# Tactical Analytics

Ed Morris
Dr. Jaime Carbonell
Ben Bradshaw
Kevin Pitstick
Petar Stojonov
Daegun Won

# Distribution Statements



This material is based upon work funded and supported by the Department of Defense under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.



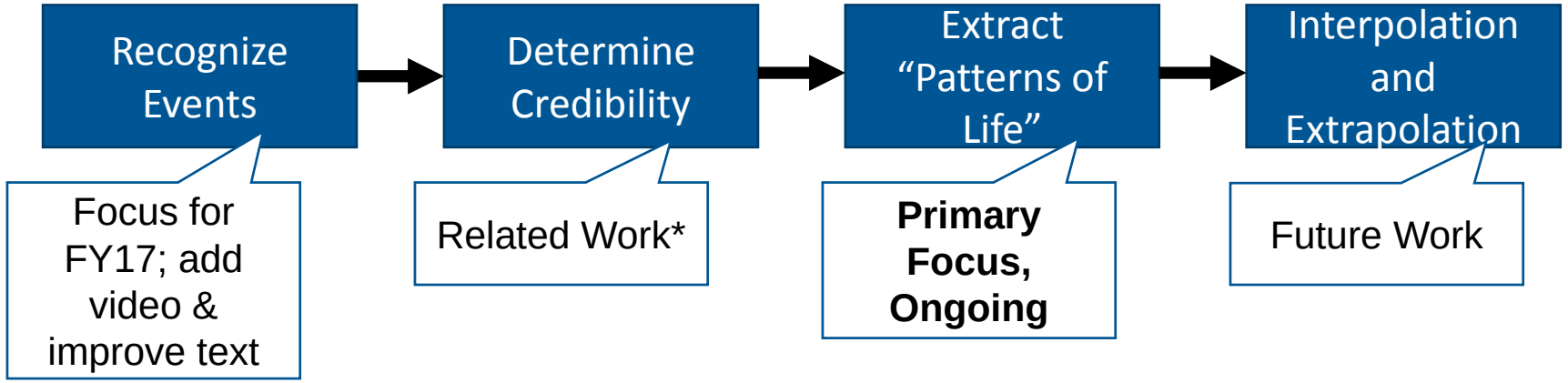
DM-0004127

# Edge Analytics Pipeline for Streaming Situational Awareness

## Previous Work:

- Developed a platform for building and testing data analytics for streaming textual data
  - Tested multiple analytics in public safety settings; Multi-day music festivals, Little League World Series, Visit of Pope Francis
  - Retrospective analysis: Cairo and Benghazi

## Current Focus:




*Information about this work is available during poster session

# Patterns of Life and Scripts

- A “Pattern of Life” represents stereotypical sequence of events and interactions in a particular context
- The research community calls these “scripts” (Schank & Abelson, 1977)
- Scripts help analysts relate emerging situations to what is already known
  - “I have seen that before” (recognition of instance of a script)
  - “He probably entered through Turkey” (what happened previously – interpolation)
  - “Since he is an American with good language skills, they will probably use him as a spokesperson” (what may happen next -- extrapolation)
- Scripts are a good way to introduce data to a new analyst or an analyst in a new job

# Pattern of Life examples

Author: Day Donaldson
License: Creative Commons Attribution 2.0

ISIL script for takeover of a village[1]:

- ❑ List the powerful families
- ❑ Name the powerful individuals
- ❑ Find out income sources
- ❑ Identify names, sizes, and control of rebel brigades
- ❑ Identify illegal activities (Sharia Law) that could be used for blackmail

Other Examples[2]:

- Russian aggression in Crimea[2]:
  - Transporting or erecting missile launcher: Apr 6 2012: "[Russian] military has begun deploying S-400 mobile surface-to-air missiles in Kaliningrad"
  - Mobilization of forces; Russian military activity increased in the 12 months leading up to the Crimean invasion. Information was available on the web indicating this activity
- North Korean nuclear test preparation e.g. vehicle activities, site activities, etc.

[1]From notes of Samir Abd Muhammad al-Khlifawi (Haji Balr), considered the architect of ISIS, killed in a firefight with Syrian rebels. Found with org charts, lists, & schedules describing ISIS strategy <http://www.spiegel.de/international/world/islamic-state-files-show-structure-of-islamist-terror-group-a-1029274.html>

[2] Recorded Future Blog <https://www.recordedfuture.com/russian-military-activity/>

# Goals and Challenges

Long term goal: build the pipeline to recognize then validates events, recognizes patterns (scripts) and allows for interpolation (previous events) and extrapolation (prediction of future events)

Objective for FY16: Understand the challenges of script learning

- Event recognition and ordering
  - Recognizing events in free-form text
    - DARPA DEFT is improving single and multiple sentence event recognition
    - No viable solution for multiple document event recognition
  - Establishing relationships among events (e.g., order, causality)
- Credibility analysis of events (more information available)
- Script creation and modification
  - Determining if event sequences represent a new script or an instance of an existing script
  - Preventing invalid pathways from being incorporated into scripts

# Our Approach

Events = {Actor, Activity, Time, Location … }

Problems with event recognition forced us to use a proxy dataset with easily recognizable events

- Box scores for major league baseball are readily available and events are very easy to recognize (runs, outs, etc.)

Deal with script extraction and matching; not on performance and volume issues (simple scripts aren't small)

Learn constraints based on initial portion of the dataset in real-time

Generate the script that represents the data

- Structure of the DAG (directed acyclic graph) representing ½ inning
- frequency for branches in the DAG (based on box score data)


Author: Flickr User alpineinc
License: Creative Commons Attribution 2.0

**Scripts are built in real-time using streaming data with no previous analysis besides events being properly recognized**

# Pseudocode

**Algorithm 1** Script Building Algorithm

**Input:** a list $S$ of $N$ sequences of events, distance metric threshold $R$, timer threshold $T$
**Output:** a resulting list of scripts, $L$

```
S = ∅
for s ∈ S[1 − 500] do
    Generate uniqueness constraints U and pairwise order constraints P.
end for
for s ∈ S[501 − end] do
    if L == ∅ then
        add s as a script to L
    else
        for l ∈ L do
            for traversal t in l do
                if dist(t, s) ≤ R then
                    mark l for adding
                end if
            end for
        end for
        if an l ∈ L is marked for adding then

            for traversal t in l do
                if dist(t, s) is smallest and no constraints in U and P are violated
                then
                    add s to l
                end if
            end for
        else
            add s as a script to L
        end if
    end if
    for l ∈ L do
        l.timer++
        if l.timer > T then
            remove l from L
        end if
    end for
end for
return L
```

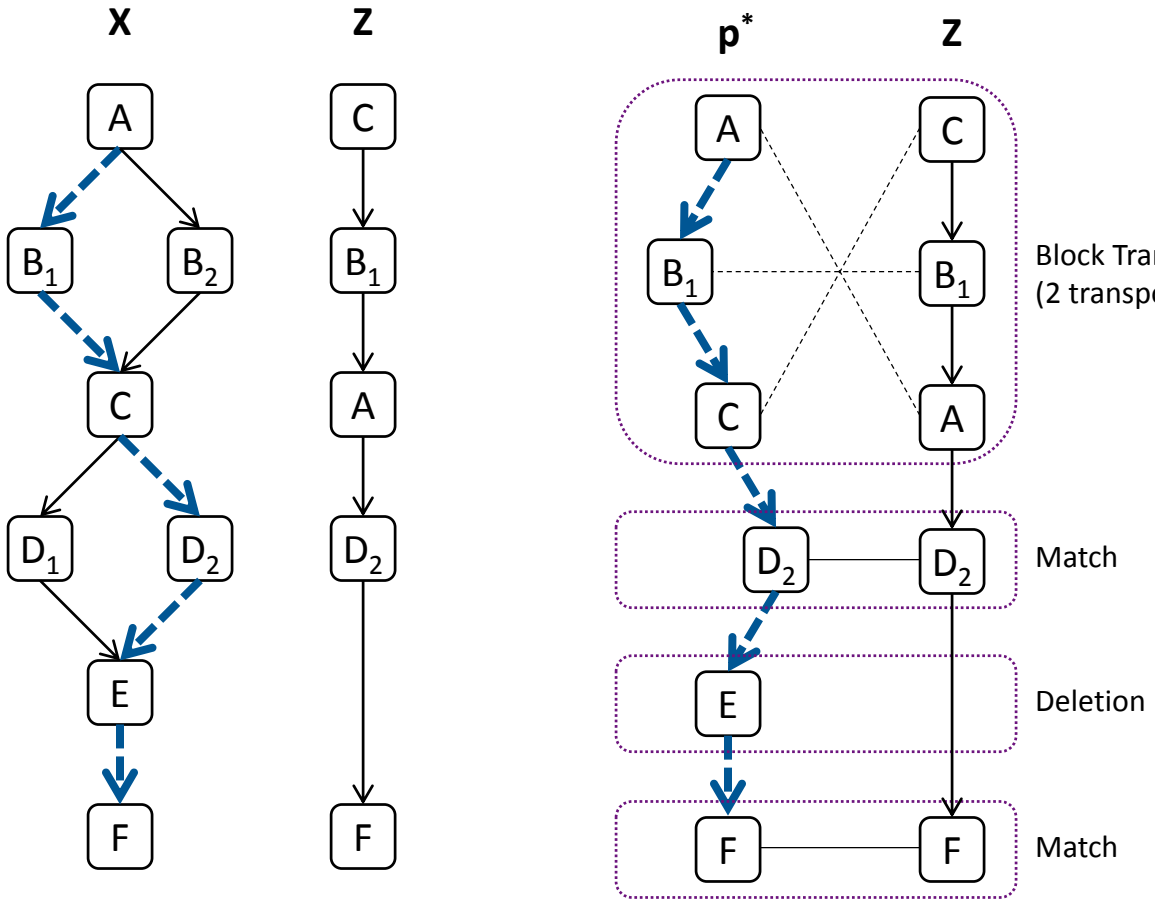
**Algorithm 2** Script Building Algorithm

**Input:** a list $S$ of $N$ sequences of events, distance metric threshold $R$, timer threshold $T$
**Output:** a resulting list of scripts, $L$

```
S = ∅
for the first 500 sequences s do
    Generate uniqueness constraints U and pairwise order constraints P.
end for
for s ∈ the remaining sequences in S do
    if Working list of scripts L is empty then
        add s as a script to L
    else
        for each script l ∈ L do
            Check if any of the traversals of l have a small enough distance metric
            compared to candidate s, and if so, mark l for addition.
        end for
        if any l ∈ L is marked for adding then
            Add s to the minimum distance traversal of l if no constraints are
            violated
        else
            add s as a script to L
        end if
    end if
    Remove any scripts from L that have not been incremented in T iterations.
end for
return L
```

# Recognize if a New Sequence Represents a New Branch of an Existing Script

## Script X and sequence Z



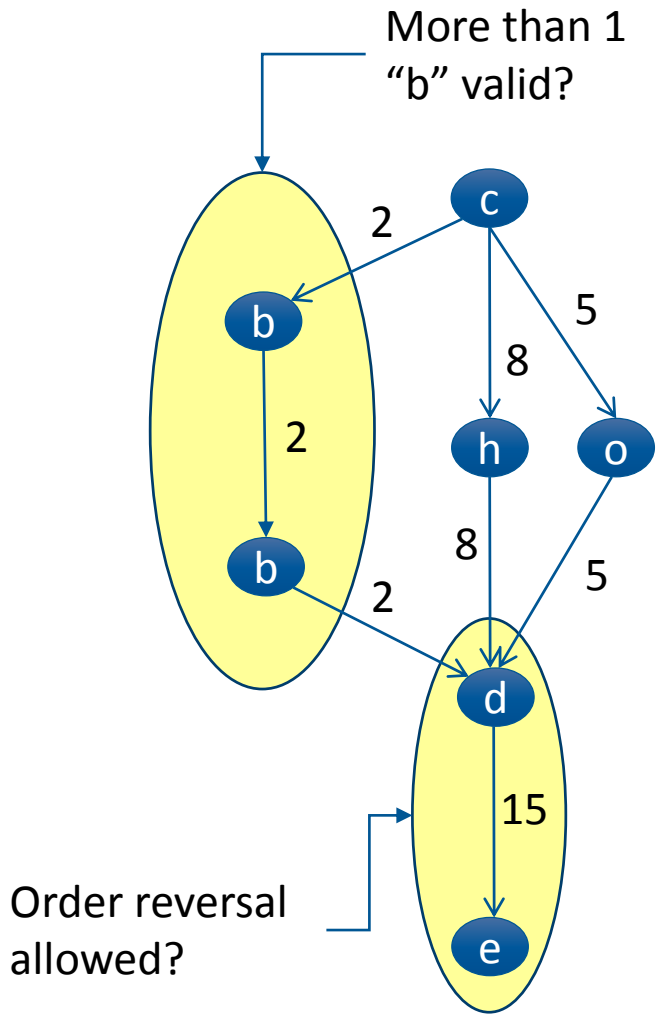

## Similarity

$$s(X,Z) = 1 - \arg\min_{p \in Path(X)} \left[ \frac{\sum_{x \in p} \beta_x \delta(x,Z)}{\sum_{x \in p} \beta_x} \right]$$

$$\delta(x,Z) = \begin{cases} 0 & \text{a match} \\ \alpha_1 & \text{if B.T.} \\ \alpha_2 & \text{insertion} \\ \alpha_3 & \text{deletion} \end{cases}$$

$\beta_x = 1$ unless specified otherwise by the user

# Preventing Invalid Pathways in Scripts

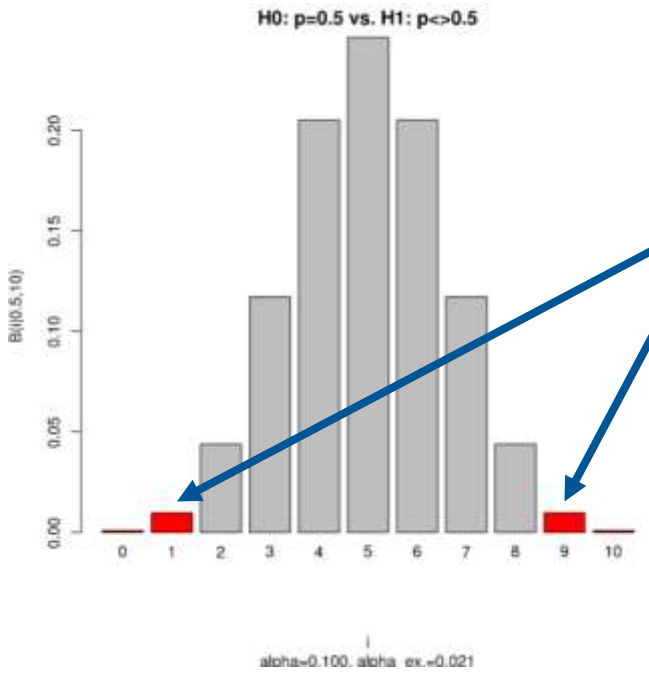



Apply constraints to prevent invalid scripts:

- Input from an expert analyst
  - Direct feedback
  - Learned from curated reports
- Learn from the data (*constraint induction*)
  - Joint learning (learn constraints and scripts while processing operational data) is ideal, but hard
  - Observe initial M sequences to infer constraints before inferring scripts
  - Learn from labeled data

When updating a script, new pathways are checked against *constraints*

# Constraints Induction: Order and Uniqueness




Order: Pairwise Constraints

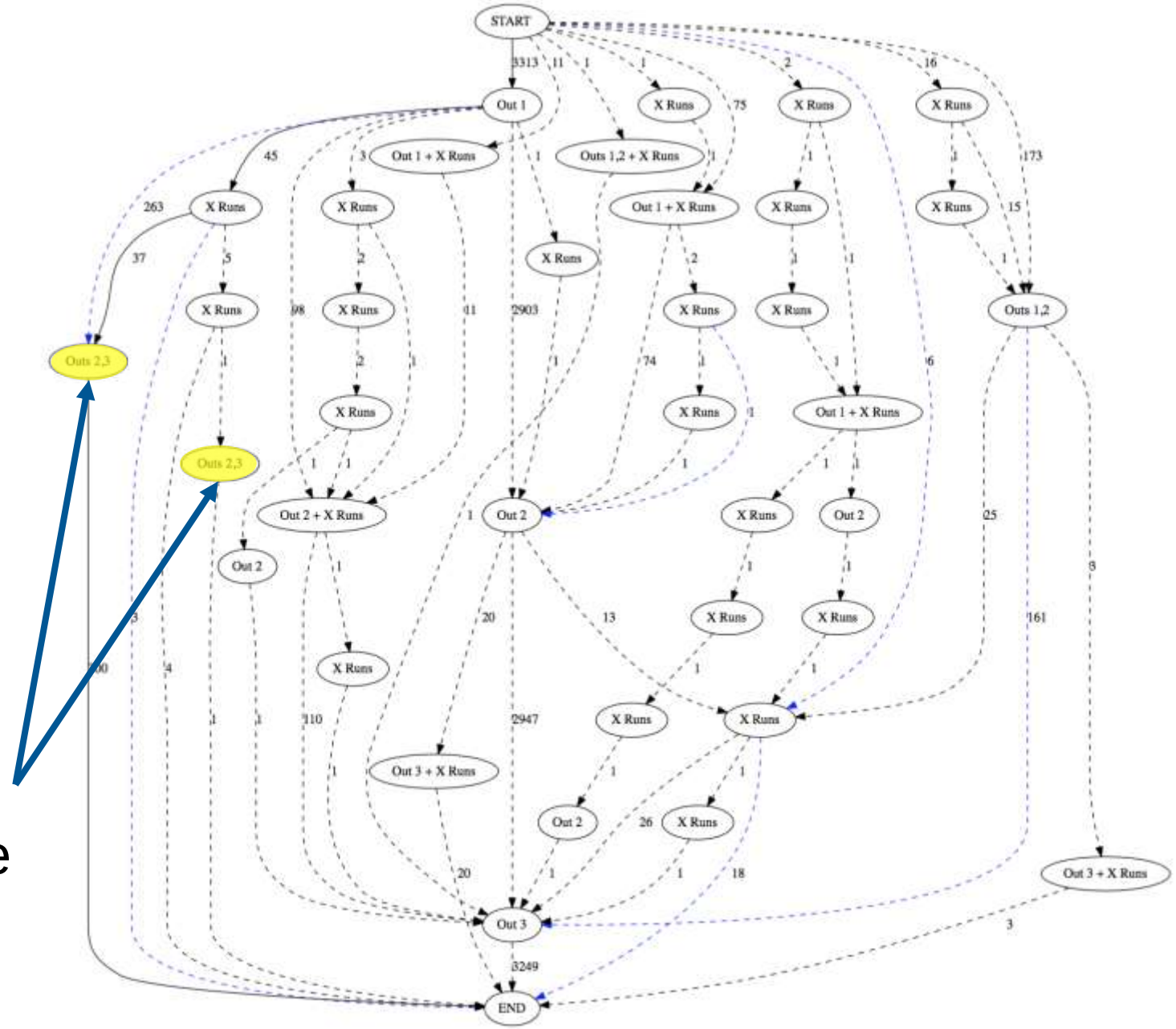
ex) Event X happens before event Y “X >> Y”

- If no order exists we should see each pair ~50/50
- If order does exist we should see much higher frequency for “X >> Y” or “Y >> X”

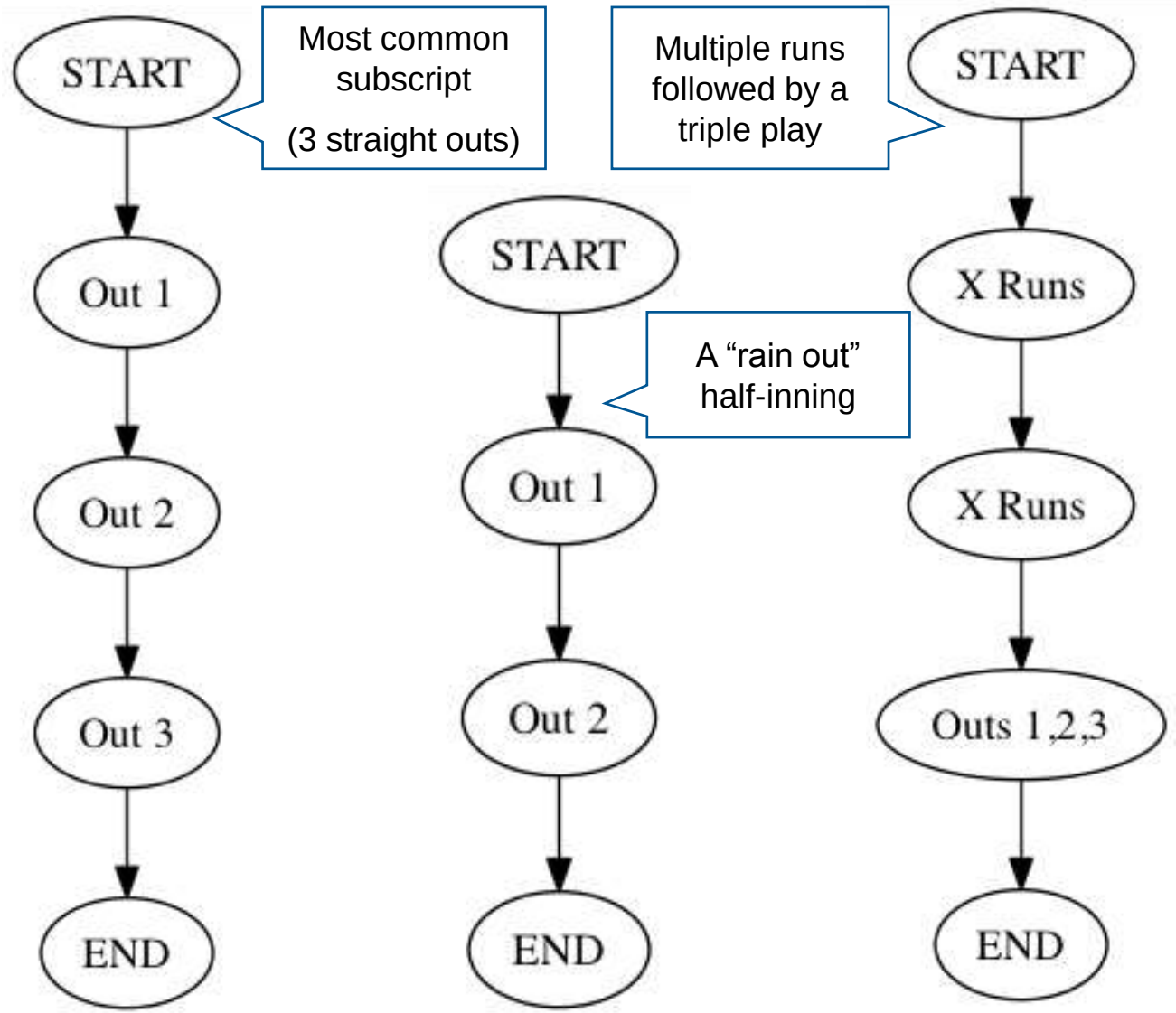
- Uniqueness: Similar strategy can be applied by looking at the probability of a specific event.
  - When X happens, X happens only once some high percentage of time
  - P(X happened only once in >= M sequences ; N Observations) > Threshold(~ 0.01)?

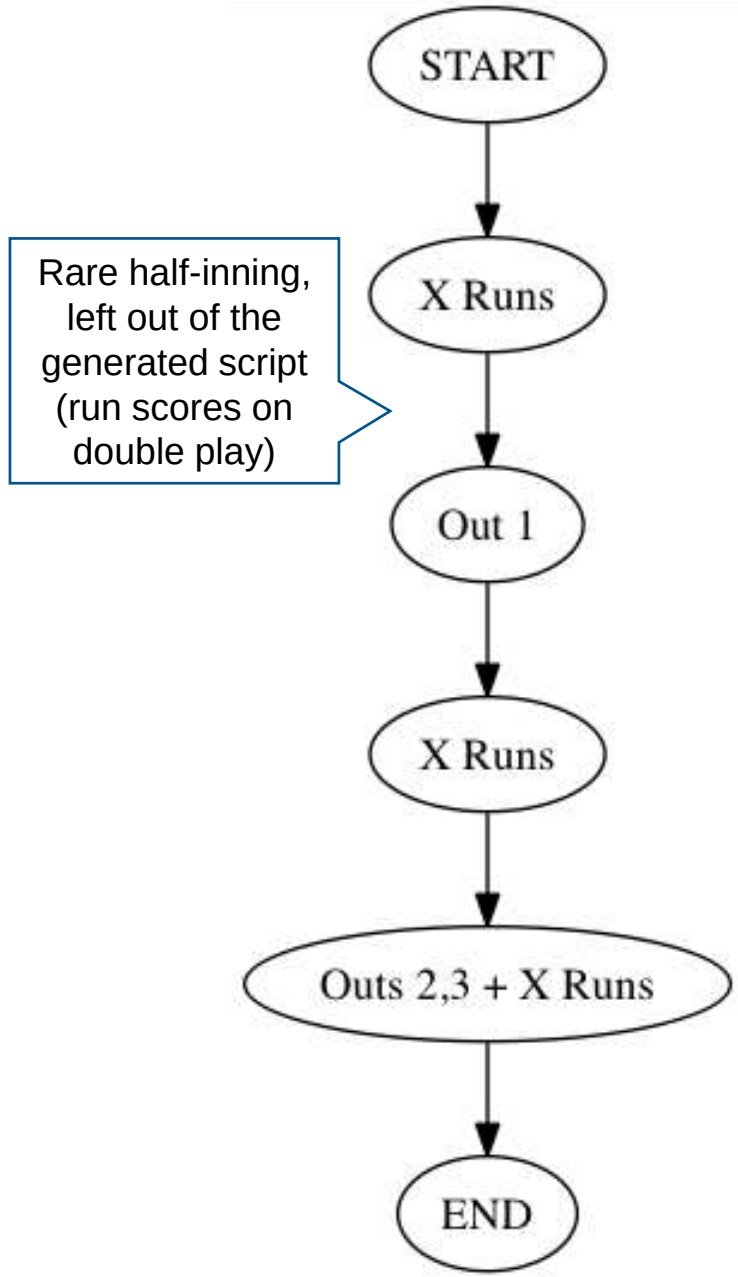
# Example Using Box Score Data

- Generated using streaming events extracted from 5000 baseball half-innings
- Constraints identified in first 500 half-innings, rest of data used to generate script
- Some imperfections in the algorithm (e.g., some pathways should be collapsed



# Subscript Instances



# Lessons Learned

Scripts can be induced from streaming data

- Assuming events are correctly identified

Constraints are necessary to avoid obviously invalid pathways

- Some of them can be learned directly from the data
- More needed to improve accuracy

Even a simple test case is very complicated

- We chose a situation with easily identifiable and correctly tagged events
- We simplified the types of events we considered (e.g. X Runs scored instead of 1 Run, 2 Runs, etc.)
- We focused on events that were directly related to the length of the script
- We duplicated sequences of events rather than allowing cycles -- DAGs

# FY17 Focus

- Event recognition and extraction
  - Use curated data sets to train classifiers to recognize events in un-curated text
  - Develop approach to recognize events across multiple sources
    - Build on work performed under DARPA contract at CMU and elsewhere
- Extending script learning algorithm to more meaningful data
  - More sophisticated approaches to learning constraints (constraint induction)
  - Direct analyst input for constraints and feedback during script generation
- Recognize that a partial set of events is part of an existing script
  - Necessary for both interpolation and extrapolation