Unsupervised Learning of Narrative Event Chains

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

Abstract

Hand-coded *scripts* were used in the 1970-80s as knowledge backbones that enabled inference and other NLP tasks requiring deep semantic knowledge. We propose unsupervised induction of similar schemata called *narrative event chains* from raw newswire text.

A narrative event chain is a partially ordered set of events related by a common protagonist. We describe a three step process to learning narrative event chains. The first uses unsupervised distributional methods to learn narrative relations between events sharing coreferring arguments. The second applies a temporal classifier to partially order the connected events. Finally, the third prunes and clusters self-contained chains from the space of events. We introduce two evaluations: the *narrative cloze* to evaluate event relatedness, and an *order coherence* task to evaluate narrative order. We show a 36% improvement over baseline for narrative prediction and 25% for temporal coherence.

1 Introduction

This paper induces a new representation of structured knowledge called **narrative event chains** (or narrative chains). Narrative chains are partially ordered sets of events centered around a common **protagonist**. They are related to structured sequences of participants and events that have been called **scripts** (Schank and Abelson, 1977) or *Fillmorean frames*. These participants and events can be filled in and instantiated in a particular text situation to draw inferences. Chains focus on a single actor to facilitate learning, and thus this paper addresses the three tasks of chain induction: *narrative event induction*, *temporal ordering of events* and *structured selection* (pruning the event space into discrete sets).

Learning these prototypical schematic sequences of events is important for rich understanding of text. Scripts were central to natural language understanding research in the 1970s and 1980s for proposed tasks such as summarization, coreference resolution and question answering. For example, Schank and Abelson (1977) proposed that understanding text about restaurants required knowledge about the Restaurant Script, including the participants (Customer, Waiter, Cook, Tables, etc.), the events constituting the script (entering, sitting down, asking for menus, etc.), and the various preconditions, ordering, and results of each of the constituent actions.

Consider these two distinct narrative chains.

_ accused X	**W joined _**
X claimed _	**W served _**
X argued	**W oversaw _**
_ dismissed X	**W resigned**

It would be useful for question answering or textual entailment to know that '**X denied _**' is also a likely event in the left chain, while '**_ replaces W**' temporally follows the right. Narrative chains (such as *Firing of Employee* or *Executive Resigns*) offer the structure and power to directly infer these new subevents by providing critical background knowledge. In part due to its complexity, automatic induction has not been addressed since the early non-statistical work of Mooney and DeJong (1985).

The first step to narrative induction uses an entity-based model for learning narrative relations by fol-

lowing a protagonist. As a narrative progresses through a series of events, each event is characterized by the grammatical role played by the protagonist, and by the protagonist's shared connection to surrounding events. Our algorithm is an unsupervised distributional learning approach that uses coreferring arguments as evidence of a narrative relation. We show, using a new evaluation task called **narrative cloze**, that our protagonist-based method leads to better induction than a verb-only approach.

The next step is to order events in the same narrative chain. We apply work in the area of temporal classification to create partial orders of our learned events. We show, using a coherence-based evaluation of temporal ordering, that our partial orders lead to better coherence judgements of real narrative instances extracted from documents.

Finally, the space of narrative events and temporal orders is clustered and pruned to create discrete sets of narrative chains.

2 Previous Work

While previous work hasn't focused specifically on learning narratives[1], our work draws from two lines of research in summarization and anaphora resolution. In summarization, **topic signatures** are a set of terms indicative of a topic (Lin and Hovy, 2000). They are extracted from hand-sorted (by topic) sets of documents using log-likelihood ratios. These terms can capture some narrative relations, but the model requires topic-sorted training data.

Bean and Riloff (2004) proposed the use of **caseframe networks** as a kind of contextual role knoweldge for anaphora resolution. A caseframe is a verb/event and a semantic role (e.g. *<patient> kidnapped*). Caseframe networks are relations between caseframes that may represent synonymy (*<patient> kidnapped* and *<patient> abducted*) or related events (*<patient> kidnapped* and *<patient> released*). Bean and Riloff learn these networks from two topic-specific texts and apply them to the problem of anaphora resolution. Our work can be seen as an attempt to generalize the intuition of caseframes (finding an entire set of events rather than just pairs of related frames) and apply it to a different task (finding a coherent structured narrative in non-topic-specific text).

More recently, Brody (2007) proposed an approach similar to caseframes that discovers high-level relatedness between verbs by grouping verbs that share the same lexical items in subject/object positions. He calls these shared arguments *anchors*. Brody learns pairwise relations between clusters of related verbs, similar to the results with caseframes. A human evaluation of these pairs shows an improvement over baseline. This and previous caseframe work lend credence to learning relations from verbs with common arguments.

We also draw from lexical chains (Morris and Hirst, 1991), indicators of text coherence from word overlap/similarity. We use a related notion of protagonist overlap to motivate narrative chain learning.

Work on semantic similarity learning such as Chklovski and Pantel (2004) also automatically learns relations between verbs. We use similar distributional scoring metrics, but differ with our use of a protagonist as the indicator of relatedness. We also use typed dependencies and the entire space of events for similarity judgements, rather than only pairwise lexical decisions.

Finally, Fujiki et al. (2003) investigated script acquisition by extracting the 41 most frequent pairs of events from the first paragraph of newswire articles, using the assumption that the paragraph's textual order follows temporal order. Our model, by contrast, learns entire event chains, uses more sophisticated probabilistic measures, and uses temporal ordering models instead of relying on document order.

3 The Narrative Chain Model

3.1 Definition

Our model is inspired by Centering (Grosz et al., 1995) and other entity-based models of coherence (Barzilay and Lapata, 2005) in which an entity is in focus through a sequence of sentences. We propose to use this same intuition to induce narrative chains.

We assume that although a narrative has several participants, there is a central actor who characterizes a narrative chain: the **protagonist**. Narrative chains are thus structured by the protagonist's grammatical roles in the events. In addition, narrative

[1] We analyzed FrameNet (Baker et al., 1998) for insight, but found that very few of the frames are event *sequences* of the type characterizing narratives and scripts.

events are ordered by some theory of time. This paper describes a partial ordering with the *before* (no overlap) relation.

Our task, therefore, is to learn events that constitute narrative chains. Formally, a **narrative chain** is a partially ordered set of narrative events that share a common actor. A **narrative event** is a tuple of an event (most simply a verb) and its participants, represented as *typed dependencies*. Since we are focusing on a single actor in this study, a narrative event is thus a tuple of the event and the typed dependency of the protagonist: *(event, dependency)*. A narrative chain is a set of narrative events $\{e_1, e_2, ..., e_n\}$, where n is the size of the chain, and a relation $B(e_i, e_j)$ that is true if narrative event e_i occurs strictly before e_j in time.

3.2 The Protagonist

The notion of a protagonist motivates our approach to narrative learning. We make the following assumption of **narrative coherence**: *verbs sharing coreferring arguments are semantically connected by virtue of narrative discourse structure.* A single document may contain more than one narrative (or topic), but the narrative assumption states that a series of argument-sharing verbs is more likely to participate in a narrative chain than those not sharing.

In addition, the narrative approach captures grammatical constraints on *narrative coherence*. Simple distributional learning might discover that the verb *push* is related to the verb *fall*, but narrative learning can capture additional facts about the participants, specifically, that the object or patient of the *push* is the subject or agent of the *fall*.

Each focused protagonist chain offers one perspective on a narrative, similar to the multiple perspectives on a commercial transaction event offered by buy and sell.

3.3 Partial Ordering

A narrative chain, by definition, includes a partial ordering of events. Early work on scripts included ordering constraints with more complex preconditions and side effects on the sequence of events. This paper presents work toward a partial ordering and leaves logical constraints as future work. We focus on the *before* relation, but the model does not preclude advanced theories of temporal order.

4 Learning Narrative Relations

Our first model learns basic information about a narrative chain: the protagonist and the constituent subevents, although not their ordering. For this we need a metric for the relation between an event and a narrative chain.

Pairwise relations between events are first extracted unsupervised. A distributional score based on how often two events share grammatical arguments (using pointwise mutual information) is used to create this pairwise relation. Finally, a global narrative score is built such that *all* events in the chain provide feedback on the event in question (whether for inclusion or for decisions of inference).

Given a list of observed verb/dependency counts, we approximate the pointwise mutual information (PMI) by:

$$pmi(e(w,d), e(v,g)) = log \frac{P(e(w,d), e(v,g))}{P(e(w,d))P(e(v,g))} \quad (1)$$

where $e(w,d)$ is the verb/dependency pair w and d (e.g. e(*push*,*subject*)). The numerator is defined by:

$$P(e(w,d), e(v,g)) = \frac{C(e(w,d), e(v,g))}{\sum_{x,y} \sum_{d,f} C(e(x,d), e(y,f))} \quad (2)$$

where $C(e(x,d), e(y,f))$ is the number of times the two events $e(x,d)$ and $e(y,f)$ had a coreferring entity filling the values of the dependencies d and f. We also adopt the 'discount score' to penalize low occuring words (Pantel and Ravichandran, 2004).

Given the debate over appropriate metrics for distributional learning, we also experimented with the t-test. Our experiments found that PMI outperforms the t-test on this task by itself and when interpolated together using various mixture weights.

Once pairwise relation scores are calculated, a global narrative score can then be built such that *all* events provide feedback on the event in question. For instance, given all narrative events in a document, we can find the next most likely event to occur by maximizing:

$$\max_{j:0<j<m} \sum_{i=0}^{n} pmi(e_i, f_j) \quad (3)$$

where n is the number of events in our chain and e_i is the ith event. m is the number of events f in our training corpus. A ranked list of guesses can be built from this summation and we hypothesize that

Known events:
(pleaded subj), (admits subj), (convicted obj)

Likely Events:

sentenced obj	0.89	indicted obj	0.74
paroled obj	0.76	fined obj	0.73
fired obj	0.75	denied subj	0.73

Figure 1: Three narrative events and the six most likely events to include in the same chain.

the more events in our chain, the more informed our ranked output. An example of a chain with 3 events and the top 6 ranked guesses is given in figure 1.

4.1 Evaluation Metric: Narrative Cloze

The **cloze task** (Taylor, 1953) is used to evaluate a system (or human) for language proficiency by removing a random word from a sentence and having the system attempt to fill in the blank (e.g. *I forgot to ___ the waitress for the good service*). Depending on the type of word removed, the test can evaluate syntactic knowledge as well as semantic. Deyes (1984) proposed an extended task, **discourse cloze**, to evaluate discourse knowledge (removing phrases that are recoverable from knowledge of discourse relations like *contrast* and *consequence*).

We present a new cloze task that requires narrative knowledge to solve, the **narrative cloze**. The narrative cloze is a sequence of narrative events in a document from which one event has been removed. The task is to predict the missing verb and typed dependency. Take this example text about American football with McCann as the protagonist:

1. McCann threw two interceptions early.
2. Toledo pulled McCann aside and told him he'd start.
3. McCann quickly completed his first two passes.

These clauses are represented in the narrative model as five events: **(threw subject), (pulled object), (told object), (start subject), (completed subject)**. These verb/dependency events make up a narrative cloze model. We could remove (**threw subject**) and use the remaining four events to rank this missing event. Removing a single such pair to be filled in automatically allows us to evaluate a system's knowledge of narrative relations and coherence. We do not claim this cloze task to be solvable even by humans, but rather assert it as a comparative measure to evaluate narrative knowledge.

New York Times Editorial

occupied subj	brought subj	rejecting subj
projects subj	met subj	appeared subj
offered subj	voted pp_for	offer subj
thinks subj		

Figure 2: One of the 69 test documents, containing 10 narrative events. The protagonist is President Bush.

4.2 Narrative Cloze Experiment

We use years 1994-2004 (1,007,227 documents) of the Gigaword Corpus (Graff, 2002) for **training**[2]. We parse the text into typed dependency graphs with the Stanford Parser (de Marneffe et al., 2006)[3], recording all verbs with subject, object, or prepositional typed dependencies. We use the OpenNLP[4] coreference engine to resolve the entity mentions. For each document, the verb pairs that share coreferring entities are recorded with their dependency types. Particles are included with the verb.

We used 10 news stories from the 1994 section of the corpus for *development*. The stories were hand chosen to represent a range of topics such as business, sports, politics, and obituaries. We used 69 news stories from the 2001 (year selected randomly) section of the corpus for **testing** (also removed from training). The test set documents were randomly chosen and not preselected for a range of topics. From each document, the entity involved in the most events was selected as the protagonist. For this evaluation, we only look at verbs. All verb clauses involving the protagonist are manually extracted and translated into the narrative events *(verb,dependency)*. Exceptions that are not included are verbs in headlines, quotations (typically not part of a narrative), "be" properties (e.g. *john is happy*), modifying verbs (e.g. *hurried to leave*, only *leave* is used), and multiple instances of one event.

The original test set included 100 documents, but

[2]The document count does not include duplicate news stories. We found up to 18% of the corpus are duplications, mostly AP reprints. We automatically found these by matching the first two paragraphs of each document, removing exact matches.

[3]http://nlp.stanford.edu/software/lex-parser.shtml

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

those without a narrative chain at least five events in length were removed, leaving 69 documents. Most of the removed documents were not stories, but genres such as interviews and cooking recipes. An example of an extracted chain is shown in figure 2.

We evalute with Narrative Cloze using leave-one-out cross validation, removing one event and using the rest to generate a ranked list of guesses. The test dataset produces 740 cloze tests (69 narratives with 740 events). After generating our ranked guesses, the position of the correct event is averaged over all 740 tests for the final score. We penalize unseen events by setting their ranked position to the length of the guess list (ranging from 2k to 15k).

Figure 1 is an example of a ranked guess list for a short chain of three events. If the original document contained (*fired obj*), this cloze test would score 3.

4.2.1 Baseline

We want to measure the utility of the protagonist and the *narrative coherence assumption*, so our baseline learns relatedness strictly based upon verb co-occurence. The PMI is then defined as between all occurrences of two verbs in the same document. This baseline evaluation is verb only, as dependencies require a protagonist to fill them.

After initial evaluations, the baseline was performing very poorly due to the huge amount of data involved in counting all possible verb pairs (using a protagonist vastly reduces the number). We experimented with various count cutoffs to remove rare occurring pairs of verbs. The final results use a baseline where all pairs occurring less than 10 times in the training data are removed.

Since the verb-only baseline does not use typed dependencies, our narrative model cannot directly compare to this abstracted approach. We thus modified the narrative model to ignore typed dependencies, but still count events with shared arguments. Thus, we calculate the PMI across verbs *that share arguments*. This approach is called **Protagonist**. The full narrative model that includes the grammatical dependencies is called **Typed Deps**.

4.2.2 Results

Experiments with varying sizes of training data are presented in figure 3. Each ranked list of candidate verbs for the missing event in Baseline/Protagonist contained approximately 9 thousand candidates. Of the 740 cloze tests, 714 of the removed events were present in their respective list of guesses. This is encouraging as only 3.5% of the events are unseen (or do not meet cutoff thresholds).

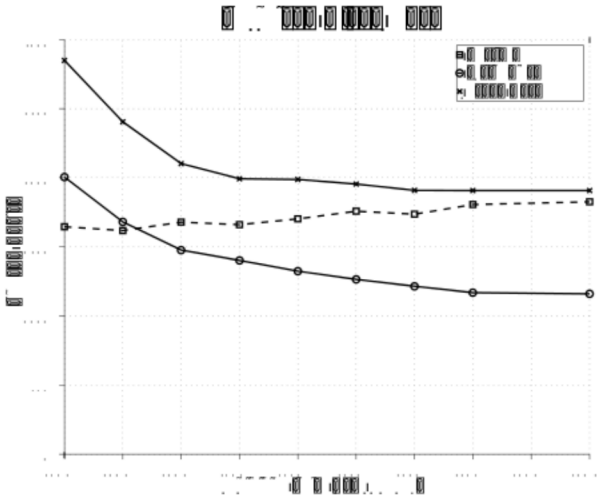

Figure 3: Results with varying sizes of training data. Year 2003 is not explicitly shown because it has an unusually small number of documents compared to other years.

When all training data is used (1994-2004), the average ranked position is 1826 for **Baseline** and 1160 for **Protagonist** (1 being most confident). The Baseline performs better at first (years 1994-5), but as more data is seen, the Baseline worsens while the Protagonist improves. This verb-only narrative model shows a **36.5%** improvement over the baseline trained on all years. Results from the full **Typed Deps** model, not comparable to the baseline, parallel the Protagonist results, improving as more data is seen (average ranked position of 1908 with all the training data). We also ran the experiment without OpenNLP coreference, and instead used exact and substring matching for coreference resolution. This showed a 5.7% decrease in the verb-only results. These results show that a protagonist greatly assists in narrative judgements.

5 Ordering Narrative Events

The model proposed in the previous section is designed to learn the major subevents in a narrative chain, but not how these events are ordered. In this section we extend the model to learn a partial temporal ordering of the events.

There are a number of algorithms for determining the temporal relationship between two events (Mani et al., 2006; Lapata and Lascarides, 2006; Chambers et al., 2007), many of them trained on the TimeBank Corpus (Pustejovsky et al., 2003) which codes events and their temporal relationships. The currently highest performing of these on raw data is the model of temporal labeling described in our previous work (Chambers et al., 2007). Other approaches have depended on hand tagged features.

Chambers et al. (2007) shows 59.4% accuracy on the classification task for six possible relations between pairs of events: *before, immediately-before, included-by, simultaneous, begins* and *ends*. We focus on the *before* relation because the others are less relevant to our immediate task. We combine *immediately-before* with *before*, and merge the other four relations into an *other* category. At the binary task of determining if one event is *before* or *other*, we achieve 72.1% accuracy on Timebank.

The above approach is a two-stage machine learning architecture. In the **first stage**, the model uses supervised machine learning to label temporal attributes of events, including tense, grammatical aspect, and aspectual class. This first stage classifier relies on features such as neighboring part of speech tags, neighboring auxiliaries and modals, and WordNet synsets. We use SVMs (Chambers et al. (2007) uses Naive Bayes) and see minor performance boosts on Timebank. These imperfect classifications, combined with other linguistic features, are then used in a **second stage** to classify the temporal relationship between two events. Other features include event-event syntactic properties such as the syntactic dominance relations between the two events, as well as new bigram features of tense, aspect and class (e.g. "present past" if the first event is in the present, and the second past), and whether the events occur in the same or different sentences.

5.1 Training a Temporal Classifier

We use the entire Timebank Corpus as supervised training data, condensing the *before* and *immediately-before* relations into one *before* relation. The remaining relations are merged into *other*.

The vast majority of potential event pairs in Timebank are unlabeled. These are often *none* relations (events that have no explicit relation) or as is often the case, *overlap* relations where the two events have no Timebank-defined ordering but overlap in time. Even worse, many events do have an ordering, but they were not tagged by the human annotators. This could be due to the overwhelming task of temporal annotation, or simply because some event orderings are deemed more important than others in understanding the document. We consider all untagged relations as *other*, and experiment with including none, half, and all of them in training.

Taking a cue from Mani et al. (2006), we also increased Timebank's size by applying transitivity rules to the hand labeled data. The following is an example of the applied transitive rule:

if run BEFORE fall **and** fall BEFORE injured
then run BEFORE injured

This increases the number of relations from 37519 to 45619. Perhaps more importantly for our task, of all the added relations, the *before* relation is added the most. We experimented with original vs. expanded Timebank and found the expanded performed slightly worse. The decline may be due to poor transitivity additions, as several Timebank documents contain inconsistent labelings. All reported results are from training *without* transitivity.

5.2 Temporal Classifier in Narrative Chains

We classify the Gigaword Corpus in two stages, once for the temporal features on each event (tense, grammatical aspect, aspectual class), and once between all pairs of events that share arguments. This allows us to classify the before/other relations between all potential narrative events.

The first stage is trained on Timebank, and the second is trained using the approach described above, varying the size of the *none* training relations. Each pair of events in a gigaword document that share a coreferring argument is treated as a separate ordering classification task. We count the resulting number of labeled *before* relations between each verb/dependency pair. Processing the entire corpus produces a database of event pair counts where confidence of two generic events A and B can be measured by comparing how many *before* labels have been seen versus their inverted order B and A[5].

[5]Note that we train with the *before* relation, and so transposing two events is similar to classifying the *after* relation.

5.3 Temporal Evaluation

We want to evaluate temporal order at the narrative level, across all events within a chain. We envision narrative chains being used for tasks of coherence, among other things, and so it is desired to evaluate temporal decisions within a coherence framework. Along these lines, our test set uses actual narrative chains from documents, hand labeled for a partial ordering. We evaluate coherence of these true chains against a random ordering. The task is thus deciding which of the two chains is most coherent, the original or the random (baseline 50%)? We generated up to 300 random orderings for each test document, averaging the accuracy across all.

Our evaluation data is the same 69 documents used in the test set for learning narrative relations. The chain from each document is hand identified and labeled for a partial ordering using only the *before* relation. Ordering was done by the authors and all attempts were made to include every before relation that exists in the document, or that could be deduced through transitivity rules. Figure 4 shows an example and its full reversal, although the evaluation uses random orderings. Each edge is a distinct before relation and is used in the judgement score.

The coherence score for a partially ordered narrative chain is the sum of all the relations that our classified corpus agrees with, weighted by how certain we are. If the gigaword classifications disagree, a weighted negative score is given. Confidence is based on a logarithm scale of the difference between the counts of before and after classifications. Formally, the score is calculated as the following:

$$\sum_{E:x,y} \begin{cases} log(D(x,y)) & \text{if } x\beta y \text{ and } B(x,y) > B(y,x) \\ -log(D(x,y)) & \text{if } x\beta y \text{ and } B(y,x) > B(x,y) \\ -log(D(x,y)) & \text{if } !x\beta y \;\&\; !y\beta x \;\&\; D(x,y) > 0 \\ 0 & \text{otherwise} \end{cases}$$

where E is the set of all event pairs, $B(i,j)$ is how many times we classified events i and j as *before* in Gigaword, and $D(i,j) = |B(i,j) - B(j,i)|$. The relation $i\beta j$ indicates that i is temporally before j.

5.4 Results

Out approach gives higher scores to orders that coincide with the pairwise orderings classified in our gigaword training data. The results are shown in figure 5. Of the 69 chains, 6 did not have any ordered events and were removed from the evaluation. We

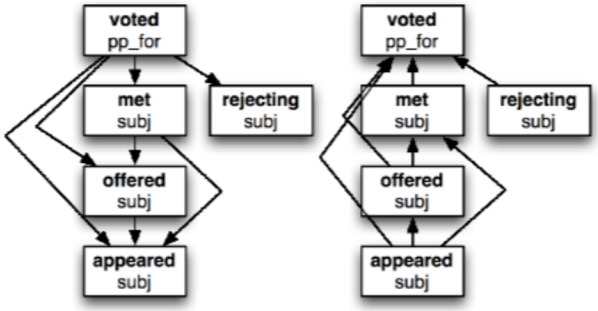

Figure 4: A narrative chain and its reverse order.

	All	$\geq$ **6**	$\geq$ **10**
correct	8086 **75%**	7603 **78%**	6307 **89%**
incorrect	1738	1493	619
tie	931	627	160

Figure 5: Results for choosing the correct ordered chain. ($\geq$ 10) means there were at least 10 pairs of ordered events in the chain.

generated (up to) 300 random orderings for each of the remaining 63. We report 75.2% accuracy, but 22 of the 63 had 5 or fewer pairs of ordered events. Figure 5 therefore shows results from chains with more than 5 pairs, and also 10 or more. As we would hope, the accuracy improves the larger the ordered narrative chain. We achieve 89.0% accuracy on the 24 documents whose chains most progress through time, rather than chains that are difficult to order with just the *before* relation.

Training without *none* relations resulted in high recall for *before* decisions. Perhaps due to data sparsity, this produces our best results as reported above.

6 Discrete Narrative Event Chains

Up till this point, we have learned narrative relations across all possible events, including their temporal order. However, the discrete lists of events for which Schank scripts are most famous have not yet been constructed.

We intentionally did not set out to reproduce explicit self-contained *scripts* in the sense that the 'restaurant script' is complete and cannot include other events. The name *narrative* was chosen to imply a *likely order* of events that is common in spoken and written retelling of world events. Discrete sets have the drawback of shutting out unseen and un-

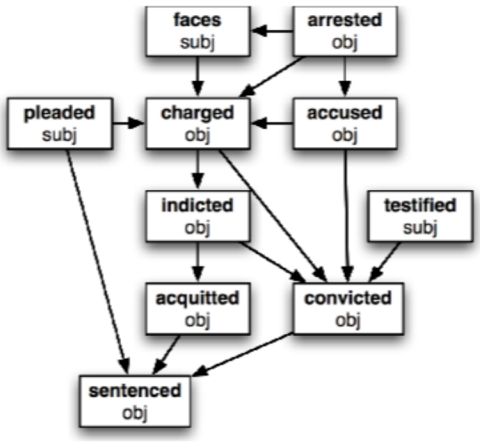

Figure 6: An automatically learned Prosecution Chain. Arrows indicate the *before* relation.

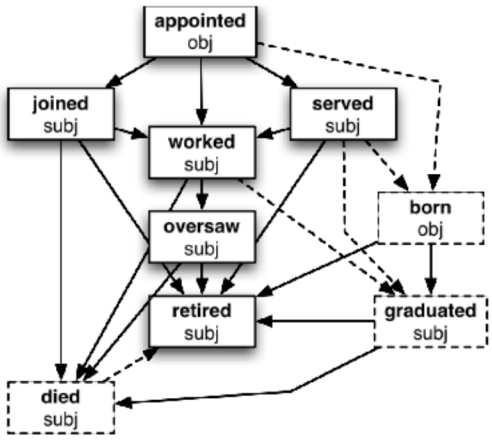

Figure 7: An Employment Chain. Dotted lines indicate incorrect *before* relations.

likely events from consideration. It is advantageous to consider a space of possible narrative events and the ordering within, not a closed list.

However, it is worthwhile to construct discrete narrative chains, if only to see whether the combination of event learning and ordering produce script-like structures. This is easily achievable by using the PMI scores from section 4 in an agglomerative clustering algorithm, and then applying the ordering relations from section 5 to produce a directed graph.

Figures 6 and 7 show two learned chains after clustering and ordering. Each arrow indicates a before relation. Duplicate arrows implied by rules of transitivity are removed. Figure 6 is remarkably accurate, and figure 7 addresses one of the chains from our introduction, the employment narrative. The core employment events are accurate, but clustering included life events (born, died, graduated) from obituaries of which some temporal information is incorrect. The Timebank corpus does not include obituaries, thus we suffer from sparsity in training data.

7 Discussion

We have shown that it is possible to learn narrative event chains unsupervised from raw text. Not only do our narrative relations show improvements over a baseline, but narrative chains offer hope for many other areas of NLP. Inference, coherence in summarization and generation, slot filling for question answering, and frame induction are all potential areas.

We learned a new measure of similarity, the narrative relation, using the protagonist as a hook to extract a list of related events from each document. The 37% improvement over a verb-only baseline shows that we may not need presorted topics of documents to learn inferences. In addition, we applied state of the art temporal classification to show that sets of events can be partially ordered. Judgements of coherence can then be made over chains within documents. Further work in temporal classification may increase accuracy even further.

Finally, we showed how the event space of narrative relations can be clustered to create discrete sets. While it is unclear if these are better than an unconstrained distribution of events, they do offer insight into the quality of narratives.

An important area not discussed in this paper is the possibility of using narrative chains for semantic role learning. A narrative chain can be viewed as defining the semantic roles of an event, constraining it against roles of the other events in the chain. An argument's class can then be defined as the set of narrative arguments in which it appears.

We believe our model provides an important first step toward learning the rich causal, temporal and inferential structure of scripts and frames.

Acknowledgment: This work is funded in part by DARPA through IBM and by the DTO Phase III Program for AQUAINT through Broad Agency Announcement (BAA) N61339-06-R-0034. Thanks to the reviewers for helpful comments and the suggestion for a non-full-coreference baseline.

References

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.

Regina Barzilay and Mirella Lapata. 2005. Modeling local coherence: an entity-based approach. *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 141–148.

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

Samuel Brody. 2007. Clustering Clauses for High-Level Relation Detection: An Information-theoretic Approach. *Proceedings of the 43rd Annual Meeting of the Association of Computational Linguistics*, pages 448–455.

Nathanael Chambers, Shan Wang, and Dan Jurafsky. 2007. Classifying temporal relations between events. In *Proceedings of ACL-07*, Prague, Czech Republic.

Timothy Chklovski and Patrick Pantel. 2004. Verbocean: Mining the web for fine-grained semantic verb relations. In *Proceedings of EMNLP-04*.

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.

Tony Deyes. 1984. Towards an authentic 'discourse cloze'. *Applied Linguistics*, 5(2).

Toshiaki Fujiki, Hidetsugu Nanba, and Manabu Okumura. 2003. Automatic acquisition of script knowledge from a text collection. In *EACL*, pages 91–94.

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

Barbara J. Grosz, Aravind K. Joshi, and Scott Weinstein. 1995. Centering: A framework for modelling the local coherence of discourse. *Computational Linguistics*, 21(2).

Mirella Lapata and Alex Lascarides. 2006. Learning sentence-internal temporal relations. In *Journal of AI Research*, volume 27, pages 85–117.

C.Y. Lin and E. Hovy. 2000. The automated acquisition of topic signatures for text summarization. *Proceedings of the 17th conference on Computational linguistics-Volume 1*, pages 495–501.

Inderjeet Mani, Marc Verhagen, Ben Wellner, Chong Min Lee, and James Pustejovsky. 2006. Machine learning of temporal relations. In *Proceedings of ACL-06*, July.

Raymond Mooney and Gerald DeJong. 1985. Learning schemata for natural language processing. In *Ninth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 681–687.

Jane Morris and Graeme Hirst. 1991. Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational Linguistics*, 17:21–43.

Patrick Pantel and Deepak Ravichandran. 2004. Automatically labeling semantic classes. *Proceedings of HLT/NAACL*, 4:321–328.

James Pustejovsky, Patrick Hanks, Roser Sauri, Andrew See, David Day, Lisa Ferro, Robert Gaizauskas, Marcia Lazo, Andrea Setzer, and Beth Sundheim. 2003. The timebank corpus. *Corpus Linguistics*, pages 647–656.

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

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