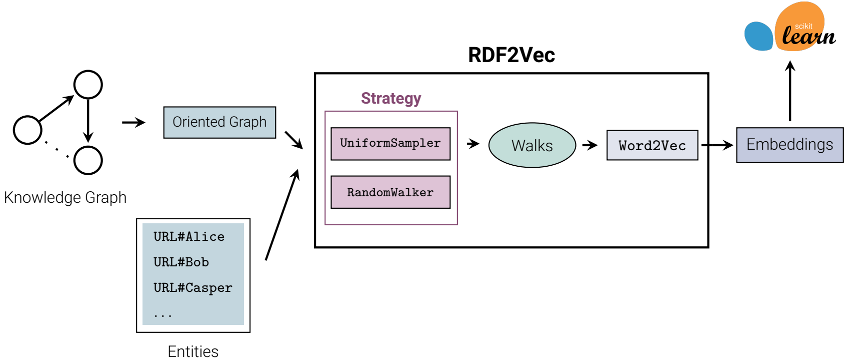


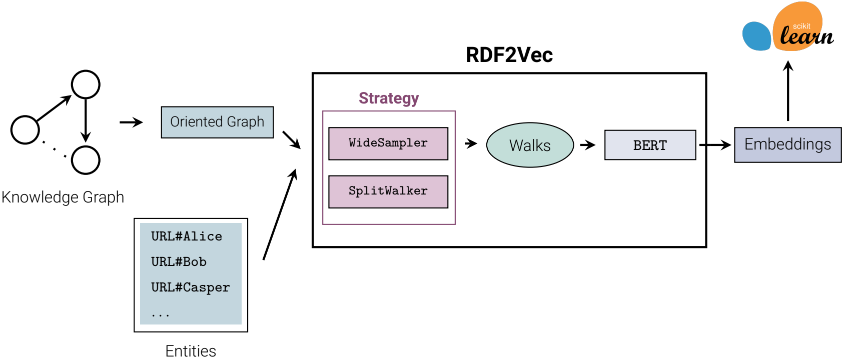
Figure: Walk Extraction for an Oriented Graph (Part III).

Walk Extraction: Attention is All you Need → hasAuthor → VASWANI

# RDF2Vec


RDF2Vec
Knowledge Graph
Oriented Graph
URL#Alice
URL#Bob
URL#Casper
...
Entities
Strategy
UniformSampler
RandomWalker
Walks
Word2Vec
Embeddings
scikit learn

# RDF2Vec


RDF2Vec
Knowledge Graph
Oriented Graph
URL#Alice
URL#Bob
URL#Casper
...
Entities
Strategy
WideSampler
SplitWalker
Walks
BERT
Embeddings
scikit learn

# Scope of the Master's Thesis

Mainly consists in evaluating BERT[7] with Word2Vec and implementing other strategies:

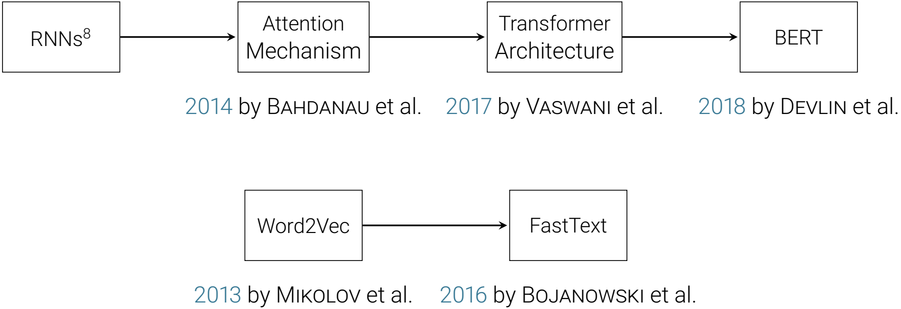
1. Supported BERT and FastText for comparison purposes.
2. Evaluated the impact of BERT with Word2Vec and FastText.
3. `SplitWalker` as a new walking strategy in RDF2Vec.
4. `WideSampler` as a new sampling strategy in RDF2Vec.

BONUS: has unintentionally improved Word2Vec and `pyRDF2Vec`.

[7] Only the traditional BERT model is concerned by this scope.

# BERT

# BERT



RNNs[8] → Attention Mechanism → Transformer Architecture → BERT

2014 by Bahdanau et al. 2017 by Vaswani et al. 2018 by Devlin et al.

Word2Vec → FastText

2013 by Mikolov et al. 2016 by Bojanowski et al.

Figure: Summary Timeline.

[8] Recurrent Neural Networks.

# BERT

Main characteristics:

- Allows one or two input sentences.
- Uses a WordPiece tokenization.

Table: Example of Tokenization With BERT.

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | The | machine | loves | embeddings. | | | | | |
| [CLS] | the | machine | loves | em | ##bed | ##ding | ##s | . | [SEP] |
| 101 | 1996 | 3698 | 7459 | 7861 | 8270 | 4667 | 2015 | 1012 | 102 |

- Uses two pre-training phases:
  1. Masked Language Model (MLM);
  2. Next Sentence Prediction (NSP).
- Contextualizes embeddings using bidirectional representations.

# BERT Implementation[9] With KGs

- Build the vocabulary.
- Fit the BERT model in two ways:
  1. node tokenization: ensuring not to split the nodes;
  2. training: only done with MLM using a data collator.
- Get entity embeddings.

[9] https://github.com/IBCNServices/pyRDF2Vec/blob/feature/bert/pyrdf2vec/embedders/bert.py

# Improve the Model’s Accuracy of Word2Vec

# Improve the Model's Accuracy of Word2Vec

Table: Context Words Determination for a Window Size of 2.

| Input Text | Target Word | Context Words |
|---|---|---|
| I will always remember her | i | will<br>always |
| I will always remember her | will | i<br>always<br>remember |
| I will always remember her | always | i<br>will<br>remember<br>her |
| I will always remember her | remember | will<br>always<br>her |
| I will always remember her | her | always<br>remember |

# Improve the Model's Accuracy of Word2Vec

In two ways:

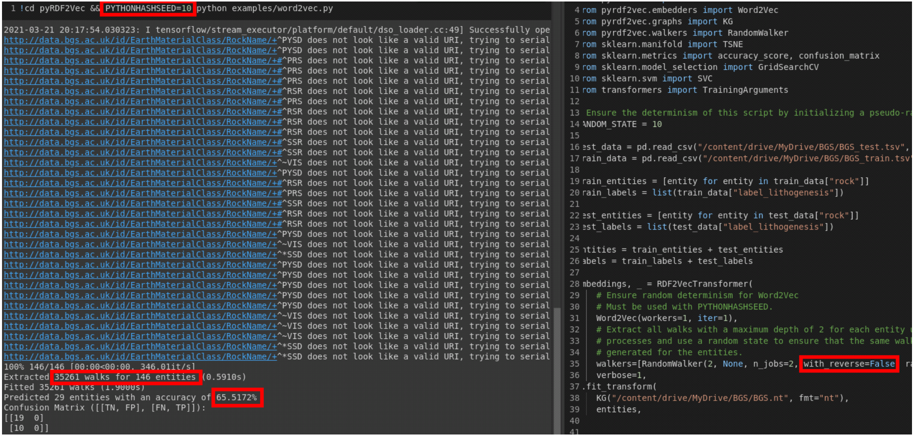
1. Extract the parent nodes of an entity.
2. Better positioning the entity in the walks to maximize its occurrence in the training samples.

For example:

```
("URL#Alice", "URL#knows", "URL#Bob"),
("URL#Alice", "URL#loves", "URL#Bob"),
                 ...
```

May become:

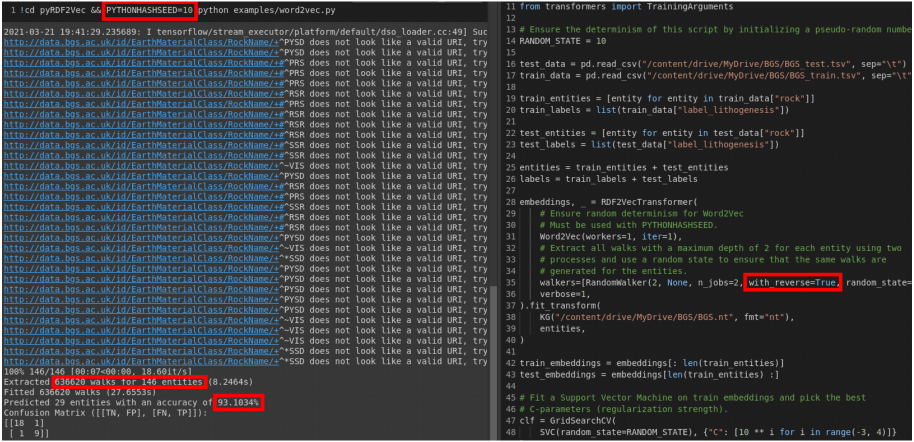
```
 ("URL#Casper", "URL#raised", "URL#Alice", "URL#knows", "URL#Bob"),
("URL#Mathilde", "URL#raised", "URL#Alice", "URL#loves", "URL#Bob"),
                              ...
```



```
!cd pyRDF2Vec && PYTHONHASHSEED=10 python examples/word2vec.py

2021-03-21 20:17:54.030323: I tensorflow/stream_executor/platform/default/dso_loader.cc:49] Successfully ope
http://data.bgs.ac.uk/id/EarthMaterialClass/RockName/+^PYSD does not look like a valid URI, trying to serial
http://data.bgs.ac.uk/id/EarthMaterialClass/RockName/+^PYSD does not look like a valid URI, trying to serial
http://data.bgs.ac.uk/id/EarthMaterialClass/RockName/+#^PRS does not look like a valid URI, trying to serial
http://data.bgs.ac.uk/id/EarthMaterialClass/RockName/+#^PRS does not look like a valid URI, trying to serial
http://data.bgs.ac.uk/id/EarthMaterialClass/RockName/+#^PRS does not look like a valid URI, trying to serial
http://data.bgs.ac.uk/id/EarthMaterialClass/RockName/+#^RSR does not look like a valid URI, trying to serial
http://data.bgs.ac.uk/id/EarthMaterialClass/RockName/+#^PRS does not look like a valid URI, trying to serial
http://data.bgs.ac.uk/id/EarthMaterialClass/RockName/+#^RSR does not look like a valid URI, trying to serial
http://data.bgs.ac.uk/id/EarthMaterialClass/RockName/+#^RSR does not look like a valid URI, trying to serial
http://data.bgs.ac.uk/id/EarthMaterialClass/RockName/+#^RSR does not look like a valid URI, trying to serial
http://data.bgs.ac.uk/id/EarthMaterialClass/RockName/+#^SSR does not look like a valid URI, trying to serial
http://data.bgs.ac.uk/id/EarthMaterialClass/RockName/+#^SSR does not look like a valid URI, trying to serial
http://data.bgs.ac.uk/id/EarthMaterialClass/RockName/+^~VIS does not look like a valid URI, trying to serial
http://data.bgs.ac.uk/id/EarthMaterialClass/RockName/+^PYSD does not look like a valid URI, trying to serial
http://data.bgs.ac.uk/id/EarthMaterialClass/RockName/+#^RSR does not look like a valid URI, trying to serial
http://data.bgs.ac.uk/id/EarthMaterialClass/RockName/+#^PRS does not look like a valid URI, trying to serial
http://data.bgs.ac.uk/id/EarthMaterialClass/RockName/+#^SSR does not look like a valid URI, trying to serial
http://data.bgs.ac.uk/id/EarthMaterialClass/RockName/+#^RSR does not look like a valid URI, trying to serial
http://data.bgs.ac.uk/id/EarthMaterialClass/RockName/+#^RSR does not look like a valid URI, trying to serial
http://data.bgs.ac.uk/id/EarthMaterialClass/RockName/+^PYSD does not look like a valid URI, trying to serial
http://data.bgs.ac.uk/id/EarthMaterialClass/RockName/+^~VIS does not look like a valid URI, trying to serial
http://data.bgs.ac.uk/id/EarthMaterialClass/RockName/+^*SSD does not look like a valid URI, trying to serial
http://data.bgs.ac.uk/id/EarthMaterialClass/RockName/+^PYSD does not look like a valid URI, trying to serial
http://data.bgs.ac.uk/id/EarthMaterialClass/RockName/+^PYSD does not look like a valid URI, trying to serial
http://data.bgs.ac.uk/id/EarthMaterialClass/RockName/+^PYSD does not look like a valid URI, trying to serial
http://data.bgs.ac.uk/id/EarthMaterialClass/RockName/+^PYSD does not look like a valid URI, trying to serial
http://data.bgs.ac.uk/id/EarthMaterialClass/RockName/+^PYSD does not look like a valid URI, trying to serial
http://data.bgs.ac.uk/id/EarthMaterialClass/RockName/+^~VIS does not look like a valid URI, trying to serial
http://data.bgs.ac.uk/id/EarthMaterialClass/RockName/+^~VIS does not look like a valid URI, trying to serial
http://data.bgs.ac.uk/id/EarthMaterialClass/RockName/+^~VIS does not look like a valid URI, trying to serial
http://data.bgs.ac.uk/id/EarthMaterialClass/RockName/+^*SSD does not look like a valid URI, trying to serial
http://data.bgs.ac.uk/id/EarthMaterialClass/RockName/+^*SSD does not look like a valid URI, trying to serial
100% 146/146 [00:00<00:00, 346.01it/s]
Extracted 35261 walks for 146 entities (0.5910s)
Fitted 35261 walks (1.9000s)
Predicted 29 entities with an accuracy of 65.5172%
Confusion Matrix ([[TN, FP], [FN, TP]]):
[[19  0]
 [10  0]]
```

```python
rom pyrdf2vec.embedders import Word2Vec
rom pyrdf2vec.graphs import KG
rom pyrdf2vec.walkers import RandomWalker
rom sklearn.manifold import TSNE
rom sklearn.metrics import accuracy_score, confusion_matrix
rom sklearn.model_selection import GridSearchCV
rom sklearn.svm import SVC
rom transformers import TrainingArguments

 Ensure the determinism of this script by initializing a pseudo-ra
ANDOM_STATE = 10

est_data = pd.read_csv("/content/drive/MyDrive/BGS/BGS_test.tsv",
rain_data = pd.read_csv("/content/drive/MyDrive/BGS/BGS_train.tsv"

rain_entities = [entity for entity in train_data["rock"]]
rain_labels = list(train_data["label_lithogenesis"])

est_entities = [entity for entity in test_data["rock"]]
est_labels = list(test_data["label_lithogenesis"])

ntities = train_entities + test_entities
abels = train_labels + test_labels

nbeddings, _ = RDF2VecTransformer(
   # Ensure random determinism for Word2Vec
   # Must be used with PYTHONHASHSEED.
   Word2Vec(workers=1, iter=1),
   # Extract all walks with a maximum depth of 2 for each entity u
   # processes and use a random state to ensure that the same wal
   # generated for the entities.
   walkers=[RandomWalker(2, None, n_jobs=2, with_reverse=False  ra
   verbose=1,
.fit_transform(
   KG("/content/drive/MyDrive/BGS/BGS.nt", fmt="nt"),
   entities,
```

Figure: Word2Vec (Before Improvement).



```
!cd pyRDF2Vec && PYTHONHASHSEED=10 python examples/word2vec.py

2021-03-21 19:41:29.235689: I tensorflow/stream_executor/platform/default/dso_loader.cc:49] Suc
http://data.bgs.ac.uk/id/EarthMaterialClass/RockName/+^PYSD does not look like a valid URI, try
http://data.bgs.ac.uk/id/EarthMaterialClass/RockName/+^PYSD does not look like a valid URI, try
http://data.bgs.ac.uk/id/EarthMaterialClass/RockName/+#^PRS does not look like a valid URI, try
http://data.bgs.ac.uk/id/EarthMaterialClass/RockName/+#^PRS does not look like a valid URI, try
http://data.bgs.ac.uk/id/EarthMaterialClass/RockName/+#^PRS does not look like a valid URI, try
http://data.bgs.ac.uk/id/EarthMaterialClass/RockName/+#^RSR does not look like a valid URI, try
http://data.bgs.ac.uk/id/EarthMaterialClass/RockName/+#^PRS does not look like a valid URI, try
http://data.bgs.ac.uk/id/EarthMaterialClass/RockName/+#^RSR does not look like a valid URI, try
http://data.bgs.ac.uk/id/EarthMaterialClass/RockName/+#^RSR does not look like a valid URI, try
http://data.bgs.ac.uk/id/EarthMaterialClass/RockName/+#^RSR does not look like a valid URI, try
http://data.bgs.ac.uk/id/EarthMaterialClass/RockName/+#^SSR does not look like a valid URI, try
http://data.bgs.ac.uk/id/EarthMaterialClass/RockName/+#^SSR does not look like a valid URI, try
http://data.bgs.ac.uk/id/EarthMaterialClass/RockName/+^~VIS does not look like a valid URI, try
http://data.bgs.ac.uk/id/EarthMaterialClass/RockName/+^PYSD does not look like a valid URI, try
http://data.bgs.ac.uk/id/EarthMaterialClass/RockName/+#^RSR does not look like a valid URI, try
http://data.bgs.ac.uk/id/EarthMaterialClass/RockName/+#^PRS does not look like a valid URI, try
http://data.bgs.ac.uk/id/EarthMaterialClass/RockName/+#^SSR does not look like a valid URI, try
http://data.bgs.ac.uk/id/EarthMaterialClass/RockName/+#^RSR does not look like a valid URI, try
http://data.bgs.ac.uk/id/EarthMaterialClass/RockName/+#^RSR does not look like a valid URI, try
http://data.bgs.ac.uk/id/EarthMaterialClass/RockName/+^PYSD does not look like a valid URI, try
http://data.bgs.ac.uk/id/EarthMaterialClass/RockName/+^~VIS does not look like a valid URI, try
http://data.bgs.ac.uk/id/EarthMaterialClass/RockName/+^*SSD does not look like a valid URI, try
http://data.bgs.ac.uk/id/EarthMaterialClass/RockName/+^PYSD does not look like a valid URI, try
http://data.bgs.ac.uk/id/EarthMaterialClass/RockName/+^PYSD does not look like a valid URI, try
http://data.bgs.ac.uk/id/EarthMaterialClass/RockName/+^PYSD does not look like a valid URI, try
http://data.bgs.ac.uk/id/EarthMaterialClass/RockName/+^PYSD does not look like a valid URI, try
http://data.bgs.ac.uk/id/EarthMaterialClass/RockName/+^PYSD does not look like a valid URI, try
http://data.bgs.ac.uk/id/EarthMaterialClass/RockName/+^~VIS does not look like a valid URI, try
http://data.bgs.ac.uk/id/EarthMaterialClass/RockName/+^~VIS does not look like a valid URI, try
http://data.bgs.ac.uk/id/EarthMaterialClass/RockName/+^~VIS does not look like a valid URI, try
http://data.bgs.ac.uk/id/EarthMaterialClass/RockName/+^*SSD does not look like a valid URI, try
http://data.bgs.ac.uk/id/EarthMaterialClass/RockName/+^*SSD does not look like a valid URI, try
100% 146/146 [00:07<00:00, 18.60it/s]
Extracted 636620 walks for 146 entities (8.2464s)
Fitted 636620 walks (27.6553s)
Predicted 29 entities with an accuracy of 93.1034%
Confusion Matrix ([[TN, FP], [FN, TP]]):
[[18  1]
 [ 1  9]]
```

```
from transformers import TrainingArguments

# Ensure the determinism of this script by initializing a pseudo-random numbe
RANDOM_STATE = 10

test_data = pd.read_csv("/content/drive/MyDrive/BGS/BGS_test.tsv", sep="\t")
train_data = pd.read_csv("/content/drive/MyDrive/BGS/BGS_train.tsv", sep="\t"

train_entities = [entity for entity in train_data["rock"]]
train_labels = list(train_data["label_lithogenesis"])

test_entities = [entity for entity in test_data["rock"]]
test_labels = list(test_data["label_lithogenesis"])

entities = train_entities + test_entities
labels = train_labels + test_labels

embeddings, _ = RDF2VecTransformer(
    # Ensure random determinism for Word2Vec
    # Must be used with PYTHONHASHSEED.
    Word2Vec(workers=1, iter=1),
    # Extract all walks with a maximum depth of 2 for each entity using two
    # processes and use a random state to ensure that the same walks are
    # generated for the entities.
    walkers=[RandomWalker(2, None, n_jobs=2, with_reverse=True, random_state=
    verbose=1,
).fit_transform(
    KG("/content/drive/MyDrive/BGS/BGS.nt", fmt="nt"),
    entities,
)

train_embeddings = embeddings[: len(train_entities)]
test_embeddings = embeddings[len(train_entities) :]

# Fit a Support Vector Machine on train embeddings and pick the best
# C-parameters (regularization strength).
clf = GridSearchCV(
    SVC(random_state=RANDOM_STATE), {"C": [10 ** i for i in range(-3, 4)]}
```

Figure: Word2Vec (After Improvement).

# FastText

# FastText

Table: Sub-word Generation for Character *N*-Grams of Length 3, 4, 5, and 6.

| **Word** | **Length** | **Character *n*-grams** |
|---|---|---|
| flying | 3 | <fl, fly, lyi, yin, ing, ng> |
| | 4 | <fly, flyi, lyin, ying, ing> |
| | 5 | <flyi, lying, ying> |
| | 6 | <flyin, flying, lying> |

Mainly solves the embeddings creation for Out-Of-Vocabulary words.

# FastText Implementation[11] With KGs

Use the **`gensim`** library by:

1. Removing the **`min_n`** and **`max_n`** parameters for *n*-grams splitting.
2. Allowing to compute *n*-grams for walks only[10] by separating the objects and predicates URIs with a splitting function.
3. Avoiding dependency on Cython.

[10] The object nodes in **`pyRDF2Vec`** are encoded in MD5 to reduce their storage in RAM.

[11] https://github.com/IBCNServices/pyRDF2Vec/blob/feature/bert/pyrdf2vec/embedders/fasttext.py

# Benchmarks

# Benchmarks

- Varied the maximum:
  1. number of walks per entity;
  2. depth per walk.
- Performed on `MUTAG`.
- Use 320 training entities and attempt to predict 68 test entities.
- Take the mean and standard deviation of five tests.

Embedding techniques and walking strategies:

- IDLab's servers with 4 CPUs, 64 GB RAM, and one GPU.

Sampling strategies:

- ThinkPad machine with 4 CPUs, 16 GB RAM.

# Benchmarks for Embedding Techniques

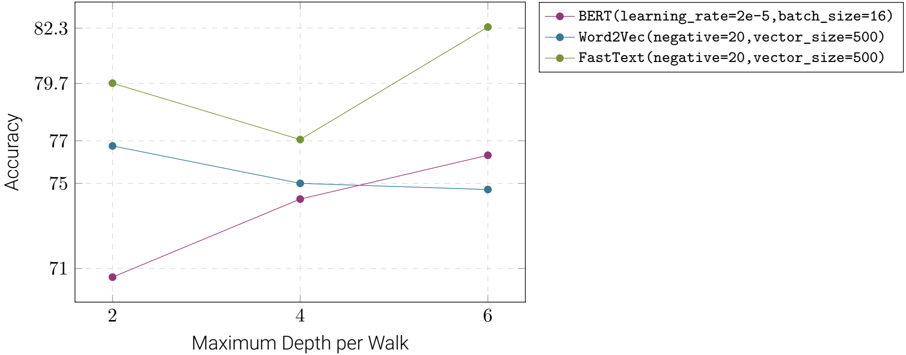


Figure: Embedding Techniques for `MUTAG` (Maximum Depth per Walk).

# Benchmarks for Embedding Techniques

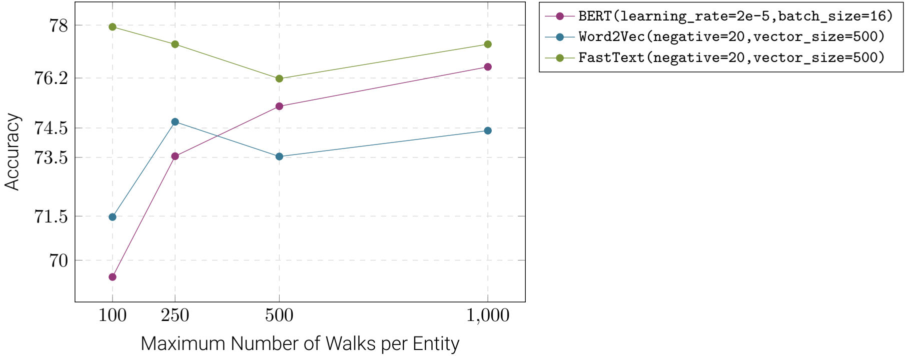


Figure: Embedding Techniques for `MUTAG` (Maximum Number of Walks).

# Strategies

# SplitWalker[12]

- Extension of the `RandomWalker`.
- Based on the principle of preprocessing.
- Customizable by a splitting function.
- Inspired by the way FastText works.

[12] https://github.com/IBCNServices/pyRDF2Vec/blob/master/pyrdf2vec/walkers/split.py

# SplitWalker

|  | Initial Node | Node After Splitting |
|---|---|---|
| Walk 1 | http://dl-learner.org/carcinogenesis#d19 | http://dl-learner.org/carcinogenesis#d19 |
|  | http://dl-learner.org/carcinogenesis#hasBond | has<br>bond |
|  | http://dl-learner.org/carcinogenesis#bond3209 | 3209 |
| Walk 2 | http://dl-learner.org/carcinogenesis#d36 | http://dl-learner.org/carcinogenesis#d36 |
|  | http://www.w3.org/1999/02/22-rdf-syntax-ns#type | type |
|  | http://dl-learner.org/carcinogenesis#bond | bond |

Table: Walk Transformation With `SplitWalker`.

# Benchmarks for Walking Strategies

| Walker | Average Rank |
|---|---|
| `HALKWalker(freq_threshold=0.01)` | 1 |
| `SplitWalker` | 2 |
| `RandomWalker` | 3 |
| `NGramWalker(grams=3)` | 4 |
| `WalkletWalker` | 5 |
| `AnonymousWalker` | 6 |

Table: Average Rank of the Walking Strategies.

# WideSampler[13]

- Maximizes the extraction of shared features between entities.
- Assumes that single entity-specific features have a negligible impact on the quality of the generated embeddings.
- Assigns higher weights to edges that lead to the largest number of:
  1. predicates and objects in the neighborhood;
  2. occurrence of predicates and objects in a graph.

[13] https://github.com/IBCNServices/pyRDF2Vec/blob/master/pyrdf2vec/samplers/wide.py

# Benchmarks for Sampling Strategies

| Sampler | Average Rank |
|---|---|
| WideSampler | 1 |
| ObjPredFreqSampler(inverse=True) | 2 |
| PredFreqSampler | 3 |
| ObjFreqSampler | 4 |
| ObjFreqSampler(inverse=True) | 5 |
| PageRankSampler(inverse=True,alpha=0.85) | 6 |
| PageRankSampler(inverse=True,split=True,alpha=0.85) | 7 |
| ObjFreqSampler(inverse=True,split=True) | 8 |
| PageRankSampler(split=True,alpha=0.85) | 9 |
| PredFreqSampler(inverse=True) | 10 |
| ObjPredFreqSampler | 11 |
| UniformSampler | 11 |
| PageRankSampler(alpha=0.85) | 12 |

Table: Average Rank of the Sampling Strategies.

# Future Work

# Future Work

- Inject `(subject,object)` pairs to BERT instead of pairs of walks.
- Evaluate:
    - the KG-oriented BERT models with other embedding techniques;
    - BERT, `SplitWalker` and `WideSampler` on larger KGs.
- If necessary, implement a new KG-oriented BERT model.

# Conclusion

# Conclusion

- The traditional BERT model has not been as effective as expected: small gain for the model's accuracy in counterpart of a more important training time and storage space.
- `WideSampler` and `SplitWalker` finished first and second respectively in average rank of the `MUTAG` benchmarks.
- Improved the accuracy of the Word2Vec model.
- Gain in popularity for `pyRDF2Vec`.

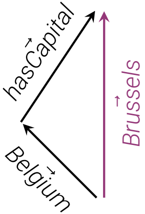
# Question Time 🐈

# Appendices

# Related Work

Three categories of conversion algorithms based:

1. Direct encoding (e.g., TransE[14]);



$\overrightarrow{hasCapital}$
$\overrightarrow{Brussels}$
$\overrightarrow{Belgium}$

Figure: Embedding Computations With TransE.

2. Deep Learning (e.g., R-GCNs[15]);
3. Path/Walk (e.g., RDF2Vec[16]).

[14] Translating Embeddings.

[15] Modeling Relational Data with Graph Convolutional Networks.

[16] Resource Description Framework To Vector.

# Continuous Bag-of-Words Model

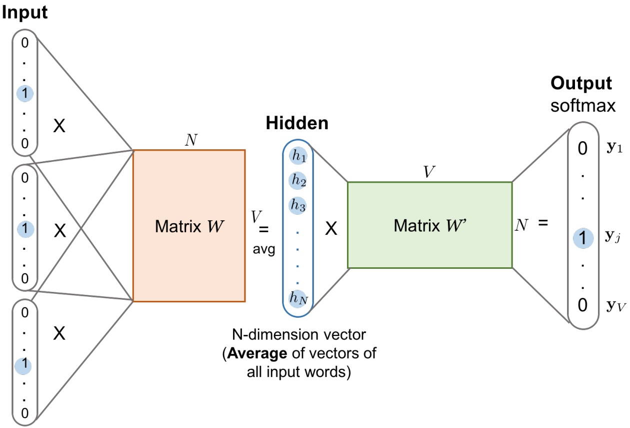


Figure: CBOW Model Architecture.

# Skip-Gram Model

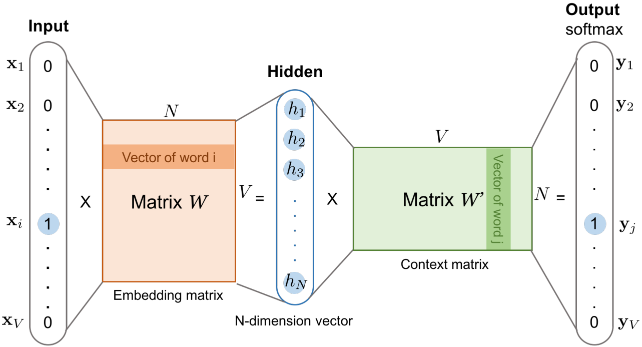


Figure: Skip-Gram Model Architecture.

# Recurrent Neural Networks

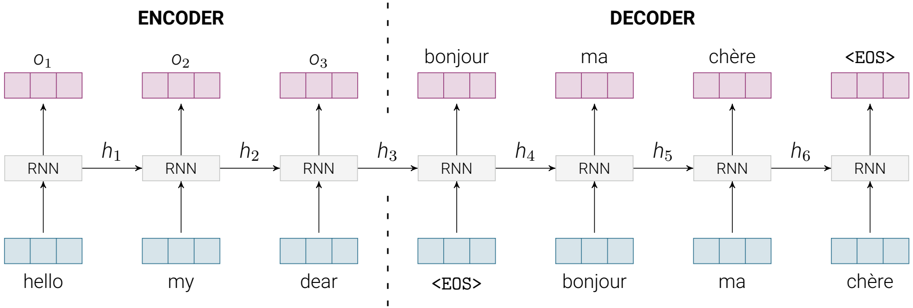


Figure: Sequence-to-Sequence Learning With RNNs.

# Attention Mechanism

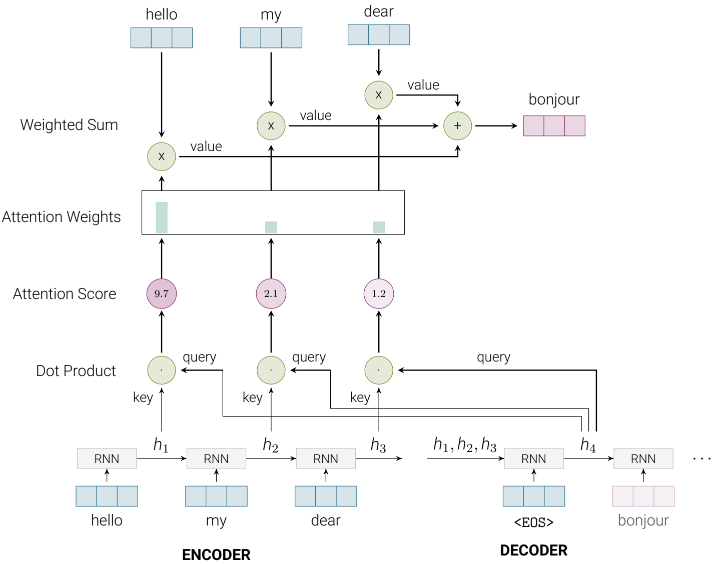


Figure: Seq2Seq Learning With Attention Mechanism.

# Transformer

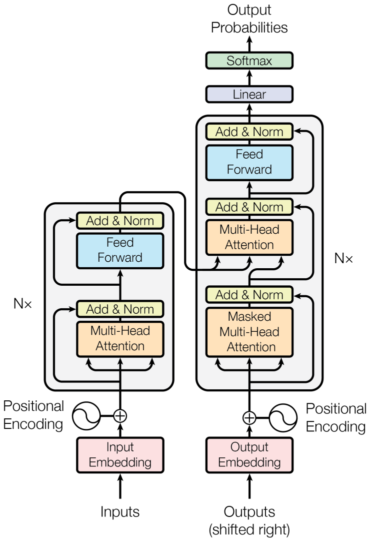
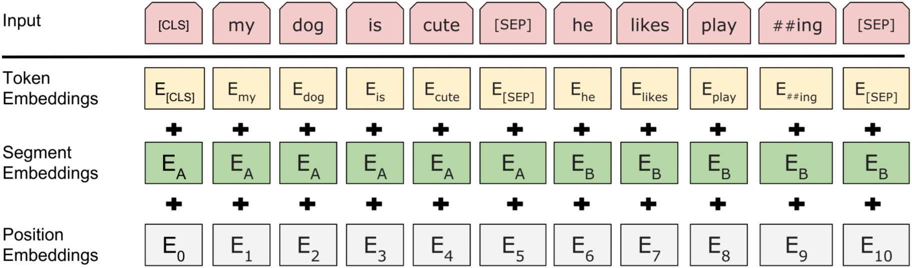


Figure: Transformer Model Architecture.

Source: VASWANI et al. – Attention Is All You Need

# Input Embeddings



| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Input | [CLS] | my | dog | is | cute | [SEP] | he | likes | play | ##ing | [SEP] |
| Token Embeddings | $E_{\text{[CLS]}}$ | $E_{\text{my}}$ | $E_{\text{dog}}$ | $E_{\text{is}}$ | $E_{\text{cute}}$ | $E_{\text{[SEP]}}$ | $E_{\text{he}}$ | $E_{\text{likes}}$ | $E_{\text{play}}$ | $E_{\text{\#\#ing}}$ | $E_{\text{[SEP]}}$ |
| | + | + | + | + | + | + | + | + | + | + | + |
| Segment Embeddings | $E_A$ | $E_A$ | $E_A$ | $E_A$ | $E_A$ | $E_A$ | $E_B$ | $E_B$ | $E_B$ | $E_B$ | $E_B$ |
| | + | + | + | + | + | + | + | + | + | + | + |
| Position Embeddings | $E_0$ | $E_1$ | $E_2$ | $E_3$ | $E_4$ | $E_5$ | $E_6$ | $E_7$ | $E_8$ | $E_9$ | $E_{10}$ |

Figure: Example of Sentence Pair Encoding.

Source: Devlin et al. – BERT

# Masked Language Model

Table: Example of MLM With BERT.

| | | | | | | |
|---|---|---|---|---|---|---|
| | | | | best | | |
| | | | | ↑ | | |
| GNU | Emacs | is | the | [MASK] | text | editor |

Each input sequence usually has 15 % of its tokens hidden where:

- a token is replaced by a [MASK] token in 80 % of the cases;
- a token is replaced randomly in 10 % of the cases;
- a token remains unchanged in 10 % of the cases.

# Next Sentence Prediction

Table: Example of NSP With BERT.

| **Sentence A** | **Sentence B** | **Label** |
|---|---|---|
| GNU Emacs is the best text editor. | I should use it. | `isNextSentence` |
| GNU Emacs is the best text editor. | I have a cat. | `NotNextSentence` |

# WideSampler

```
Algorithm get_weight(h, d, c)
Require: a $\mathcal{H}$ 2-tuple that contains a predicate and an object.
Require: a $\mathcal{D}$ array of $n \geq 1$ degree indexed from $0$ to $n - 1$
Require: a $\mathcal{C}$ array of $n \geq 1$ counter of neighbors indexed from $0$ to $n - 1$
Ensure: The weight of the hop for this predicate
if $\mathcal{D}_{preds}$ and $\mathcal{D}_{objs}$ and $\mathcal{C}$ then
    return $(\mathcal{C}[\mathcal{H}[0]_{name}] + \mathcal{C}[\mathcal{H}[1]_{name}]) \left(\frac{\mathcal{D}_{preds}[\mathcal{H}[0]_{name}] + \mathcal{D}_{objs}[\mathcal{H}[1]_{name}]}{2}\right)$
end if
```

Listing: Splits Nodes of Random Walks with `SplitWalker` (Part I).

```python
def basic_split(self, walks):
  canonical_walks = set()
  for walk in walks:
    canonical_walk = [walk[0].name]
    for i, _ in enumerate(walk[1::], 1):
      vertices = []
      if "http" in walk[i].name:
        vertices = " ".join(re.split("[\#]", walk[i].name)).split()
      if i % 2 == 1:
        name = vertices[1] if vertices else walk[i].name
        preds = [
          sub_name
          for sub_name in re.split(r"([A-Z][a-z]*)", name)
          if sub_name
        ]
        for pred in preds:
        canonical_walk += [pred.lower()]
```

Listing: Splits Nodes of Random Walks with `SplitWalker` (Part II).

```python
...
    else:
      name = vertices[-1] if vertices else walk[i].name
      objs = []
      try:
        objs = [str(float(name))]
      except ValueError:
        objs = re.sub("[^A-Za-z0-9]+", " ", name).split()
        if len(objs) == 1:
          match = re.match(
            r"([a-z]+)([0-9]+)", objs[0], re.I
          )
          if match:
            objs = list(match.groups())
      for obj in objs:
        canonical_walk += [obj.lower()]
  canonical_walk = list(dict(zip(canonical_walk, canonical_walk)))
  canonical_walks.add(tuple(canonical_walk))
  return canonical_walks
```