Understanding and Reasoning of Humans and Agents

Tianyi Zhang

Self Introduction

- Expertises:
  - Education and Cognitive Science (6 years of experience, B.S., M.Ed)
    Natural and Symbolic Language Understanding and Reasoning (3 years, MSE)
- Passion and Goal:
  - Devise intelligent agents that **emulate human understanding and reasoning** (in PhD) to facilitate seamless **interaction with humans** (Multimodal, PhD and beyond), that will ultimately enhance human life, e.g. a partner and assistant for the elder.
  - Future work:
    - Topic: multimodal structured knowledge acquisition and application
    - Methodology: RL and GNN

Projects Overview

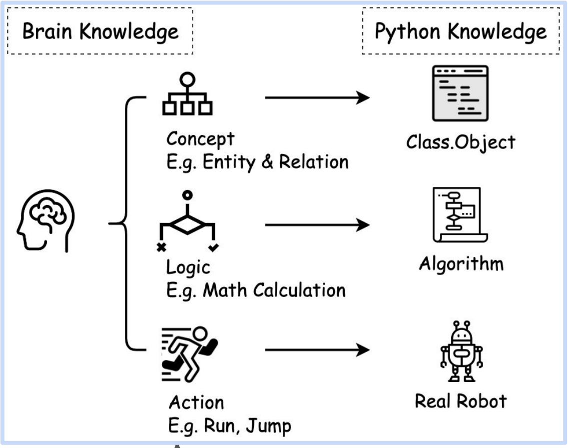
- Generative Symbolic Reasoning for Itinerary Planning (plan, python generation)
  - 23 fall - now, independent research, publication [4]: on working and writing
- wikHow2PDDL: Event Entity-State Tracking (robotic plan, text2pddl generation)
  - AI2, 23 spring, member & leader, publication [3]: submitted to LREC-Coling 2024
- Human-in-the-loop Event Schema Induction
  - DARPA KAIROS, 22-23, leader, publication [2]: accepted by ACL Demo 2023
- Event Extraction w/ QA Data Augmentation
  - DARPA BETTER, 20-22, member, publication [1]: on personal webpage

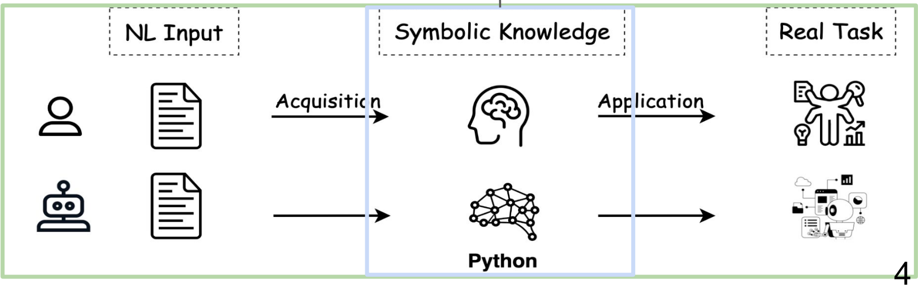
1.Generative Symbolic Reasoning for Itinerary Planning – Foundation

- Human Symbolic Knowledge can be efficiently represented in Symbolic Language (e.g. Python)

- Domains of Human Learning: (Bloom, B. S., 1956, 1973)
  - Cognitive Knowledge (concepts and procedures)
  - Physical Skills (actions)
  - Affective Attitude (emotions)

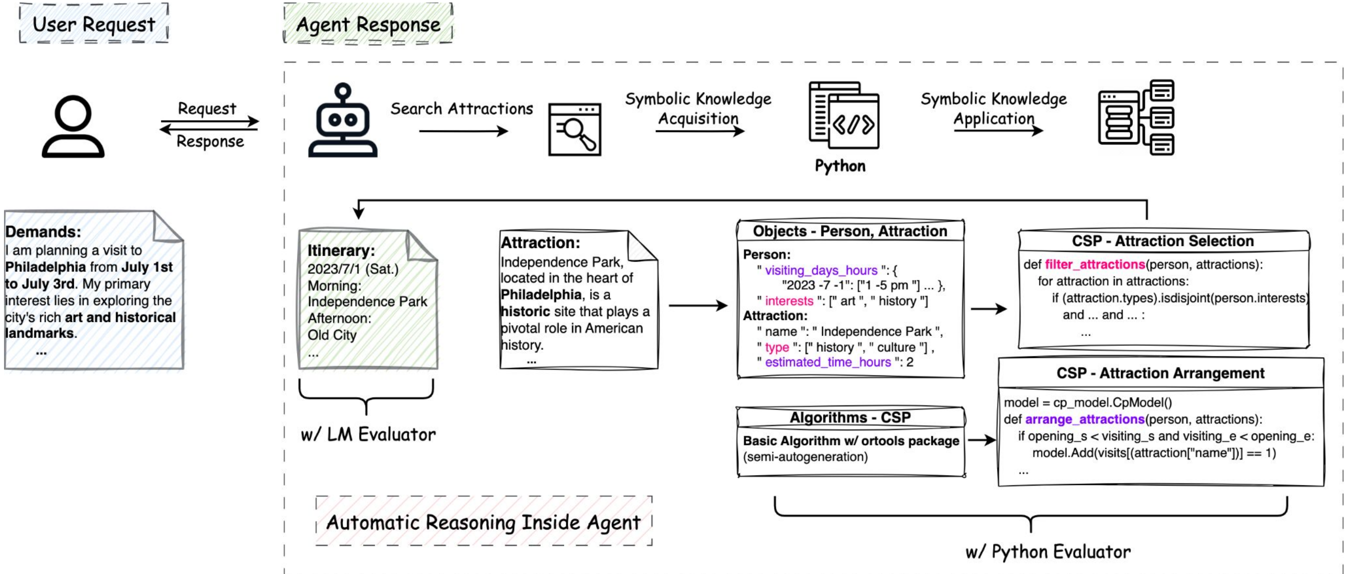
- Procedures of Human Learning: (Piaget, J., 1952)
  - Inputs
    – Acquisition →
  - Structured Symbolic Knowledge
    – Application →
  - Outputs





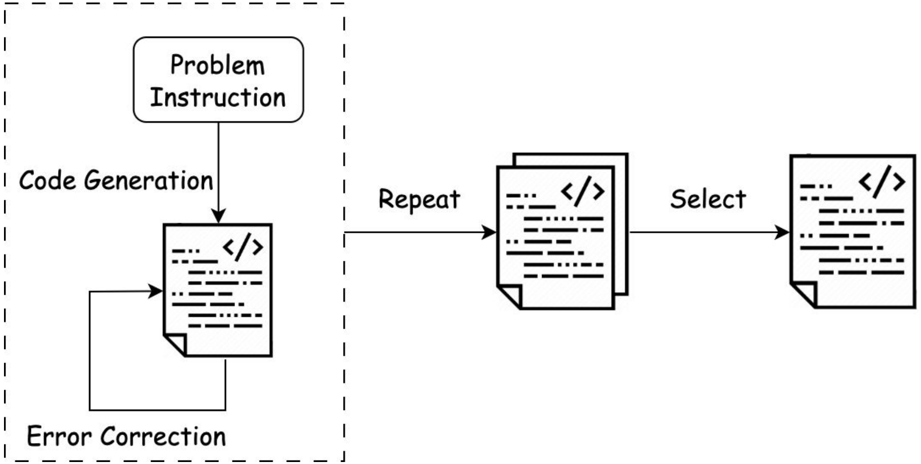
1.Generative Symbolic Reasoning for Itinerary Planning – Methodology

- Agent **acquires** symbolic knowledge including **attraction objects** and similar constraint satisfaction **algorithms** (e.g. job shop).
- Agent **applies** it to specific tasks by **dynamically generating codes** according to user's requirements (e.g., interests, time constraints).



1.Generative Symbolic Reasoning for Itinerary Planning – Methodology

- Knowledge Acquisition and Application Prompts:
  - Clarify the data structure, constraints and goals, a relevant task→
  - Generate code and correct it step by step →
  - Repeat 3-5 times →
  - Choose the most robust and extensible version (succinct, easy to add/remove constraints)



1.Generative Symbolic Reasoning for Itinerary Planning – Contribution

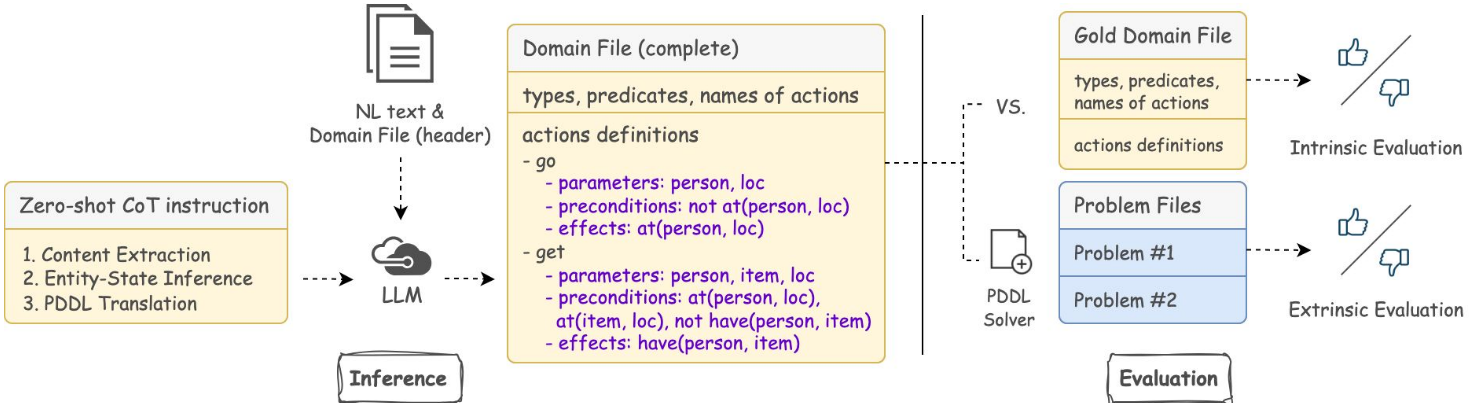
- vs. Natural Language Reasoning
  - Black-box, unfaithful, generic suggestion
- vs. Symbolic Language Reasoning
  - Simplistic, fixed to specific questions
- Our Generative Symbolic Reasoning
  - Symbolic Acquisition-Application framework is versatile
  - Interpretable and controllable, mutable and flexible, personalized suggestion

2.wikHow2PDDL: Event Entity-State Tracking – Motivation

- Importance:
  - PDDL, with its pre- and post-conditions for events, is a useful tool for robot planning and human causal reasoning.
- Relevant works:
  - Robotics: Obtain action-state sequences to infer the underlying domain actions.
  - NLP: Condition on natural language text to generate segments of a problem file.
- Our work:
  - Automatically convert open-domain natural language procedure (e.g. wikiHow) into domain actions.

2.wikHow2PDDL: Event Entity-State Tracking – Methodology

- Approach:
  - Zero-shot 3-step proximal development scaffolding
  - Entity-State Inference and Translation
- Intuitions:
  - Abundant action descriptions in NL vs. Limited domains and actions in PDDL
  - LMs' strong common sense knowledge and faithful planning of PDDL



2.wikHow2PDDL: Event Entity-State Tracking – Evaluation

- Analysis:
  - Entity-state inference overall is good but translation performance is poor (e.g. semantic equivalence of existing predicates and natural language expressions)
  - Explicit inference on the entity-states benefits the parameters
  - Precondition is harder to predict than effect (complex and less obvious)

Model %	Intrinsic action acc.	Extrinsic ℙ𝔽 solve	Extrinsic exact plan
`gpt-3.5`	0.2	1.0	1.0
`gpt-4`	15.9	33.7	4.2
`gpt-4` + CoT	**18.1**	**35.8**	**6.3**
gold	100.0	100.0	100.0

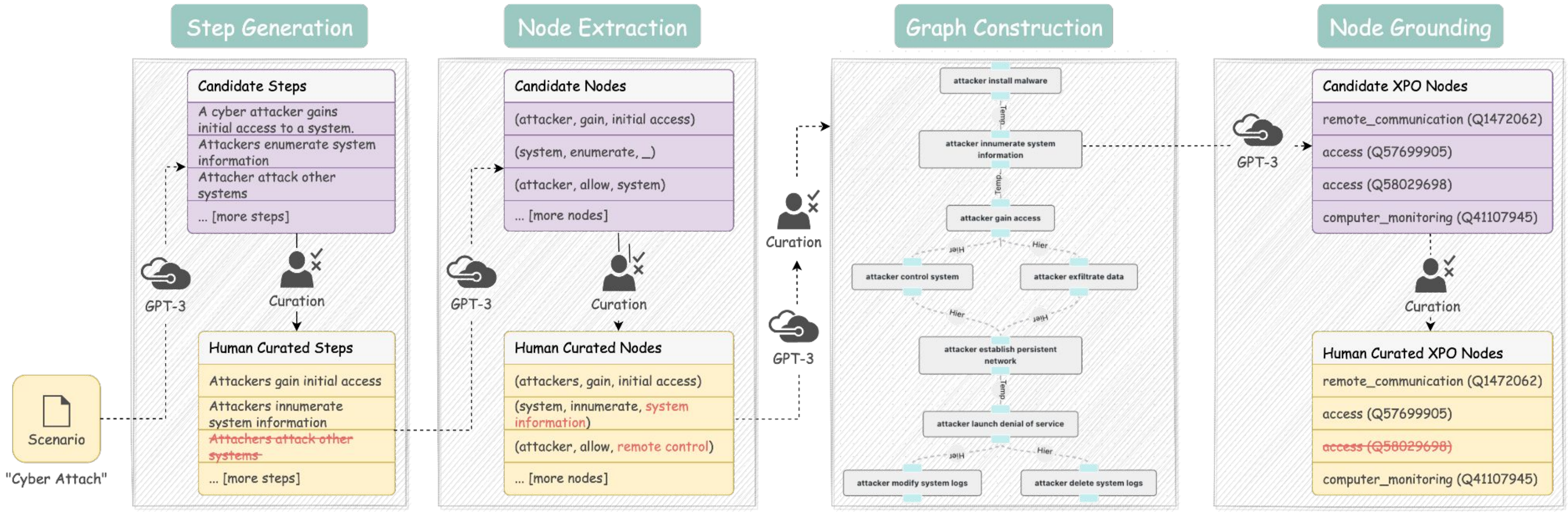
Model %	Parameter	Precondition	Effect
`gpt-4`	36.7	31.1	53.0
`gpt-4` + CoT	42.2	29.7	48.1

3.Human-in-the-loop Schema Induction – Motivation

- Importance:
    - Event schema is essential for understanding complex processes (an outline in a book).
- Difficulties:
    - Given its highly structured and complicated nature
      It's hard to generate directly by LMs and laborious for humans.
- Contributions:
    - Construct a schema in 4 stages from scratch, by leveraging both LM's robust commonsense knowledge and the precision of human modifications.

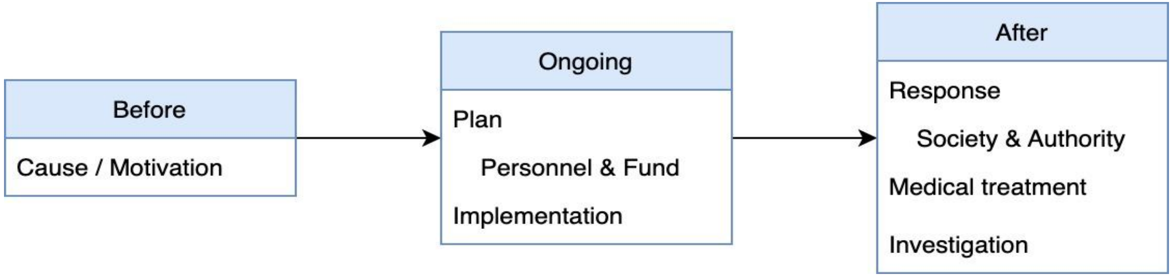
3.Human-in-the-loop Schema Induction – Methodology

- Divide schema generation into 4 stages and in each stage:
  - machine generates results → human corrects them → inputs to the next stage



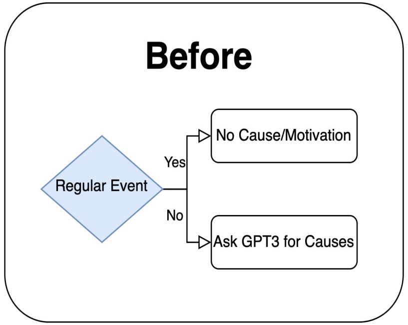
3.Human-in-the-loop Schema Induction – Methodology

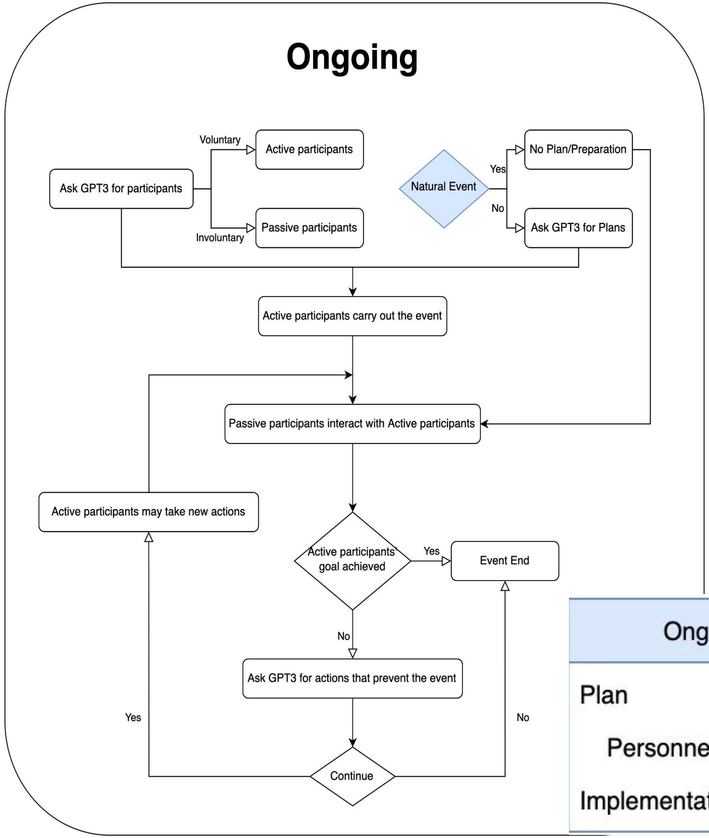
- Design prompts to foster inclusive steps:
  - Dissect a schema into 3 stages: Before, Ongoing, After
  - Summarize the common components
  - Prompt the components guided by a flowchart

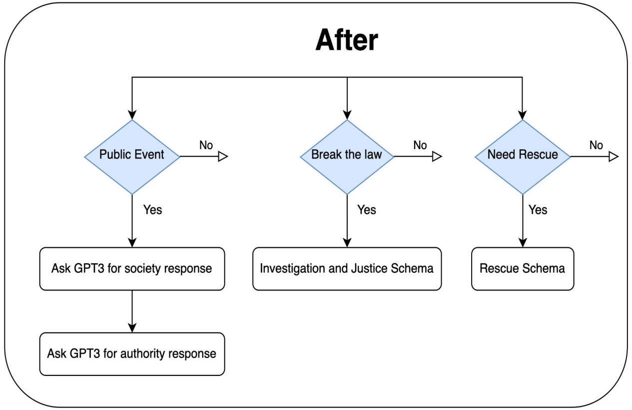


Regular Event: Election	Natural Event: Disease Outbreak	Harmless Event: Election
Attack Disease Outbreak	Attack Election	Attack Disease Outbreak

3.Human-in-the-loop Schema Induction – Methodology







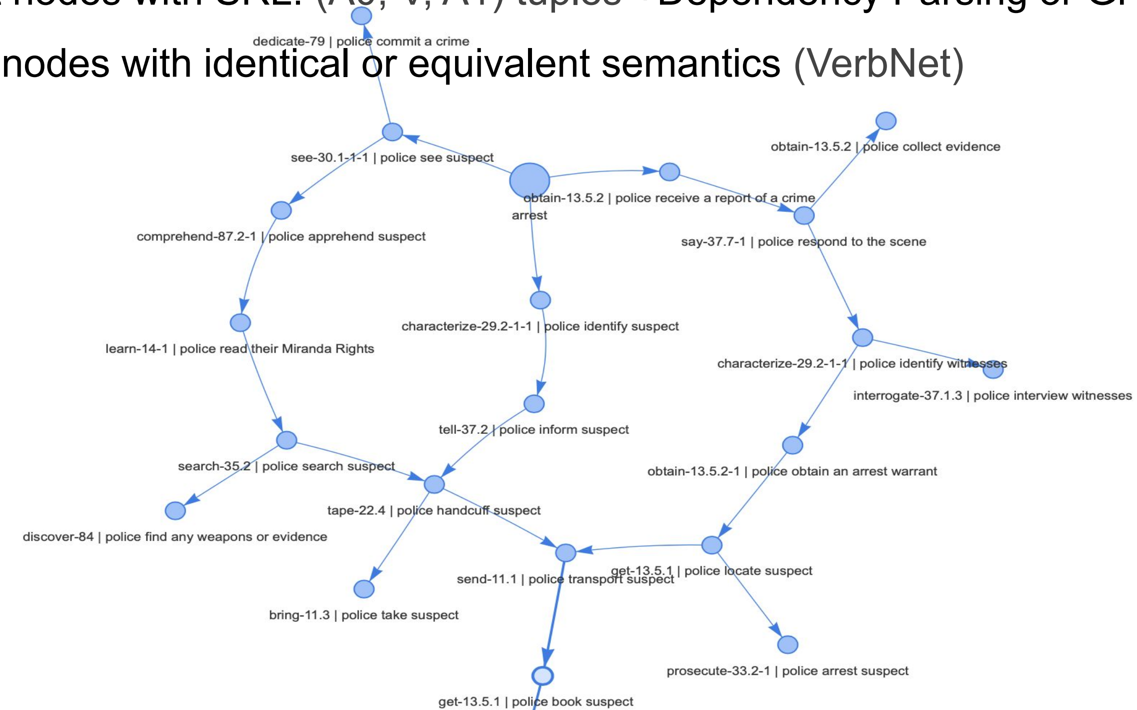
Before
Cause / Motivation

Ongoing
Plan Personnel & Fund Implementation

After
Response Society & Authority Medical treatment Investigation

3.Human-in-the-loop Schema Induction – Methodology

- Node Extraction and Merging:
  - Extract nodes with SRL: (A0, V, A1) tuples +Dependency Parsing or GPT-3
  - Merge nodes with identical or equivalent semantics (VerbNet)



3.Human-in-the-loop Schema Induction – Evaluation

- Analysis:
  - strong commonsense knowledge of GPTs
  - human improvements made on auto generations
  - the time and effort efficiency of our approach

	EVC	FOD	JOB	MED	MRG
Step Acc	11/12	7/8	10/10	10/10	12/12
Node Acc	13/15	10/10	11/12	12/12	12/14
Graph Node ED	1	0	0	0	0
Graph Edge ED	8	0	7	3	16
Grouding Success Rate	5/12	3/10	3/11	6/12	9/12
Self-reported time (min)	15	10	11	10	14

EVC: Evacuation
FOD: Ordering Food in a Restaurant
JOB: Finding and Starting a New Job
MED: Obtaining Medical Treatment
MRG: Corporate Merger or Acquisition

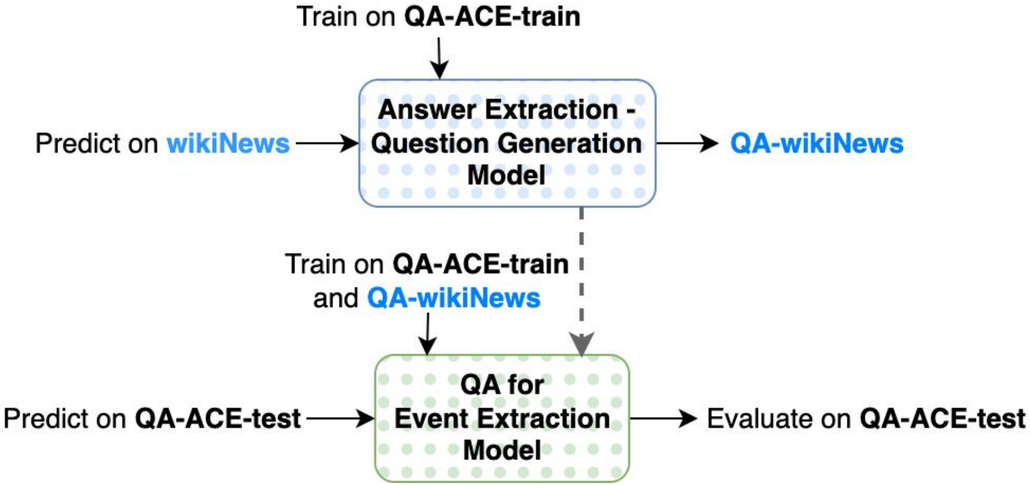
Acc: Accuracy
ED: Editing Distance

4.Event Extraction w/ QA Data Augmentation – Motivation

- Importance:
  - Event is the backbone of natural language understanding
- Difficulties:
  - Human annotation is expensive to obtain
- Other works:
  - **BIO sequence tagging**: multiclass classification lack semantic information sharing
  - **QA transfer learning**: transfer learning data with reduced efficiency
- Our work:
  - **QA data augmentation**: efficiently train event models with abundant in-domain data

4.Event Extraction w/ QA Data Augmentation – Methodology

- Approach:
  - Train an **AE-QG** model (Bert-T5) on domain specific data (ACE)
  - Augment unlabeled data (wikiNews QA)
  - Human annotations + Augmented QA pairs train a **QA** model (RoBerta)



Text: April 7, 2014, writer Peaches Geldof was found dead in her home near Wrotham.
AE input: extract answers: April 7, 2014, ...
AE output: Peaches Geldof <sep> Wrotham <sep>
SRL input: ["April" . . . "Peaches", "Geldof". . . "found", "dead". . . "Wrotham", "."]
SRL output: ["11:B-TMP". . . "11:B-A1", "11: I-A1". . . "[prd]","11:B-A3". . . "11:I-LOC",""]
QG input: generate question: ...writer <hl> Peaches Geldof <hl> was...
prd-aware QG input: generate question: ...<hl> Peaches Geldof <hl> was # found # dead...
QG output: Who is killed?
QA input: ...Peache... [SEP] Who is killed?
QA output: Peaches Geldof

4.Event Extraction w/ QA Data Augmentation – Evaluation

- Analysis:
  - Augmented QA pairs exceed the performance of other QA transfer learning datasets.
  - Augmented QA pairs + gold annotations demonstrate superior performance.

	QG Model			QA Model	
Approach	Dataset1	Num of QA pairs	Test result	Dataset2	Test result
Main	WikiNews-finetuned	8080	**60.91**	ACE	**72.05**
Test1	WikiNews	8060	47.49	ACE	70.07
Test2	SQuAD	87599	52.86	ACE	71.85
Baseline	-	-	-	ACE	70.25
Du et al	-	-	-	ACE-context	**72.20**
Main + Du	WikiNews-finetuned	8080	59.20	ACE-context	**72.84**

- 6895 QA pairs for ACE;
- 6935 QA pairs for ACE-context