Script Induction as Association Rule Mining

Anton Belyy and **Benjamin Van Durme**
Department of Computer Science
Johns Hopkins University
{abel,vandurme}@jhu.edu

Abstract

We show that the count-based Script Induction models of Chambers and Jurafsky (2008) and Jans et al. (2012) can be unified in a general framework of narrative chain likelihood maximization. We provide efficient algorithms based on Association Rule Mining (ARM) and weighted set cover that can discover interesting patterns in the training data and combine them in a reliable and explainable way to predict the missing event. The proposed method, unlike the prior work, does not assume full conditional independence and makes use of higher-order count statistics. We perform the ablation study and conclude that the inductive biases introduced by ARM are conducive to better performance on the narrative cloze test.

1 Introduction

The goal of this paper is to demonstrate how the efforts in Script Induction (SI), up until recently dominated by statistical approaches (Chambers and Jurafsky, 2008; Jans et al., 2012; Pichotta and Mooney, 2014; Rudinger et al., 2015a,b), can be productively framed and extended as a special case of Association Rule Mining (ARM), a well-established problem in Data Mining (Agrawal et al., 1993, 1994; Han et al., 2000).

We start by introducing SI and ARM and then demonstrate a unification under a general chain likelihood maximization framework. We discuss how the existing count-based SI models tackle this maximization problem using naïve Bayes assumptions. We provide an alternative: mining higher-order count statistics using ARM and picking the most reliable rules using the weighted set cover algorithm. We validate the proposed approach and demonstrate improved performance over other count-based approaches. We conclude with a discussion on the implications and potential extensions of the proposed framework.

ARM term	SI term
Transaction t	Narrative chain
Itemset I	Co-occurring events
$\text{sup}(\{i_1, i_2\})$	$C(i_1, i_2)$
$\text{int}(\{a\} \to \{e\})$	$P(a\|e) = \frac{C(a,e)}{C(*,e)}$
$\text{sup}(I), \|I\| > 2$	**Eq. 5**
$\text{int}(A \to \{e\}), \|A\| > 1$	**Eq. 12**

Table 1: Mapping between ARM and Count-based SI terminology. **Bolded** are contributions of this paper. Namely, we make use of *frequent itemsets* and *interesting rules*, or higher-order count statistics that can be efficiently mined and used in the narrative cloze test.

Our intent in this work is not to establish new state of the art results in the area of SI. Rather, our primary contribution is retrospective, drawing a connection between a sub-topic in Computational Linguistics (CL) with a major pre-existing area of Computer Science, i.e., Data Mining. In the case one approached SI through counting co-occurrence statistics, then the existing tools of ARM lead naturally to solutions that had not been previously considered within CL.

2 Background

2.1 Association Rule Mining

ARM is a prevalent problem in Data Mining, introduced by Agrawal et al. (1993). The task is often referred to as *market basket analysis* due to its widespread usage for discovering interesting patterns in consumer purchases. The applicability of ARM extends far beyond this specific scenario, where examples of ARM usage for NLP applications include detecting annotation inconsistencies (Novák and Razímová, 2009), discovering strongly-related events (Shibata and Kurohashi, 2011), adding missing knowledge to the KB

Proceedings of the 1st Joint Workshop on Narrative Understanding, Storylines, and Events, pages 55–62

(Galárraga et al., 2013), as well as understanding clinical narratives (Boytcheva et al., 2017).

ARM aims to extract interesting patterns from a transactional database $\mathcal{D}$. A transaction is a set of *items*, and a non-empty subset of a transaction is called an *itemset*. We define *support* as the number of transactions we observe an itemset I in:

$$\sup(I) = |\{t|t \in \mathcal{D}, I \subseteq t\}|. \quad (1)$$

We say that an itemset I is *frequent*, if its support (on a given database $\mathcal{D}$) exceeeds a user-defined threshold t_{sup}: $\sup(I) \geq t_{sup}$.

A pair of itemsets A, B is called a *rule* if $A \cap B = \varnothing$ and is denoted as $A \rightarrow B$. We say that a rule $A \rightarrow B$ is *interesting* if 1) both A and B are frequent, 2) the *interestingess* of the rule exceeds a user-defined threshold t_{int}: $\text{int}(A \rightarrow B) \geq t_{int}$. The definition of the interestingness function $\text{int}(\cdot)$ is problem-specific.

ARM is thus concerned with:

1. mining frequent itemsets from a transactional database,

2. discovering interesting rules from frequent itemsets.

2.2 Script Induction

The concept of *script knowledge* in AI, along with early knowledge-based methods to learn scripts were introduced by Minsky (1974); Schank and Abelson (1977); Mooney and DeJong (1985).

With the rise of statistical methods, the next generation of algorithms made use of co-occurrence statistics and distributional semantics for script learning (Chambers and Jurafsky, 2008, 2009; Jans et al., 2012; Pichotta and Mooney, 2014). Our primary focus is on drawing connections between ARM and this body of work.

Following Chambers and Jurafsky (2008), we define a *narrative chain* as "a partially ordered set of narrative events that share a common actor", where the partial ordering typically represents temporal or causal order of events, and a *narrative event* is "a tuple of an event and its participants, represented as typed dependencies". Formally, we define a narrative event $e := (v, d)$, where v is a verb lemma, and d is a dependency arc between the verb and the common actor (dobj or nsubj). An example of a narrative chain is given in Figure 1.

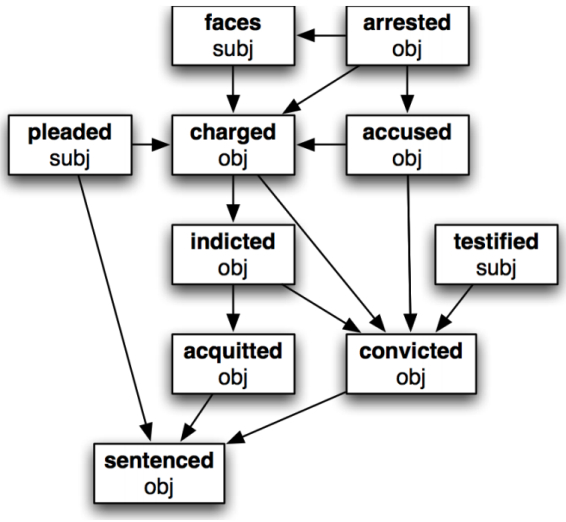

Figure 1: Graphical depiction of a Prosecution narrative chain learned by Chambers and Jurafsky (2008). Arrows indicate partial temporal ordering.

SI is thus concerned with:

1. automatic mining of commonly co-occurring sets of narrative events from text,

2. partially ordering those sets.

The *narrative cloze* test (Chambers and Jurafsky, 2008) is a standard extrinsic evaluation procedure for Task 1 of SI. In this test, a sequence of narrative events is automatically extracted from a document, and one event is removed; the goal is to predict the missing event. Formally, given an incomplete narrative chain $\{e_1, e_2, \ldots, e_L\}$ and an *insertion point* $k \in [L]$, we would like to predict the most likely missing event $\hat{e}$ to complete the chain:

$$\{e_1, e_2, \ldots, e_k, \hat{e}, e_{k+1}, \ldots e_L\}.$$

Although the recent work in SI (Rudinger et al., 2015b; Pichotta and Mooney, 2016; Peng and Roth, 2016; Weber et al., 2018) has focused on a Language Modeling (LM) approach for the narrative cloze test, it is fundamentally different from ARM in that it makes use of the total ordering of events and is thus incomparable to ARM, which does not assume any ordering of events within a chain.

In the next section, we survey two of the most influential count-based SI models, showing how each of them is related to ARM.

3 Count-based Script Induction

3.1 Unordered PMI model

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

$$\text{pmi}(e_1, e_2) \propto \log \frac{C(e_1, e_2)}{C(e_1, *)C(*, e_2)}, \quad (2)$$

where $C(e_1, e_2)$ is defined as the number of narrative chains where e_1 and e_2 both occurred and

$$C(e, *) := \sum_{e' \in E} C(e, e'),$$

where E is a fixed vocabulary of narrative events.

The model selects the missing event $\hat{e}$ in the narrative cloze test according to the score

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

assuming that the missing event $\hat{e}$ is inserted at the end of the existing chain ($k = L$).

From (2) and (3) we observe that

$$\begin{aligned}
\hat{e} &= \arg\max_{e \in E} \sum_{i=1}^{L} \text{pmi}(e, e_i) \\
&= \arg\max_{e \in E} \sum_{i=1}^{L} \log \frac{C(e, e_i)}{C(e, *)C(*, e_i)} \\
&= \arg\max_{e \in E} \log \prod_{i=1}^{L} \frac{C(e, e_i)}{C(e, *)} \\
&= \arg\max_{e \in E} \log \prod_{i=1}^{L} P(e_i|e) \\
&= \arg\max_{e \in E} \prod_{i=1}^{L} P(e_i|e). \quad (4)
\end{aligned}$$

One way to interpret Eq. 4 is to say that it was obtained from the following model with the *naïve Bayes assumption*:

$$\hat{e} = \arg\max_{e \in E} P(e_1, e_2, \ldots, e_L|e). \quad (5)$$

Importantly, in the above equation, no assumptions are made about the order in which events $e_1, \ldots, e_L$ happened and we treat the narrative chain as a document, where individual events are features (the "bag of events" assumption).

3.2 Bigram Probability model

The bigram probability model was proposed by Jans et al. (2012) and was also used by Pichotta and Mooney (2014). It utilizes positional information between co-occurring events. It selects the missing event $\hat{e}$ according to the score

$$\hat{e} = \arg\max_{e \in E} \left(\prod_{i=1}^{k} P(e|e_i) \right) \cdot \left(\prod_{i=k+1}^{L} P(e_i|e) \right),$$

where k is the insertion point of the missing event $\hat{e}$, $P(e_2|e_1) = \frac{C_{ord}(e_1, e_2)}{C_{ord}(e_1, *)}$, and counts $C_{ord}(e_1, e_2)$ are ordered, e.g. $C_{ord}(e_1, e_2) \neq C_{ord}(e_2, e_1)$.

Similarly to the Unordered PMI model, we can relax the conditional independence assumption. However, to apply Bayes' theorem, we would need (e_1, e_2) and (e_2, e_1) to be the same events in the outcome space, thus we have to assume unordered counts: $C(e_1, e_2) = C_{ord}(e_1, e_2) + C_{ord}(e_2, e_1)$. Proceeding with this, we get:

$$\begin{aligned}
\hat{e} &= \arg\max_{e \in E} \left(\prod_{i=1}^{k} P(e|e_i) \right) \cdot \left(\prod_{i=k+1}^{L} P(e_i|e) \right) \\
&= \arg\max_{e \in E} \left(\prod_{i=1}^{L} P(e_i|e) \right) \cdot (P(e))^k \\
&= \arg\max_{e \in E} \log \left(\left(\prod_{i=1}^{L} P(e_i|e) \right) \cdot (P(e))^k \right) \\
&= \arg\max_{e \in E} \log P(e_1, \ldots, e_L|e) + k \cdot \log P(e), \quad (6)
\end{aligned}$$

where the last equality is obtained by relaxing the full conditional independence assumption (similar to Eq. 5). It follows that the Bigram Probability model with unordered counts is exactly the Unordered PMI model augmented with the prior probability of a missing event multiplied by its position in a chain. Additionally, note that if $k = 1$, this model is equivalent to maximizing the posterior probability of a missing event (rather than the likelihood of a narrative chain in Eq. 5):

$$\begin{aligned}
\hat{e} &= \arg\max_{e \in E} \log P(e_1, \ldots, e_L|e) + \log P(e) \\
&= \arg\max_{e \in E} \log \left(P(e_1, \ldots, e_L|e) \cdot P(e) \right) \\
&= \arg\max_{e \in E} \log P(e|e_1, \ldots, e_L). \quad (7)
\end{aligned}$$

Similar to Eq. 5, we view the narrative chain $e_1, \ldots, e_n$ as a set, and thus Eq. 6 is not a language model in the traditional NLP sense.

4 SI as ARM

The models defined by Eqs. 5, 6, and 7 are hard to compute directly: without simplifying assumptions, they would require huge number of parameters and large training sets (Jurafsky and Martin, 2019). A common approach in the existing Count-based SI work is to assume full conditional independence. A viable and less restrictive alternative, as we show in this section, is estimating higher-order count statistics via mining association rules (Section 4.1) and combining the most confident rules to predict the missing event with a simple weighted set cover algorithm (Section 4.2).

More formally, during the training phase, we would like to populate the set of interesting rules $\mathbb{S} = \{S \rightarrow \{e\}\}$, whose antecedents are sub-sets of the event space $S \subset E$, and consequents are single events e, $e \notin S$. We denote as $\mathbb{S}_e$ all the rules with the same consequent event e.

During the test phase, where we have an incomplete narrative chain $\{e_1, e_2, \ldots, e_L\}$ and want to predict a missing event, we will use rules from $\mathbb{S}_e$ to efficiently decompose $P(e_1, e_2, \ldots, e_L|e)$ into $P(S_1|e) \cdot P(S_2|e) \cdot \ldots \cdot P(S_t|e)$ for each candidate event e. Naturally, this means selecting a set of rules whose antecedents $\{S_1, S_2, \ldots, S_t\}$ (we call this set a candidate cover) are pairwise disjoint ($S_i \cap S_j = \varnothing \ \forall i, j \in [t]$), and cover the event chain fully ($S_1 \cup S_2 \cup \ldots \cup S_t = \{e_1, e_2, \ldots, e_L\}$).

To quantify the goodness of the decomposition, we define a score function for a candidate cover $\{S_1, \ldots, S_t\}$ and a candidate event e as follows:

$$\text{score}(S_1, S_2, \ldots, S_t; e) = \prod_{i=1}^{t} P(S_i|e). \quad (8)$$

For each candidate event e, we select the best candidate cover $\hat{S}_e$ according to the score function:

$$\hat{S}_e = \underset{\{S'_1,\ldots,S'_t\} \in \mathbb{S}_e}{\arg\max} \ \text{score}(S'_1, \ldots, S'_t; e). \quad (9)$$

This allows to rewrite Eq. 5 as:

$$\hat{e} = \underset{e \in E}{\arg\max} \ \hat{S}_e. \quad (10)$$

In Section 4.1, we explain how the set of rules $\mathbb{S}$ is populated from the SI training corpus. In Section 4.2, we provide a randomized algorithm that solves problem 9 with a provably bounded error.

4.1 Mining interesting rules

As discussed in Section 2.1, in order to discover the set of interesting rules $\mathbb{S}$, we need to mine frequent itemsets first. This can be achieved by any frequent itemset mining algorithm, such as Apriori (Agrawal et al., 1994), Eclat (Zaki, 2000), or FP-growth (Han et al., 2000).

Next, for the rule mining step we define an interestingness function $\text{int}(S \rightarrow E)$ over a rule $S \rightarrow E$:

$$\text{int}(S \rightarrow E) = \frac{\sup(S \cup E)}{\sum_{S'} \sup(S' \cup E)}, \quad (11)$$

where S' ranges over all itemsets of size $|S|$ and is disjoint with E.

Note that $\text{int}(S \rightarrow E)$ provides a maximum likelihood estimate of $P(S|E)$ for the probability space defined over sets of events, and $\sup(\cdot)$ is a generalization of the previously defined $C(\cdot, \cdot)$ for event sets of size larger than two.

The denominator of (11) requires calculating the support over exponentially many itemsets. We can instead use the following simpler formula:

$$\text{wsup}_\text{k}(I) = \sum_{t \in \mathcal{D}} \binom{|t| - |I|}{k} \cdot \mathbf{1}_{I \subseteq t},$$

where $\mathcal{D}$ is a transactional database of narrative event chains.

Lemma 1. $\sum_{S'} \sup(S' \cup I) = \text{wsup}_\text{k}(I)$, *where* S' *ranges over all itemsets of size* k*, disjoint with* I*.*

Proof. By definition of support from Eq. 1,

$$\begin{aligned}
&\sum_{S'} \sup(S' \cup I) \\
&= \sum_{S'} |\{t | t \in \mathcal{D}, S' \cup I \subseteq t\}| \\
&= \sum_{S'} |\{t | t \in \mathcal{D}, (S' \subseteq t/I) \ \wedge \ (I \subseteq t)\}| \\
&= \sum_{t \in \mathcal{D}} \mathbf{1}_{I \subseteq t} \cdot \sum_{S'} \mathbf{1}_{S' \subseteq t/I} \\
&= \sum_{t \in \mathcal{D}} \mathbf{1}_{I \subseteq t} \cdot \binom{|t| - |I|}{k} \\
&= \text{wsup}_\text{k}(I).
\end{aligned}$$

□

Algorithm 1 Mining interesting rules

Input: A set of high-support itemsets $\mathbb{I}$,
Output: A set of interesting rules $\mathbb{S}$.
Initialization: $\mathbb{S} = \varnothing$
for $I \in \mathbb{I}$ **do**
  for $e \in I$ **do**
    $S = I \backslash \{e\}$
    if $\mathrm{int}(S \to \{e\}) \geq t_{int}$ **then**
      $\mathbb{S} = \mathbb{S} \cup \{S \to \{e\}\}$
    end if
  end for
end for
Return $\mathbb{S}$.

Our intent is to use the above interestingness function to score rules from $\mathbb{S}$ that have a single event as a consequent, and thus Eq. 11 can be further simplified:

$$\mathrm{int}(S \to \{e\}) = \frac{\sup(S \cup \{e\})}{\mathrm{wsup}_{|S|}(\{e\})}. \qquad (12)$$

Assuming that for each rule $S \to \{e\}$ the antecedent is bounded in size and small, we can precompute $\mathrm{wsup_k}(\{e\})$ for each $e \in E$ and each $k \in [|S|]$ in a single pass over the database. Note also that $\mathrm{wsup}_0(I) = \sup(I)$ and thus $\mathrm{wsup}_k(\cdot)$ is a generalization of support (1).

Given an interestingness function, we can now proceed to mine interesting rules over frequent event sets. The rule mining process is shown in Algorithm 1.

After a set of interesting rules $\mathbb{S}$ is populated, we can perform test-time inference on new narrative chains with Eqs. 9 and 10. To facilitate this, we frame the inference problem as the weighted set cover problem. The latter was known to be NP-complete by Karp (1972), but there is a simple greedy algorithm by Chvatal (1979) that provides an approximate solution. To make it applicable to the search problem 9, we will run it (for each candidate event e) on the set $\mathbb{S}$, mined by Algorithm 1, with the following weight function:

$$\begin{aligned} w(S) &= -\ln \mathrm{int}(S \to \{e\}) \\ &= -\ln P(S|e). \end{aligned}$$

The following lemma provides a lower bound on the score of the candidate cover obtained by Algorithm 2.

Algorithm 2 Greedy weighted set cover

Input:
- A set of interesting rules $\mathbb{S}_e$,
- A narrative chain $e_1, e_2, \ldots, e_L$.

Output: An approximation (within a $O(\log L)$ factor) of the best cover $\{S_1, S_2, \ldots, S_t\}$.
Initialization:
  $U_0 = \{e_1, e_2, \ldots, e_L\}$
  $t = 0$
while $U_t \neq \varnothing$ **do**
  $t = t + 1$
  $S_t = \arg\max_{S' \in \mathbb{S}_e} \frac{|S' \cap U_{t-1}|}{w(S')}$
  $U_t = U_{t-1} \backslash S_t$
end while
Return $\{S_1, S_2, \ldots, S_t\}$.

4.2 Score estimation via weighted set cover

Lemma 2. *Algorithm 2 finds a candidate cover $\{S_1, \ldots, S_t\}$ for a narrative chain $\{e_1, \ldots, e_L\}$ and a candidate event e, such that $\mathrm{score}(S_1, \ldots, S_t; e) \geq OPT^{\ln L + 1}$, where OPT is the score of the best candidate cover $\hat{S}_e$.*

Proof. Chvatal (1979) showed that Algorithm 2 finds a weighted set cover $\{S_1, \ldots, S_t\}$, such that $OPT_{cover} \leq \sum_{i=1}^{t} w(S_i) \leq (\ln L + 1) OPT_{cover}$. Since the weight $w(\cdot)$ is a negative log probability:

$$\begin{aligned} \sum_{i=1}^{t} w(S_i) &= -\sum_{i=1}^{t} \ln P(S_i|e) \\ &= -\ln \mathrm{score}(S_1, \ldots, S_t; e) \\ &\leq (\ln L + 1) OPT_{cover}. \end{aligned}$$

By exponentiating left and right-hand sides and noting that $OPT = e^{-OPT_{cover}}$ (by definition of the weight and score functions), we get:

$$\begin{aligned} \mathrm{score}(S_1, \ldots, S_t; e) &\geq e^{-(\ln L + 1) OPT_{cover}} \\ &\geq OPT^{\ln L + 1}. \end{aligned}$$

□

If we group the rules $S \to \{e\}$ by the consequent event and order by $\frac{|S|}{w(S)}$ within each group, then step 8 in Algorithm 2 becomes equivalent to iterating over ordered rules in $\mathbb{S}_e$. The overall running time to score the candidate event e is $O(L + |\mathbb{S}_e|)$.

Additionally, $O(\sum_{e \in E} |\mathbb{S}_e| \log |\mathbb{S}_e|)$ preprocessing time is needed to group and order the rules in $\mathbb{S}$.

5 Experiments

5.1 Dataset

We perform experiments on the New York Times part of the Annotated Gigaword dataset by (Napoles et al., 2012). Chains of narrative events are constructed from the (automatically generated) in-document coreference chains: from each document in the dataset, we extract all coreference chains and retain the longest one, with length two or greater. We also filter top-10 occurring events which are mostly reporting verbs such as "say" and "think" and convey little meaning for SI task.

Training is done on the 1994–2006 portion (1.3M chains with 8.7M narrative events), development set is a subset of 2007–2008 portion (10K chains with 62K narrative events), and test set is a subset of 2009–2010 portion (5K chains with 31K narrative events).

5.2 Model setup

We implement and compare models described in Sections 3 and 4, along with a strong baseline Unigram model by Pichotta and Mooney (2014), which ranks each event according to its unigram probability in the training corpus.

For testing the Unordered PMI and Bigram models, we use implementations from the Nachos software package (Rudinger et al., 2015a). Both models are tuned to use skip-grams (as defined by Jans et al. (2012)) of size up to the chain length, which allows to reduce data sparsity and is consistent with the set of rules (of size two) generated by ARM.

ARM consists of 1) mining frequent itemsets and 2) obtaining interesting rules from those itemsets. For frequent itemsets mining, we use the FP-growth algorithm by Han et al. (2000) with a $t_{sup} = 100$ threshold. For rule mining, we implement Algorithm 1. Since the rule mining step is much less computationally intensive than itemset mining, we can use a more permissive $t_{int} = 10^{-5}$ threshold. We use the same thresholds across all models by applying the following back-off strategy in the Unordered PMI and Bigram models:

$$P(e_i|e) = \begin{cases} \frac{C(e_i,e)}{C(*,e)} & \text{if } C(e_i,e) \geq t_{ARM}, \\ \frac{1}{|E|+1} & \text{otherwise,} \end{cases}$$

where $t_{ARM} = \max\left(t_{sup}, C(*,e) \cdot t_{int}\right)$.

Ablation	R@50
ARM (posterior, (7))	0.36
ARM (bigram, (6))	0.34
ARM (UOP, (5))	0.30
ARM (UOP, binary rules only, (4))	0.28
UOP (both t_{sup} & t_{int} pruning, (4))	0.28
UOP (only t_{sup} pruning, (4))	0.28
UOP (only t_{int} pruning, (4))	0.03
UOP (no t_{int} & t_{sup} pruning, (4))	0.03

Table 2: Ablation experiments on NYTimes dev set. R@50 stands for Recall@50.

6 Experimental Results

We perform two experiments, comparing existing count-based SI models with three variants of the proposed ARM model. The performance is measured using Recall@50 and Mean Reciprocal Rank.

In the first experiment, we establish that the count-based pruning, introduced by ARM support and interestingness thresholds (t_{sup} and t_{int}, respectively) for reducing the search space during rule mining, does contribute to better performance on the narrative cloze test. We also validate empirically that the ARM model with binary (of size two) rules is equivalent to the UOP model by Chambers and Jurafsky (2008). Finally, we compare variants of the ARM model, which vary in a way of incorporating a prior probability of the missing event. We conclude that the posterior ARM model, given by Eq. 7, achieves the best performance. The results of this experiment are outlined in Table 2.

In the second experiment, we compare the best-scoring ARM model and other baseline models on 5,000 test chains. We achieve 5% relative improvement for Mean Reciprocal Rank (MRR) and 10% for Recall@50, which can be attributed to using higher-order count statistics and the selection of the prior for the missing event. The scalability of both rule mining and inference algorithms suggests that the performance may be further improved as the training corpus size grows and more reliable higher-order statistics become available. The results of this experiment are shown in Table 3.

Similar to Rudinger et al. (2015b), we also note that all models tend to improve their performance on longer chains, which may be explained by the availability of additional contextual information.

Len	UNI	UOP	BG	ARM	Tests
1	0.050	0.034	0.047	**0.060**	642
2	0.044	0.040	0.060	**0.061**	764
3	0.045	0.046	0.058	**0.063**	659
4	0.053	0.047	0.065	**0.070**	568
5	0.068	0.059	**0.087**	0.076	423
6	0.067	0.048	**0.074**	**0.074**	324
7	0.051	0.050	0.056	**0.063**	288
8	0.074	0.054	**0.088**	0.075	205
9	0.048	0.048	**0.068**	0.066	179
10+	0.044	0.064	0.062	**0.068**	948
ALL	0.051	0.049	0.063	**0.066**	5000

(a) Mean Reciprocal Rank (MRR)

Len	UNI	UOP	BG	ARM	Tests
1	0.34	0.17	0.24	**0.36**	642
2	0.28	0.22	0.28	**0.32**	764
3	0.30	0.28	0.32	**0.34**	659
4	0.32	0.29	0.34	**0.36**	568
5	0.33	0.30	0.35	**0.36**	423
6	0.33	0.33	0.36	**0.37**	324
7	0.30	0.32	0.33	**0.35**	288
8	0.33	0.34	0.36	**0.39**	205
9	0.35	0.35	**0.37**	**0.37**	179
10+	0.32	0.36	0.35	**0.36**	948
ALL	0.32	0.29	0.32	**0.35**	5000

(b) Percent Recall at 50

Table 3: Narrative cloze results bucketed by incomplete narrative chain length for each model and scoring function with best results in bold. The models are Unigram Model (UNI), Unordered PMI (UOP), Bigram Probability Model (BG), and proposed ARM model (ARM).

7 Conclusion

Our decision to approach count-based SI as ARM was motivated by a previously under-explored similarity of these well-established areas, which we outlined in this paper. Drawing similarities from the existing work on Classification using Association Rules (CAR) (Liu et al., 1998; Thabtah et al., 2005), we proposed a scoring function that uses ARM-based count statistics to reliably predict the missing event in the narrative cloze test.

One downside of relying solely on count-based statistics is the low support of longer itemsets due to data sparsity. On the other hand, modern contextual encoders (Devlin et al., 2018) mitigate this via parameter sharing. Reliably mining rules whose support and interestingness are based on both counts and properties of dense embeddings can be a promising direction of future work.

Acknowledgments

This work was supported by DARPA KAIROS. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes. The views and conclusions contained in this publication are those of the authors and should not be interpreted as representing official policies or endorsement.

We would like to thank Suzanna Sia, Kenton Murray, Noah Weber, and three anonymous reviewers for their feedback.

References

Rakesh Agrawal, Tomasz Imieliński, and Arun Swami. 1993. Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM SIGMOD international conference on Management of data*, pages 207–216.

Rakesh Agrawal, Ramakrishnan Srikant, et al. 1994. Fast algorithms for mining association rules. In *Proc. 20th int. conf. very large data bases, VLDB*, volume 1215, pages 487–499.

Svetla Boytcheva, Ivelina Nikolova, and Galia Angelova. 2017. Mining association rules from clinical narratives. In *RANLP*, pages 130–138.

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

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: Volume 2-Volume 2*, pages 602–610. Association for Computational Linguistics.

Vasek Chvatal. 1979. A greedy heuristic for the set-covering problem. *Mathematics of operations research*, 4(3):233–235.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Luis Antonio Galárraga, Christina Teflioudi, Katja Hose, and Fabian Suchanek. 2013. Amie: association rule mining under incomplete evidence in ontological knowledge bases. In *Proceedings of the 22nd*

international conference on World Wide Web, pages 413–422.

Jiawei Han, Jian Pei, and Yiwen Yin. 2000. Mining frequent patterns without candidate generation. *ACM sigmod record*, 29(2):1–12.

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. Association for Computational Linguistics.

Dan Jurafsky and James H Martin. 2019. Speech and language processing (draft). *Chapter 4: Naive Bayes and Sentiment Classification (Draft of October 2, 2019)*.

Richard M Karp. 1972. Reducibility among combinatorial problems. In *Complexity of computer computations*, pages 85–103. Springer.

Bing Liu, Wynne Hsu, Yiming Ma, et al. 1998. Integrating classification and association rule mining. In *KDD*, volume 98, pages 80–86.

Marvin Minsky. 1974. A framework for representing knowledge. mit-ai laboratory memo 306. *Massachusetts Institute of Technology*.

Raymond J Mooney and Gerald DeJong. 1985. Learning schemata for natural language processing. In *IJCAI*, pages 681–687.

Courtney Napoles, Matthew Gormley, and Benjamin Van Durme. 2012. Annotated gigaword. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction*, pages 95–100. Association for Computational Linguistics.

Václav Novák and Magda Razímová. 2009. Unsupervised detection of annotation inconsistencies using apriori algorithm. In *Proceedings of the Third Linguistic Annotation Workshop*, pages 138–141. Association for Computational Linguistics.

Haoruo Peng and Dan Roth. 2016. Two discourse driven language models for semantics. *arXiv preprint arXiv:1606.05679*.

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.

Karl Pichotta and Raymond J Mooney. 2016. Learning statistical scripts with lstm recurrent neural networks. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Rachel Rudinger, Vera Demberg, Ashutosh Modi, Benjamin Van Durme, and Manfred Pinkal. 2015a. Learning to predict script events from domain-specific text. In *Proceedings of the Fourth Joint Conference on Lexical and Computational Semantics*, pages 205–210.

Rachel Rudinger, Pushpendre Rastogi, Francis Ferraro, and Benjamin Van Durme. 2015b. Script induction as language modeling. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1681–1686.

Roger Schank and Robert Abelson. 1977. Scripts, plans, goals and understanding: An inquiry into human knowledge structures.

Tomohide Shibata and Sadao Kurohashi. 2011. Acquiring strongly-related events using predicate-argument co-occurring statistics and case frames. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 1028–1036.

Fadi Thabtah, Peter Cowling, and Yonghong Peng. 2005. Mcar: multi-class classification based on association rule. In *The 3rd ACS/IEEE International Conference onComputer Systems and Applications, 2005.*, page 33. IEEE.

Noah Weber, Leena Shekhar, Niranjan Balasubramanian, and Nathanael Chambers. 2018. Hierarchical quantized representations for script generation. *arXiv preprint arXiv:1808.09542*.

Mohammed Javeed Zaki. 2000. Scalable algorithms for association mining. *IEEE transactions on knowledge and data engineering*, 12(3):372–390.