Narrative Schemas

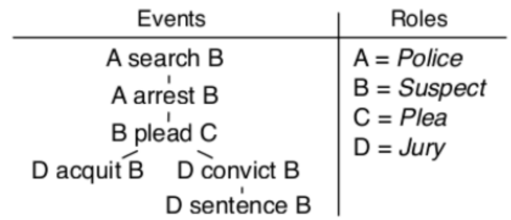
- Narrative event chains consist of events that share a single participant, the protagonist.
- A **Narrative Schema** [Chambers & Jurafsky, 2009] is a set of narrative chains that share arguments, so a schema can include *multiple* participants.
- The schema learning process also can induce types for the arguments (role fillers) of events.
- This work focuses on identifying related events and inducing their arguments. The task of temporally ordering the events is left for future work.

Sets of Events and Participants

The goal is to learn:

(1) sets of events that correspond to common, co-occurring, and (ideally) partially ordered sets of events.

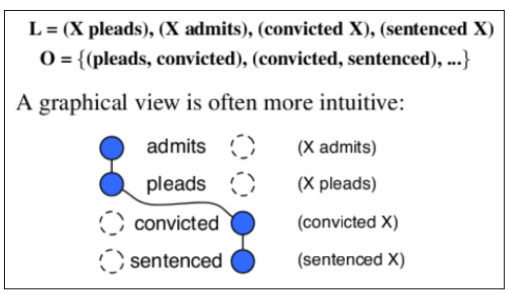
(2) the types of role fillers (arguments) common to the events.



Narrative Chain Terminology

A *narrative event chain* is represented by a tuple (L, O), where:

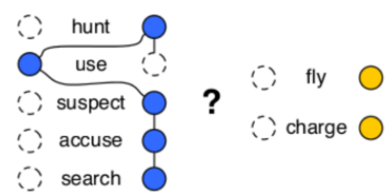
L is a set of event slots: (verb, dependency) pairs
O is a partial (temporal) ordering



Motivation for Argument Types

Representing argument types creates richer schemas. But argument types can also help to identify related events more accurately.

For example:



Without argument types, *(fly X)* rated higher than *(charge X)* because it occurred with all 5 event types in the training data. But few argument types of (*fly* X) are shared with the events!

(*charge* X) shares many argument types with (*accuse* X), (*search* X), and *suspect* X), e.g., *"criminal"* and *"suspect"*.

Typed Narrative Chains

A Typed Narrative Chain is a partially ordered set of event slots that share an argument (role) which is a member of a set of types R.

Argument types can be lexical units (e.g., head words), noun clusters, or another semantic representation.

Formally, a typed narrative chain is a tuple (L, P, O) where:

- L is a set of event slots: (verb, dependency) pairs
- O is a partial (temporal) ordering
- P is a set of argument types representing a single role

For example:

- L = {(*hunt* X), (X *use*), (*suspect* X), (*accuse* X), (*search* X)
- O = {(*use, hunt*), (*suspect, search*), (*suspect, accuse*), …}
- P = {person, government, company, criminal, …}

Learning Argument Types

The training corpus is parsed, a coreference resolver is applied, and events that share participants are extracted.

For each pair of event slots, coreferring arguments are identified, and the most frequent head word in the coreference chain is used as the role filler.

For example: (the underlined words are coreferent)

But for a growing proportion of U.S. workers, the troubles really set in when they apply for unemployment benefits. Many workers find their benefits challenged.

→ <(X *apply*), (X *find*), workers> *is learned.*

Example of Learned Typed Events

The following event chain (but the ordering information is omitted) was induced for a crime scenario.

The four top arguments are shown:

L = {(X *arrest*), (X *charge*), (X *raid*), (X *seize*), (X *confiscate*), (X *detain*), (X *deport*)

P = {police, agent, authority, government}

Event Slot Similarity with Arguments

The similarity of a new event slot <f, g> with an untyped narrative chain C of size N is the sum of scores between all pairs:

$$\textit{chainsim}(C, \langle f, g\rangle)) = \sum_{i=1}^{N} \mathrm{sim}(\langle e_i, d_i\rangle, \langle f, g\rangle)$$

The similarity of a new event slot <f, g> with a typed narrative chain C of size N is based on the argument that maximizes the chain's score:

$$\textit{chainsim-new}(C, \langle f, g\rangle)) = \max_{a}(\mathrm{score}(C, a) + \sum_{i=0}^{N} \mathrm{sim}(\langle e_i, d_i\rangle, \langle f, g\rangle, a)$$

Argument-based Similarity

The similarity function is adapted to be specific to an argument *a*:

$$\mathrm{sim}(\langle e, d\rangle, \langle e', d'\rangle, a) = \mathrm{PMI}(\langle e,d\rangle,\langle e',d'\rangle) + \lambda \log \mathrm{freq}(\langle e,d\rangle,\langle e',d'\rangle,a)$$

where:
λ is a weighting parameter
freq(b,b',*a*) is the frequency of *a* occurring as arguments to b and b'

The score for a chain C of size N for argument *a* is:

$$\mathrm{score}(C, a) = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \mathrm{sim}(\langle e_i,d_i\rangle, \langle e_j, d_j\rangle, a)$$

Event Relatedness with Schemas

For narrative chains, an event is added if its slot (argument) is consistent with other events in the chain.

For narrative schemas, an event is added only if both its Subject and Object are consistent with arguments in existing chains.

The similarity function for an event (v) with respect to a narrative scheme N is computed as:

$$\mathit{narsim}(N,v) = \sum_{d \,\varepsilon\, D_v} \max(\beta, \max_{c \,\varepsilon\, C_N} \mathit{chainsim\text{-}new}(c, \langle v, d\rangle))$$

where:

C_N is the set of chains in the narrative scheme N
β is a base score, in case no existing chain is a good match

Adding Events to Schemas

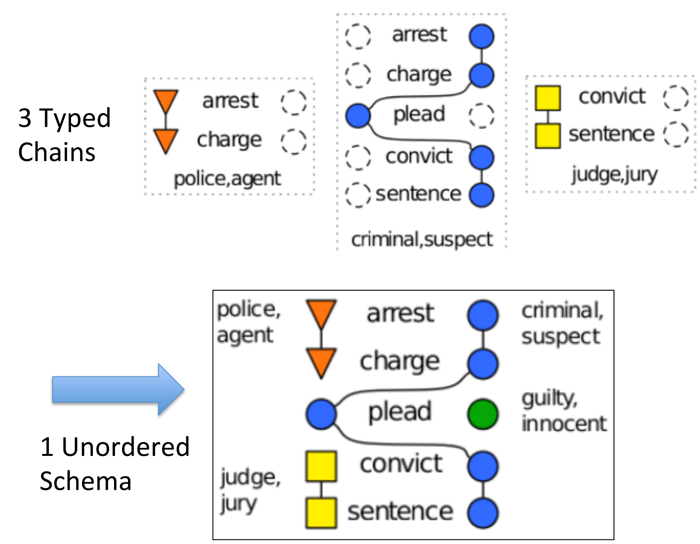
For narrative chains, the best event is identified by comparing individual event slots with the existing chain, where *m* is the number of event slots in the corpus:

$$\max_{0 < j < m} \mathit{chainsim}(c, \langle v_j, g_j\rangle)$$

For narrative schemas, the best event is identified by comparing all slots for a verb using the narrative similarity function, where |v| is the number of verbs in the corpus:

$$\max_{0 < j < |v|} \mathit{narsim}(N, v_j)$$

Merging Typed Chains



Evaluation Details

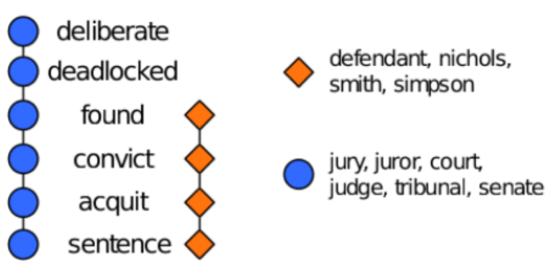
- Training Corpus: NYT portion of the Gigaword Corpus. A dependency parser and coreference resolver were applied.
- The clustering procedure was stopped ("and sometimes continued" ?) at 6 events per schema, because the mean number of verbs in FrameNet frames is 5-6.
- A low β value was chosen to limit chain splitting.
- A new schema was built starting with each verb that occurred in > 3,000 but < 50,000 documents (1,800 verbs in total).
- The top 20 highest scoring schemas were analyzed. These generally concerned business, politics, crime, or food.

Learned Schema Example

The following schema was automatically learned starting with the verb "arrest":

Another Learned Schema Example

The following schema was automatically learned starting with the verb "convict":



Three of Top Six Scored Schemas

A produce B *A sell B* A manufacture B A *market B *A distribute B* A -develop B	A ∈ {company, inc, corp, microsoft, iraq, co, unit, maker, ...} B ∈ {drug, product, system, *test*, software, funds, movie, ...}
A boil B A *slice* B A -peel B A saute B A cook B A *chop* B	A ∈ {*wash, heat, thinly, onion, note*} B ∈ {potato, onion, mushroom, clove, orange, gnocchi }
A *uphold B A *challenge B A rule B A *enforce* B A *overturn B A *strike_down B	A ∈ {court, judge, justice, panel, osteen, circuit, nicolau, sporkin, majority, ...} B ∈ {law, ban, rule, constitutionality, conviction, ruling, *lawmaker*, tax, ...}

Next Three of Top Six Scored Schemas

B trade C	A ∈ {}
B fell C	B ∈ {dollar, share, index, mark, currency,
A *quote B	stock, yield, price, pound, ...}
B fall C	C ∈ {friday, _most_, year, _percent_, thursday
B -slip C	monday, _share_, week, _dollar_, ...}
B rise C	
A _detain_ B	A ∈ {police, agent, officer, authorities,
A _confiscate_ B	troops, official, investigator, ... }
A _seize_ B	
A _raid_ B	B ∈ {suspect, government, journalist,
A _search_ B	_monday_, member, citizen, client, ... }
A _arrest_ B	
A _own_ B	A ∈ {company, investor, trader, _corp_,
A *borrow B	enron, _inc_, government, bank, itt, ...}
A sell B	
A buy_back B	B ∈ {share, stock, stocks, bond, company,
A buy B	security, team, funds, house, ... }
A *repurchase B	

Qualitative Evaluation

- Schemas and frames both represent groups of events (verbs), but for different purposes:
  - schemas group verbs that co-occur in narrative discourse structures
  - frames group verbs that share core participants (semantic roles)
- Nevertheless, FrameNet's frame definitions are a source of lexical information that can be used to qualitatively assess some aspects of the learned schemas.
- Three evaluations are performed: verb groupings, linking struture, and argument roles.
- The verb groupings are based on mapping to the most consistent frame. But only 13 map, many verbs were not present in FrameNet, and the evaluation was highly subjective. It's really an apples and oranges comparison anyway.

Linking Structure Evaluation

- The **linking structure evaluation** assessed the quality of the linked grammatical relations in a schema.
- For each chain in 13 schemas, they identified the FrameNet frame element that could correctly fill the most verb arguments in the chain. The remaining arguments were considered incorrect.
- These chains had 78 verbs, containing 156 arguments (*subject* and *object* for each verb).
- 151 arguments were correctly clustered
  → 96.8% accuracy.

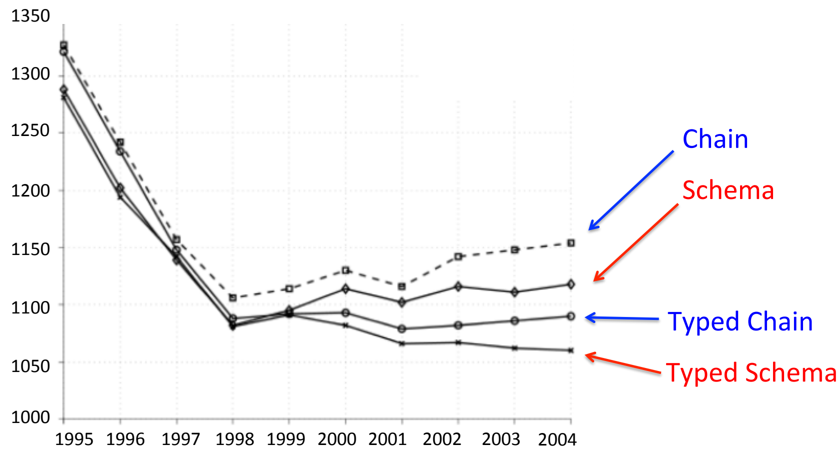
Argument Role Evaluation

- The argument role evaluation assesses the quality of the entities that fill the argument slots.
- First, they manually identify the best frame element for each argument. For example:

  "A produces B" → *Manufacturing* frame

  B → *Product* frame element
- The top 10 arguments are evaluated. For example:
  - "drug" is a correct Product argument, "test" is incorrect
- 289 of 400 possible arguments were judged correct
  → 72% precision

Narrative Cloze Evaluation Results



Training Data Size, based on years 1994 – X

Conclusions

- This work represents a nice first step toward learning sets of related events and their shared participants.
- However, much research remains until we can learn coherent, accurate, and useful event knowledge structures on a large scale.
  - event representation is still very weak (just verbs)
  - arguments are limited to subjects and objects
  - weak semantic representation of the argument types
  - temporal ordering of the events is largely absent
  - assessment of whether the learned events are important
  - evaluating knowledge structures is a challenge in its own right!