Code4Struct: Code Generation for Few-Shot Event Structure Prediction

Xingyao Wang, Sha Li, Heng Ji
{xingyao6, shal2, hengji}@illinois.edu

UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN

Motivation

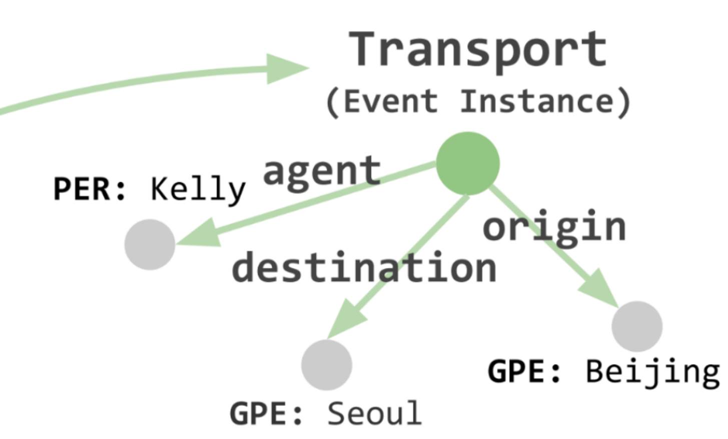
- Large Language Model (LLM) trained on a mixture of text and code can translate natural language (NL) instructions into structured code.
- Some semantic structures (e.g., output event-entity graph in event argument extraction) can be easily translated into code.

Can we leverage such text-to-code capability of LLM to tackle structured prediction problems?

```
Translate the following sentence into an instance of Transport.
The trigger word(s) of the event is marked with **trigger word**.
"Kelly , the US assistant secretary for East Asia and Pacific
Affairs , **arrived** in Seoul from Beijing Friday to brief Yoon ,
the foreign minister ."
"""
transport_event = Transport(
    artifact=[
        PER("Kelly"),
    ],
    destination=[
        GPE("Seoul"),
    ],
    origin=[
        GPE("Beijing"),
    ],
)
```

Input Sentence

Code



Method

Ontology Code Representation: We convert the existing event type ontology to Python class definitions.
Task Prompt: Conditioned on these definitions, we put the input sentence into a docstring to prompt LLM.

```
class Movement(Event): # Inherit from `Event` class
    ... # omitted for space
class Transport(Movement):
    """
    self.agent transported self.artifact in self.vehicle vehicle from
self.origin place to self.destination place.
    """
    def __init__(
        self,
        agent: List[GPE | ORG | PER] = [],
        artifact: List[FAC | ORG | PER | VEH | WEA] = [],
        destination: List[FAC | GPE | LOC] = [],
        origin: List[FAC | GPE | LOC] = [],
        vehicle: List[VEH] = [],
    ):
        self.agent = agent
        self.artifact = artifact
        self.destination = destination
        self.origin = origin
        self.vehicle = vehicle
```

Hierarchical Ontology

Event template

Entity Type Annotation

```
"""
Translate the following sentence into an instance of Transport. The trigger
word(s) of the event is marked with **trigger word**.
"Renowned Hollywood madam Heidi Fleiss has been **flown** to Melbourne as guest
of honour at Thursday's market debut and , according to Harris , has already
played a key role in attracting worldwide media attention to the event ."
"""
transport_event = Transport(
    artifact=[PER("Heidi Fleiss"),],
    destination=[GPE("Melbourne"),],
)
```

Groundtruth Code

LLM Prompt

```
from typing import List
class Entity:
    def __init__(self, name: str):
        self.name = name
class Event:
    def __init__(self, name: str):
        self.name = name
```

Base Class Definition

```
class ORG(Entity):

class GPE(Entity):
    """Geopolitical entities such as countries, provinces,
states, cities, towns, etc. GPEs are composite entities,
consisting of ..."""
    def __init__(self, name: str):
        super().__init__(name=name)
```

Relevant Entity Definition(s)

Event Definition

Ontology Code Representation

(optional) k In-context Examples

```
"""
Translate the following sentence into an instance of
Transport. The trigger word(s) of the event is marked
with **trigger word**.
"Kelly , the US assistant secretary for East Asia and
Pacific Affairs , **arrived** in Seoul from Beijing
Friday to brief Yoon , the foreign minister ."
"""
transport_event = Transport(
```

Trigger Marking

Task Prompt

Evaluation

Arg-Identification F1: Whether LLM can identify an argument correctly (e.g., Kelly, Seoul, Beijing)
Arg-Classification F1: Whether LLM can match correctly identified argument with a correct role (e.g. agent=Kelly)

Comparison with Supervised Approaches

20-shot Code4Struct rivals fully-supervised approaches trained on >4k training instances.

Model	Data	Arg-I F1	Arg-C F1
DyGIE++	Full	66.2	60.7
BERT-QA	Full	68.2	65.4
OneIE	Full	73.2	69.3
TANL	Full	65.9	61.0
BART-Gen	Full	69.9	66.7
DEGREE	Full	**76.0**	**73.5**
CODE4STRUCT$_{\texttt{text-davinci-003}}$	0-shot	49.9	37.8
Text2Event	20-shot*	23.1	19.1
DEGREE	20-shot*	33.0	30.9
CODE4STRUCT$_{\texttt{text-davinci-003}}$	20-shot*	**65.0**	**60.4**
Text2Event	50-shot*	30.6	26.0
DEGREE	50-shot*	40.8	37.3
CODE4STRUCT$_{\texttt{code-davinci-002}}$	50-shot*	**62.3**	**58.1**

It surpass current SOTA by 29.5% under 20-shot

Code Representation Allows Cross-Sibling Transfer

```
class Transfer_Money(Transaction):
    """self.giver gave money to self.recipient for the benefit of
self.beneficiary in self.place place."""
    def __init__(
        self,
        beneficiary: List[GPE | ORG | PER] = [],
        giver: List[GPE | ORG | PER] = [],
        place: List[FAC | GPE | LOC] = [],
        recipient: List[GPE | ORG | PER] = [],
    ):
        super().__init__(
            beneficiary=beneficiary,
            giver=giver,
            place=place,
            recipient=recipient,
        )

class Transfer_Ownership(Transaction):
    """self.seller gave self.artifact to self.buyer for the benefit of
self.beneficiary at self.place place."""
    def __init__(
        self,
        artifact: List[FAC | ORG | PER | VEH | WEA] = [],
```

Child event Transfer_Money inherit from parent Transaction

Transfer_Ownership is a sibling event type of Transfer_Money

Child event Transfer_Ownership inherit from parent Transaction

- **same-type**: using examples from the testing event type itself
- **non-sibling type:** using examples from a random non-sibiling

	Arg-I	Arg-C
0-shot	52.8	42.9
1-shot (same type)	54.3	50.2
1-shot (sibling type)	**57.2**	**51.9**
1-shot (non-sibling type)	56.3	50.3
10-shot (same type)	58.7	**55.2**
10-shot (sibling type)	**60.8**	54.9
10-shot (non-sibling type)	58.5	51.0

Using sibling examples help: they are just as useful as annotated example from the predict event type!

Is code prompt any better than text prompt?

```
Translate the following sentence into an instance of Transport event. The trigger
word(s) of the event is marked with **trigger word**.
"Kelly , the US assistant secretary for East Asia and Pacific Affairs ,
**arrived** in Seoul from Beijing Friday to brief Yoon , the foreign minister ."
1. agent: () ""
2. artifact: (PER) "Kelly"
3. destination: (GPE) "Seoul"
4. origin: (GPE) "Beijing"
5. vehicle: () ""
```

(4) Event Instantiation

Text Prompt T1 (code-prompt-style)

```
Translate the following sentence into an instance of Transport event. The trigger
word(s) of the event is marked with **trigger word**.
"Kelly , the US assistant secretary for East Asia and Pacific Affairs ,
**arrived** in Seoul from Beijing Friday to brief Yoon , the foreign minister ."
In this event: [] transported ["Kelly"] in [] vehicle from ["Beijing"] place to
["Seoul"] place.
```

(4) Event Instantiation

Text Prompt T2 (BART-Gen-style)

- Code prompt is generally more effective with sufficient in-context examples.
- Text prompt performance have higher variances: T2 has poor low-shot perf, while being slightly better than code prompt on an LLM finetuned with RLHF.

Model	code-davinci-002						text-davinci-002						text-davinci-003					
k-shot	Arg-I	$\Delta^{(1)}_{C-T}$	$\Delta^{(2)}_{C-T}$	Arg-C	$\Delta^{(1)}_{C-T}$	$\Delta^{(2)}_{C-T}$	Arg-I	$\Delta^{(1)}_{C-T}$	$\Delta^{(2)}_{C-T}$	Arg-C	$\Delta^{(1)}_{C-T}$	$\Delta^{(2)}_{C-T}$	Arg-I	$\Delta^{(1)}_{C-T}$	$\Delta^{(2)}_{C-T}$	Arg-C	$\Delta^{(1)}_{C-T}$	$\Delta^{(2)}_{C-T}$
0	50.6	0.7	50.6	36.0	-2.2	36.0	48.9	-2.6	20.2	35.0	-2.4	13.1	49.9	-2.1	15.3	37.8	-1.4	12.6
1	57.3	0.1	4.7	47.8	-1.0	4.7	55.8	1.8	5.3	45.2	3.0	4.9	56.0	-1.5	1.1	44.7	-3.2	1.1
5	58.0	1.1	1.9	52.5	2.9	1.1	56.0	-2.0	1.0	48.8	3.0	1.4	59.2	-0.9	-0.7	51.7	1.4	-2.1
10	57.2	-1.4	-0.2	52.8	0.8	0.1	60.6	2.7	2.9	53.9	6.4	5.0	62.8	3.1	0.6	56.3	5.0	-1.2
20	62.1	1.7	0.2	58.5	3.6	2.4	59.9	0.9	3.7	56.5	8.0	5.8	65.0	3.5	0.7	60.4	7.8	-0.4

Code

ILLINOIS

Paper