Generating Disentangled Arguments With Prompts: A Simple Event Extraction Framework That Works

PAPER ID: 2248

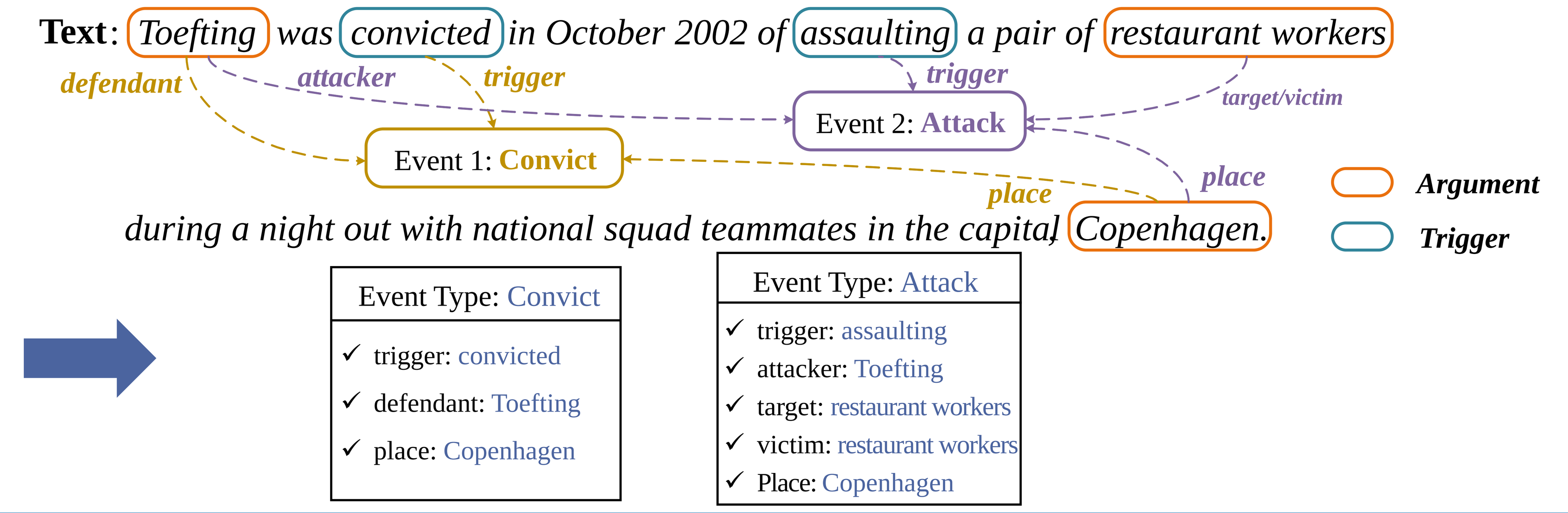
Jinghui Si*, Xutan Peng*, Chen Li, Haotian Xu, Jianxin Li

{sijh, lichen, lijx}@act.buaa.edu.cn x.peng@shef.ac.uk albert.xht@alibaba-inc.com

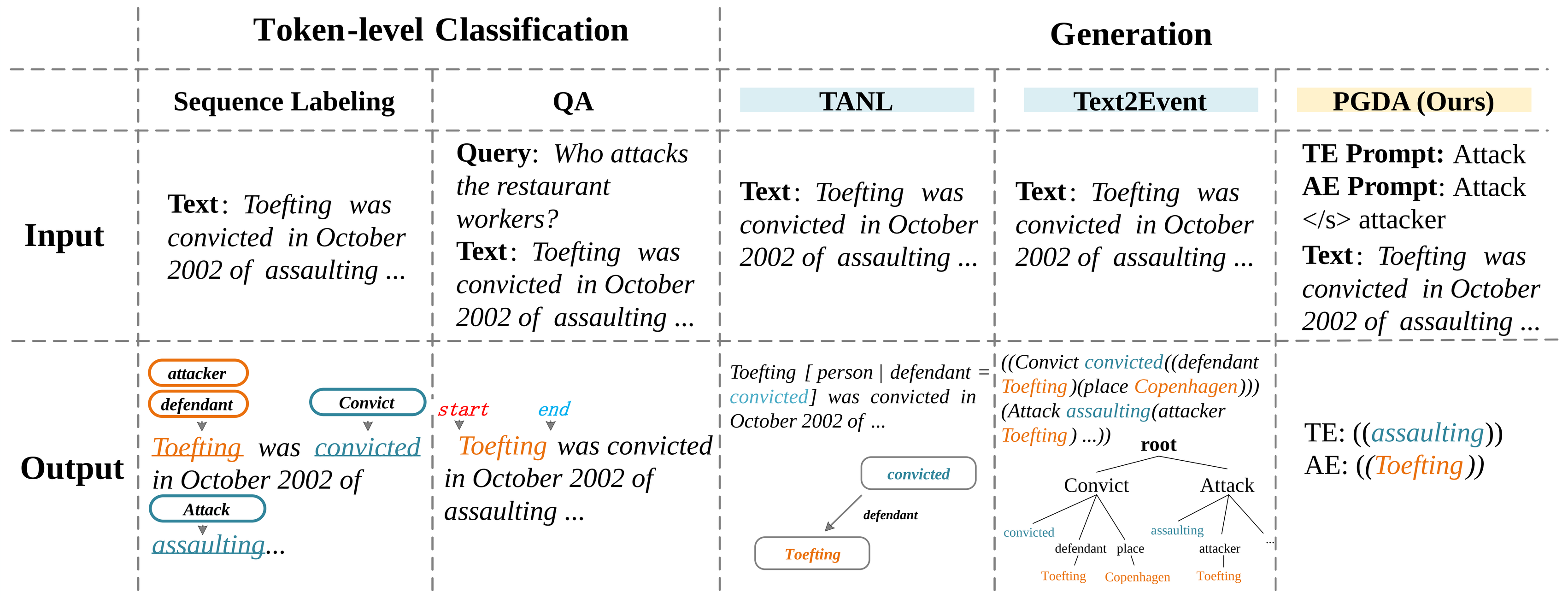
ICASSP 2022 SINGAPORE 22-27 MAY 2022

1. Event Extraction Task

Event Extraction, which aims to extract structured event signals from plain text, is a crucial but hallenging Information Extraction task.



2. Related Work



- Token-level Classification Methods

  Limitations:
  - Insufficient use of labelling knowledge. (Sequence Labeling)
  - The design of the question templates requires high-level expertise and massive human labour. (QA)
- Generation Methods (TANL, Text2Event)

  Limitations:
  - Fail to exploit the lebal semantics on the encoding side.
  - The dependency between Trigger and Argument Extractions can be unnecessary.
  - The outputs are redundant or complex.

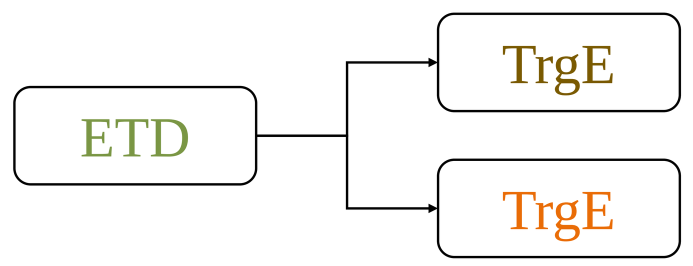
3. Contribution

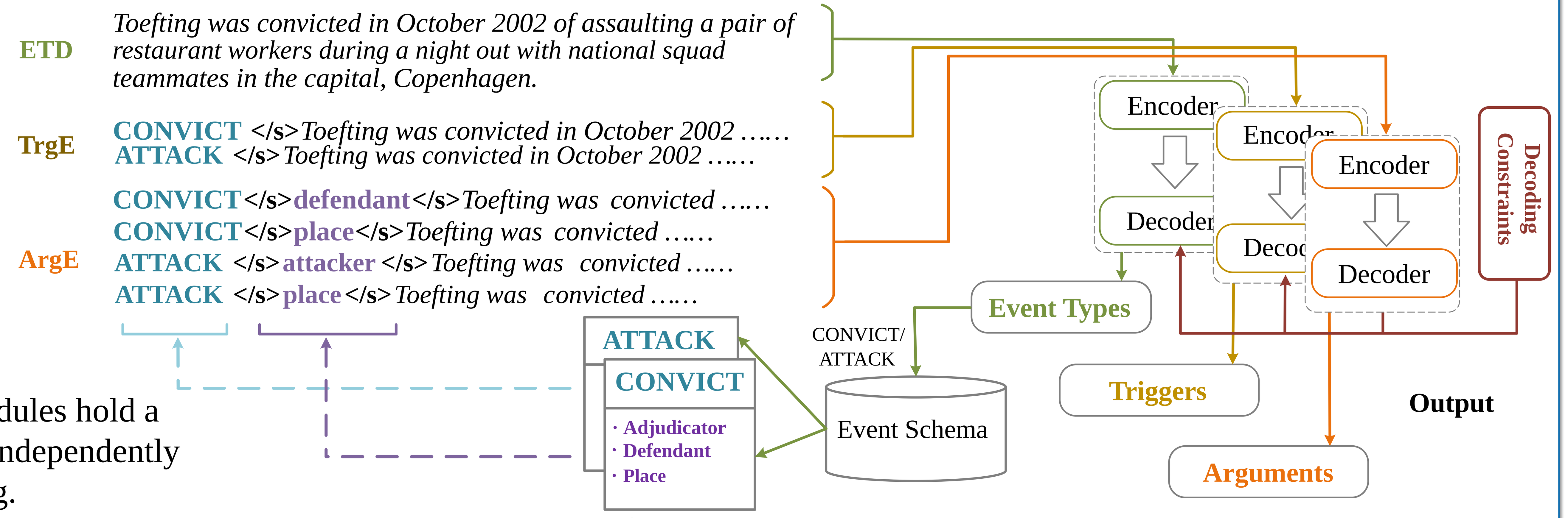
We propose a novel framework that **G**enerates **D**isentangled **A**rguments with **P**rompts (**GDAP**).

- Introduce prompt-based learning to effectively inject knowledge via various label semantics.
- Disentangles the extraction of triggers and arguments.
- Simplify hugely both the architecture and the output forma.

4. Method

- **Architecture**
  - ✓ ETD: Event Type Detection.
  - ✓ TrgE: Trigger Extraction.
  - ✓ ArgE: Argument Extraction.



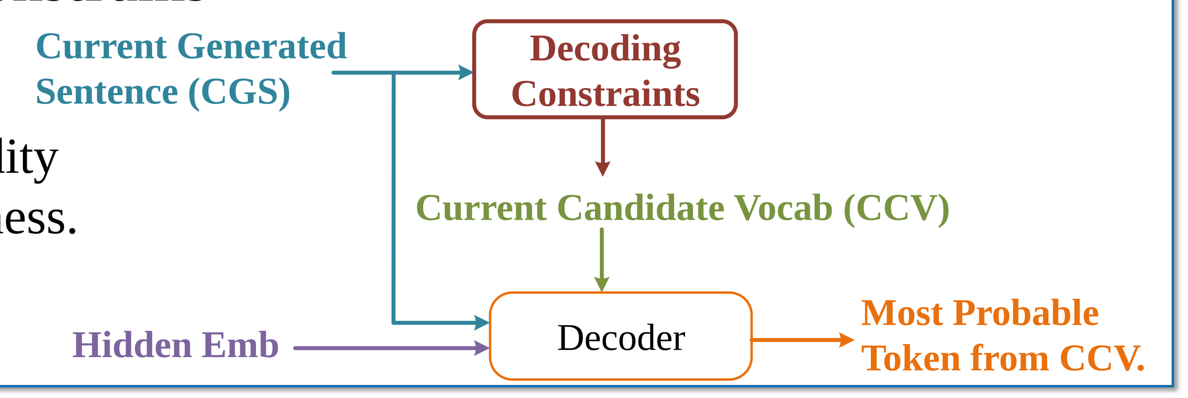


- ✓ For simplicity, all these three modules hold a similar architecture while being independently trained without parameter sharing.

Task	Input	Output	Output Example
ETD	Sent	$((ET_1)(ET_2)\ldots(ET_x))$	((Convict)(Attack))
TrgE	ET_i [SEP] Sent	$((Trg_1)(Trg_2)\ldots(Trg_y))$	((convicted))
ArgE	ET_i [SEP] RT_i [SEP] Sent	$((Arg_1)(Arg_2)\ldots(Arg_z))$	((Toefting))

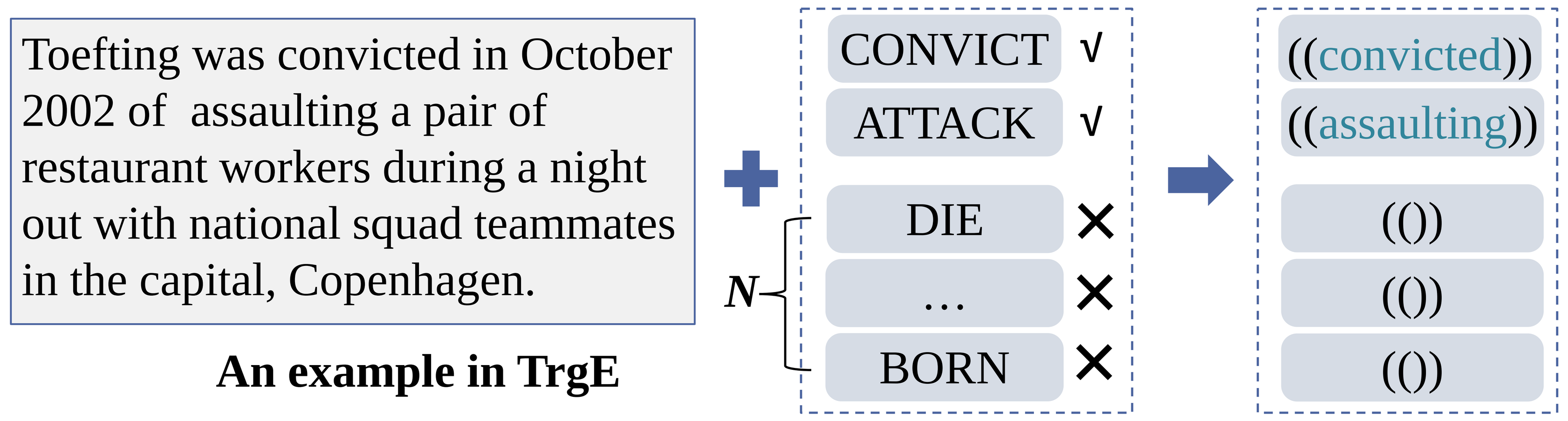
- **Decoding Constrains**

  ✓Guarantee structural validity & token validness.



5. Negtive Sampling

- When training the modules for Trigger and Argument Extractions, we introduce a simple yet effective negative sampling mechanism that **makes our model more fault-tolerant.**
- For each Sent, we randomly select N event types that have not appeared. The model should learn not to extract triggers or arguments when such negative samples appear in the prompt.



An example in TrgE

6. Experiment Settings

- **DataSet:** ACE 2005 (English)
- **Metrics:** we report the precision (P), recall (R), and F1 score (F1) of Trigger and Argument Extractions.
- **Configurations:** we set N in negative sampling at 4 and 2 for Trigger and Argument Extractions, respectively. The learning rate is set at 5e-5 and the epochs are set within {20, 25, 30}.

7. Experiment Results

Model	PLM	Trigger (%) Pre	Trigger (%) Recall	Trigger (%) F1	Argument (%) ☆ Pre	Argument (%) ☆ Recall	Argument (%) ☆ F1
Classification-based Methods							
dbRNN	-	-	-	69.6	-	-	50.1
JMEE	-	-	-	-	-	-	50.4
Joint3EE	-	-	-	69.8	52.1	52.1	52.1
DYGIE++	BERT-L	-	-	69.7	-	-	48.8
GAIL	ELMO	**74.8**	69.4	72.0	**61.6**	45.7	52.4
BERT_QA	BERT-L	71.1	73.7	**72.4**	56.8	50.2	53.3
MQAEE	BERT-L	-	-	71.7	-	-	53.4
RCEE_ER	BERT-L	-	-	-	-	-	**58.7**
Generative-based Methods							
TANL	T5-B	-	-	68.4	-	-	47.6
Text2Event	T5-B	67.5	71.2	69.2	46.7	53.4	49.8
Text2Event	T5-L	<u>69.6</u>	74.4	<u>71.9</u>	<u>52.5</u>	55.2	53.8
GDAP	T5-B	66.1	**<u>75.3</u>**	70.4	47.3	59.1	52.6
GDAP	T5-L	65.6	74.7	69.9	48.0	**<u>61.6</u>**	<u>54.0</u>

- ✓ GDAP (T5-L) hits the highest F1 score among all generative methods in ArgE..
- ✓ The T5-B variant still outperforms TANL(T5-B) and Text2Event(T5-B).
- ✓ GDAP (T5-L) ranks 2nd F1 score in ArgE among all the 13 methods.
- ✓ GDAP (both the T5-B and T5-L versions) achieves the highest recall in ArgE and TriE.

This work is supported by the NSFC through grant No.61872022. We also thank the Google Cloud TPU team for providing TPU machine access.