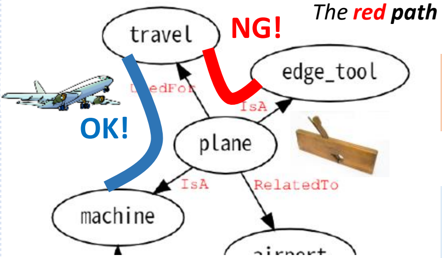
LREC 2022 Marseille

Towards the Detection of a Semantic Gap in the Chain of Commonsense Knowledge Triples

Yoshihiko Hayashi (Waseda University, Japan)

Mind the *semantic* gap in commonsense KG



*The **red path** should be avoided in reasoning!*

< **plane**, IsA, edge_tool>
< **plane** , UsedFor, travel>

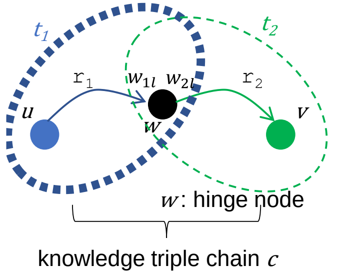
< **plane**, IsA, machine>
< **plane** , UsedFor, travel>

Contributions

- We created a pilot dataset in which knowledge triple chains sampled from *ConceptNet* were annotated whether each contains a semantic gap.
- We devised a few baseline methods for detecting the semantic gaps and compared them in small-scale preliminary experiments.
- We achieved several insights from preliminary experiments: the potential efficacy of sense embeddings and contextualized word embeddings.

Notations and definitions

knowledge triples: t_1, t_2



- A **knowledge triple chain** c is formed by two adjacent knowledge triples, t_1 and t_2.
  - A knowledge triple chain is a primary component of an arbitrary length knowledge path.
- A **semantic gap** exists in c if the intended meaning of the hinge word w in t_1 may be different from that in t_2.

ConceptNet
An open, multilingual knowledge graph

https://conceptnet.io/

"ConceptNet is a multilingual knowledge base, representing words and phrases that people use and the common-sense relationships between them. The knowledge in *ConceptNet* is collected from a variety of resources, including crowd-sourced resources, games with a purpose, and expert-created resources".

Pilot dataset

- Selection of the hinge words: from "List of English homographs", as well as well-known polysemous words (e.g., river, bank, plane, etc.)
- Sampling of chains: random sampling, but excluded some concept relations (e.g., RelatedTo, HasContext, etc.)
- Labeling of semantic gaps:
  - initially annotated by an English native speaker; revised by the author
  - checked invalid triples by POS-level checking rules
  - allowed mutually associated derivative meanings (e.g., red as a noun or as an adjective) and systematic polysemy (e.g., school)

Item	Count	
total # of chains	3,000	
# of chains *without-gap*	1,313	*relatively well-balanced*
# of chains *with-gap*	1,425	
# of chains incl. invalid triples	262	
# of unique triples	4,316	
# of triples flagged invalid	196	*noisy! (~5% erroneous data)*
# of unique hinge words	255	
- average degree of polysemy	10.5	WordNet 3.0: *too fine-grained?*

```
>>> semgap_df = pd.read_csv('./data/semgap_april_2022.tsv', sep='\t')
>>> semgap_df
     semgap           u            r1          w              r2                  v w1_pos w2_pos inv_triple
0       Yes      rotate   --MannerOf--> alternate   <--MannerOf--              spell      v      v         --
1       Yes  instrument   <--Synonym-->   arrange   --MannerOf-->           organize      v      v         --
2        No       agree   <--Synonym-->   arrange   <--MannerOf--             settle      v      v         --
3        No      string   --MannerOf-->   arrange   <--MannerOf--              drape      v      v         --
4       Yes     cascade   --MannerOf-->   arrange   <--Synonym-->          negotiate      v      v         --
...     ...         ...           ...       ...             ...                ...    ...    ...        ...
2995     No  schoolhouse  <--Synonym-->    school  <--AtLocation--            binder      n      n         --
2996    Yes      prague  <--AtLocation--   school        <--IsA--  secretarial_school    n,v      n       NG_L
2997     No    homework  --AtLocation-->   school  <--AtLocation--             track      n      n         --
2998     No     janitor  --AtLocation-->   school        <--IsA--     dancing_school      n      n         --
2999     No      violin  --AtLocation-->   school  --AtLocation-->              town      n    n,v         --

[3000 rows x 9 columns]
```

https://bit.ly/3vt7Re9

Concluding remarks

- ARES-based second-order similarities would be effective (when combined with other features)
- Simple BERT-based classifier outperformed other baselines (the aggregation method could/should be improved)
- Weakly or self-supervised method is required (to address the data issue)

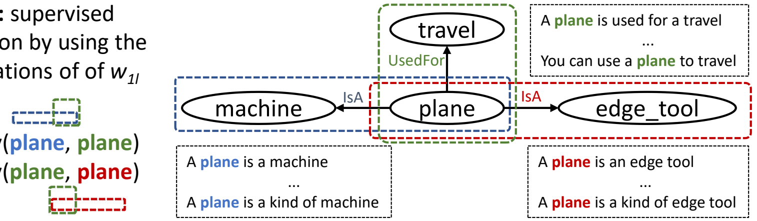
Selected references

- [ConceptNet & NumberBatch] Speer et al., (2017). ConceptNet 5.5: An Open Multilingual Graph of General Knowledge, *AAAI 2017*.
- [ARES] Scarlini et al., (2020). With More Contexts Comes Better Performance: Contextualized Sense Embeddings for All-Round Word Sense Disambiguation, *EMNLP 2020*.

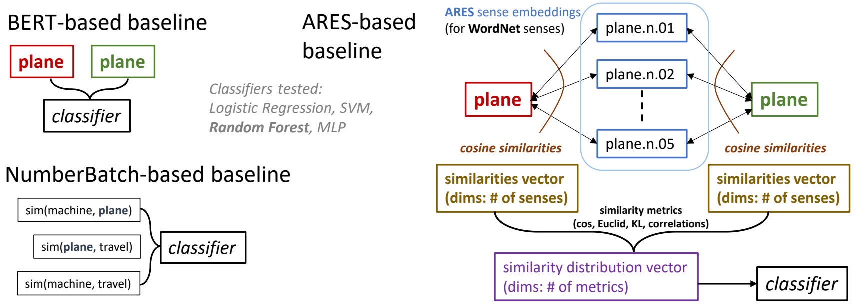
Baseline detection methods

Basic idea: supervised classification by using the representations of of w_{1l} and w_{2l}

c_1: classify(**plane**, **plane**)
c_2: classify(**plane**, **plane**)



- Pseudo sentences are generated by applying hand-coded templates
- Representations for **plane**, **plane**, and **plane** are obtained by pooling BERT vectors



Preliminary experiments and the Results

Semantic gap detection is not a trivial task!

BERT	ARES	NumBat	P	R	F1
✓			0.68	0.65	0.65
	✓		0.64	0.61	0.60
		✓	0.66	0.65	0.65
✓	✓		0.69	0.65	0.66
	✓	✓	0.69	**0.68**	**0.68**
✓	✓	✓	**0.70**	0.67	0.67

Table 2: Experimental results with the pre-trained BERT. P and R stand for precision and recall, respec-

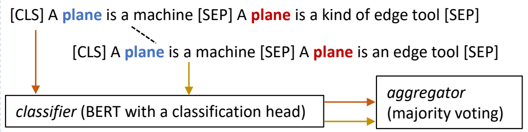
- BERT > NumberBatch > ARES
- ARES however plays a role when combined with NumberBatch/BERT

BERT	ARES	NumBat	P	R	F1
✓			0.67	0.66	0.66
	✓		0.65	0.65	0.64
✓	✓		0.67	0.66	0.66
	✓	✓	**0.68**	0.66	0.66
✓	✓	✓	0.67	**0.67**	**0.67**

Table 3: Experimental results with the fine-tuned BERT. The NumberBatch baseline is excluded from this table, as it does not use BERT-originated vectors.

- fine-tuning is *not* effective (in the comparison with ARES embeddings)

Alternative baseline: Simple BERT-based classifier



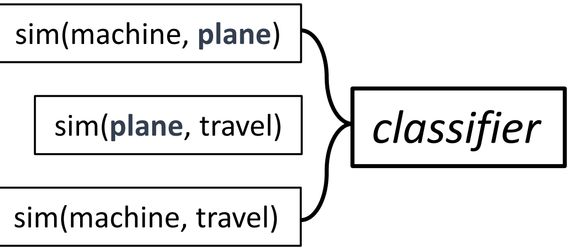
Level	P	R	F1
Sentence pair-wise	0.73	0.74	0.73
Chain-level	**0.70**	**0.70**	**0.70**

Table 5: Experimental results with fine-tuned BERT-based classifier.

Logistic Regression, SVM,
Random Forest, MLP

c_1: classify(**plane**, **plane**)
c_2: classify(**plane**, **plane**)

https://bit.ly/3vt7Re9


sim(machine, **plane**)
sim(**plane**, travel)
sim(machine, travel)
classifier