Node Resolution and Relation Classification

FILIP ILIEVSKI;

HANZHI ZHANG;

Background

- Most CSKG nodes are lexical (e.g., "hit")
  - we do have disambiguated nodes in WordNet
  - **Q1: Can we disambiguate the lexical nodes to WordNet?**

Background

- Most CSKG nodes are lexical (e.g., “hit”)
  - we do have disambiguated nodes in WordNet
  - **Q1: Can we disambiguate the lexical nodes to WordNet?**
- Many relations are underspecified (e.g., HasProperty)
  - WebChild provides us with specific property relations
  - **Q2: Can we specify the property relations as in WebChild?**

Sources

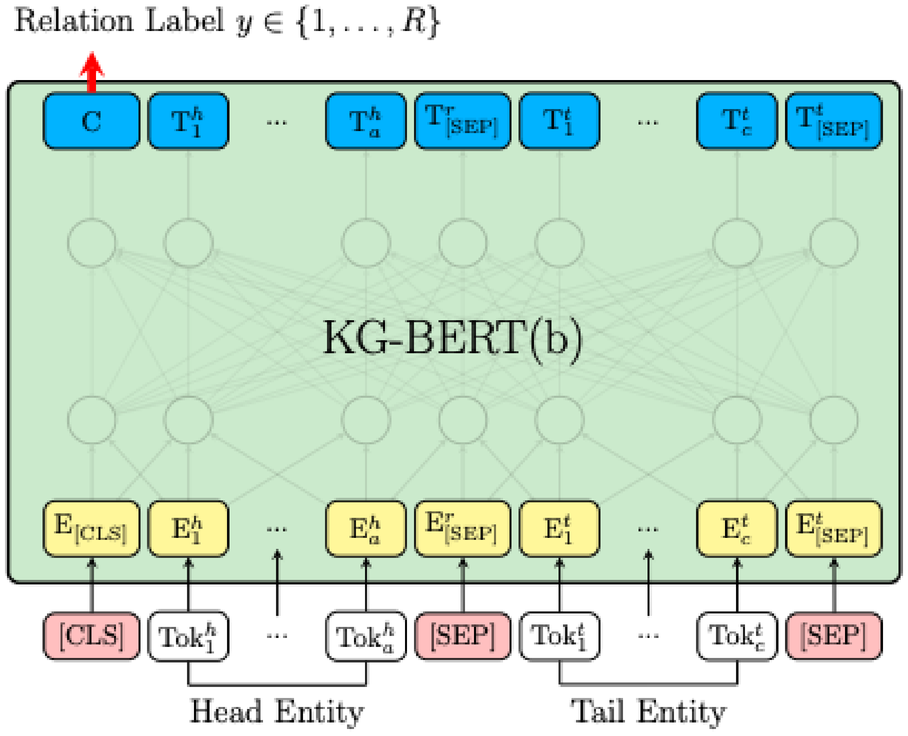
- Training data:
  - **WebChild – disambiguated nodes and property relations**
  - **WordNet – disambiguated nodes**
- Test data:
  - **Subset of CSKG**
  - **relation==HasProperty**
  - **subject and object are ambiguous**

Task definition

- Relation classification
  - given an edge, specify its relation
  - HasProperty -> temperature, shape, color, …
- Node classification
  - given an edge, disambiguate its subject or object
  - “hit” -> hit.v.02
- Joint specification [future work]

Method

- adapt [KG-BERT [Yao et al., 2019]]
- originally, it has been applied to:
  - link prediction
  - relation classification
  - triple classification
- input is a description or a label for <h>, <r>,< t>



Some Assumptions (that might not hold)

- The subject and the object have exactly one correct synset in WordNet

- The relation of HasProperty is exactly one of the 27 properties (synsets) listed in WebChild's property file

Relation Classification – Task definition

Task: Predict the relation of the triple, given the labels or the descriptions of the subject and the object. (Example: use the subject, “mandarin orange” and the object “orange” to predict relation “color”)

27 Unique Relations:
wn:quality.n.1, wn:trait.n.1, wn:age.n.1, wn:color.n.1, wn:beauty.n.1, wn:shape.n.2, wn:size.n.1, wn:state.n.2, wn:weight.n.1, wn:emotion.n.1, wn:strength.n.1, wn:motion.n.4, wn:physical_property.n.1, wn:temperature.n.1, wn:feeling.n.1, wn:sensitivity.n.2, wn:tactile_property.n.1, wn:manner.n.1, wn:sound.n.1, wn:ability.n.1, wn:appearance.n.1, wn:sustainability.n.1, wn:personality.n.1, wn:taste_property.n.1, wn:disposition.n.4, wn:length.n.1, wn:structure.n.2

Relation Classification - Setup

Task: Predict the relation of the triple, given the labels or the descriptions of the subject and the object. (Example: use the subject, “mandarin orange” and the object “orange” to predict relation “color”)

Setup:

1. Use **WebChild** as train, development and test data.
2. Learn to predict the correct relation id (**wn:color.n.01**) given the **descriptions** of the node **synsets** (e.g., **wn:mandarin.n.01 and wn:orange.a.01**).
3. Use different baselines to make prediction and calculate the accuracy.
4. Analyze the results and pick the most reliable baseline to classify HasProperty edges in **CSKG**.

Relation Classification - Data

Task: Predict the relation of the triple, given the labels or the descriptions of the subject and the object. (Example: use the subject, “mandarin orange” and the object “orange” to predict relation “color”)

Setup:

1. Use **WebChild** as train, development and test data.
2. Learn to predict the correct relation id (**wn:color.n.01**) given the **descriptions** of the node **synsets** (e.g., **wn:mandarin.n.01 and wn:orange.a.01**).
3. Use different baselines to make prediction and calculate the accuracy.
4. Analyze the results and pick the most reliable baseline to classify HasProperty edges in **CSKG**.

```
aardvark#n#1    manner#n#1  adorable#a#1    aardvark    adorable    130 1   1   2gms,   1   a
aardvark#n#1    quality#n#1 general#a#1 aardvark    general 44  1   1   2gms,   0.453608    a
```

Data sampling: Randomly Choose 100k lines which from WebChild’s ***property*** subset. Split to train-dev-test at 80%: 10%: 10% ratio

The number of triples without relations	3673697
The number of triples with relation	2836191
The number of unique entities	48076
The number of unique relations	27
The most frequent entity	"wn:new.a.1"
The most frequent relation	"wn:quality.n.1"

Data sampling

Data sampling: Randomly Choose 100k lines which from WebChild's ***property*** subset.
Split to train-dev-test at 80%: 10%: 10% ratio

then filter duplicates, edges with no candidates, and edges where the ground truth is not in the candidates

	Total Line	Duplicate Line	Remaining Line
Train	80000	22	79978
Dev	10000	7	9993
Test	10000	9	9991

	Total Line	No Candidates
Train	79978	49854
Dev	9993	6206
Test	9991	6153

	Total Line	Ground Truth not in Candidates (Left)	Ground Truth not in Candidates (Right)
Train	30124	3685	0
Dev	3787	447	0
Test	3838	508	0

Relation Classification-Baselines

Baselines:

Random Baseline:

Randomly choose one relation from 27 unique relations.

Frequency Baseline (MFS):

Choose the relation, “wn:quality.n.1”, which is most frequent on the training set.

Sentence Embedding Baseline:

1. For each triple, create a sentence by concatenating the labels of the node1+relation+node2 as **label sentence**.

2. For each relation candidate, take its relation definition (**candidate sentence)**.
Example *for wn:quality.n.1*: *an essential and distinguishing attribute of something or someone*

3. Embed the label sentence and each candidate sentence by **sentence-transformer-bert Baseline (STB)** and **sentence-transformer-RoBERTa Baseline (STR)**.

4. Compute the cosine similarity between **label sentence** and **candidate sentence**.

5. Pick the candidate with max similarity.

Relation Classification-Baselines

Kg-Bert: BERT-based neural network

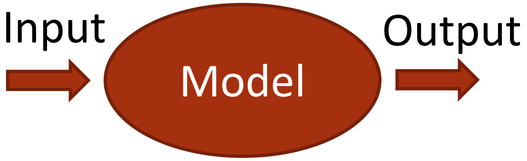
1. Training time Input: definitions of the subject and the object synset
2. Training time Output: One of the 27 relation labels
3. Test Input: definitions of the subject and the object synset
4. Test Output: score for each of the 27 labels

Example:

wn:food.n.1-~~wn:quality.n.1~~-wn:expensive.a.1

any substance that can be metabolized by an animal to give energy and build tissue;

high in price or charging high prices;



wn:quality.n.1	13.8
wn:trait.n.1	-1.1
wn:age.n.1	-2.0
wn:color.n.1	-3.2
wn:beauty.n.1	-3.3
wn:shape.n.2	-1.9
wn:size.n.1	-1.2
wn:state.n.2	-0.1
wn:weight.n.1	-2.1
wn:emotion.n.1	-2.8
wn:strength.n.1	-2.5
wn:motion.n.4	-3.0
wn:physical_property.n.1	-2.0
wn:temperature.n.1	-2.0
wn:feeling.n.1	-1.7
wn:sensitivity.n.2	-5.7
wn:tactile_property.n.1	-3.4
wn:manner.n.1	-1.7
wn:sound.n.1	-3.7
wn:ability.n.1	-2.2
wn:appearance.n.1	-2.9
wn:sustainability.n.1	-5.3
wn:personality.n.1	-4.7
wn:taste_property.n.1	-2.6
wn:disposition.n.4	-4.8
wn:length.n.1	-4.4
wn:structure.n.2	-4.6

Relation Classification on WebChild - Results

WebChild:

Relation Prediction							
Accuracy	MRS	MFS	STB	STR	STB II	STR II	Kg-Bert
Train	3.75%	38.30%	10.44%	11.58%	7.33%	7.95%	99.94%
Dev	3.86%	38.70%	11.21%	11.17%	7.47%	8.35%	99.39%
Test	3.85%	37.70%	10.77%	11.50%	8.17%	7.90%	99.48%

STB and STB II:
If we have a triple: “Entity1, Relation1, Entity2”
The label sentence of STB is “Entity1 label, Relation label, Entity2 label”
The label sentence of STB II is “Entity1 definition, Relation label, Entity2 definition”

Findings:

Kg-Bert has the best performance

Relation Classification on CSKG

Application to CSKG:

1. **CSKG Test Data:** sample 100 /r/HasProperty edges for which we can find 2+ candidates for the subject and 1+ candidates for the object in NLTK Corpus--WordNet.
2. **KG-BERT:** As the CSKG nodes are lexical, disambiguate the subject and the object to their most-frequent sense as node id.

   this is correct more often than not, but it is causing some errors

   ```
   Q190024 /r/HasProperty  Q39338  mandarin orange orange
   /c/en/time  /r/HasProperty  /c/en/endless   time    endless
   ```

3. Combine KG-BERT's prediction with the MFS prediction using different ratios.
4. Pick 100 edges to manually inspect whether the prediction is right.
5. Calculate the accuracy.
6. Annotate the incorrect edges manually to be able to compare to the other baselines.

Relation Classification on CSKG - Results

$$Score(s) = \alpha \frac{1}{f(s)} + (1 - \alpha) Model(s)$$

Score(s) is the final score of a word *s*.

α is ratio coefficient, to balance between:

- $f(s)$ = the frequency rank of a candidate (1 for most frequent sense).
- $Model(s)$ = KG-BERT score

Alpha	Relation Accuracy
0	77%
0.1	74%
0.2	77%
0.3	77%
0.4	75%
0.5	77%
0.6	79%
0.7	78%
0.8	**80%**
0.9	78%
1	65%

Relation Classification on CSKG – Analysis of misclassified edges

- The most frequent sense for a node is not always correct. Wrong node id can result in incorrect prediction.

Predicted Result Example: /c/en/glass, wn:temperature.n.1, /c/en/solid;

S: (n) **solid** (matter that is solid at room temperature and pressure) (Most Frequency sense)

S: (n) **solid**, solidness, solid state (the state in which a substance has no tendency to flow under moderate stress; resists forces (such as compression) that tend to deform it; and retains a definite size and shape) (Second most Frequency sense)

- Sometimes the answer is not bad, but there is a better one.

  /c/en/glass, wn:shape.n.2, /c/en/empty; (wn:state.n.2 or wn.quality.n.1 may be better)

- Sometimes it is hard to determine the correct answer. (/c/en/curtains,wn:temperature.n.1,/c/en/dusty)

- The 27 WebChild relations may not cover all cases (e.g., *6 - /r/HasProperty - six*).

- Rest: the model is wrong.

Relation Classification – Findings, challenges, future work

- Incorrect disambiguation of nodes with MFS
  - **Way forward: Combine node resolution and relation classification into a joint model.**

- Generation of ground truth in CSKG is time consuming
  - **Way forward: Find/generate a CSKG test set.**

- KG-BERT scores suspiciously high on WebChild, but only about 80% on CSKG

- The baselines score very low.

- **Options for joint prediction**
  - message passing – would require additional constraints
  - multi-task prediction – might reduce generalizability

Node Resolution

Node Resolution - Introduction

Task: Predict the subject of the triple, given its relation and object.

Steps:

1. Use **WordNet** as training data.

```
node1;label relation    node2;label node1   node2
tendinitis  /r/IsA  inflammation    wn:tendinitis.n.01  wn:inflammation.n.01
```

2. Predict the correct subject id (wn:tendinitis.a.01) by the node and relation id (e.g., wn:inflammation.n.01 and /r/IsA).
3. Use different baselines to make prediction and calculate the accuracy.
4. Analyse the results and pick the most reliable baseline to classify nodes in **CSKG**.
5. In the step of **CSKG** prediction, disambiguate the object to their most-frequent sense as node id.
6. Pick 100 edges to manually inspect whether the prediction is right. And calculate the accuracy.

Node Resolution-Data Summary

WordNet:

```
node1	relation	node2	node1;label	node2;label	relation;label	relation;dimension	source	sentence
wn:physical_entity.n.01	/r/IsA	wn:entity.n.01	"physical entity"	"entity"	"is a"		"WN"
```

The number of triples without relations	0
The number of triples with relation	111,276
The number of unique entities	15,052
The number of unique relations	3
The most frequent entity	"wn:bird_genus.n.01"
The most frequent relation	"/r/IsA"

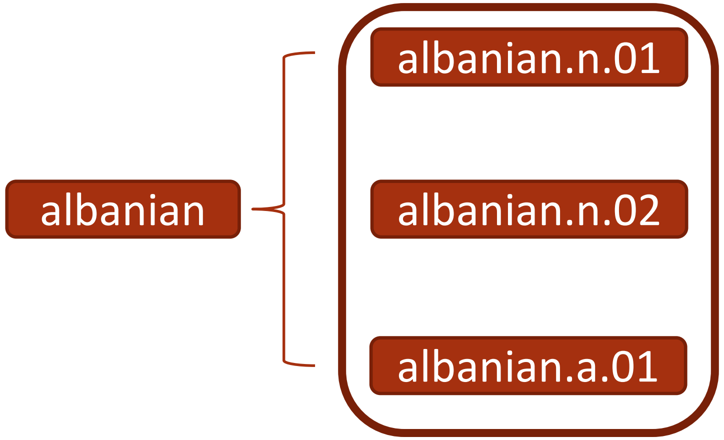
WordNet Train, Dev, Test: Randomly Choose 10k lines which has relation in triple from WordNet.

Divide the data at ratio 0.8: 0.1: 0.1.
Then remove duplicates, cases with no candidates, and cases with no ground truth in the candidates

CSKG:
Same 100 edges as in Relation Prediction

Node Resolution - Baselines

Candidates Generation: Candidates are the all subjects and objects in train dataset.
Example: albanian, /r/PartOf, albania (subject prediction)



Get synsets with lemma "albanian" from WordNet.

Baselines:

Random Baseline:
Randomly choose one candidate as the prediction result. (One of albanian.n.01, albanian.n.02, and albanian.a.01)

Frequency Baseline (MFS):
Candidates are the same as Random Baseline. Choose the most frequent sense (albanian.n.01 in example).

Node Resolution-Baselines

Sentence Embedding Baseline:

1. Candidates are the same as previous baseline. For each triple, create a sentence by "node1;label+relation+node2;label" as **label sentence**.

2. Then for each subject candidate, create a sentence by subject definition as **candidate sentence**.

3. Use **STB** and **STR** model to transfer sentences to different sentence embedding. Compute the cosine similarity between **label sentence** and **candidate sentence**.

4. Pick the candidate whose sentence has the largest similarity.

Example: albanian, /r/PartOf, albania (subject prediction)

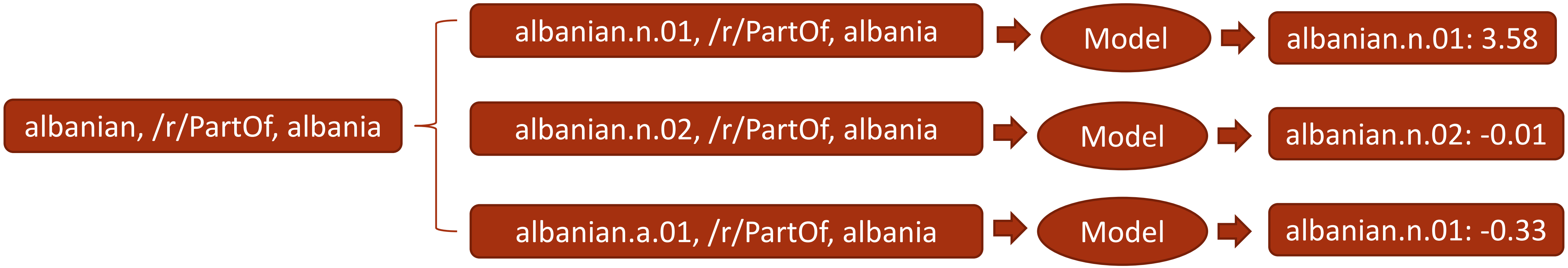
label sentence:
albanian part of albania

Candidate sentence:
albanian.n.01: *a native or inhabitant of Albania*
albanian.n.02: *the Indo-European language spoken by the people of Albania*
albanian.a.01: *of or relating to Albania or its people or language or culture*

Node Resolution – KG-BERT

1. Negative sampling: For each ground truth, we need to add some false label data.
   1. V1: Randomly choose some subjects in the whole data file to replace the correct subject.
   2. V2: Use other concurrent synsets that share the lemma.
2. Training Input: Build a sentence embedding from the definition of subject and object as input data. Relation id is 0 for false, 1 for true.
3. Training Output: A Classification Model with the score for triples
4. Test Input: Build all sentence embedding from the definition of all subject candidates from NLTK WordNet Corpus and object as input data
5. Test Output: Scores for all input triples. Pick the subject synset who has the highest score.

Example: albanian, /r/PartOf, albania (subject prediction)



Node Resolution – KG-BERT variants

Kg-Bert and Kg-Bert II:

- The difference between Kg-Bert and Kg-Bert II is in the negative sampling

albanian, /r/PartOf, albania

- albanian.n.01, /r/PartOf, Albania; label: 1 → Kg-Bert Model
- airhead.n.02, /r/PartOf, albania ; label: 0 → Kg-Bert Model
- dullness.n.02, /r/PartOf, albania ; label: 0 → Kg-Bert Model

albanian, /r/PartOf, albania

- albanian.n.01, /r/PartOf, albania ; label: 1 → Kg-Bert II Model
- albanian.n.02, /r/PartOf, albania ; label: 0 → Kg-Bert II Model
- albanian.a.01, /r/PartOf, albania ; label: 0 → Kg-Bert II Model

Node Resolution-Result & Finding

WordNet:

Link Prediction							
Dataset Type	Location	Baseline Type					
		MRS	MFS	STB	STR	Kg-Bert	Kg-Bert II
Train	Left Entity	70.7%	73.2%	79.7%	80.7%	88.3%	91.2%
	Right Entity	58.5%	72.3%	63.3%	62.3%	79.1%	81.3%
Dev	Left Entity	71.3%	73.3%	81.0%	83.0%	84.5%	85.6%
	Right Entity	58.6%	72.6%	62.9%	64.9%	76.3%	77.5%
Test	Left Entity	71.4%	73.2%	81.3%	85.2%	85.9%	86.9%
	Right Entity	59.6%	72.6%	64.8%	67.1%	74.9%	77.9%

WebChild:

Link Prediction						
Dataset Type	Location	Baseline Type				
		MRS	MFS	STB	STR	Kg-Bert
Train	Left Entity	37.8%	65.6%	40.3%	41.3%	54.7%
	Right Entity	28.6%	46.4%	31.1%	32.2%	41.3%
Dev	Left Entity	39.2%	66.3%	42.3%	43.7%	55.2%
	Right Entity	29.2%	46.3%	32.1%	31.2%	39.7%
Test	Left Entity	37.9%	65.0%	42.1%	41.9%	53.9%
	Right Entity	29.1%	47.6%	33.2%	34.3%	41.0%

Findings:

- Kg-Bert has the best performance on WordNet
- In WordNet, 58% of the subjects has only one candidate.

- MFS has the best performance on WebChild. However, WebChild dataset is obtained by MFS method. Therefore, we still apply KG-BERT on **CSKG**.

Node Resolution – CSKG Results

Alpha	Subject Accuracy	Object Accuracy
0	0.38	0.39
0.1	0.4	0.39
0.2	0.42	0.42
0.3	0.4	0.41
0.4	0.48	0.4
0.5	0.47	0.49
0.6	0.52	0.45
0.7	0.64	0.49
0.8	0.71	0.52
0.9	**0.73**	0.5
1	0.7	0.49

WordNet

Alpha	Subject Accuracy	Object Accuracy
0	0.4	0.3
0.1	0.37	0.34
0.2	0.52	0.36
0.3	0.53	0.4
0.4	0.64	0.43
0.5	0.65	0.49
0.6	0.70	0.49
0.7	0.70	0.50
0.8	0.70	0.49
0.9	0.70	0.49
1	**0.70**	0.49

WebChild

Findings:
Prediction is sometimes unstable
Example: for “mandarin orange”

[3.7036562 3.6185894]

[3.42065907 3.47564721]

mandarin orange (two candidates):
mandarin_orange.n.01:
shrub or small tree having flattened globose fruit with very sweet aromatic pulp and thin yellow-orange to flame-orange rind that is loose and easily removed; native to southeastern Asia

mandarin_orange.n.02:
a somewhat flat reddish-orange loose skinned citrus of China

Node Classification – Error analysis

- For ambiguous subject/object, we pick the most frequent sense. However, most frequent is not always correct. Wrong node id can result in incorrect prediction.

Predicted Result Example: wn:yield.n.03*,* /r/HasProperty*, /c/en/sour,* fruit, sour*;*

wn:yield.n.03: an amount of a product

S: (n) **sour** (a cocktail made of a liquor (especially whiskey or gin) mixed with lemon or lime juice and sugar) (Most Frequency sense)

S: (n) **sour**, sourness, tartness (the taste experience when vinegar or lemon juice is taken into the mouth) (Second most Frequency sense)

- Sometimes it is hard to determine the correct answer. (/c/en/chip, /r/HasProperty,/c/en/dead)

- Rest: model is wrong. Sometimes we find a reasonable one but not the best.

Predicted Result Example: wn:hair.n.03*,* /r/HasProperty, /c/en/thin*;*

hair.n.03: filamentous hairlike growth on a plant

Node Resolution-Challenges & Ongoing Work

Challenges:

- Unclear why the MFS results are better than KG-BERT alone on CSKG.
  - **way forward: further analysis**
- Incorrect disambiguation of nodes with MFS
  - **Way forward: Combine node resolution and relation classification into a joint model.**
- Generation of ground truth in CSKG is time consuming
  - **Way forward: Find/generate a CSKG test set.**
- Hard to pick the best node when two candidates have similar classification scores.
- Many subjects/objects have no obvious candidates in WordNet

Update: Feb 4th

FILIP ILIEVSKI;

HANZHI ZHANG;

Node Resolution-Subject vs object accuracy?

WordNet:

Link Prediction							
Dataset Type	Entity	Baseline Type					
		MRS	MFS	STB	STR	Kg-Bert	Kg-Bert II
Train	Subject	70.7%	73.2%	79.7%	80.7%	88.3%	91.2%
	Object	58.5%	72.3%	63.3%	62.3%	79.1%	81.3%
Dev	Subject	71.3%	73.3%	81.0%	83.0%	84.5%	85.6%
	Object	58.6%	72.6%	62.9%	64.9%	76.3%	77.5%
Test	Subject	71.4%	73.2%	81.3%	85.2%	85.9%	86.9%
	Object	59.6%	72.6%	64.8%	67.1%	74.9%	77.9%

WebChild:

Link Prediction						
Dataset Type	Entity	Baseline Type				
		MRS	MFS	STB	STR	Kg-Bert
Train	Subject	37.8%	65.6%	40.3%	41.3%	54.7%
	Object	28.6%	46.4%	31.1%	32.2%	41.3%
Dev	Subject	39.2%	66.3%	42.3%	43.7%	55.2%
	Object	29.2%	46.3%	32.1%	31.2%	39.7%
Test	Subject	37.9%	65.0%	42.1%	41.9%	53.9%
	Object	29.1%	47.6%	33.2%	34.3%	41.0%

We saw that subjects are easier to disambiguate than objects.

Q: Is this correlated with the number of candidates?

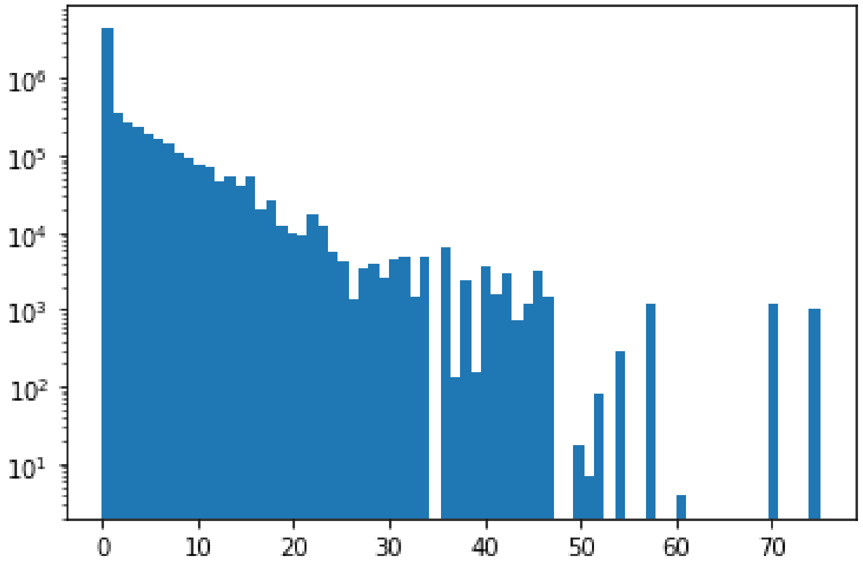
Node Resolution-Subject vs object candidates

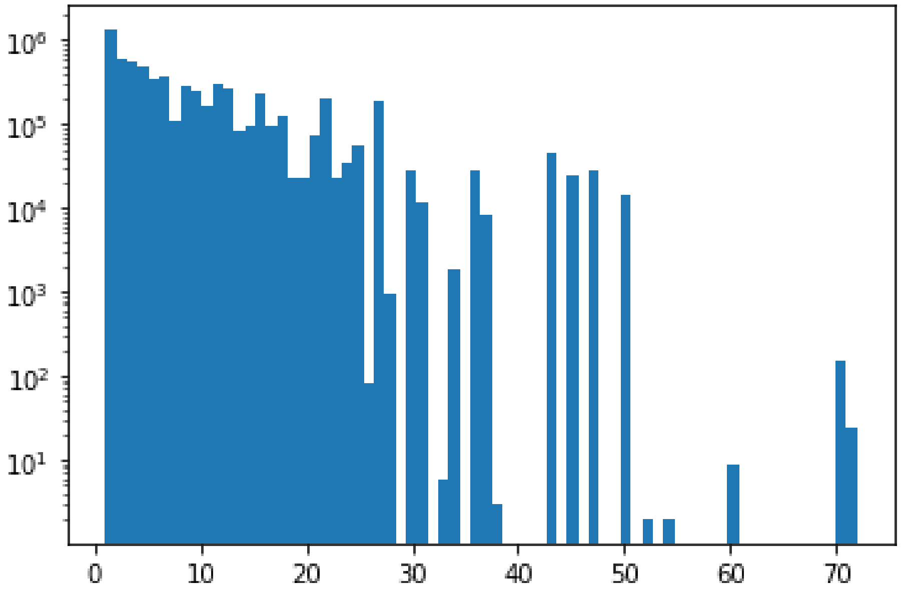
Subjects have less candidates

No object lemmas have zero candidates, while many subjects have zero candidates

Edges with no subject candidates have more object candidates (not sure why this is)

	Avg No. Candidates	Avg No. Candidates (c(s)>0 and c(o)>0)	Avg No. Candidates (c(s)==0 or c(o)==0)
Subject	**2.5**	5.8	0.0
Object	**9.2**	6.7	11.2





WebChild MFS based on training data frequency

MFS based on the WebChild training data performs better than WN MFS on WebChild. But on CSKG???

	Accuracy (With 0)	Accuracy (Without 0)	Accuracy (Having correct answer)
Left Head	35.02%	81.15%	**89.37%**
Right Head	91.01%	87.32%	**87.24%**

MFS Baseline on WebChild

	WN MFS Accuracy	WC MFS Accuracy
Left Head	70%	63%
Right Head	49%	59%

MFS Baseline on CSKG

Possible contributions

1. We define a task of *commonsense knowledge specification*, which aims to reframe commonsense knowledge triples with ambiguous nodes and generic relations to semantically well-defined ones. We study three variants of this task: relation classification, node disambiguation, and triple specification.
2. We design and collect a benchmark for evaluating these tasks automatically.
3. We adapt a state-of-the-art link prediction technique, KG-BERT, and implement informative baselines: random baseline, most-frequent-sense baseline, and unsupervised BERT and RoBERTa baselines. Extensive experiments show the promise of state-of-the-art systems to perform on specification tasks, by combining structural and textual information found in commonsense knowledge graphs.
4. We analyze and discuss the limitations of state-of-the-art systems on this task, as well as the limitations of our current experimental setup. Based on these observations, we propose a way forward for comprehensive specification of commonsense knowledge graphs.

1.Task

Node disambiguation and relation classification are ok

- though in practice we evaluate on a single property class
- we evaluate on nodes for which we can find candidates easily

We haven't defined/tried the triple classification yet

2.Data for training and evaluation

- We train on WebChild-prop and WordNet
  - WebChild not ideal for training node resolution
  - WordNet not useful for property classification
- We test on 100 examples from CSKG
  - too small and ad-hoc – how to expand to a proper systematic benchmark?

3. Evaluation - setup

- various baselines, unsupervised transformers, and KG-BERT
  - maybe another system should be added later
- KG-BERT relies on sentences, which are originally from WordNet
  - how to generate sentences for arbitrary nodes?

3. Evaluation

- Relation classification
  - Supervision with KG-BERT performs best on the source corpus (WebChild)
  - KG-BERT+MFS best on the test corpus, but performance drops 20%
- Node disambiguation
  - on WordNet, KG-BERT || performs best on the source corpus
  - WebChild – best performance by MFS (artifact of the data)
  - On CSKG for both datasets, best performance with 90%MFS + 10% KG-BERT
  - max performance 73% when training on WN
  - subject performance much higher, partially because subjects have less candidates

4. Discussion and limitations

- not urgent

Current obstacles

- How to create a good benchmark for testing?
- How to increase the set of properties?
- How to deal with more complex node phrases (for which we get no candidates at first)?
- how to generate sentences for arbitrary nodes?
- Is triple classification needed?

Direction 2: Spatial knowledge about household items

- How much spatial and part-whole knowledge about household items do we find in CSKG?

Method:

1. filter CSKG nodes based on a set of ~50 labels in the EQA
2. filter CSKG edges based on dimensions
3. Compute statistics

Source	Number
VG	3,446
CN	265
CN|WN	9
WD	14

Statistics

Number Edges: 3,734

cup	406
mirror	284
vase	240
bed	229
toilet	223
computer	191
sink	166
desk	160
pan	148
television	122
microwave	101
refrigerator	97
rug	93
fireplace	71
bathtub	69
plates	65
shower	63
sofa	54
dresser	53
bookshelf	50
kettle	45
heater	41
piano	41
ottoman	40
cutting board	35
dishwasher	28
dryer	22

towel rack	17
washer	15
fish tank	10
food processor	7
ironing board	6
vacuum cleaner	4
loudspeaker	3
fruit bowl	3
tv stand	3
coffee machine	2
whiteboard	1
xbox	1
shoe rack	1
chessboard	1
wardrobe cabinet	0
knife rack	0
range oven	0
utensil holder	0
dressing table	0
playstation	0
stereo set	0
water dispenser	0

Relation distribution

Data quality impressions

Direction 3: Constraint-based benchmark

Aspects like transitivity or symmetry have not been part of current benchmarks – is this something worth pursuing (by us)?