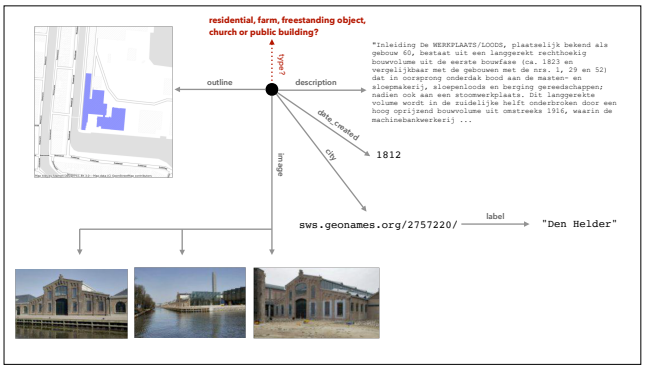
kgbench

a collection of knowledge graph datasets for evaluating relational and multimodal machine learning

Peter Bloem Xander Wilcke Lucas van Berkel Victor de Boer

Knowledge representation & reasoning
User centric data science

VU VRIJE UNIVERSITEIT AMSTERDAM



Imagine if you were told about a building, and shown the outline of its floorplan. Could you guess what its function was?

You might guess that it's not a church (or at least not an old one), and that it's not a single residential house. If I gave you more information, perhaps a description in natural language, you might be able to discern more. I could also show you a picture, and perhaps give you some relational data, including when the building was built and where it's located.

By integrating all of these source of information: visual, natural language, geographical, you can **learn** a lot more than from just a single source.

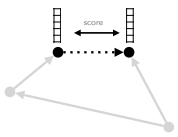
most knowledge is multimodal and relational

That is the motivation behind this research. Most knowledge that we have, looks like this: various modalities, strung together with relational connections. Knowledge graphs are a very natural representation for this kind of data, as the previous slide shows, so if we want to do machine learning in this kind of setting---learning from all available modalities---and the relational connections between them, we could do worse than to investigate machine learning on knowledge graphs.

what's available?

embedding methods

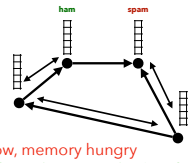
DistMult, ComplEx, SimplE, etc.



fast
link prediction only
one-hop "reasoning"

message passing methods

R-GCN, RGAT, etc.



slow, memory hungry
link prediction, node classification
multi-hop "reasoning"

So let's look at what kind of methods are available so far. Probably the most popular approach are the *embedding models*. These learn a single vector representation for each node in the graph, and crucially, to predict whether a triple is true, look only at the embeddings for the subject and object of that triple. Any information beyond this one-hop neighborhood needs to be included implicitly through the embeddings.

An alternative approach is **message passing**. This extends the embedding vectors representing the nodes by pooling information from deeper in the graph. Such models are slower and more memory hungry than the simple embedding models, but they can use information from deeper in the graph more explicitly.

They can also be used to solve **node classification**

problems, for which simple embedding models are not suitable. We believe that for integrating information in a rich multimodal knowledge, message passing models are the best candidates and that node classification is the simplest testing ground for evaluation.

MP for relational and multimodal data **what's stopping us?**

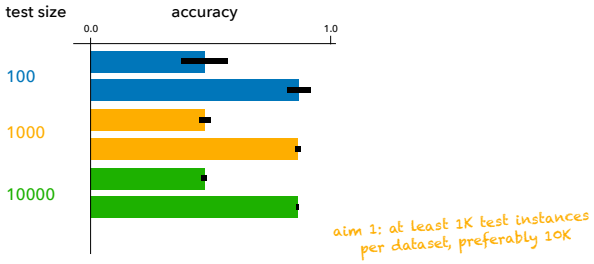
Dataset	AIFB	MUTAG	BGS*	AM*
Entities	8 285	23 644	87 688	246 728
Relations	45	23	70	122
Edges	29 043	74 227	230 698	875 946
Labeled	176	340	146	1 000
Classes	4	2	2	11

So, if message passing is such a great framework, what's stopping us? Right, now: a lack of good data. Here are four of the most popular datasets.

The main problem is not the quality of the data, but the number of labeled nodes available as test data. This is the entirety of the labeled data, so if we split off a test set of 20% that can go as low as 30 instances.

---

1. small test sets



The reason that this is a problem is that how well we can compare models depends on the size of the test set. We may compute classification accuracy and be confident that our model is far better than our baseline, but with only 100 instances in our test data, we can see that the error bars are actually very big. If our baseline is 50% accurate and our model is 60% accurate, 100 instances is not enough to then claim that our model has a better performance. Or, if our baseline scores 90% and we score 95%, we may still get overlapping error bars.

At 1000 instances in our data, things start to look a little better, but really for a long-lasting benchmark, we would want something like 10000 instances in our test set (and our validation set as well).

This is our first aim: to create datasets with at leats 1000 test instances, but preferably an order of magnitude more.

---

2. large volume per instance

Dataset	AIFB	MUTAG	BGS*	AM*
Entities	8 285	23 644	87 688	246 728
Relations	45	23	70	122
Edges	29 043	74 227	230 698	875 946
Labeled	176	340	146	1 000
Classes	4	2	2	11

GPU memory limit

aim 2: a medium-sized, information-rich knowledge graph

Now, to achieve this, we can't just take this kind of dataset and scale it up 50 times to get a test set of 10K instances. If we did that, just training a single message passing model would become a great engineering challenge. We do eventually want to train on such data, of course, but for easy model development, we need small datasets, so that we can iterate fast, and allow people with limited hardware resources to perform research.

So, our second aim is to achieve a modest total size of graph, that is still rich in information. To achieve this, we make a trade-off. We remove a lot of information that we judge to be unlikely to be helpful for the classification task in hand. This makes the data less representative of knowledge graphs in the wild, but better suited for fast iteration and model development.

3. lack of multimodal data (in the graph)

AM*
246 728
122
875 946
1 000
11

aim 3: (optional) multimodal information embedded in the graph

Some of these earlier benchmarks do contain multimodal information. The Amsterdam Museum dataset, for instance contains links to images for many items in the collection. However, these are given as URLs to images on the original collection website. This makes it difficult to run reproducible experiments. Each practitioner needs to download all images, and there is no guarentee that these will be stable or always available.

For a proper machine learning task, all data should be self-contained. Ideally, we feel, inside the graph. That way different practitioners can be sure that experiments were run on exactly the dame data.

---

embedded multimodal data

<https://data.labs.pdok.nl/rce/id/image/20349547> <http://dbpedia.org/ontology/thumbnail>
"_9j_4AAQSkZJRgABAQAAAQABAAD_2wBDAAgGBgcGBQgHBwcJCQgKDBQNDAsLDBkSEw8UHRofIh0aHBwgJC4nICIsIxwcKDc
pLDAxNDQ0Hyc5PTgyPC4zNDL_2wBDAQkJCQwLDBgNDRgyIRwhMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjIyMjI
yMjIyMjIyMjIyMjL_wAARCADIAJY{...jeuaC5yRnPpVbzcmjf3olHYsE5phY5qPfzszhh8AO3EUUwsPWigCEkHkcUH7
YA_Wiiuy5jYrBBiScGowrE9cUUUmwJAXUDuKRn67gRRRRAjMuDwDU8M4Jw1FPTIpFsSLjiguDxRWTKGFgKj3AmiipGB5GBT
MEd6XKQIAxHeniQg80UUFDg460bxnNFFBINZaKKKBn_9k="^^<http://kgbench.info/dt#base64Image> .


```
:base64Image a rdfs:Datatype ;
    rdfs:subClassOf xsd:base64Binary ;
    rdfs:label "Base64-encoded image"@en ;
    rdfs:comment "An image encoded as a base64 string"@en .

:base64Video a rdfs:Datatype ;
    rdfs:subClassOf xsd:base64Binary ;
    rdfs:label "Base64-encoded video"@en ;
    rdfs:comment "A video encoded as a base64 string"@en .

:base64Audio a rdfs:Datatype ;
    rdfs:subClassOf xsd:base64Binary ;
    rdfs:label "Base64-encoded audio"@en ;
    rdfs:comment "An audio sequence encoded as a base64 string"@en .
```

To make the datasets self-contained, we embed the multimodal data inside the RDF as literals. For many data types, such as natural language, dates and measurements, there are literal annotations available.

For the images, we encode these in base64. There is an XSD datatype available., but it offers insufficient information for quite how to interpret the literal. We introduce the following micro-ontology to annotate these strings with enough information to encode them successfully (only the Image datatype is currently used in our data).

---

kgbench

Dataset		amplus	dmgfull	dmg777k	dblp	mdgenre	mdgender*
Triples		2 521 046	1 850 451	777 124	21 985 048	1 252 247	1 203 789
Relations		33	62	60	68	154	154
Nodes.		1 153 679	842 550	341 270	4 470 778	349 344	349 347
	entities	1 026 162	262 494	148 127	4 231 513	191 135	191 138
	literals	127 517	580 056	192 143	239 265	158 209	158 209
Density		$2{\cdot}10^{-6}$	$3{\cdot}10^{-6}$	$7{\cdot}10^{-6}$	$1{\cdot}10^{-6}$	$1{\cdot}10^{-5}$	$1{\cdot}10^{-5}$
Degree	avg	4.37	4.47	4.53	9.83	7.17	6.89
	min	1	1	1	1	1	1
	max	154 828	121 217	65 576	3 364 084	57 363	57 363
Classes		8	14	5	2	12	9
Labeled	total	73 423	63 565	8 399	86 535	8 863	57 323
	train	13 423	23 566	5 394	26 535	3 846	27 308
	valid	20 000	10 001	1 001	20 000	1 006	10 005
	test	20 000	20 001	2 001	20 000	3 005	10 003
	meta	20 000	10 001		20 000	1 006	10 007
Source		[5]	see text		[29][31][21]	[31][12]	[31][12]
GPU friendly		✓		✓			
CPU friendly		✓	✓	✓		✓	✓
Datatypes[6]							
Numerical		8 418	64 184	8 891		1 387	1 387
Temporal		6 676.	463	290		37 442	37 442
Textual		56 202	340 396	117 062	239 265	51 852	51 852
Visual		56 130	58 791	46 061		67 528	67 528
Spatial			116 220	20 837			

So , here are the datasets we came up with. As you can see, the amount of test and validation data is never less than 1000 instances, and usually more than 10 000.

We tested whether a simple 2-layer RGCN could be trained on each. If this could be done 12 Gb of memory, we call the dataset GPU friendly and if it could be done in 64Gb of memory we call the dataset CPU friendly. This is pretty arbitrary of course, but it gives a general idea of what can be achieved with a single CPU or GPU compute node.

And finally, most datasets have rich multimodal data, including images, and even some geographical outlines as shown in the first slide.

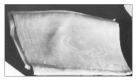
Amsterdam Museum (`amplus`)

AM*
246 728
122
875 946
1 000
11

- include all AM artifacts: 1 000 to 73 423 instances
- Reduce artifact classes: 17 to 8
- Reduce number of relations: 122 to 33
- Include images as base64 literals
- Annotate other literals with datatype

classes:
Decorative art
Prints
Historical artifacts
Drawings
Metallic art
Books & documents
Paintings
Photographs



AMSTERDAM MUSEUM

Dutch Monument Graph (`dmgfull`, `dmg777k`)

integrates information from:

- the Dutch Cultural Heritage Agency
  www.cultureelerfgoed.nl
- the Dutch Cadastre, Land Registry and Mapping Agency
  www.kadaster.nl
- Statistics Netherlands
  www.cbs.nl
- Geonames
  www.geonames.org

classes:

residential	windmill
farm	road, waterway
freestanding object	castle manor
church	charity
public building	catering industry
defensive	part of church
archaeological	part of building

"Inleiding De WERKPLAATS/LOODS, plaatselijk bekend als gebouw 60, bestaat uit een langgerekt rechthoekig bouwvolume uit de eerste bouwfase (ca. 1823 en vergelijkbaar met de gebouwen met de nrs. 1, 29 en 52) dat in oorsprong onderdak bood aan de masten- en sloepmakerij, sloepenloods en berging gereedschappen; nadien ook aan een stoomwerkplaats. Dit langgerekte volume wordt in de zuidelijke helft onderbroken door een hoog

The Dutch Monument graph is the data we saw in the first slide. It contains a large amount of multimodal data about monumental buildings in the data.
The dmg777k version is limited to only the top 5 classes. This reduces the available test and validation data, but it provides a smaller number of classes, and a more uniform class distribution, which is beneficial for training.

---

movie genres (`mdgenre`)

integrates information from:

- Wikidata (movies and actors)
- IMDb (portraits and posters)

classes:

drama
action
literature-based
fiction
comedy-drama
romance
fantasy
musical
documentary
genre movie
comedy

`<http://www.wikidata.org/entity/Q34012> <http://www.wikidata.org/prop/direct/P26> <http://www.wikidata.org/entity/Q269037>`

Marlon Brando — spouse — Movita Castaneda

Actor gender (`mdgender*`)

- Sanity-check task. *Not* an official part of the benchmark.
- Potential for study of bias in classification.

classes:

female
male
cisgender
transgender person
transgender female
transgender male
non-binary
cisgender male
gender fluid

`<http://www.wikidata.org/entity/Q34012> <http://www.wikidata.org/prop/direct/P21> <https://www.wikidata.org/wiki/Q6581097>`

Marlon Brando — has sex or gender — male

For most of these tasks, we have little direct evidence of exactly how much the multimodal data contributes to the problem. As a potential sanity check, we include a gender classification task. The benefit of this task is that we have a reasonable confidence that the gender or sex can be guessed from the portrait image with some accuracy, so this task can be used to make sure that information present in the multimodal literals is propagating through the graph.

However, we emphasize that we do not consider this a suitable benchmark task. The task of classifiying sex or gender from appearance or other information is very sensitive, and since benchmarks have a guiding effect for what the community focuses on, we do not believe that this is a suitable general purpose task to train our models on. While wikidata offers sex and gender categories beyond male and female, their frequencies in this data are far too low to be cast in a simple classification framework.

We expect there may be some value to this task in studying gender bias, but for general purpose model evaluation this task should only be used as a sanity check.

CS publications and authors (`dblp`)

classes : nr of citations (above or below median)

regression:	classes:
nr of citations	1 citation >1 citation

Integrates data from:

- DBLP (paper names, authors, metadata)
- Wikidata (extra author metadata)
- OpenCitations (citation counts)

This dataset is low on multimodal information, but it does provide a well-balanced binary classification objective and a very large amount fo test and validation data.

---

for your convenience

Available in `.hdt`, `.nt.gz`. Pre-transformed to CSV:

- Easy to load into numpy, torch, sk-learn, etc.
- Python data-loader
- Baselines and example code

---

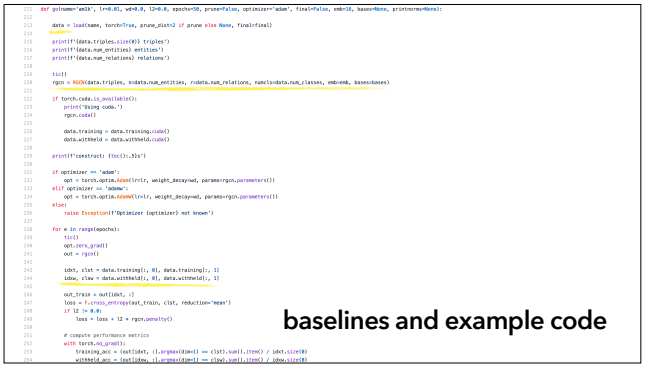
data loader

```python
import kgbench as kg

data = kg.load('amplus') # Load with numpy arrays, and train/validation split

data = kg.load('amplus', torch=True) # Load with pytorch arrays

data = kg.load('amplus', final=True) # Load with numpy arrays and train/test split
```



baselines and example code

so...

kgbench.info

Five knowledge graph node classification tasks with

- large test sets,
- manageable size,
- rich, multimodal information

and

- a friendly dataloader,
- baselines/reference implementations.

```
pip install git+https://github.com/pbloem/kgbench.git
```
