Computerized Coding of Event Data

Javier Osorio

School of Government and Public Policy
University of Arizona

Prepared for the
Winter Institute in Computational Social Science (WICSS)
University of Arizona

Virtual event, January 7th, 2021

Route

MOTIVATION

Motivation

We want to measure (and explain) social behavior in a systematic way

Systematic measurement

Systematic measurement of political events:

- Concept validity
- Consistent application of coding method
- Transparent and verifiable
- Minimize measurement error
- Enable explanation

Systematic measurement

Systematic measurement of political events:

- Concept validity
- Consistent application of coding method
- Transparent and verifiable
- Minimize measurement error
- Enable explanation

Is feasible:

- Massive availability of online text
- Real-time information
- Global coverage

WHAT IS EVENT DATA?

Event Data Elements

Event data is a categorical description of:

- Someone
- Doing something
- To someone else
- At a given moment
- In a certain place

Event Data Elements

Event data is a categorical description of:

- Someone → Source
- Doing something → Action
- To someone else → Target
- At a given moment → Date/time
- In a certain place → Location

Event Data Elements

Event data is a categorical description of:

- Someone	→ Source	→ Subject
- Doing something	→ Action	→ Verb
- To someone else	→ Target	→ Object
- At a given moment	→ Date/time	→ Date/time
- In a certain place	→ Location	→ Toponyms

Example

Example of event coding:

- Event statement ✓
- Event description
- Event coding

```
"Yesterday,  Police  in Denver  arrested  BLM protesters"
```

Example

Example of event coding:

- Event statement ✓
- Event description ✓
- Event coding

`"Yesterday,`	`Police`	`in Denver`	`arrested`	`BLM protesters"`
date	source	location	action	target

Example

Example of event coding:

- Event statement ✓
- Event description ✓
- Event coding ✓

"Yesterday,	Police	in Denver	arrested	BLM protesters"
date	source	location	action	target
01062021	101	08031	215	405

MANUAL AND COMPUTERIZED EVENT CODING

Types of Event Coding

- Manual Coding
- Computerized Coding

Manual Event Coding

Manual coding:

- Develop codebook
- Train coders
- Gather information (news)
- Annotate
- Validate

Computerized Event Coding

Computerized coding:

- Develop codebook (dictionaries or GSR)
- Gather information (scraping)
- Annotate (coder or ML model)
- Validate

Pros and Cons

Criteria	Manual coding	Computerized coding
Coding project		
Volume of documents	Small	Large
Coding period	Once	Repeated or continuous
Recoding possibility	Limited	Easy
Updating possibility	Limited	Easy
Dictionary modification	Not recommended	Easy
Content of interest		
Coding unit	Entire document	Sentence or paragraph
Syntax characteristics	Complex	Simple
Content of interest	Metaphoric or idiomatic	Literal
Bias concerns		
Sources of coder bias	Multiple	Unique
Inter-coder reliability	Problematic	Not an issue
Coder fatigue	Difficult	Not an issue
Feasibility of the coding project		
Coding time	Slow	Fast
Labor	High	Low

Brief History of Event Data

1970s	- World Event/Interaction Survey (WEIS)
1980s	- Protocol for the Assessment of Nonviolent Action (PANDA) - Conflict and Peace Data Bank (COPDAB) - NSF's Data Development in International Relations (DDIR)
1990s	- Kansas Event Data System (KEDS) - Global Event Data Systems (GEDS) - Protocol for the Assessment of Nonviolent Direct Action (PANDA) - Integrated Data for Events Analysis (IDEA) - Virtual Research Associates - VRA Reader and later on VRA Reporter
2000s	- Textual Analysis By Augmented Replacement Instructions (TABARI) - Conflict and Mediation Event Observations (CAMEO) Ontology - VRA Prospects - Integrated Crisis Early Warning System (ICEWS)
2010s	- Python Engine for Text Resolution And Related Coding Hierarchy (PETRARCH), later on PETRARCH-2, and UniversalPETRARCH - Phoenix Event Data - Multi-lingual: Eventus ID (Spanish) and Hadath (Arabic) - ACCENT event data using JABARI - TERRIER event data
2020s	- Machine Learning models

TYPES OF COMPUTERIZED EVENT CODING

Types of Computerized Event Coding

Types of Computerized Event Coding:

- Rule-based coding:
  - Shallow parsing
  - Deep parsing
- Example-based coding:
  - Machine Learning

Rule-based event coding

Natural Language Processing:

- Information extraction task
- Rely on dictionaries (actors, actions, locations) and rules
- Recognize patterns in the text

Shallow parsing:

- Dictionaries and rules provide search criteria
- Find exact match in the text while ignoring everything else
- Works well for narrow tasks in relatively structured text

Deep parsing:

- Still uses dictionaries and rules as search criteria
- Uses syntactical elements and structures of the text
- Parts of Speech tagging (POS), Named Entity Recognition (NER), Universal Dependencies (UD)
- Better suited for broad coding tasks with unstructured data

Deep parsing example

"Yesterday, Police in Denver arrested BLM protesters"

Deep parsing example

```
"Yesterday, Police in Denver arrested BLM protesters"
```

`Yesterday_RB ,_,`	← Adverb
`([ Police_NNP ])`	← Proper noun plural
`in_IN`	← Preposition
`([ Denver_NNP ])`	← Proper noun plural
`<:  arrested_VBD :>`	← Verb past tense
`([ BLM_NNP ])`	← Proper noun plural
`([ protesters_NNS ])`	← Noun plural

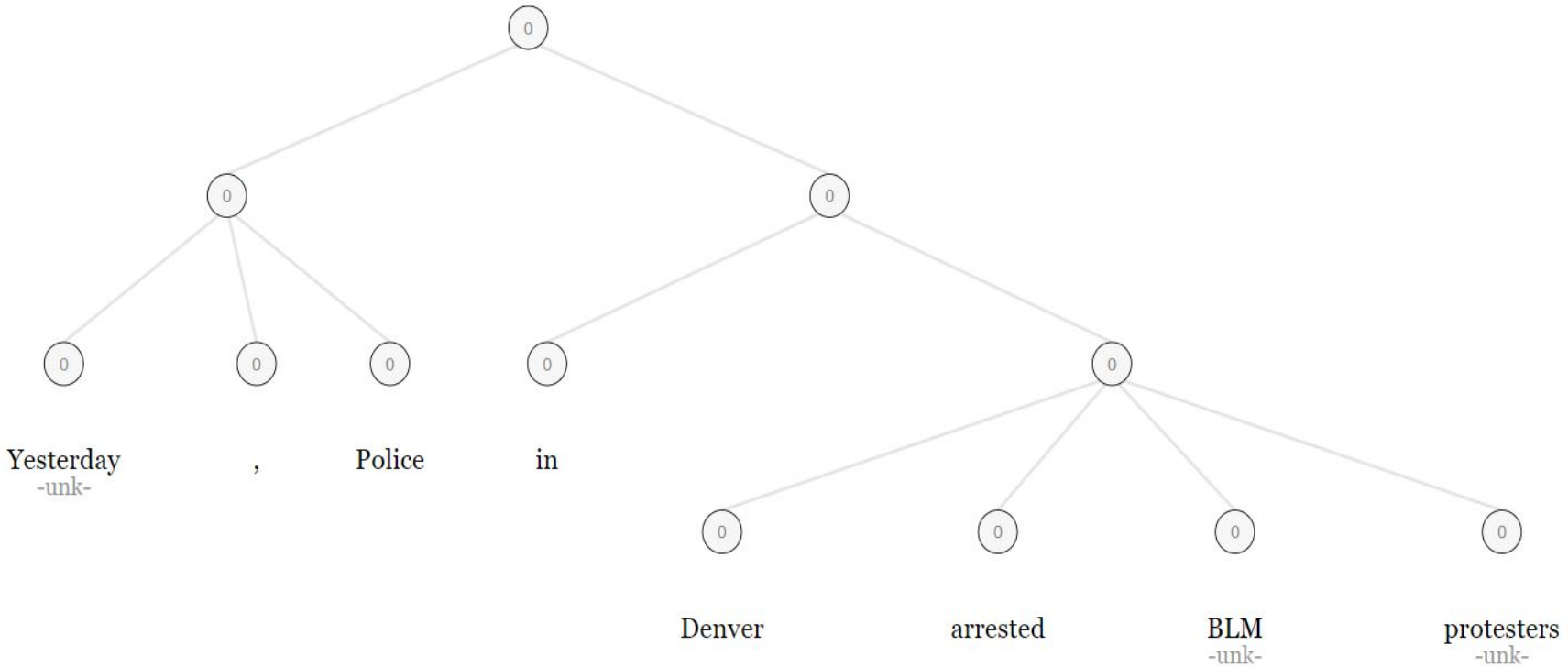
Deep parsing example

"Yesterday, Police in Denver arrested BLM protesters"

Yesterday_RB ,_,	← Adverb
([Police_NNP])	← Proper noun plural
in_IN	← Preposition
([Denver_NNP])	← Proper noun plural
<: arrested_VBD :>	← Verb past tense
([BLM_NNP])	← Proper noun plural
([protesters_NNS])	← Noun plural

Deep parsing example

Deep parsing example



Deep parsing example

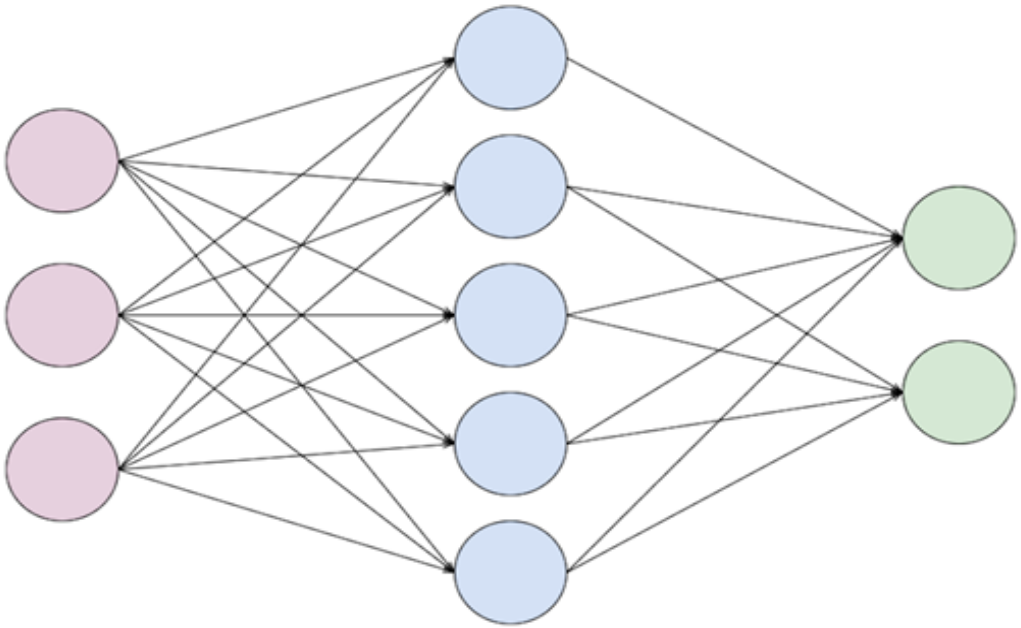
Challenges in deep parsing:

- Universal dependencies do not work well for long documents
- Anaphora is an enormous problem
- Toponyms: identifying locations is really hard
- Multi-lingual limitations

Example-based event coding

Machine Learning models:

- Document level analysis
- Identify the relationships between features
- Output classification



EXAMPLE-BASED EVENT CODING PROCESS

Example-Based Event Coding Process

Simplified process:

- Information gathering
- Annotation
- Machine Learning models

Information gathering

Information gathering is NOT a trivial task:

- Need to process massive amounts of information at a global scale, from multiple sources, in real-time
  - Manual gathering
  - Supervised gathering
  - Automated gathering
- Beware of coverage bias
- Duplicates
- Foreign languages
- Fake news

Information gathering

Supervised extraction:

- Develop a codebook and train coders
- Human coders select relevant information
- Is slow, tedious, and costly
- Automated downloading

Automatic extraction:

- Plenty of resources: API, Selenium, Scrapy, Rvest
- Quick and dirty automated info gathering
- Probably too dirty
  - Post-gathering classification (find the needle in a haystack)
  - Considerable human effort
  - Duplicates, fake news
  - Validity of information gathered (garbage in, garbage out)

Information gathering

Document classification using Machine Learning:

- Apply after large automatic data gathering
- Treat this as a classification task for ML:
  - Classify document as relevant or not relevant (0/1)
  - Develop a codebook
  - Train human coders
  - Need a large amount of annotated documents
  - Complex and nuanced classification criteria are difficult
  - Inter-coder reliability
- Imbalanced data could be a considerable problem

Applications:

Tagtog, Prodigy, Brat, ezTag, WebAnno

Event Annotation

Classify event elements:

- Someone → Source
- Doing something → Action
- To someone else → Target
- At a given moment → Date/time
- In a certain place → Location

Event Annotation

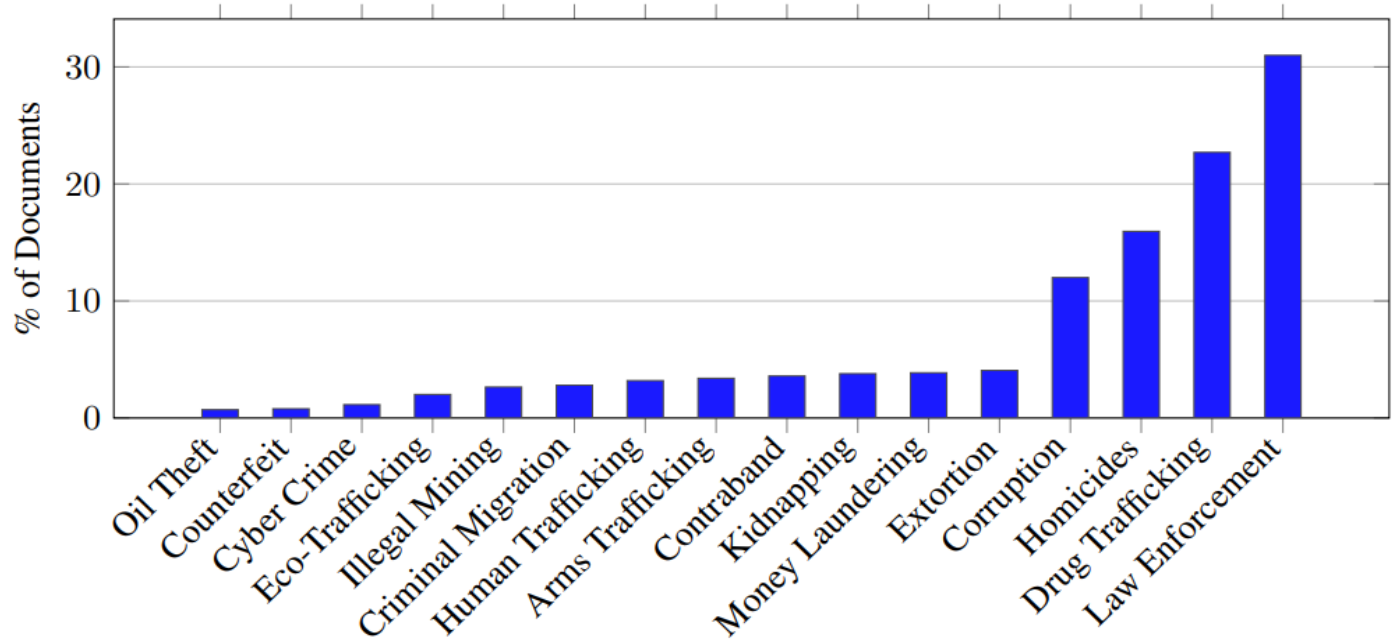
Classify event elements:

- Someone → Source
- Doing something → Action
- To someone else → Target
- At a given moment → Date/time
- In a certain place → Location

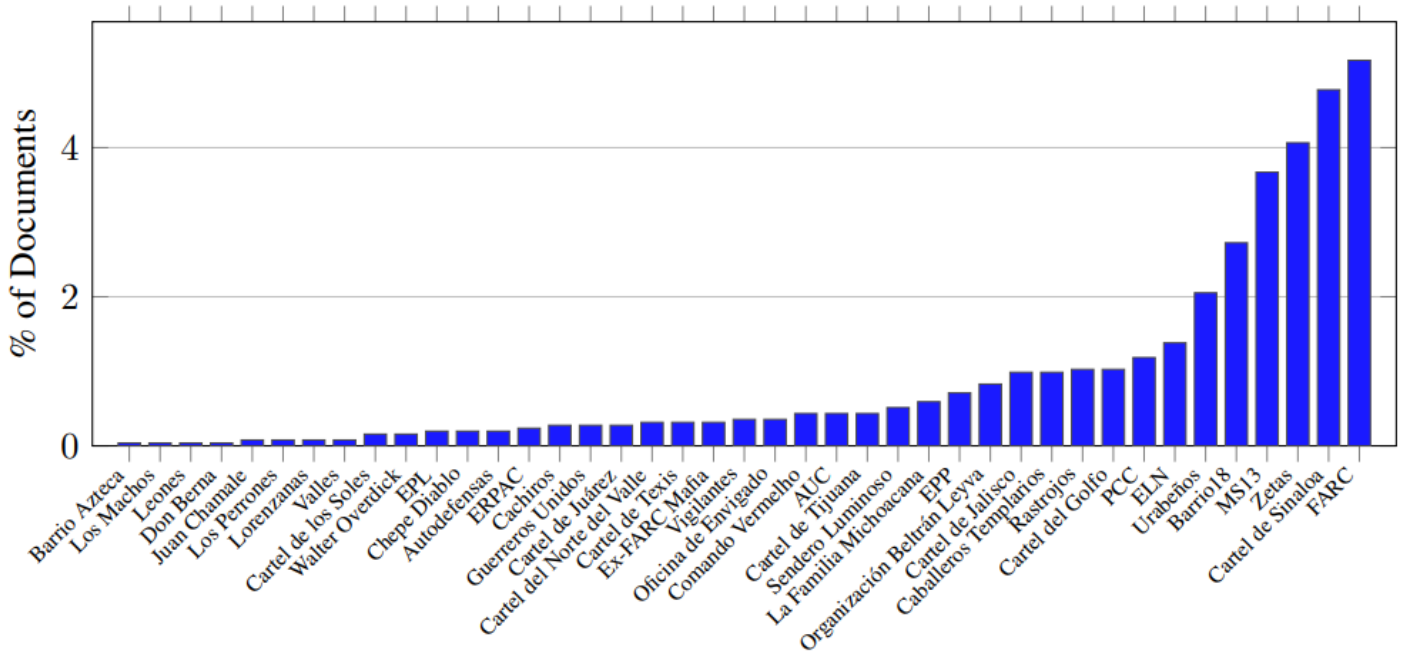
Challenges:

- Multi-class and multi-label annotations
- Imbalanced annotations
- Requires large amounts of data
- Inter-coder reliability (?!)

Distribution of Crime Category Annotations.



Distribution of Criminal Entities Annotations.



Machine Learning models

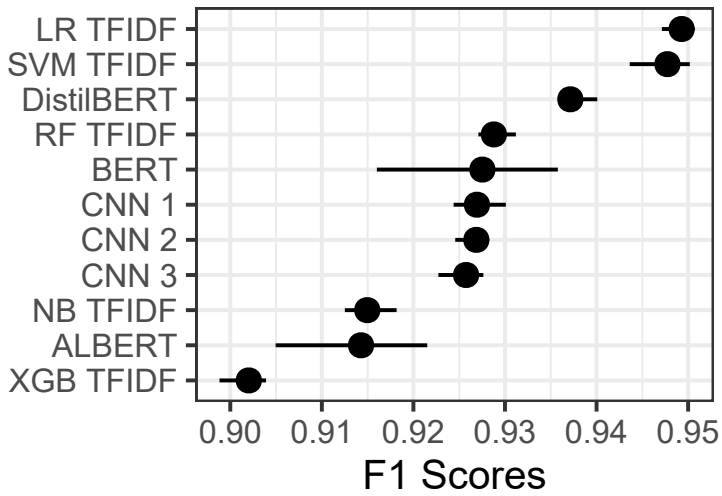
Machine Learning:

- Data-driven approach:
  - Underlying relationships in the text are unknown
  - Need to remain agnostic about ML model selection
  - Run different models
  - Pick the one with the best performance

Machine Learning models

Machine Learning:

- Data-driven approach:
  - Underlying relationships in the text are unknown
  - Need to remain agnostic about ML model selection
  - Run different models
  - Pick the one with the best performance



Machine Learning models

Machine Learning challenges:

- Are Gold Standard Records made of gold, copper, or tin?
  - Largely under-looked in CS
  - CSS should not take GSRs at face value
- More data is always better, but it is labor intensive
- Simple tasks are easier to classify & need less data
  - Clear and distinct concepts
  - That may be hard in CSS
- Few labels are easier to classify & need less data
  - Multi-class multi-label tasks are hard
- Balanced data is easier to classify

Machine Learning models

Machine Learning as a black box:

- ML models work well for classifying outcomes
- But are difficult to understand
- Classification at document level (in most models)
- Difficult to identify the specific event elements
- Some approaches may help:
    - Shapley values
    - HAN models
- But are limited for complex docs or classification tasks

Machine Learning models

Machine Learning for event coding:

- Powerful tools with great potential
- But ML is not a silver bullet
- Always validate the quality of the input and output
- Event coding with ML from scratch is not labor/cost free
- ML is good for classifying documents, but need additional steps to extract events

Thanks :)

Javier Osorio

School of Government and Public Policy

University of Arizona