Script Induction as Language Modeling

Rachel Rudinger[1], **Pushpendre Rastogi**[1], **Francis Ferraro**[1], **and Benjamin Van Durme**[1,2]
[1]Center for Language and Speech Processing
[2]Human Language Technology Center of Excellence
Johns Hopkins University

Abstract

The *narrative cloze* is an evaluation metric commonly used for work on automatic script induction. While prior work in this area has focused on count-based methods from distributional semantics, such as pointwise mutual information, we argue that the narrative cloze can be productively reframed as a language modeling task. By training a discriminative language model for this task, we attain improvements of up to 27 percent over prior methods on standard narrative cloze metrics.

1 Introduction

Although the concept of *scripts* in artificial intelligence dates back to the 1970s (Schank and Abelson, 1977), interest in this topic has renewed with recent efforts to automatically induce scripts from text on a large scale. One particularly influential work in this area, Chambers and Jurafsky (2008), treats the problem of script induction as one of learning *narrative chains*, which they accomplish using simple textual co-occurrence statistics. For the novel task of learning narrative chains, they introduce a new evaluation metric, the *narrative cloze* test, which involves predicting a missing event from a chain of events drawn from text. Several follow-up works (Chambers and Jurafsky, 2009; Jans et al., 2012; Pichotta and Mooney, 2014; Rudinger et al., 2015) employ and extend Chambers and Jurafsky (2008)'s methods for learning narrative chains, each using the narrative cloze to evaluate their work. [1]

In this paper, we take the position that the narrative cloze test, which has been treated predominantly as a method for evaluating script knowledge, is more productively thought of simply as a language modeling task.[2] To support this claim, we demonstrate a marked improvement over previous methods on this task using a powerful discriminative language model – the Log-Bilinear model (LBL). Based on this finding, we believe one of the following conclusions must follow: either discriminative language models are a more effective technique for script induction than previous methods, or the narrative cloze test is not a suitable evaluation for this task.[3]

2 Task Definition

Following the definitions of Chambers and Jurafsky (2008), a **narrative chain** is "a partially ordered set of narrative events that share a common actor," where a **narrative event** is "a tuple of an event (most simply a verb) and its participants, represented as *typed dependencies*." (De Marneffe et al., 2006) Formally, $e := (v, d)$, where e is a narrative event, v is a verb lemma, and d is the syntactic dependency (*nsubj* or *dobj*) between v and the protagonist. As an example, consider the following narrative:

> John studied for the exam and aced it.
> His teacher congratulated him.

With John as protagonist, we have a sequence of three narrative events: $(study, nsubj)$, $(ace, nsubj)$, and $(congratulate, dobj)$.

In the **narrative cloze** test, a sequence of narrative events (like the example provided here) is extracted automatically from a document, and one

[1]A number of related works on script induction use alternative task formulations and evaluations. (Chambers, 2013; Cheung et al., 2013; Cheung and Penn, 2013; Frermann et al., 2014; Manshadi et al., 2008; Modi and Titov, 2014; Regneri et al., 2010)

[2]Manshadi et al. (2008) also take a language modeling approach to event prediction, although their experiments are not directly comparable.

[3]We note that, whether the narrative cloze was originally intended as a rigorous evaluation of script induction techniques or merely a preliminary metric, we are motivated by the observation that this evaluation has nonetheless become a standard metric for this task.

Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 1681–1686,
Lisbon, Portugal, 17-21 September 2015.

narrative event is removed; the task is to predict the missing event.

Data Each of the models discussed in the following section are trained and tested on chains of narrative events extracted from stories in the New York Times portion of the Gigaword corpus (Graff et al., 2003) with Concrete annotations (Ferraro et al., 2014). Training is on the entirety of the 1994–2006 portion (16,688,422 chains with 58,515,838 narrative events); development is a subset of the 2007–2008 portion (10,000 chains with 35,109 events); and test is a subset of the 2009–2010 portion (5,000 chains with 17,836 events). All extracted chains are of length two or greater.

Chain Extraction To extract chains of narrative events for training and testing, we rely on the (automatically-generated) coreference chains present in Concretely Annotated Gigaword. Each narrative event in an extracted chain is derived from a single mention in the corresponding coreference chain, i.e., it consists of the verb and syntactic dependency ($nsubj$ or $dobj$) that governs the head of the mention, if such a dependency exists. Overlapping mentions within a coreference chain are collapsed to a single mention to avoid redundant extractions.

3 Models

In this section we present each of the models we train for the narrative cloze evaluation. In a single narrative cloze test, a sequence of narrative events, $(e_1, \cdots, e_L)$, with an insertion point, k, for the missing event is provided. Given a fixed vocabulary of narrative events, $\mathcal{V}$, a candidate sequence is generated for each vocabulary item by inserting that item into the sequence at index k. Each model generates a score for the candidate sequences, yielding a ranking over the vocabulary items. The rank assigned to the actual missing vocabulary item is the score the model receives on that cloze test. In this case, we set $\mathcal{V}$ to include all narrative events, e, that occur at least ten times in training, yielding a vocabulary size of 12,452. All out-of-vocabulary events are converted to (and scored as) the symbol UNK.

3.1 Count-based Methods

Unigram Baseline (UNI) A simple but strong baseline introduced by Pichotta and Mooney (2014) for this task is the unigram model: candidates are ranked by their observed frequency in training, without regard to context.

Unordered PMI (UOP) The original model for this task, proposed by Chambers and Jurafsky (2008), is based on the pointwise mutual information (PMI) between events.

$$pmi(e_1, e_2) \propto \log \frac{C(e_1, e_2)}{C(e_1, *)C(*, e_2)} \quad (1)$$

Here, $C(e_1, e_2)$ is the number of times e_1 and e_2 occur in the same narrative event sequence, i.e., the number of times they "had a coreferring entity filling the values of [their] dependencies," and the ordering of e_1 and e_2 is not considered. In our implementation, individual counts are defined as follows:

$$C(e, *) := \sum_{e' \in \mathcal{V}} C(e, e') \quad (2)$$

This model selects the best candidate event in a given cloze test according to the following score:

$$\hat{e} = \arg\max_{e \in \mathcal{V}} \sum_{i=1}^{L} pmi(e, e_i) \quad (3)$$

We tune this model with an option to apply a modified version of discounting for PMI from Pantel and Ravichandran (2004).

Ordered PMI (OP) This model is a slight variation on Unordered PMI introduced by Jans et al. (2012). The only distinction is that $C(e_1, e_2)$ is treated as an asymmetric count, sensitive to the order in which e_1 and e_2 occur within a chain.

Bigram Probability (BG) Another variant introduced by Jans et al. (2012), the "bigram probability" model uses conditional probabilities rather than PMI to compute scores. In a cloze test, this model selects the following event:

$$\hat{e} = \arg\max_{e \in \mathcal{V}} \prod_{i=1}^{k} p(e|e_i) \prod_{i=k+1}^{L} p(e_i|e) \quad (4)$$

where $p(e_2|e_1) = \frac{C(e_1, e_2)}{C(e_1, *)}$ and $C(e_1, e_2)$ is asymmetric. We tune this model with an option to perform absolute discounting. Note that this model is not a bigram model in the typical language modeling sense.

Len	UNI	UOP	OP	BG	LBL2	LBL4	Tests
2	490	1887	2363	1613	**369**	371	5668
3	452	1271	1752	1009	**330**	334	2793
4	323	806	1027	502	**229**	232	1616
5	364	735	937	442	254	**243**	1330
6	347	666	891	483	257	**249**	942
7	330	629	838	468	241	**237**	630
8	259	466	510	278	208	**201**	512
9	299	610	639	348	198	**195**	396
10+	331	472	397	277	240	**229**	3949
ALL	400	1115	1382	868	294	**292**	17836

(a) Average Rank

Len	UNI	UOP	OP	BG	LBL2	LBL4	Tests
2	.148	.053	.077	.149	**.205**	.204	5668
3	.179	.043	.065	.164	**.217**	.215	2793
4	.226	.042	.064	.195	**.253**	.253	1616
5	.225	.049	.076	.213	.261	**.266**	1330
6	.213	.054	.079	.214	.254	**.263**	942
7	.213	.061	.092	.215	.243	**.247**	630
8	.235	.063	.091	.244	.268	**.278**	512
9	.259	.058	.107	.252	**.280**	.278	396
10+	.191	.082	.113	.193	.198	**.205**	3949
ALL	.186	.057	.083	.181	.221	**.223**	17836

(b) Mean Reciprocal Rank (MRR)

Len	UNI	UOP	OP	BG	LBL2	LBL4	Tests
2	23.9	09.4	11.9	23.8	34.0	**34.1**	5668
3	28.8	08.2	11.1	28.0	**36.3**	35.6	2793
4	33.9	07.7	14.4	32.2	**38.7**	**38.7**	1616
5	33.4	10.1	18.7	34.0	39.6	**40.3**	1330
6	34.8	10.9	22.2	36.8	40.5	**41.9**	942
7	32.5	12.2	24.0	34.9	**39.4**	39.2	630
8	36.7	13.7	21.7	38.7	41.6	**43.2**	512
9	37.9	15.2	28.5	39.1	41.7	**43.2**	396
10+	31.4	18.5	24.0	32.7	**35.7**	**35.7**	3949
ALL	29.5	11.6	16.8	29.8	36.5	**36.6**	17836

(c) Percent Recall at 10

Len	UNI	UOP	OP	BG	LBL2	LBL4	Tests
2	41.7	16.9	25.5	38.6	**51.2**	51.0	5668
3	46.8	20.2	30.2	45.0	**54.8**	54.0	2793
4	53.8	25.3	37.8	54.0	59.0	**60.0**	1616
5	52.5	29.9	40.5	54.3	59.1	**61.1**	1330
6	53.9	33.2	40.7	55.2	60.6	**61.7**	942
7	51.8	34.3	42.7	56.5	61.6	**63.8**	630
8	58.2	42.2	47.7	61.3	**67.2**	67.0	512
9	58.1	42.2	47.7	60.1	66.2	**67.0**	396
10+	49.9	47.4	50.1	54.2	58.4	**59.8**	3949
ALL	48.0	28.6	36.4	48.3	56.3	**56.8**	17836

(d) Percent Recall at 50

Table 1: Narrative cloze results bucketed by chain length for each model and scoring metric with best results in bold. The models are Unigram Model (UNI), Unordered PMI (UOP), Ordered PMI (OP), Bigram Probability Model (BG), Log-Bilinear Model N=2 (LBL2), Log-Bilinear Model N=4 (LBL4)

Skip N-gram We tune the previous three models (UOP, OP, and BG) with the skip n-gram counting methods introduced by Jans et al. (2012) for this task, varying the ways in which the counts, $C(e_1, e_2)$, are collected. Using skip-n counting, $C(e_1, e_2)$ is incremented every time e_1 and e_2 co-occur within a window of size n. We experiment with **skip-0** (consecutive events only), **skip-3** (window size 3), and **skip-all** (entire chain length) settings.

For each of the four narrative cloze scoring metrics we report on (average rank, mean reciprocal rank, recall at 10, and recall at 50), we tune the Unordered PMI, Ordered PMI, and Bigram Probability models over the following parameter space: $\{$**skip-0**, **skip-3**, **skip-all**$\}$ $\times$ $\{$**discount**, **no-discount**$\}$ $\times$ $\{$**T=4**, **T=10**, **T=20**$\}$, where **T** is a pairwise count threshold.

3.2 A Discriminative Method

Log-Bilinear Language Model (LBL) The Log-Bilinear language model is a language model that was introduced by Mnih and Hinton (2007). Like other language models, the LBL produces a probability distribution over the next possible word given a sequence of N previously observed words. N is a hyper-parameter that determines the size of the context used for computing the probabilities. While many variants of the LBL have been proposed since its introduction, we use the simple variant described below.

Formally, we associate one context vector $\mathbf{c}_e \in \mathbb{R}^d$, one bias parameter $b_e \in \mathbb{R}$, and one target vector $\mathbf{t}_e \in \mathbb{R}^d$ to each narrative event $e \in \mathcal{V} \cup \{$ UNK, BOS, EOS $\}$. $\mathcal{V}$ is the vocabulary of events and BOS, EOS, and UNK are the beginning-of-sequence, end-of-sequence, and out-of-vocabulary symbols, respectively. The probability of an event e that appears after a sequence $s = [s_1, s_2, \ldots, s_N]$ of context words is defined as:

$$p(e|s) = \frac{\exp(\mathbf{t}_e^\intercal \hat{\mathbf{t}}_s + b_e)}{\sum_{e' \in \mathcal{V} \cup \{ \text{ UNK, EOS } \}} \exp(\mathbf{t}_{e'}^\intercal \hat{\mathbf{t}}_s + b_{e'})}$$

$$\text{where } \hat{\mathbf{t}}_s = \sum_{j=1}^{N} \mathbf{m}_j \odot \mathbf{c}_{s_j}$$

The $\odot$ operator performs element-wise multiplication of two vectors. The parameters that are optimized during training are $\mathbf{m}_j \; \forall j \in [1, \ldots, N]$ and $\mathbf{c}_e, \mathbf{t}_e \; \forall e \in \mathcal{V} \cup \{$ UNK, BOS, EOS $\}$. To calculate the log-probability of a sequence of narrative events $E = (e_1, \ldots, e_L)$ we compute:

$$l(S) = \left(\sum_{i=1}^{n} \log(p(e_i | f_E(e_i))) \right) + \log(p(\text{EOS} | f_E(\text{EOS}))) \quad (5)$$

Here f_E is a function that returns the sequence of N words that precede the event e_i in the se-

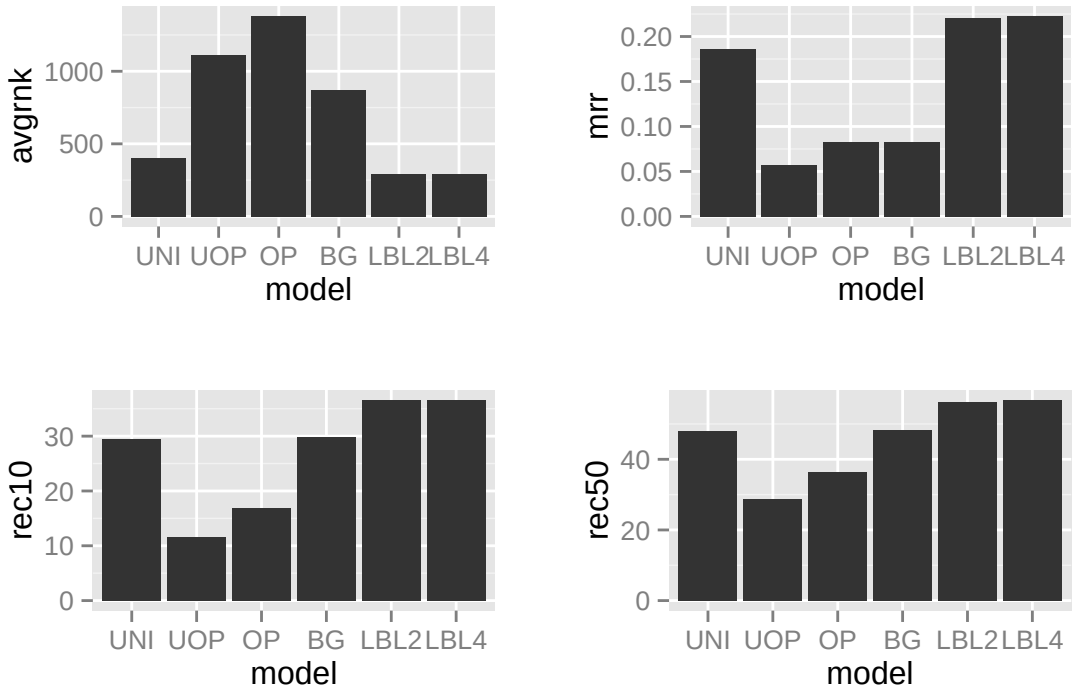

Figure 1: Narrative cloze results over all chain lengths. Unigram Model (UNI), Unordered PMI Model (UOP), Ordered PMI Model (OP), Bigram Probability Model (BG), Log-Bilinear Model with context size 2 or 4 (LBL2, LBL4). Average Rank (avgrnk), Mean Reciprocal Rank (mrr), % Recall at 10 (rec10), % Recall at 50 (rec50).

quence E' made by prepending N BOS tokens and appending a single EOS token to E.

The LBL models are trained by minimizing the objective described in Equation 5 for all the sequences in the training corpus. We used the OxLM toolkit (Paul et al., 2014) which internally uses Noise-Contrastive Estimation (NCE) (Gutmann and Hyvärinen, 2010) and processor parallelization for speeding up the training. For this task, we train LBL models with $N = 2$ (LBL2) and $N = 4$ (LBL4). In our experiments, increasing context size to $N = 6$ did not significantly improve (or degrade) performance.

4 Experimental Results

Table 1 shows the results of 17,836 narrative cloze tests (derived from 5,000 held-out test chains), with results bucketed by chain length. Performance is reported on four metrics: average rank, mean reciprocal rank, recall at 10, and recall at 50.

For each of the four metrics, the best overall performance is achieved by one of the two LBL models (context size 2 or 4); the LBL models also achieve the best performance on every chain length. Not only are the gains achieved by the discriminative LBL consistent across metrics and chain length, they are large. For average rank, the LBL achieves a 27.0% relative improvement over the best non-discriminative model; for mean reciprocal rank, a 19.9% improvement; for recall at 10, a 22.8% improvement; and for recall at 50, a 17.6% improvement. (See Figure 1.) Furthermore, note that both PMI models and the Bigram model have been individually tuned for each metric, while the LBL models have not. (The two LBL models are tuned only for overall perplexity on the development set.)

All models trend toward improved performance on longer chains. Because the unigram model also improves with chain length, it appears that longer chains contain more frequent events and are thus easier to predict. However, LBL performance is also likely improving on longer chains because of additional contextual information, as is evident from LBL4's slight relative gains over LBL2 on longer chains.

5 Conclusion

Pointwise mutual information and other related count-based techniques have been used widely to identify semantically similar words (Church and Hanks, 1990; Lin and Pantel, 2001; Tur-

ney and Pantel, 2010), so it is natural that these techniques have also been applied to the task of script induction. Qualitatively, PMI often identifies intuitively compelling matches; among the top 15 events to share a high PMI with $(eat, nsubj)$ under the Unordered PMI model, for example, we find events such as $(overeat, nsubj)$, $(taste, nsubj)$, $(smell, nsubj)$, $(cook, nsubj)$, and $(serve, dobj)$. When evaluated by the narrative cloze test, however, these count-based methods are overshadowed by the performance of a general-purpose discriminative language model.

Our decision to attempt this task with the Log-Bilinear model was motivated by the simple observation that the narrative cloze test is, in reality, a language modeling task. Does the LBL's success on this task mean that work in script induction should abandon traditional count-based methods for discriminative language modeling techniques? Or does it mean that an alternative evaluation metric is required to measure script knowledge? While we believe our results are sufficient to conclude that one of these alternatives is the case, we leave the task of determining which to future research.

Acknowledgments

This work was supported by the Paul Allen Institute for Artificial Intelligence (*Acquisition and Use of Paraphrases in a Knowledge-Rich Setting*), a National Science Foundation Graduate Research Fellowship (Grant No. DGE-1232825), the Johns Hopkins HLTCOE, and DARPA DEFT (FA8750-13-2-001, *Large Scale Paraphrasing for Natural Language Understanding*). We would also like to thank three anonymous reviewers for their feedback. Any opinions expressed in this work are those of the authors.

References

Nathanael Chambers and Dan Jurafsky. 2008. Unsupervised learning of narrative event chains. In *Proceedings of ACL-08: HLT*, pages 789–797, Columbus, Ohio. Association for Computational Linguistics.

Nathanael Chambers and Dan Jurafsky. 2009. Unsupervised learning of narrative schemas and their participants. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 602–610, Suntec, Singapore. Association for Computational Linguistics.

Nathanael Chambers. 2013. Event schema induction with a probabilistic entity-driven model. In *EMNLP*, volume 13, pages 1797–1807.

Jackie Chi Kit Cheung and Gerald Penn. 2013. Probabilistic domain modelling with contextualized distributional semantic vectors. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 392–401. Association for Computational Linguistics.

Jackie Chi Kit Cheung, Hoifung Poon, and Lucy Vanderwende. 2013. Probabilistic frame induction. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 837–846, Atlanta, Georgia, June. Association for Computational Linguistics.

Kenneth Ward Church and Patrick Hanks. 1990. Word association norms, mutual information, and lexicography. *Computational linguistics*, 16(1):22–29.

Marie-Catherine De Marneffe, Bill MacCartney, Christopher D Manning, et al. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC*, volume 6, pages 449–454.

Francis Ferraro, Max Thomas, Matthew R. Gormley, Travis Wolfe, Craig Harman, and Benjamin Van Durme. 2014. Concretely Annotated Corpora. In *4th Workshop on Automated Knowledge Base Construction (AKBC)*.

Lea Frermann, Ivan Titov, and Manfred Pinkal. 2014. A hierarchical bayesian model for unsupervised induction of script knowledge. *EACL 2014*, page 49.

David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2003. English gigaword. *Linguistic Data Consortium, Philadelphia*.

Michael Gutmann and Aapo Hyvärinen. 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *International Conference on Artificial Intelligence and Statistics*, pages 297–304.

Bram Jans, Steven Bethard, Ivan Vulić, and Marie-Francine Moens. 2012. Skip n-grams and ranking functions for predicting script events. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 336–344, Avignon, France. Association for Computational Linguistics.

Dekang Lin and Patrick Pantel. 2001. Dirt - discovery of inference rules from text. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 323–328. ACM.

Mehdi Manshadi, Reid Swanson, and Andrew S Gordon. 2008. Learning a probabilistic model of event sequences from internet weblog stories. In *FLAIRS Conference*, pages 159–164.

Andriy Mnih and Geoffrey Hinton. 2007. Three new graphical models for statistical language modelling. In *Proceedings of the 24th international conference on Machine learning*, pages 641–648. ACM.

Ashutosh Modi and Ivan Titov. 2014. Inducing neural models of script knowledge. *CoNLL-2014*, page 49.

Patrick Pantel and Deepak Ravichandran. 2004. Automatically labeling semantic classes. In Daniel Marcu Susan Dumais and Salim Roukos, editors, *HLT-NAACL 2004: Main Proceedings*, pages 321–328, Boston, Massachusetts, USA. Association for Computational Linguistics.

Baltescu Paul, Blunsom Phil, and Hoang Hieu. 2014. Oxlm: A neural language modelling framework for machine translation. *The Prague Bulletin of Mathematical Linguistics*, 102(1):81–92.

Karl Pichotta and Raymond Mooney. 2014. Statistical script learning with multi-argument events. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 220–229, Gothenburg, Sweden. Association for Computational Linguistics.

Michaela Regneri, Alexander Koller, and Manfred Pinkal. 2010. Learning script knowledge with web experiments. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 979–988. Association for Computational Linguistics.

Rachel Rudinger, Vera Demberg, Ashutosh Modi, Benjamin Van Durme, and Manfred Pinkal. 2015. Learning to predict script events from domain-specific text. *Lexical and Computational Semantics (* SEM 2015)*, page 205.

Roger Schank and Robert Abelson. 1977. *Scripts, plans, goals and understanding: An inquiry into human knowledge structures*. Lawrence Erlbaum Associates, Hillsdale, NJ.

Peter D. Turney and Patrick Pantel. 2010. From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research*, 37(1):141–188, January.