Carnegie Mellon University
Language Technologies Institute

Neural Language Modeling for Contextualized Temporal Graph Generation

Aman Madaan and Prof. Yiming Yang

Outline

- Temporal Graph Generation
- Methodology
- Experiments
- Conclusion

Temporal Graph Generation

Introduction

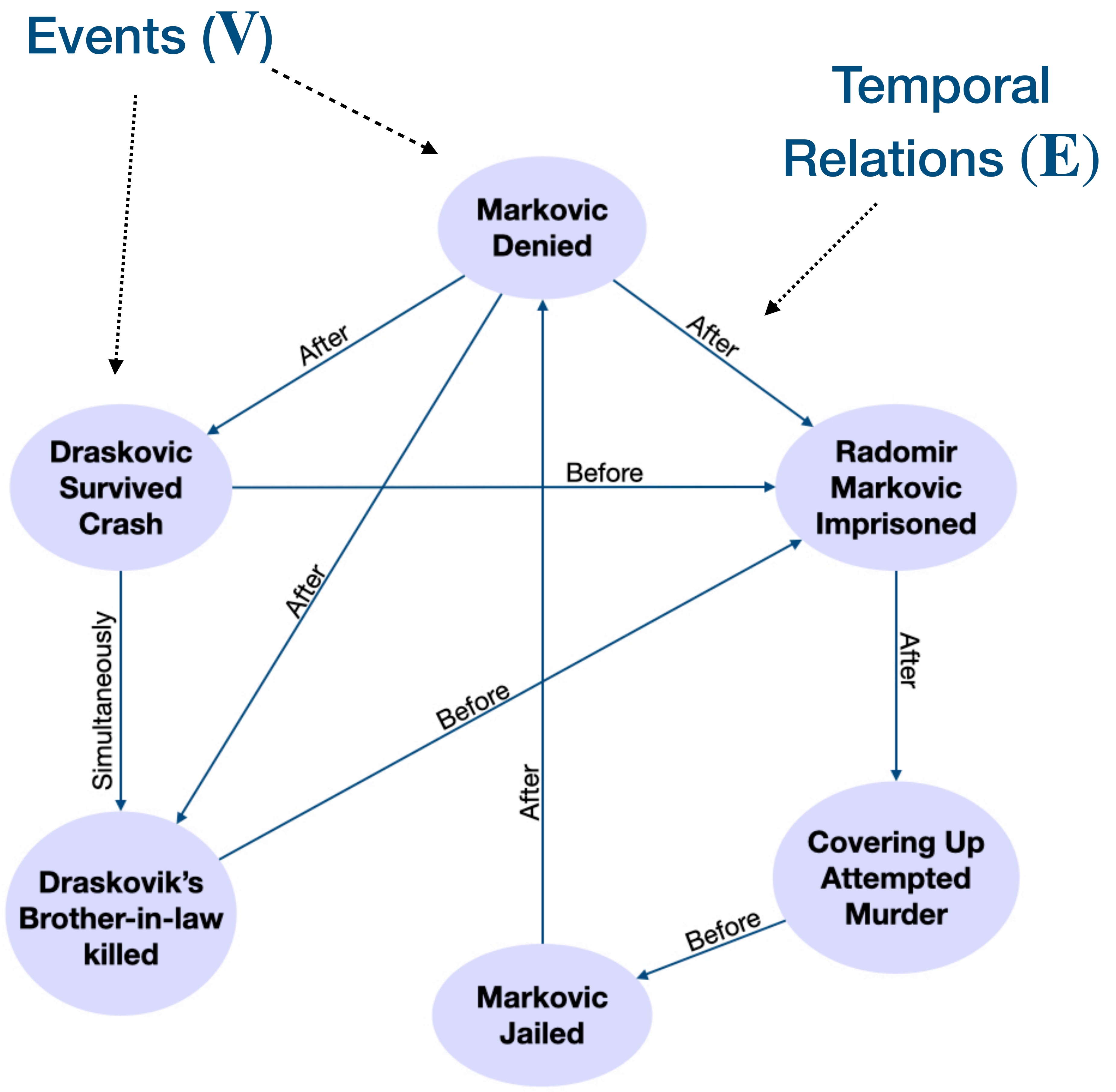
- Given a document $\mathbf{D}$, extract a graph $\mathbf{G}(\mathbf{V}, \mathbf{E})$ where the nodes $\mathbf{V}$ are the events and the edges $\mathbf{E}$ are temporal relations.
- Introduced in Tempeval-3 (UzZaman et al. 2013).
- Applications include topic detection and tracking, information extraction, parsing of clinical records, discourse analysis, and question answering.

Temporal Graph Generation

Example

Radomir Markovic, the former head of Serbian intelligence under Slobodan Milosevic, was jailed for seven years for covering up the attempted murder of a leading opposition politician in a 1999 car crash. Markovic, who has been imprisoned since 2001 for revealing state secrets, had denied there was ever a plot to kill Vuk Draskovic, the opposition leader, who survived the crash with minor injuries. Mr. Draskovic's brother-in-law and three others traveling in a convoy with him were killed.

Document ($\mathbf{D}$)



Graph ($\mathbf{G}$)

Temporal Graph Generation

Prior Work

- **CAEVO** (Chambers et al. 2014), and **Cogcomptime** (Ning et al. 2018)
  - Multi-stage approach (dependency parsing, event extraction, relation classification, etc.)
  - Different specialized system (rule based or statistical) for each stage, cascading errors.
  - Developed using a small corpus (36 documents for CAEVO, 276 for Cogcomptime).
- Lack of hand-labeled corpora is a major bottleneck in using latest developments in fine-tuning large scale language models on the Task.

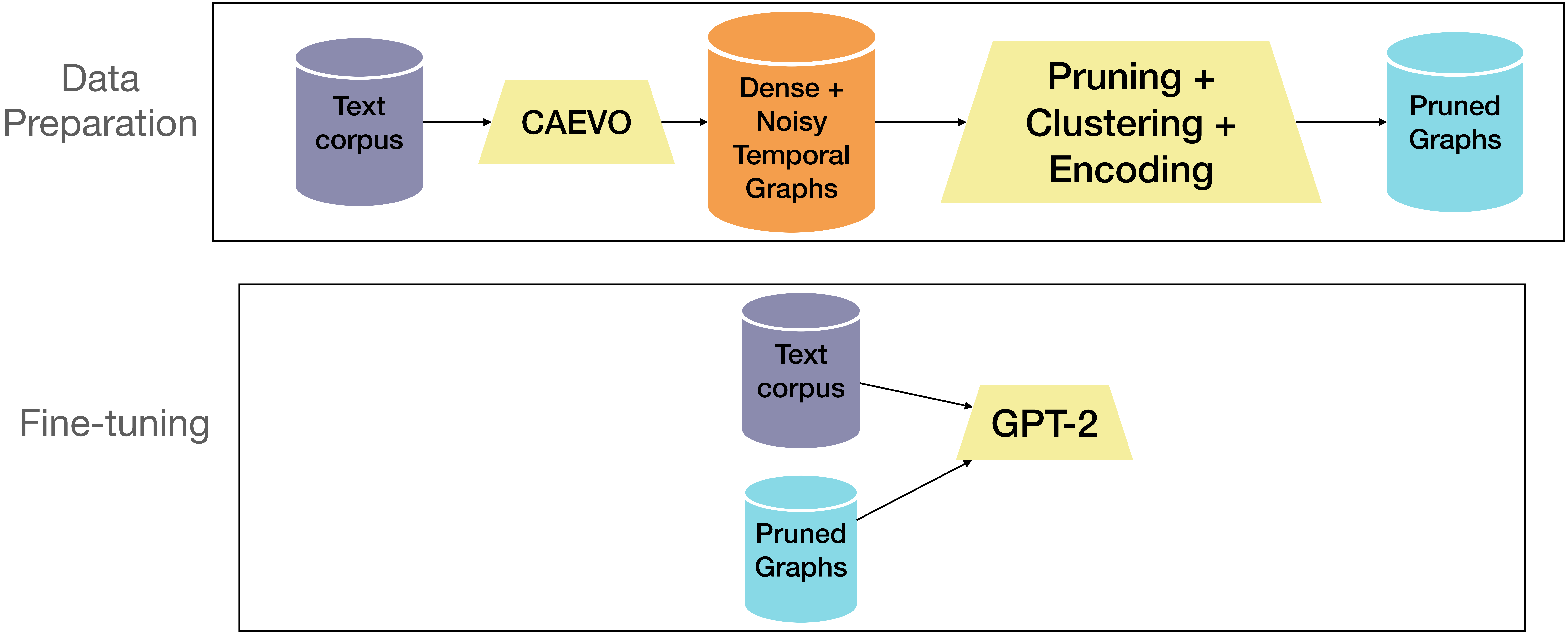
Temporal Graph Generation

Task Definition

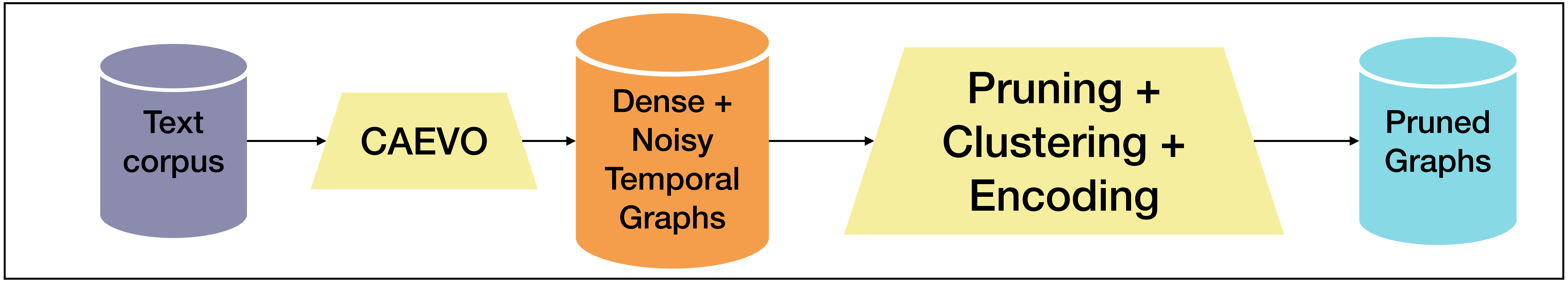
- $\mathbf{G}(\mathbf{V}, \mathbf{E})$ is the temporal graph corresponding to the given document $\mathbf{D}$.
- $r(e_p, e_q) \in \mathbf{E}$ is a temporal relation between events e_p and e_q .
- C_r be the set of sentences in the document $\mathbf{D}$ that contains the events e_p or e_q or are adjacent to them.
- We tackle two tasks of increasing complexity.
- **Node generation (Task 1)**
    - Given the context C_r, source event e_p, and a temporal relation r, generate the target event e_q.
- **Graph Generation (Task 2)**
    - Given a document $\mathbf{D}$, generate a temporal graph $\mathbf{G}$.

Methodology

Temporal Graph Generation



Dataset Preparation



Dataset Preparation

Source

- We use the New York Times Corpus to collect a dataset of document-graph pairs.
- 1.8 million articles between 1987 - 2007, each article has a hand-assigned descriptive term.
- Filter articles related to bombing, terrorism, murder, riots, hijacking, assassination, kidnapping, arson, vandalism, serial murder, manslaughter, extortion.
- ~90k articles total.

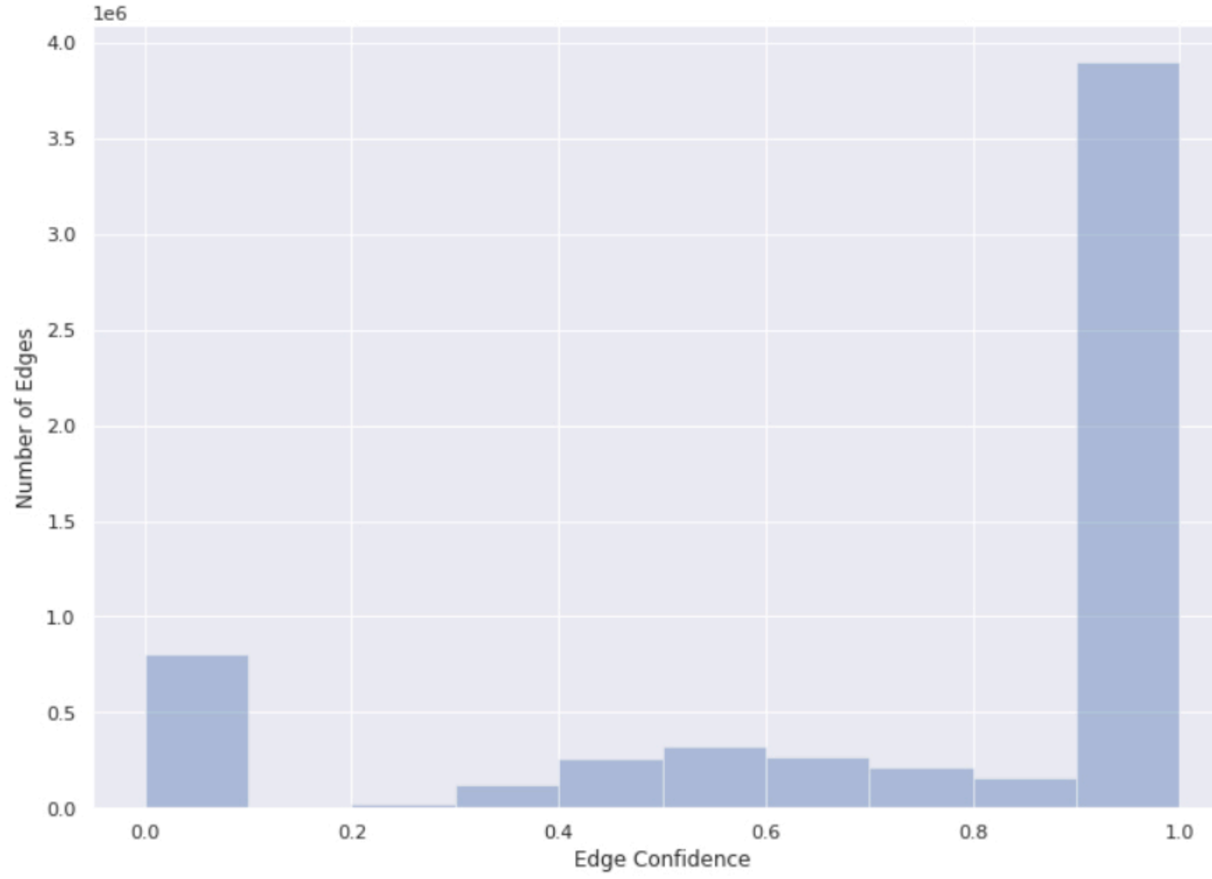
Descriptor	%Articles
Terrorism	23.69
Murders and attempted murders	14.57
US International Relations	10.28
US armament and defense	9.72
airlines and airplanes	9.32
world trade center (nyc)	8.77
demonstrations and riots	8.38
hijacking	8.38
politics and government	3.63
bombs and explosives	3.25

Dataset Preparation

Generating supervised data with CAEVO

- Use CAEVO (Chambers et al., 2014) to extract a temporal graph for each document.
- Scaleable: need to label over ~90k documents ✔
- Noisy verbs: ~10% of all the events were *said* ✘
- Relations extracted with ~0 confidence ✘
- Extract events as standalone verbs, adds ambiguity ✘
    - A *killed* B, after which C *killed* A →after(killed, killed)

Event verb	Raw frequency	% Frequency
said	647685	9.60
say	57667	0.86
had	47320	0.70
killed	43369	0.64
told	42983	0.64
found	41733	0.62
made	40544	0.60
war	35257	0.52
get	30726	0.46
make	29407	0.44



Dataset Preparation

Pruning and post-processing

- Remove verbs that have i) a low-idf, are ii) light or reporting (Liu et al. 2018; Recasens, Hovy, and Martí 2010)
    - i) low idf: “said”, “say”, “had”, “made”, “told”, ii) light: “appear”, “be”, “become”, “do”, “have”, “seem”, “get”, “give”, “go”, “have”, “keep”, “make”, “put”, “set”, “take”, iii) reporting: “argue”, “claim”, “say”, “suggest”, “tell”’
- Augment each verb with the corresponding noun phrase and object (Chambers and Jurafsky 2008)
    - A *killed* B, after which C *killed* A →after(C killed A, A killed B)
- Drop relations that have a confidence of < 0.50, all the *vague* relations, retain all the rule-based extractions. Relation set: *before*, *after*, *is included, includes*, *simultaneously*

Dataset Preparation

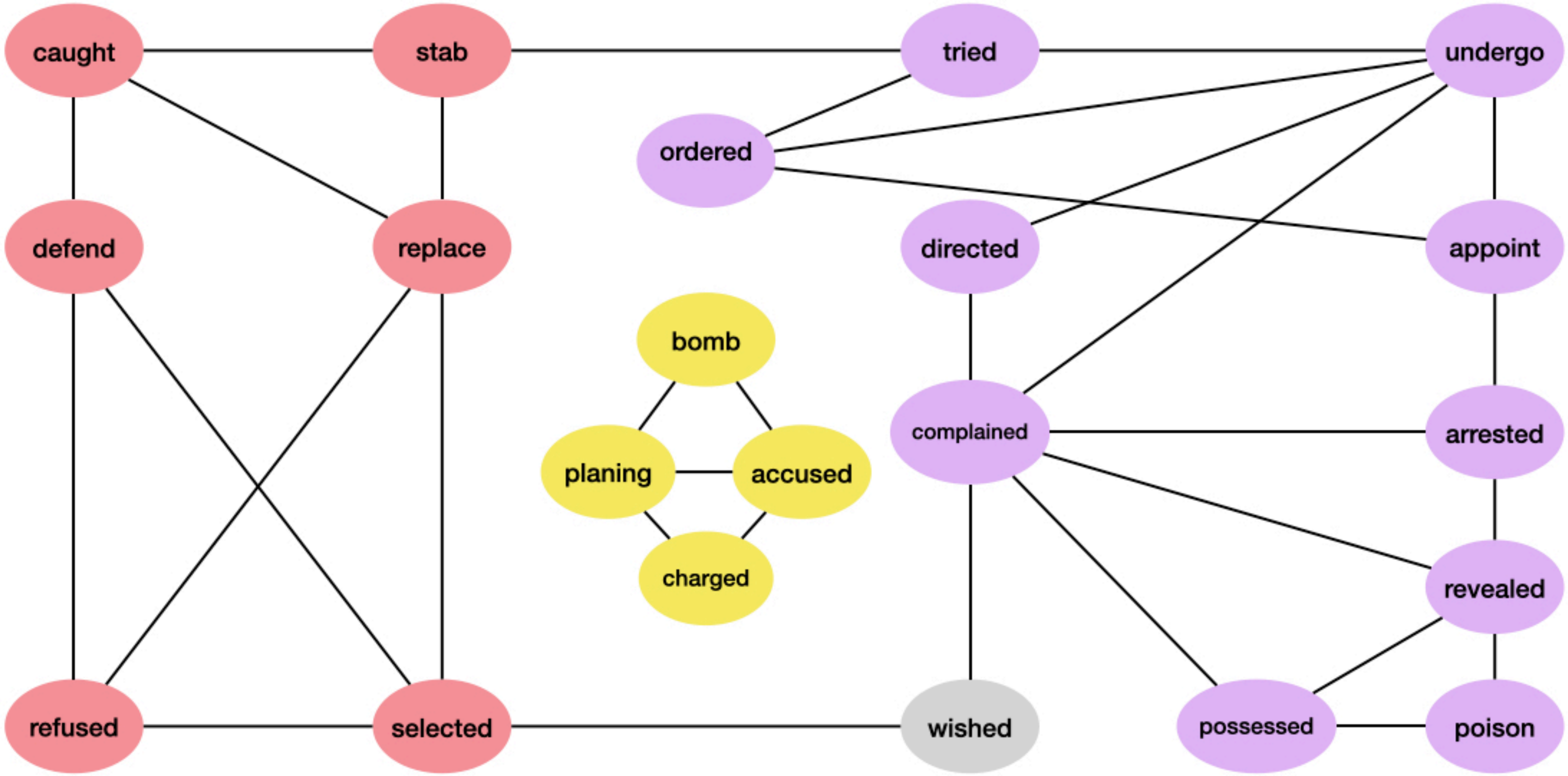
Clustering Event Communities

- **Observation**
  - Temporal graph typically has several sub-graphs that are either completely disconnected or have high intra-link density.
  - Typically arise from different topics in the narrative.
  - Each of these sub-graphs refers to a certain parts of the document.
  - Use this property to ground each sub-graph in its proper context.

- **Our approach**
  - Divide a temporal graph G into sub-graphs induced by event communities $G_1, G_2, \ldots, G_n$ (using Newman et al. 2004).
  - Let $D_1, D_2, \ldots, D_n$ be the corresponding parts in the document.
  - Add each of (D_i, G_i) to the training data.

Dataset Preparation

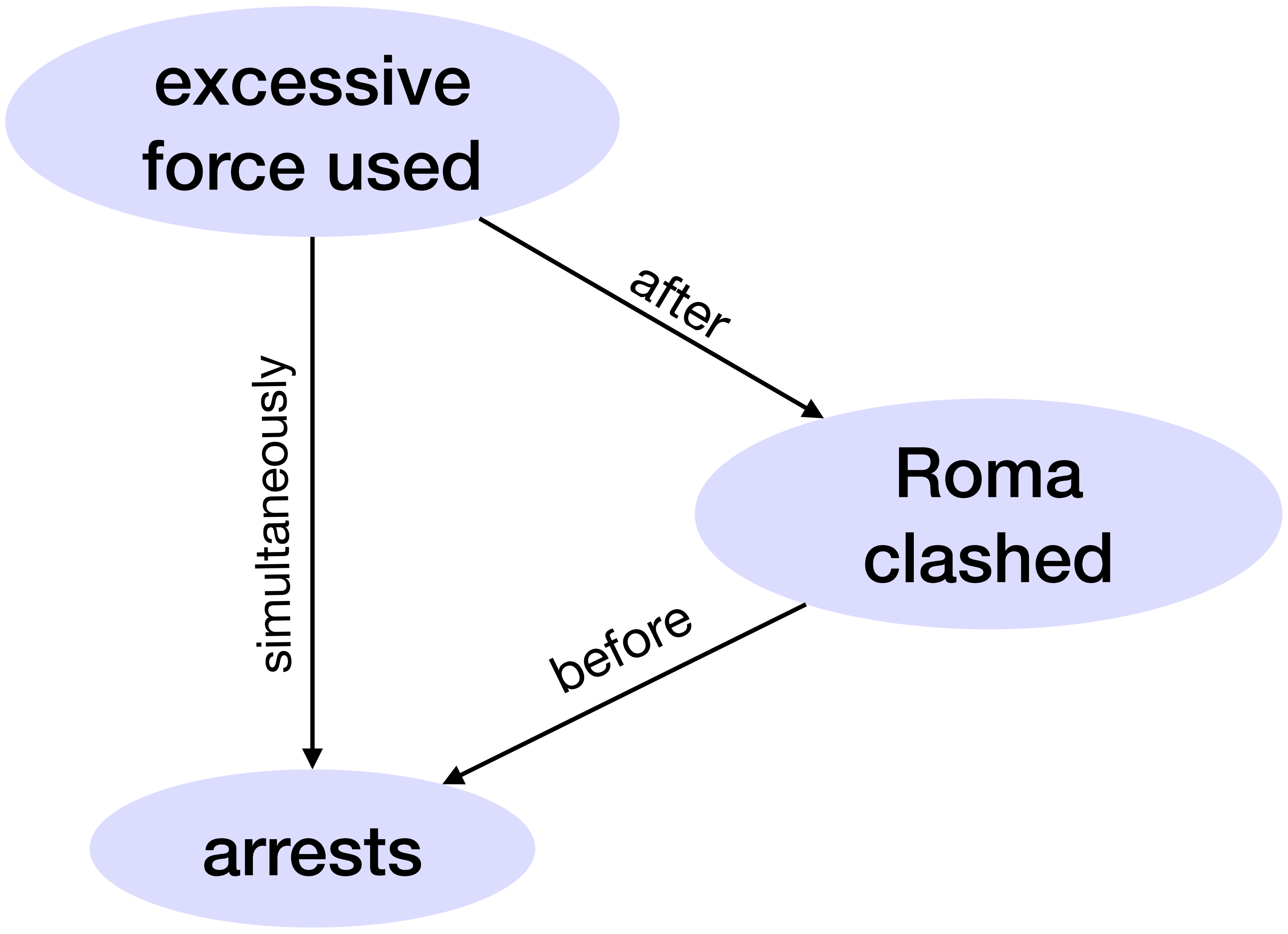
Clustering Event Communities



Dataset Preparation

Encoding Graphs

- We use DOT (Gansner, Koutsofios, and North 2006) to represent each graph as a string.
- The edges are listed in the order in which their constituent nodes appear in the document

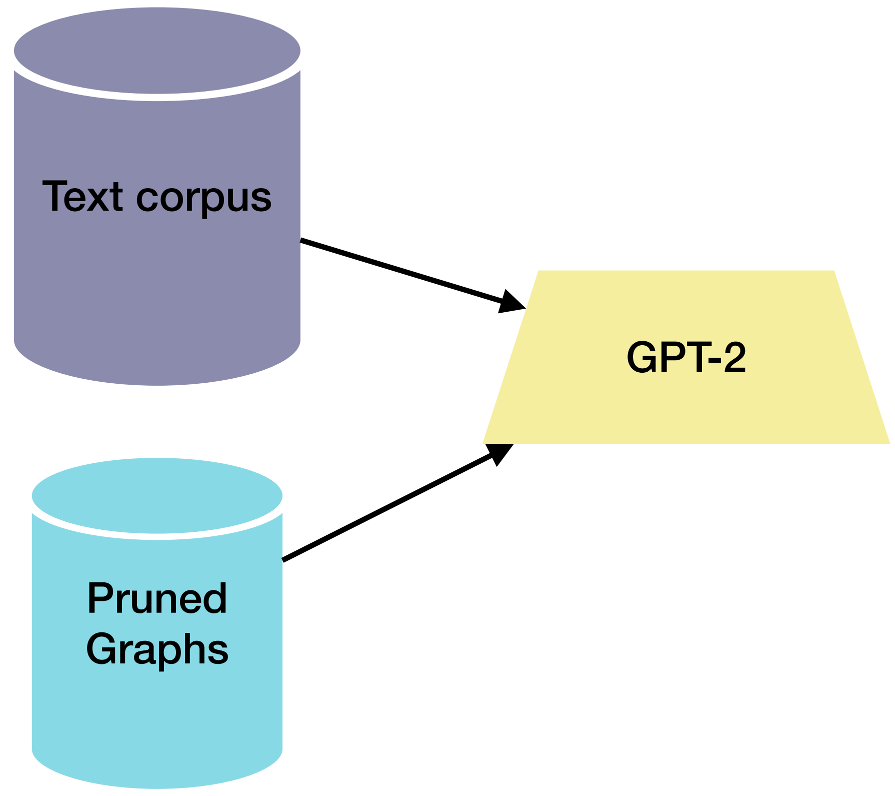


⬌

```
digraph  {
    "Roma clashed" -> arrests  [ label = before ];
    "excessive force used" -> "Roma clashed" [ label = after ];
    "excessive force used" -> arrests [ label = simultaneously ];
}
```

Fine-tuning

- The training data for both Tasks 1 and 2 comprises of tuples $\mathscr{D} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{N}$.
- We aim to estimate the distribution $p_\theta(\boldsymbol{y}_i \mid \boldsymbol{x}_i)$
- $\mathscr{L}_{\text{masked}}(\mathscr{D}) = -\sum_{i=1}^{|\mathscr{D}|} \sum_{j=1}^{|x_i|+|y_i|} m_{i,j} * \log(p_\theta(u_{i,j} \mid \boldsymbol{u}_{i,<j}))$
  - $\boldsymbol{u}_i = \boldsymbol{x}_i \| \boldsymbol{y}_i$
  - $m_{i,j} = \begin{cases} 0 & \text{if } u_{i,j} \in \{\boldsymbol{x}_i\} \\ 1 & \text{otherwise} \end{cases}$
- We experiment with LSTM and GPT-2 to parameterize p_θ



Experiments

Experiments

Datasets

- **TG-Gen:** Test split of our dataset.

- **TB-Dense:** A hand-labeled corpus of 36 documents from mixed domains, derived from TimeBank-Dense (Cassidy et al. 2014).

Task	Split	#Samples
Task 1	train	4,260,328
Task 1	valid	542,994
Task 1	test	541,844
Task 1	total	5,345,166
Task 2	train	709,929
Task 2	valid	89,407
Task 2	test	91,341
Task 2	total	890,677

TG-Gen Dataset Statistics

Experiments
GPT-2 and Baselines

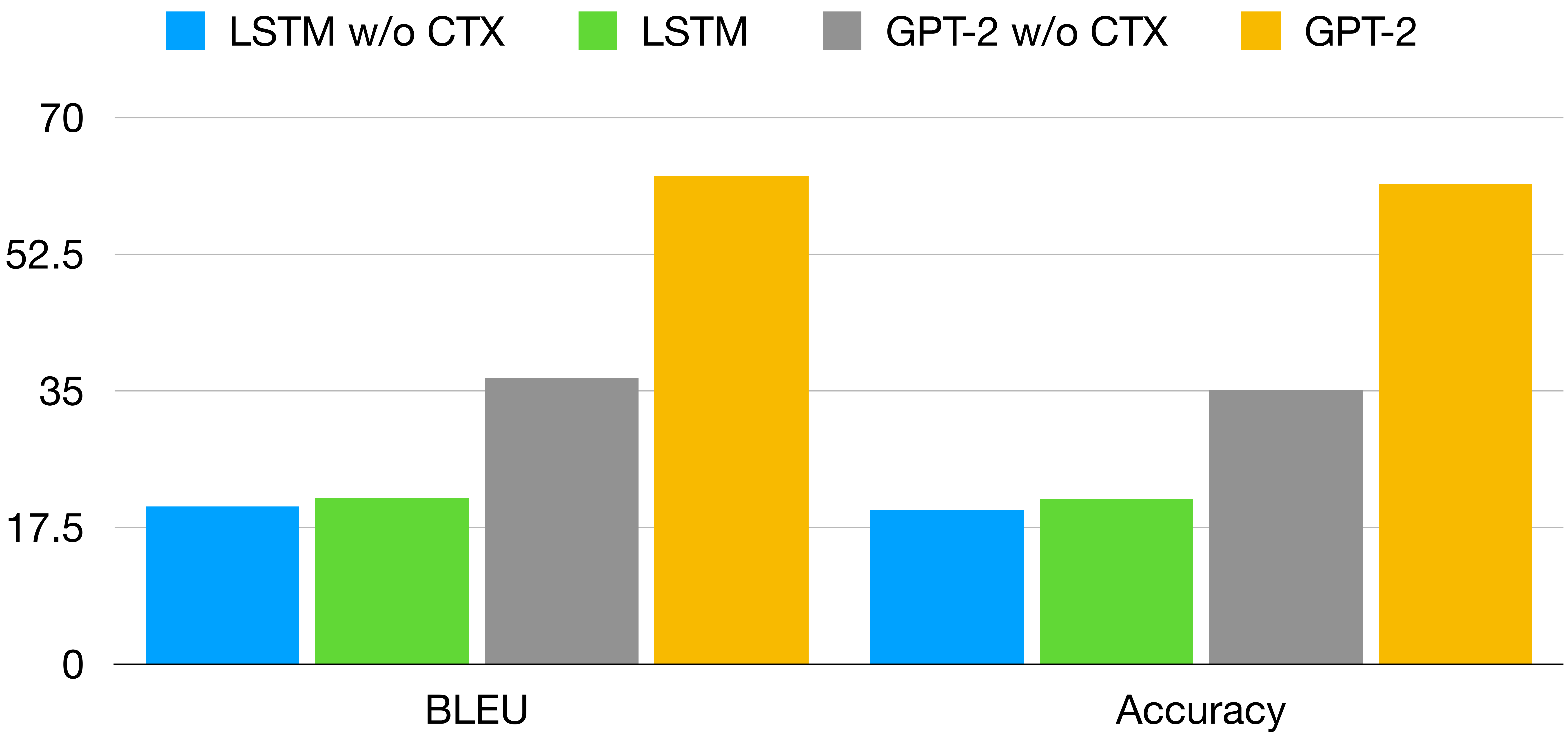
- **GPT-2**
  - GPT-2 medium (355M parameters), 24 layers, hidden-size of 1024, 16 attention heads.
  - Fine-tuned using masked language modeling loss (probability of graph tokens given the text), samples drawn using nucleus sampling (Holtzman et al. 2019).
- **LSTM**
  - Bidirectional encoder, uni-directional decoder (2-layers each), embeddings initialized with 300-dim Glove.
- **CAEVO**
  - Use to compare the performance of our system on TB-Dense.

Experiments
Task Definition

- **Node generation (Task 1)**
  - Let $r(e_p, e_q) \in E$, C_r be the set of sentences in the document $\mathbf{D}$ that contains the events e_p or e_q or are adjacent to them.
  - Given the context C_r, source event e_p, and a relation r, generate the event e_q.
- **Graph Generation (Task 2)**
  - Given a document $\mathbf{D}$, generate a temporal graph $\mathbf{G}$

Results

Node Generation (TG-Gen)



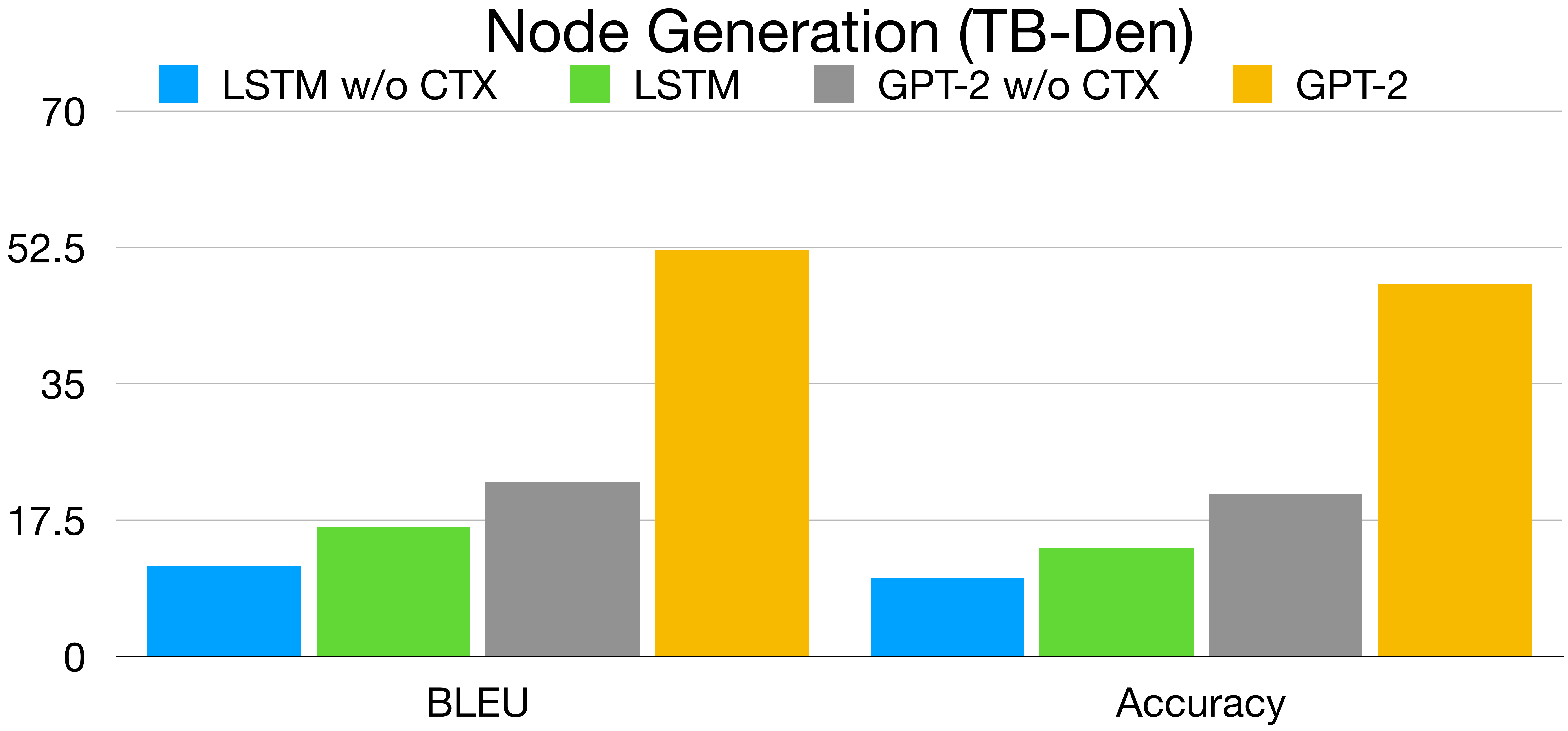
w/o context: Sentence containing the target event missing

Results

Node Generation (TB-Den: hand-labeled, out-domain)

- Pre-training helps in out-domain settings.

- Context becomes even more important on the out-of-domain corpus!



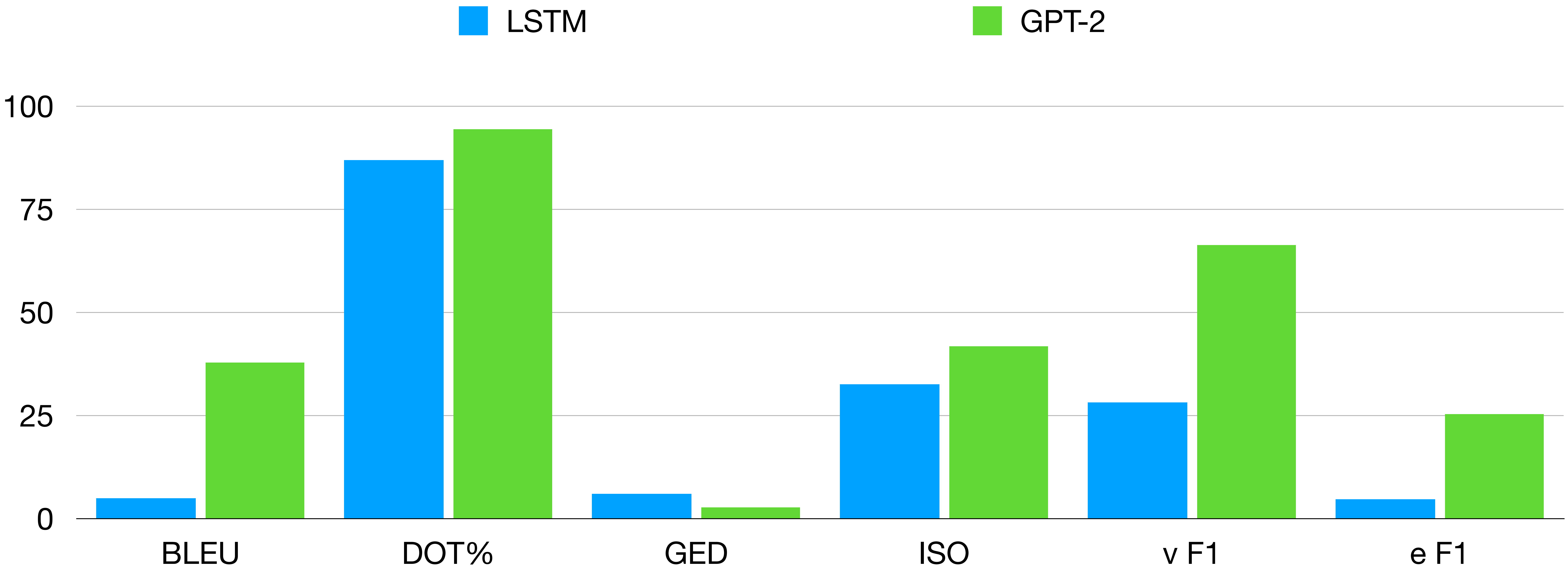
Results

Graph Generation: Metrics

- **BLEU:** String metric to evaluate the string representations of the graphs
- Isomorphism (**ISO**), graph edit distance (**GED**), average degree, size of the node and edge sets: To compare the graph structures
- Edge and node set precision (**P**), recall (**R**), F-score (**F1**)
- **DOT%**: % of generated graphs which are valid DOT files

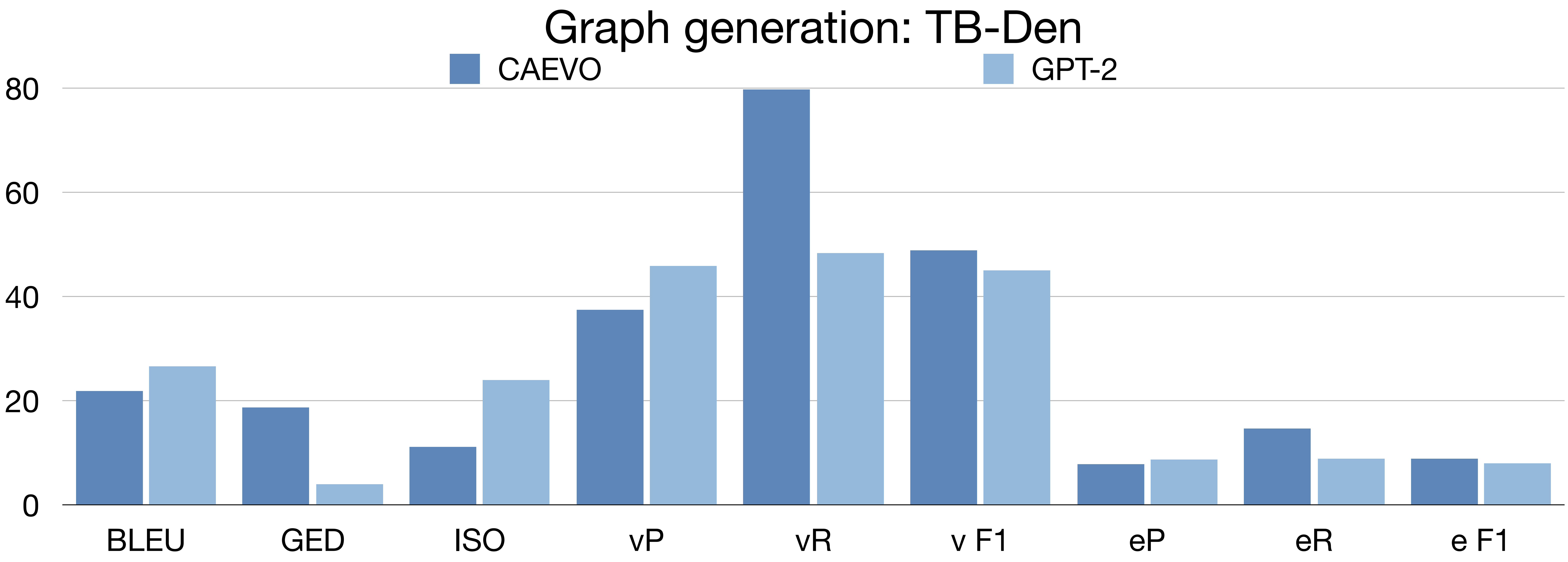
Results

Graph generation: TG-Gen



Results

Graph generation: TB-Den



Conclusion

- Difficult to of obtain large corpora of human-annotated graphs for temporal reasoning over events to fine-tune language models.

- We develop a data generation pipeline that uses existing IE/NLP/clustering techniques for automated acquisition of a large corpus of document-graph pairs.

- We propose a new formulation of the graph generation task as a sequence-to-sequence mapping task, allowing us to leverage and fine-tune pre-trained language models for our goal.

- Our experiments strongly support the effectiveness of the proposed approach, which significantly outperforms strong baselines, and is competitive with traditional IE techniques.

Thanks!