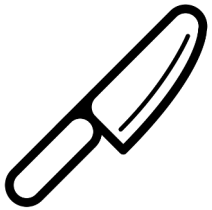
Learning Prototypical Functions for Physical Artifacts

Tianyu Jiang and Ellen Riloff
University of Utah
ACL-IJCNLP, August 2021

Introduction

Humans create things for a reason.

spears for hunting



knives for cutting

pots for cooking

Introduction

- We refer to the most typical way a physical object is often being used as its **prototypical function**.

- The prototypical function of human-made physical artifacts is a kind of **commonsense knowledge** that often plays a role in natural language understanding.

Examples

a) He killed the mayor with a ***gun***.

b) He killed the mayor with a ***knife***.

c) He killed the mayor with a ***bomb***.

Examples

a) He killed the mayor with a ***gun***. → Shooting

b) He killed the mayor with a ***knife***. → Stabbing

c) He killed the mayor with a ***bomb***. → Explosion

Introduction

- It is essential for NLP systems to “read between the lines”.
- The goal of this work is to learn the prototypical functions of human-made physical artifacts.
- We define a new NLP task to associate physical artifacts with frames from FrameNet as the representation for the function.

Dataset Creation

01 Artifact Selection

02 Function Representation

03 Human Annotation

Artifact Selection

- We focus on artifacts that are
  - Physical objects
  - Created by people
- WordNet synset "*artifact.n.01*"
- Concreteness dictionary [1]
  - We apply a threshold >= 4.5
- Intersection produces a set of 1,017 concrete physical artifacts.

Function Representation

- FrameNet 1.7 contains 1,221 frame definitions.

Cooking_creation
Def. This frame describes food and meal preparation…
LU. bake.v, cook.v, grill.v, ...

Protecting
Def. Some protection prevents a danger from harming an asset.
LU. guard.v, protect.v, ...

- Not all frames are suitable for representing functions of physical artifacts, e.g.,
  - Abstract nominals: Calendric_unit
  - High-level abstractions: Intentionally_act
  - Misc: Judgement

- We manually selected 42 frames in FrameNet

Human Annotation

Frames	Artifact examples
Wearing	*hat, shirt*
Containing	*basket, luggage*
Self_motion	*bicycle, yacht*
Protecting	*armor, helmet*
Supporting	*chair, scaffolding*
Cause_harm	*cannon, spear*
Perception_exp	*earphone, eyeglass*
Make_noise	*bell, violin*
Cause_motion	*engine, propeller*
Cutting	*knife, scissors*

Task Definition

- Multiclass classification
- Denote
  - artifacts as $a_i (i = 1..m)$,
  - frames as $f_j (j = 1..n)$,
- The task is to select f_j that represents the prototypical function for a given a_i.

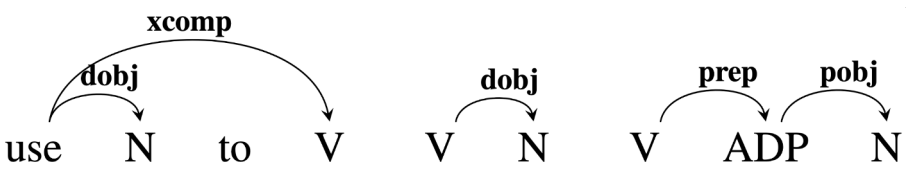
Baselines

- For each artifact in Conceptnet [2], we extract the first word under **UsedFor** relation.
- For COMET, we use its beam-10 setting to generate 10 phrases of the UsedFor relation for each artifact.
- Use the extracted words to rank candidate frames through **lexical units**.

1. **Existing knowledge base - ConceptNet and COMET [3]**

Baselines

- We use 3 dependency patterns to extract <N, V> pairs from Wikipedia.



- 3.8 million <N, V> pairs for our 938 artifacts

1. Existing knowledge base - ConceptNet and COMET
2. **Co-occurrence information**

Baselines

- Use masked language model [4] to get prediction score for every artifact and lexical unit pair.

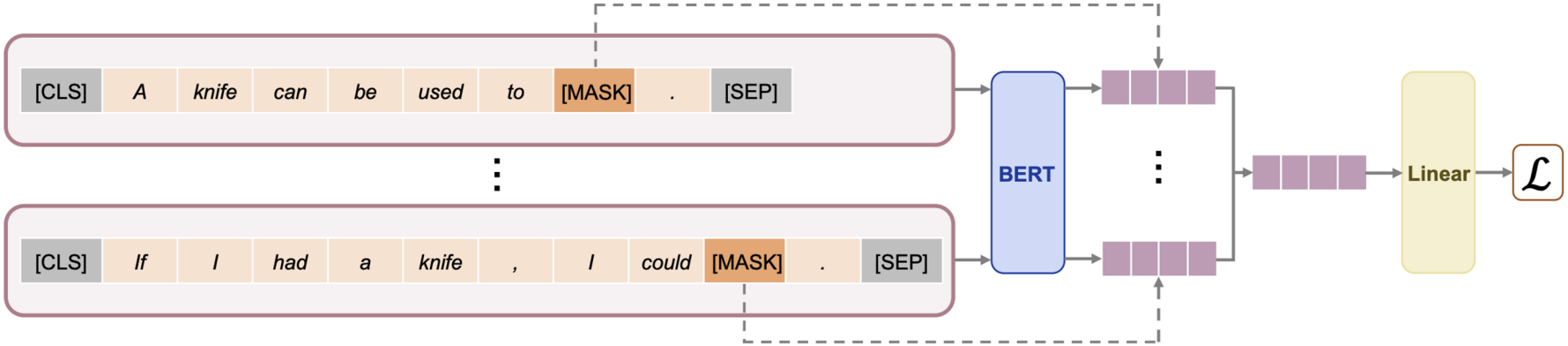
- 6 sentence templates:

  ___ can be used to ___ .
  I used ___ to ___ .
  ___ can be used for ___ .
  I used ___ for ___ .
  The purpose of ___ is to ___ .
  If I had ___ , I could ___ .

1. Existing knowledge base - ConceptNet and COMET
2. Co-occurrence information
3. **Masked language model (MLM)**

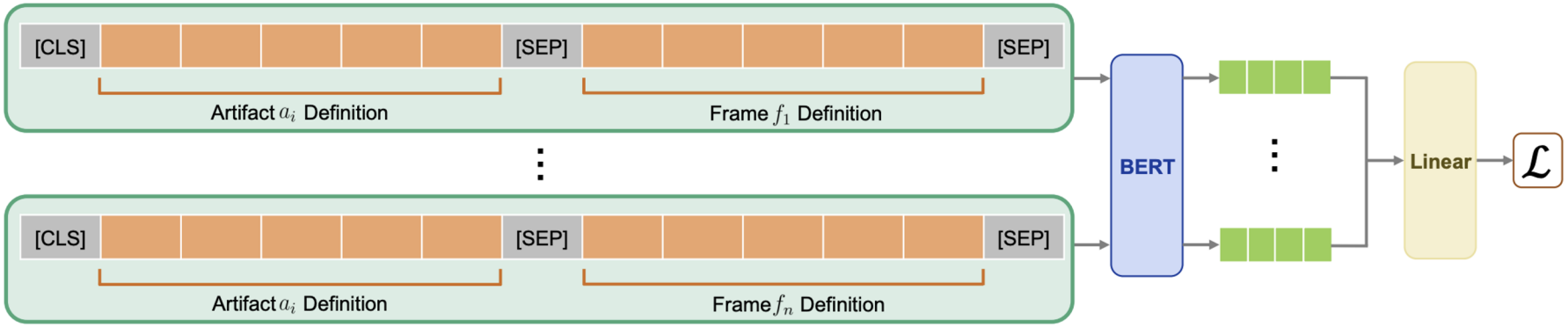
Learning from Masked Patterns – PF_{mask}

- Instead of using the predicted word for the mask token, we retrieve the last hidden state vector of the [MASK] token.
- Then pass it through a linear layer for classification.



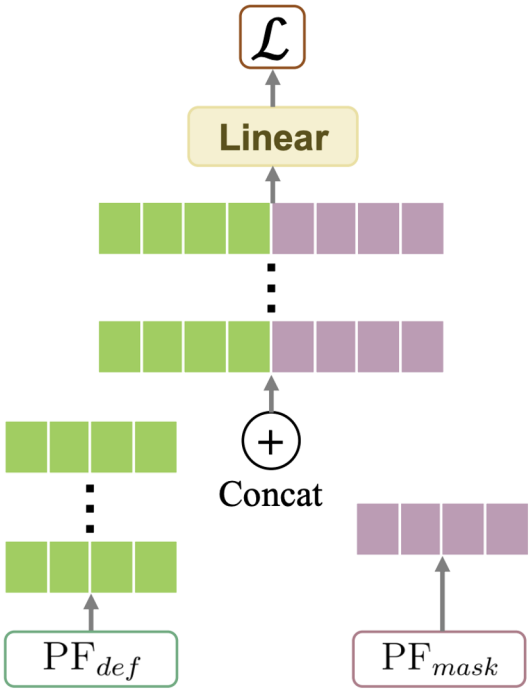
Learning from Definitions - PF_{def}

- Each green block represents an artifact a_i paired with one of the candidate frames.
- Model produces a probability distribution over all candidate frames.



Combining PF_{mask} and PF_{def}

- Left is output of PF_{def}, a matrix of dimension (# of frames, hidden vector size)
- Right is output of PF_{mask}, a matrix of dimension (1, hidden vector size)



Experimental Results

Model	Acc	Pre	Rec	F1
ConceptNet	17.5	33.6	13.5	16.4
Co-occurrence	31.9	24.1	23.9	19.9
COMET	30.7	29.7	35.6	28.2
MLM	42.8	29.5	33.8	28.2

Experimental Results

Model	Acc	Pre	Rec	F1
ConceptNet	17.5	33.6	13.5	16.4
Co-occurrence	31.9	24.1	23.9	19.9
COMET	30.7	29.7	35.6	28.2
MLM	42.8	29.5	33.8	28.2
PF_{mask}	58.5	35.7	36.5	35.4
PF_{def}	74.7	63.5	57.6	59.3
$PF_{def+mask}$	**76.8**	**65.2**	**61.1**	**62.4**

Experimental Results

Model	Acc	Pre	Rec	F1
ConceptNet	17.5	33.6	13.5	16.4
Co-occurrence	31.9	24.1	23.9	19.9
COMET	30.7	29.7	35.6	28.2
MLM	42.8	29.5	33.8	28.2
PF_{mask}	58.5	35.7	36.5	35.4
PF_{def}	74.7	63.5	57.6	59.3
$PF_{def+mask}$	**76.8**	**65.2**	**61.1**	**62.4**

Conclusion

- We introduced the new task of learning prototypical functions for human-made physical artifacts.
- A manually annotated data set.
- We built a transformer-based model using both artifact and frame definitions as well as masked pattern predictions, that can learn the function of a given artifact.
- We will release our data at https://github.com/tyjiangU/physical_artifacts_function

THANKS!

Do you have any questions?

Reference

[1] Marc Brysbaert, Amy Beth Warriner, and Victor Kuperman. 2014. Concreteness ratings for 40 thousand generally known english word lemmas. Behavior research methods, 46(3):904–911.

[2] Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In Proceedings of the AAAI Conference on Artificial Intelligence (AAAI 2017).

[3] Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 2019. COMET: Commonsense transformers for automatic knowledge graph construction. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL 2019).

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL 2019).