Unsupervised Learning of Narrative Schemas and their Participants

Nathanael Chambers and **Dan Jurafsky**
Stanford University, Stanford, CA 94305
{natec,jurafsky}@stanford.edu

Abstract

We describe an unsupervised system for learning *narrative schemas*, coherent sequences or sets of events (*arrested*(POLICE,SUSPECT), *convicted*(JUDGE, SUSPECT)) whose arguments are filled with participant semantic roles defined over words (JUDGE = {judge, jury, court}, POLICE = {police, agent, authorities}). Unlike most previous work in event structure or semantic role learning, our system does not use supervised techniques, hand-built knowledge, or predefined classes of events or roles. Our unsupervised learning algorithm uses coreferring arguments in chains of verbs to learn both rich narrative event structure and argument roles. By jointly addressing both tasks, we improve on previous results in narrative/frame learning and induce rich frame-specific semantic roles.

1 Introduction

This paper describes a new approach to event semantics that jointly learns event relations and their participants from unlabeled corpora.

The early years of natural language processing (NLP) took a "top-down" approach to language understanding, using representations like *scripts* (Schank and Abelson, 1977) (structured representations of events, their causal relationships, and their participants) and frames to drive interpretation of syntax and word use. Knowledge structures such as these provided the interpreter rich information about many aspects of meaning.

The problem with these rich knowledge structures is that the need for hand construction, specificity, and domain dependence prevents robust and flexible language understanding. Instead, modern work on understanding has focused on shallower representations like *semantic roles*, which express at least one aspect of the semantics of events and have proved amenable to supervised learning from corpora like PropBank (Palmer et al., 2005) and Framenet (Baker et al., 1998). Unfortunately, creating these supervised corpora is an expensive and difficult multi-year effort, requiring complex decisions about the exact set of roles to be learned. Even unsupervised attempts to learn semantic roles have required a pre-defined set of roles (Grenager and Manning, 2006) and often a hand-labeled seed corpus (Swier and Stevenson, 2004; He and Gildea, 2006).

In this paper, we describe our attempts to learn script-like information about the world, including both event structures and the roles of their participants, but without pre-defined frames, roles, or tagged corpora.

Consider the following *Narrative Schema*, to be defined more formally later. The events on the left follow a set of participants through a series of connected events that constitute a narrative:

Events	Roles
A search B A arrest B B plead C D acquit B D convict B D sentence B	A = *Police* B = *Suspect* C = *Plea* D = *Jury*

Being able to robustly learn sets of related events (left) and frame-specific role information about the argument types that fill them (right) could assist a variety of NLP applications, from question answering to machine translation.

Our previous work (Chambers and Jurafsky, 2008) relied on the intuition that in a coherent text, any two events that are about the same participants are likely to be part of the same story or narrative. The model learned simple aspects of narrative structure ('narrative chains') by extracting events that share a single participant, the *protagonist*. In this paper we extend this work to represent sets of situation-specific events not unlike scripts, caseframes (Bean and Riloff, 2004), and FrameNet frames (Baker et al., 1998). This paper shows that verbs in distinct narrative chains can be merged into an improved single narrative schema, while the shared arguments across verbs can provide rich information for inducing semantic roles.

2 Background

This paper addresses two areas of work in event semantics, narrative event chains and semantic role labeling. We begin by highlighting areas in both that can mutually inform each other through a narrative schema model.

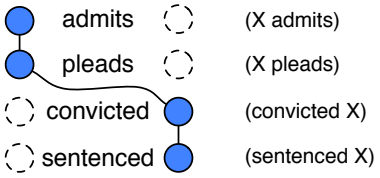
2.1 Narrative Event Chains

Narrative Event Chains are partially ordered sets of events that all involve the same shared participant, the *protagonist* (Chambers and Jurafsky, 2008). A chain contains a set of verbs representing events, and for each verb, the grammatical role filled by the shared protagonist.

An event is a verb together with its constellation of arguments. An *event slot* is a tuple of an event and a particular argument slot (grammatical relation), represented as a pair $\langle v, d \rangle$ where v is a verb and $d \in \{subject, object, prep\}$. A chain is a tuple (L, O) where L is a set of event slots and O is a partial (temporal) ordering. We will write event slots in shorthand as (X pleads) or (pleads X) for $\langle pleads, subject \rangle$ and $\langle pleads, object \rangle$. Below is an example chain modeling criminal prosecution.

L = (X pleads), (X admits), (convicted X), (sentenced X)
O = {(pleads, convicted), (convicted, sentenced), ...}

A graphical view is often more intuitive:



In this example, the protagonist of the chain is the person being prosecuted and the other unspecified event slots remain unfilled and unconstrained. Chains in the Chambers and Jurafsky (2008) model are ordered; in this paper rather than address the ordering task we focus on event and argument induction, leaving ordering as future work.

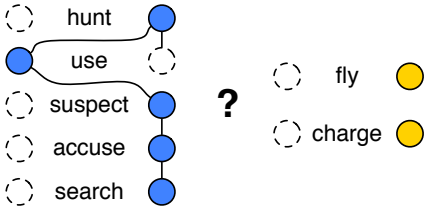
The Chambers and Jurafsky (2008) model learns chains completely unsupervised, (albeit after parsing and resolving coreference in the text) by counting pairs of verbs that share coreferring arguments within documents and computing the pointwise mutual information (PMI) between these verb-argument pairs. The algorithm creates chains by clustering event slots using their PMI scores, and we showed this use of co-referring arguments improves event relatedness.

Our previous work, however, has two major limitations. First, the model did not express any information about the protagonist, such as its *type* or *role*. Role information (such as knowing whether a filler is a location, a person, a particular class of people, or even an inanimate object) could crucially inform learning and inference. Second, the model only represents one participant (the protagonist). Representing the other entities involved in all event slots in the narrative could potentially provide valuable information. We discuss both of these extensions next.

2.1.1 The Case for Arguments

The Chambers and Jurafsky (2008) narrative chains do not specify what type of argument fills the role of protagonist. Chain learning and clustering is based only on the frequency with which two verbs share arguments, ignoring any features of the arguments themselves.

Take this example of an actual chain from an article in our training data. Given this chain of five events, we want to choose other events most likely to occur in this scenario.



One of the top scoring event slots is (fly X). Narrative chains incorrectly favor (fly X) because it is observed during training with all five event slots, although not frequently with any one of them. An event slot like (charge X) is much more plausible, but is unfortunately scored lower by the model.

Representing the *types* of the arguments can help solve this problem. Few types of arguments are shared between the chain and (fly X). However, (charge X) shares many arguments with (accuse X), (search X) and (suspect X) (e.g., *criminal* and *suspect*). Even more telling is that these arguments are jointly shared (the same or coreferent) across all three events. Chains represent coherent scenarios, not just a set of independent pairs, so we want to model *argument overlap across all pairs*.

2.1.2 The Case for Joint Chains

The second problem with narrative chains is that they make judgments only between protagonist arguments, one slot per event. All entities and slots

in the space of events should be jointly considered when making event relatedness decisions.

As an illustration, consider the verb *arrest*. Which verb is more related, *convict* or *capture*? A narrative chain might only look at the objects of these verbs and choose the one with the highest score, usually choosing *convict*. But in this case the subjects offer additional information; the subject of *arrest* (police) is different from that of *convict* (judge). A more informed decision prefers *capture* because both the objects (suspect) and subjects (police) are identical. This joint reasoning is absent from the narrative chain model.

2.2 Semantic Role Labeling

The task of semantic role learning and labeling is to identify classes of entities that fill predicate slots; semantic roles seem like they'd be a good model for the kind of argument types we'd like to learn for narratives. Most work on semantic role labeling, however, is supervised, using Propbank (Palmer et al., 2005), FrameNet (Baker et al., 1998) or VerbNet (Kipper et al., 2000) as gold standard roles and training data. More recent learning work has applied bootstrapping approaches (Swier and Stevenson, 2004; He and Gildea, 2006), but these still rely on a hand labeled seed corpus as well as a pre-defined set of roles. Grenegar and Manning (2006) use the EM algorithm to learn PropBank roles from unlabeled data, and unlike bootstrapping, they don't need a labeled corpus from which to start. However, they do require a predefined set of roles (arg0, arg1, etc.) to define the domain of their probabilistic model.

Green and Dorr (2005) use WordNet's graph structure to cluster its verbs into FrameNet frames, using glosses to name potential slots. We differ in that we attempt to learn frame-like narrative structure from untagged newspaper text. Most similar to us, Alishahi and Stevenson (2007) learn verb specific *semantic profiles* of arguments using WordNet classes to define the roles. We learn situation-specific classes of roles shared by multiple verbs.

Thus, two open goals in role learning include (1) unsupervised learning and (2) learning the roles themselves rather than relying on pre-defined role classes. As just described, Chambers and Jurafsky (2008) offers an unsupervised approach to event learning (goal 1), but lacks semantic role knowledge (goal 2). The following sections describe a model that addresses both goals.

3 Narrative Schemas

The next sections introduce *typed narrative chains* and *chain merging*, extensions that allow us to jointly learn argument roles with event structure.

3.1 Typed Narrative Chains

The first step in describing a narrative schema is to extend the definition of a narrative chain to include argument types. We now constrain the protagonist to be of a certain type or role. A *Typed Narrative Chain* is a partially ordered set of event slots that share an argument, but now the shared argument is a role defined by being a member of a set of types R. These types can be lexical units (such as observed head words), noun clusters, or other semantic representations. We use head words in the examples below, but we also evaluate with argument clustering by mapping head words to member clusters created with the CBC clustering algorithm (Pantel and Lin, 2002).

We define a typed narrative chain as a tuple (L, P, O) with L and O the set of event slots and partial ordering as before. Let P be a set of argument types (head words) representing a single role. An example is given here:

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

3.2 Learning Argument Types

As mentioned above, narrative chains are learned by parsing the text, resolving coreference, and extracting chains of events that share participants. In our new model, argument types are learned simultaneously with narrative chains by finding salient words that represent coreferential arguments. We record counts of arguments that are observed with each pair of event slots, build the referential set for each word from its coreference chain, and then represent each observed argument by the most frequent head word in its referential set (ignoring pronouns and mapping entity mentions with person pronouns to a constant PERSON identifier).

As an example, the following contains four *worker* mentions:

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.

$L = \{$X arrest, X charge, X raid, X seize, X confiscate, X detain, X deport $\}$
$P = \{police, agent, authority, government\}$

Figure 1: A typed narrative chain. The four top arguments are given. The ordering O is not shown.

The four bolded terms are coreferential and (hopefully) identified by coreference. Our algorithm chooses the head word of each phrase and ignores the pronouns. It then chooses the most frequent head word as the most salient mention. In this example, the most salient term is *workers*. If any pair of event slots share arguments from this set, we count *workers*. In this example, the pair (X find) and (X apply) shares an argument (they and workers). The pair ((X find),(X apply)) is counted once for narrative chain induction, and ((X find), (X apply), workers) once for argument induction.

Figure 1 shows the top occurring words across all event slot pairs in a criminal scenario chain. This chain will be part of a larger narrative schema, described in section 3.4.

3.3 Event Slot Similarity with Arguments

We now formalize event slot similarity with arguments. Narrative chains as defined in (Chambers and Jurafsky, 2008) score a new event slot $\langle f, g\rangle$ against a chain of size n by summing over the scores between all pairs:

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

where C is a narrative chain, f is a verb with grammatical argument g, and $sim(e, e')$ is the pointwise mutual information $pmi(e, e')$. Growing a chain by one adds the highest scoring event.

We extend this function to include argument types by defining similarity in the context of a specific argument a:

$$sim(\langle e, d\rangle, \langle e', d'\rangle, a) = pmi(\langle e, d\rangle, \langle e', d'\rangle) + \lambda \log freq(\langle e, d\rangle, \langle e', d'\rangle, a) \quad (2)$$

where λ is a constant weighting factor and $freq(b, b', a)$ is the corpus count of a filling the arguments of events b and b'. We then score the entire chain for a particular argument:

$$score(C, a) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} sim(\langle e_i, d_i\rangle, \langle e_j, d_j\rangle, a) \quad (3)$$

Using this chain score, we finally extend $chainsim$ to score a new event slot based on the argument that maximizes the entire chain's score:

$$chainsim'(C, \langle f, g\rangle) = \max_a (score(C, a) + \sum_{i=1}^{n} sim(\langle e_i, d_i\rangle, \langle f, g\rangle, a)) \quad (4)$$

The argument is now directly influencing event slot similarity scores. We will use this definition in the next section to build Narrative Schemas.

3.4 Narrative Schema: Multiple Chains

Whereas a narrative chain is a set of event slots, a **Narrative Schema** is a set of typed narrative chains. A schema thus models *all actors* in a set of events. If *(push X)* is in one chain, *(Y push)* is in another. This allows us to model a document's entire narrative, not just one main actor.

3.4.1 The Model

A narrative schema is defined as a 2-tuple $N = (E, C)$ with E a set of events (here defined as verbs) and C a set of typed chains over the event slots. We represent an *event* as a verb v and its grammatical argument positions $D_v \subseteq \{subject, object, prep\}$. Thus, each event slot $\langle v, d\rangle$ for all $d \in D_v$ belongs to a chain $c \in C$ in the schema. Further, each c must be unique for each slot of a single verb. Using the criminal prosecution domain as an example, a narrative schema in this domain is built as in figure 2.

The three dotted boxes are graphical representations of the typed chains that are combined in this schema. The first represents the event slots in which the criminal is involved, the second the police, and the third is a court or judge. Although our representation uses a set of chains, it is equivalent to represent a schema as a constraint satisfaction problem between $\langle e, d\rangle$ event slots. The next section describes how to learn these schemas.

3.4.2 Learning Narrative Schemas

Previous work on narrative chains focused on relatedness scores between pairs of verb arguments (event slots). The clustering step which built chains depended on these pairwise scores. Narrative schemas use a generalization of the entire verb with all of its arguments. A joint decision can be made such that a verb is added to a schema if *both* its subject and object are assigned to chains in the schema with high confidence.

For instance, it may be the case that *(Y pull_over)* scores well with the 'police' chain in

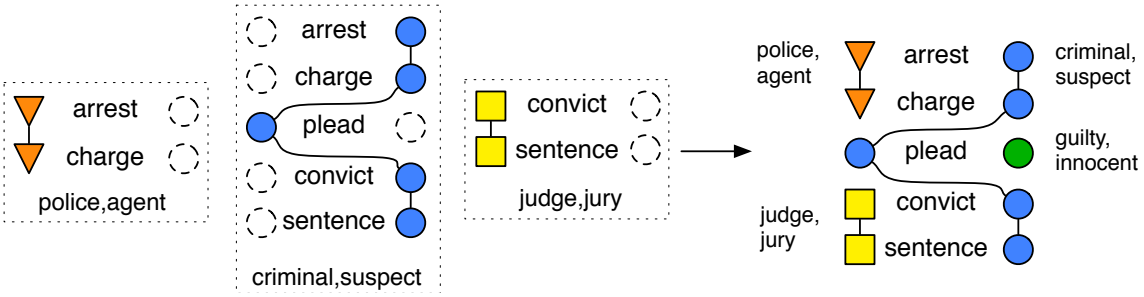


Figure 2: Merging typed chains into a single unordered Narrative Schema.

figure 3. However, the object of *(pull_over A)* is not present in any of the other chains. Police pull over *cars*, but this schema does not have a chain involving cars. In contrast, *(Y search)* scores well with the 'police' chain and *(search X)* scores well in the 'defendant' chain too. Thus, we want to favor *search* instead of *pull_over* because the schema is already modeling both arguments.

This intuition leads us to our event relatedness function for the entire narrative schema N, not just one chain. Instead of asking which event slot $\langle v, d \rangle$ is a best fit, we ask if v is best by considering all slots at once:

$$narsim(N, v) = \sum_{d \in D_v} \max(\beta, \max_{c \in C_N} chainsim'(c, \langle v, d \rangle)) \quad (5)$$

where C_N is the set of chains in our narrative N. If $\langle v, d \rangle$ does not have strong enough similarity with any chain, it creates a new one with base score β. The β parameter balances this decision of adding to an existing chain in N or creating a new one.

3.4.3 Building Schemas

We use equation 5 to build schemas from the set of *events* as opposed to the set of *event slots* that previous work on narrative chains used. In Chambers and Jurafsky (2008), narrative chains add the best $\langle e, d \rangle$ based on the following:

$$\max_{j:0<j<m} chainsim(c, \langle v_j, g_j \rangle) \quad (6)$$

where m is the number of seen event slots in the corpus and $\langle v_j, g_j \rangle$ is the jth such possible event slot. Schemas are now learned by adding events that maximize equation 5:

$$\max_{j:0<j<|v|} narsim(N, v_j) \quad (7)$$

where $|v|$ is the number of observed verbs and v_j is the jth such verb. Verbs are incrementally added to a narrative schema by strength of similarity.

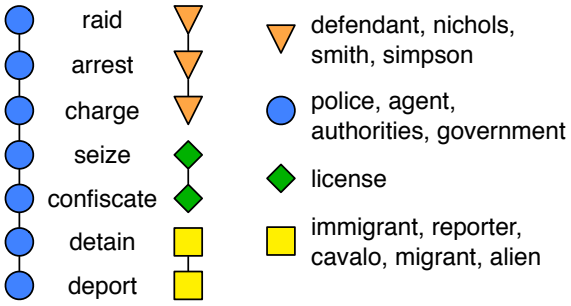


Figure 3: Graphical view of an unordered schema automatically built starting from the verb 'arrest'. A β value that encouraged splitting was used.

4 Sample Narrative Schemas

Figures 3 and 4 show two criminal schemas learned completely automatically from the NYT portion of the Gigaword Corpus (Graff, 2002). We parse the text into dependency graphs and resolve coreferences. The figures result from learning over the event slot counts. In addition, figure 5 shows six of the top 20 scoring narrative schemas learned by our system. We artificially required the clustering procedure to stop (and sometimes continue) at six events per schema. Six was chosen as the size to enable us to compare to FrameNet in the next section; the mean number of verbs in FrameNet frames is between five and six. A low β was chosen to limit chain splitting. We built a new schema starting from each verb that occurs in more than 3000 and less than 50,000 documents in the NYT section. This amounted to approximately 1800 verbs from which we show the top 20. Not surprisingly, most of the top schemas concern business, politics, crime, or food.

5 Frames and Roles

Most previous work on unsupervised semantic role labeling assumes that the set of possible

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, ...}
B trade C B fell C *A *quote B* B fall C B -slip C B rise C	A ∈ {} B ∈ {dollar, share, index, mark, currency, stock, yield, price, pound, ...} C ∈ {friday, *most*, year, *percent*, thursday monday, *share*, week, *dollar*, ...}
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 *detain* B A *confiscate* B A *seize* B A *raid* B A *search* B A *arrest* B	A ∈ {police, agent, officer, authorities, troops, official, investigator, ... } B ∈ {suspect, government, journalist, *monday*, member, citizen, client, ... }
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, ...}
A *own* B A *borrow B A sell B A buy_back B A buy B A *repurchase B	A ∈ {company, investor, trader, *corp*, enron, *inc*, government, bank, itt, ...} B ∈ {share, stock, stocks, bond, company, security, team, funds, house, ... }

Figure 5: Six of the top 20 scored Narrative Schemas. Events and arguments in italics were marked misaligned by FrameNet definitions. * indicates verbs not in FrameNet. - indicates verb senses not in FameNet.

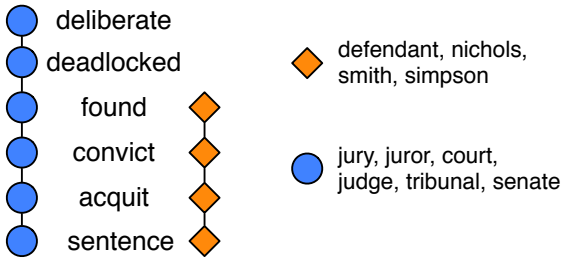


Figure 4: Graphical view of an unordered schema automatically built from the verb 'convict'. Each node shape is a chain in the schema.

classes is very small (i.e, PropBank roles ARG0 and ARG1) and is known in advance. By contrast, our approach induces sets of entities that appear in the argument positions of verbs in a narrative schema. Our model thus does not assume the set of roles is known in advance, and it learns the roles at the same time as clustering verbs into frame-like schemas. The resulting sets of entities (such as {*police, agent, authorities, government*} or {*court, judge, justice*}) can be viewed as a kind of schema-specific semantic role.

How can this unsupervised method of learning roles be evaluated? In Section 6 we evaluate the schemas together with their arguments in a cloze task. In this section we perform a more qualitative evalation by comparing our schema to FrameNet.

FrameNet (Baker et al., 1998) is a database of *frames*, structures that characterize particular situations. A frame consists of a set of events (the verbs and nouns that describe them) and a set of frame-specific semantic roles called *frame elements* that can be arguments of the lexical units in the frame. FrameNet frames share commonalities with narrative schemas; both represent aspects of situations in the world, and both link semantically related words into frame-like sets in which each predicate draws its argument roles from a frame-specific set. They differ in that schemas focus on events in a narrative, while frames focus on events that share core participants. Nonetheless, the fact that FrameNet defines frame-specific argument roles suggests that comparing our schemas and roles to FrameNet would be elucidating.

We took the 20 learned narrative schemas described in the previous section and used FrameNet to perform qualitative evaluations on three aspects of schema: verb groupings, linking structure (the mapping of each argument role to syntactic subject or object), and the roles themselves (the set of entities that constitutes the schema roles).

Verb groupings To compare a schema's event selection to a frame's lexical units, we first map the top 20 schemas to the FrameNet frames that have the largest overlap with each schema's six verbs. We were able to map 13 of our 20 narratives to FrameNet (for the remaining 7, no frame contained more than one of the six verbs). The remaining 13 schemas contained 6 verbs each for a total of 78 verbs. 26 of these verbs, however, did not occur in FrameNet, either at all, or with the correct sense. Of the remaining 52 verb mappings, 35 (67%) occurred in the closest FrameNet frame or in a frame one link away. 17 verbs (33%)

occurred in a different frame than the one chosen.

We examined the 33% of verbs that occurred in a different frame. Most occurred in related frames, but did not have FrameNet links between them. For instance, one schema includes the causal verb *trade* with unaccusative verbs of change like *rise* and *fall*. FrameNet separates these classes of verbs into distinct frames, distinguishing motion frames from caused-motion frames.

Even though *trade* and *rise* are in different FrameNet frames, they do in fact have the narrative relation that our system discovered. Of the 17 misaligned events, we judged all but one to be correct in a narrative sense. Thus although not exactly aligned with FrameNet's notion of event clusters, our induction algorithm seems to do very well.

Linking structure Next, we compare a schema's *linking structure*, the grammatical relation chosen for each verb event. We thus decide, e.g., if the object of the verb arrest *(arrest B)* plays the same role as the object of detain *(detain B)*, or if the subject of detain *(B detain)* would have been more appropriate.

We evaluated the clustering decisions of the 13 schemas (78 verbs) that mapped to frames. For each chain in a schema, we identified the frame element that could correctly fill the most verb arguments in the chain. The remaining arguments were considered incorrect. Because we assumed all verbs to be transitive, there were 156 arguments (subjects and objects) in the 13 schema. Of these 156 arguments, 151 were correctly clustered together, achieving 96.8% accuracy.

The schema in figure 5 with events *detain, seize, arrest*, etc. shows some of these errors. The object of all of these verbs is an animate theme, but *confiscate B* and *raid B* are incorrect; people cannot be confiscated/raided. They should have been split into their own chain within the schema.

Argument Roles Finally, we evaluate the learned sets of entities that fill the argument slots. As with the above linking evaluation, we first identify the best frame element for each argument. For example, the events in the top left schema of figure 5 map to the *Manufacturing* frame. Argument B was identified as the *Product* frame element. We then evaluate the top 10 arguments in the argument set, judging whether each is a reasonable filler of the role. In our example, *drug* and *product* are correct *Product* arguments. An incorrect argument is *test*, as it was judged that a test is not a product.

We evaluated all 20 schemas. The 13 mapped schemas used their assigned frames, and we created frame element definitions for the remaining 7 that were consistent with the syntactic positions. There were 400 possible arguments (20 schemas, 2 chains each), and 289 were judged correct for a precision of **72%**. This number includes Person and Organization names as correct fillers. A more conservative metric removing these classes results in 259 (65%) correct.

Most of the errors appear to be from parsing mistakes. Several resulted from confusing objects with adjuncts. Others misattached modifiers, such as including *most* as an argument. The cooking schema appears to have attached verbal arguments learned from instruction lists (wash, heat, boil). Two schemas require situations as arguments, but the dependency graphs chose as arguments the subjects of the embedded clauses, resulting in 20 incorrect arguments in these schema.

6 Evaluation: Cloze

The previous section compared our learned knowledge to current work in event and role semantics. We now provide a more formal evaluation against untyped narrative chains. The two main contributions of schema are (1) adding typed arguments and (2) considering joint chains in one model. We evaluate each using the narrative cloze test as in (Chambers and Jurafsky, 2008).

6.1 Narrative Cloze

The cloze task (Taylor, 1953) evaluates human understanding of lexical units by removing a random word from a sentence and asking the subject to guess what is missing. The narrative cloze is a variation on this idea that removes an event slot from a known narrative chain.Performance is measured by the position of the missing event slot in a system's ranked guess list.

This task is particularly attractive for narrative schemas (and chains) because it aligns with one of the original ideas behind Schankian scripts, namely that scripts help humans 'fill in the blanks' when language is underspecified.

6.2 Training and Test Data

We count verb pairs and shared arguments over the NYT portion of the Gigaword Corpus (years 1994-2004), approximately one million articles.

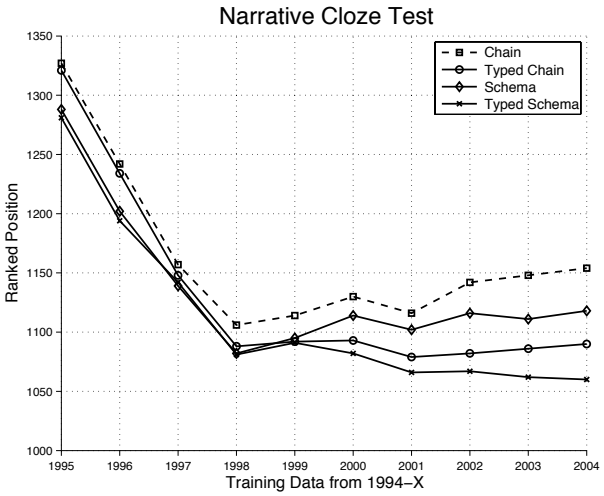


Figure 6: Results with varying sizes of training data.

We parse the text into typed dependency graphs with the Stanford Parser (de Marneffe et al., 2006), recording all verbs with subject, object, or prepositional typed dependencies. Unlike in (Chambers and Jurafsky, 2008), we lemmatize verbs and argument head words. We use the OpenNLP[1] coreference engine to resolve entity mentions.

The test set is the same as in (Chambers and Jurafsky, 2008). 100 random news articles were selected from the 2001 NYT section of the Gigaword Corpus. Articles that did not contain a protagonist with five or more events were ignored, leaving a test set of 69 articles. We used a smaller development set of size 17 to tune parameters.

6.3 Typed Chains

The first evaluation compares untyped against typed narrative event chains. The typed model uses equation 4 for chain clustering. The dotted line 'Chain' and solid 'Typed Chain' in figure 6 shows the average ranked position over the test set. The untyped chains plateau and begin to worsen as the amount of training data increases, but the typed model is able to improve for some time after. We see a **6.9%** gain at 2004 when both lines trend upwards.

6.4 Narrative Schema

The second evaluation compares the performance of the narrative schema model against single narrative chains. We ignore argument types and use untyped chains in both (using equation 1 instead of 4). The dotted line 'Chain' and solid 'Schema' show performance results in figure 6. Narrative Schemas have better ranked scores in all data sizes and follow the previous experiment in improving results as more data is added even though untyped chains trend upward. We see a **3.3%** gain at 2004.

6.5 Typed Narrative Schema

The final evaluation combines *schemas* with *argument types* to measure overall gain. We evaluated with both head words and CBC clusters as argument representations. Not only do typed chains and schemas outperform untyped chains, combining the two gives a further performance boost. Clustered arguments improve the results further, helping with sparse argument counts ('Typed Schema' in figure 6 uses CBC arguments). Overall, using all the data (by year 2004) shows a **10.1%** improvement over untyped narrative chains.

7 Discussion

Our significant improvement in the cloze evaluation shows that even though narrative cloze does not evaluate argument types, jointly modeling the arguments with events improves event clustering. Likewise, the FrameNet comparison suggests that modeling related events helps argument learning. The tasks mutually inform each other. Our argument learning algorithm not only performs unsupervised induction of situation-specific role classes, but the resulting roles and linking structures may also offer the possibility of (unsupervised) FrameNet-style semantic role labeling.

Finding the best argument representation is an important future direction. The performance of our noun clusters in figure 6 showed that while the other approaches leveled off, clusters continually improved with more data. The exact balance between lexical units, clusters, or more general (traditional) semantic roles remains to be solved, and may be application specific.

We hope in the future to show that a range of NLU applications can benefit from the rich inferential structures that narrative schemas provide.

Acknowledgments

This work is funded in part by NSF (IIS-0811974). We thank the reviewers and the Stanford NLP Group for helpful suggestions.

[1] http://opennlp.sourceforge.net/

References

Afra Alishahi and Suzanne Stevenson. 2007. A computational usage-based model for learning general properties of semantic roles. In *The 2nd European Cognitive Science Conference*, Delphi, Greece.

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet project. In Christian Boitet and Pete Whitelock, editors, *ACL-98*, pages 86–90, San Francisco, California. Morgan Kaufmann Publishers.

David Bean and Ellen Riloff. 2004. Unsupervised learning of contextual role knowledge for coreference resolution. *Proc. of HLT/NAACL*, pages 297–304.

Nathanael Chambers and Dan Jurafsky. 2008. Unsupervised learning of narrative event chains. In *Proceedings of ACL-08*, Hawaii, USA.

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC-06*, pages 449–454.

David Graff. 2002. English Gigaword. *Linguistic Data Consortium*.

Rebecca Green and Bonnie J. Dorr. 2005. Frame semantic enhancement of lexical-semantic resources. In *ACL-SIGLEX Workshop on Deep Lexical Acquisition*, pages 57–66.

Trond Grenager and Christopher D. Manning. 2006. Unsupervised discovery of a statistical verb lexicon. In *EMNLP*.

Shan He and Daniel Gildea. 2006. Self-training and co-training for semantic role labeling: Primary report. Technical Report 891, University of Rochester.

Karin Kipper, Hoa Trang Dang, and Martha Palmer. 2000. Class-based construction of a verb lexicon. In *Proceedings of AAAI-2000*, Austin, TX.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The proposition bank: A corpus annotated with semantic roles. *Computational Linguistics*, 31(1):71–106.

Patrick Pantel and Dekang Lin. 2002. Document clustering with committees. In *ACM Conference on Research and Development in Information Retrieval*, pages 199–206, Tampere, Finland.

Roger C. Schank and Robert P. Abelson. 1977. *Scripts, plans, goals and understanding*. Lawrence Erlbaum.

Robert S. Swier and Suzanne Stevenson. 2004. Unsupervised semantic role labelling. In *EMNLP*.

Wilson L. Taylor. 1953. Cloze procedure: a new tool for measuring readability. *Journalism Quarterly*, 30:415–433.