Constructing Flow Graphs from Procedural Cybersecurity Texts

Kuntal Kumar Pal*, Kazuaki Kashihara*, Pratyay Banerjee*, Swaroop Mishra, Ruoyu Wang, Chitta Baral
Motivation

- **Challenges** of procedural texts written in free natural language form
  - Hard to follow,
  - Difficult to visualize interactions between sentences
  - Difficult to extract inferences
  - Hard to track states of an object or a sub-task

- **Goal**: Provide flow-structures to free form natural language texts
  - *Cybersecurity*(CTFW), Cooking instruction(COR), maintenance manual domains(MAM)

- **Flow-Structure** of the Procedural Text:
  - Sentence level dependencies leading to a goal (action traces, effects of an action, information leading to the action, and instruction order)
Flow-Structure Example

- **CTFW (3154) (New dataset)**
  - Cybersecurity write-ups from Catch The Flag (CTF) competitions
  - Participants find and exploit vulnerabilities in a given set of software services
  - They publish the details of how they exploited the services

- $S_1, S_3, S_4$: Author’s observation about nature of service
- $S_5, S_6$: possible courses of action
- $S_6', S_7, S_8$: chosen path to exploit the vulnerability
- $S_0, S_2$: Irrelevant information

---

*Shading codes:*
- **None**
- **Action**
- **Information**
- **Both**
- **Code**
CTFW Dataset

● **How structure helps in cybersecurity?**
  ○ Automated Vulnerability Discovery and mitigation
  ○ Automated Exploit generation,
  ○ Security education in general

● **Annotations:**
  ○ Sentence Type: Action (A), Information (I), Both (A/I), Code (C), None.
  ○ Flow-Structure: Connection between a pair of sentences based on the interaction between them
Flow-Structure Generation Approach

● **Segment Document to Sentences**
  ○ Rule-based segmentation into sentences
  ○ Relevant sentence identification
  ○ $D_i = \{S_0, S_1, S_2...S_{n-1}\}$

● **Graphical Representation of Document**
  ○ Each relevant sentence as a graph node
  ○ Sentence Windowing ($W_N$) where $N = \{3, 4, 5, \text{ all}\}$
  ○ Graph Connections are directed edges from $S_i$ to $S_j$ where $i < j$.
  ○ In each window,
    ■ *Linear* : $S_i$ to $S_{i+1}$
    ■ *Semi-Complete* : $S_i$ to $\{S_{i+1}, ... S_{i+N}\}$
Approach (Contd.)

- **Node Feature learning**
  - Initial Node features from BERT/ RoBERTa
  - $h_{si} = \text{BERT}([\text{CLS}]s_0s_1...s_{n-1}[\text{SEP}])$

- **Neighbor-aware node feature learning**
  - GCN and GAT learns richer node representations through message passing
  - Linear connections learn from its predecessors
  - Semi-Complete connections learn based on all the previous nodes in a Window
Experiments - Sentence Classification Baseline

- Preprocessing and segmentation into sentences
- We modeled this as a text classification task with five classes:
  - Action (A), Information (I), Both (A/I), Code (C), None.
- We consider any sentence with Action or Information or Both as relevant and rest as irrelevant

| Model         | Val  | Test  |
|---------------|------|-------|
| BERT-Base     | 78.48±0.25 | 77.42±0.10 |
| BERT-Large    | 78.19±0.48 | 77.13±0.20 |
| RoBERTa-Base  | 78.85±0.25 | 77.37±0.11 |
| RoBERTa-Large | 79.02±0.16 | **77.66±0.12** |
Experiments - Flow Structure Prediction

| Models          | CTFW  | COR  | MAM  |
|-----------------|-------|------|------|
|                 | PRAUC | F1   | PRAUC | F1   | PRAUC | F1   |
| Baselines       |       |      |       |       |       |      |
| Random          | -     | 50.49| -     | 42.78 | -     | 47.82|
| Weighted Random | -     | 37.61| -     | 39.13 | -     | 44.10|
| BERT-NS         | 0.5751| 26.12| 0.5638| 43.14 | 0.5873| 29.73|
| RoBERTa-NS      | 0.5968| 32.44| 0.5244| 42.99 | 0.6236| 39.65|
| Ours            |       |      |       |       |       |      |
| BERT-GCN        | 0.7075| 69.26| **0.6312**| 58.13 | **0.6888**| 63.75|
| RoBERTa-GCN     | **0.7221**| 69.04| 0.6233| 61.44 | 0.6802| 65.73|
| BERT-GAT        | 0.5585| 61.93| 0.4553| 41.93 | 0.4568| 62.18|
| RoBERTa-GAT     | 0.5692| 64.51| 0.4358| 24.74 | 0.4585| 59.55|

- PRAUC scores for both LM-GCN versions are better than baseline next sentence prediction task (LM-NS)
- LM-GAT underperforms
## Effect of Graph Connection Type

For each Window, best model performs better than the baseline PRAUC scores (EP)

- Linear connections work better with smaller windows
- Semi-complete connections work better for $W_{all}$

|        | $W_3$  | $W_4$  | $W_5$  | $W_{all}$ |
|--------|--------|--------|--------|-----------|
| CTFW-SC| 0.6630 | 0.5985 | 0.5733 | 0.5590    |
| CTFW-L | 0.7221 | 0.6520 | 0.6150 | 0.3962    |
| CTFW-EP| 0.3700 | 0.2900 | 0.2400 | 0.0700    |
| COR-SC | 0.5639 | 0.5129 | 0.4731 | 0.5580    |
| COR-L  | 0.6456 | 0.6012 | 0.5274 | 0.4034    |
| COR-EP | 0.3700 | 0.3100 | 0.2600 | 0.1700    |
| MAM-SC | 0.6528 | 0.6219 | 0.6091 | 0.6718    |
| MAM-L  | 0.6888 | 0.6362 | 0.6137 | 0.4161    |
| MAM-EP | 0.4500 | 0.3700 | 0.3200 | 0.1500    |
Effect of Graph Layers

- Single GNN layer have better performance
- Increasing graph layers reduces the performance across all 3 datasets
Conclusion

- Introduced a new procedural sentence flow extraction task from natural language texts
- We create a sufficiently large procedural text dataset in the cybersecurity domain (CTFW) and construct structures from the natural form
- We empirically show that this task can be generalized across multiple domains with different natures and styles of texts
Thank You !!!

Contacts: Kuntal Kumar Pal, Kazuaki Kashihara, Pratyay Banerjee, Swaroop Mishra, Ruoyu Wang, Chitta Baral
Paper: https://arxiv.org/abs/2105.14357
Code: https://github.com/kuntalkumarpal/FlowGraph
Email: {kkpal, kkashiha, pbanerj6, srmishr1, fishw, chitta}@asu.edu