A Hierarchical Bayesian Model for Unsupervised Induction of Script Knowledge

Lea Frermann[1] l.frermann@ed.ac.uk
Ivan Titov[2] titov@uva.nl
Manfred Pinkal[3] pinkal@coli.uni-sb.de

[1] ILCC, School of Informatics, University of Edinburgh, United Kingdom
[2] ILLC, University of Amsterdam, Netherlands
[3] Department of Computational Linguistics, Saarland University, Germany

Abstract

Scripts representing common sense knowledge about stereotyped sequences of events have been shown to be a valuable resource for NLP applications. We present a hierarchical Bayesian model for unsupervised learning of script knowledge from crowdsourced descriptions of human activities. Events and constraints on event ordering are induced jointly in one unified framework. We use a statistical model over permutations which captures event ordering constraints in a more flexible way than previous approaches. In order to alleviate the sparsity problem caused by using relatively small datasets, we incorporate in our hierarchical model an informed prior on word distributions. The resulting model substantially outperforms a state-of-the-art method on the event ordering task.

1 Introduction

A *script* is a "predetermined, stereotyped sequence of actions that define a well-known situation" (Schank and Abelson, 1975). While humans acquire such common-sense knowledge over their lifetime, it constitutes a bottleneck for many NLP systems. Effective question answering and summarization are impossible without a form of story understanding, which in turn has been shown to benefit from access to databases of script knowledge (Mueller, 2004; Miikkulainen, 1995). Knowledge about the typical ordering of events can further help assessing document coherence and generating coherent text. Here, we present a general method for acquiring data bases of script knowledge.

Our work may be regarded as complementary to existing work on learning script knowledge from natural text (cf. (Chambers and Jurafsky, 2008)), as not all types of scripts are elaborated in natural text – being left implicit because of assumed readers' world knowledge. Our model, operating on data obtained in a cheap way by crowdsourcing, is applicable to any kind of script and can fill this gap. We follow work in inducing script knowledge from explicit instantiations of scripts, so-called *event sequence descriptions* (ESDs) (Regneri et al., 2010). Our data consists of sets of ESDs, each set describing a well-known situation we will call *scenario* (e.g., "washing laundry"). An ESD consists of a sequence of *events*, each describing an action defining part of the scenario (e.g., "place the laundry in the washing machine"). We refer to descriptions of the same event across ESDs as *event types*. We refer to entities involved in a scenario as *participants* (e.g., a "washing machine" or a "detergent"), and to sets of participant descriptions describing the same entity as *participant types*.

For each type of scenario, our model clusters descriptions which refer to the same type of event, and infers constraints on the temporal order in which the events types occur in a particular scenario. Common characteristics of ESDs such as event optionality and varying degrees of temporal flexibility of event types make this task nontrivial. We propose a model which, in contrast to previous approaches, explicitly targets these characteristics. We develop a Bayesian formulation of the script learning problem, and present a generative model for *joint* learning of event types and ordering constraints, arguing that the temporal position of an event in an ESD provides a strong cue for its type, and vice versa. Our model is unsupervised in that no event- or participant labels are required for training.

We model constraints on the order of event types using a statistical model over permutations, the Generalized Mallows Model (GMM; Fligner

and Verducci (1986)). With the GMM we can flexibly model apparent characteristics of scripts, such as event type-specific temporal flexibility. Assuming that types of participants provide a strong cue for the type of event they are observed in, we use participant types as a latent variable in our model. Finally, by modeling event type occurrence using Binomial distributions, we can model event optionality, a characteristic of scripts that previous approaches did not capture.

We evaluate our model on a data set of ESDs collected via web experiments from non-expert annotators by Regneri et al. (2010) and compare our model against their approach. Our model achieves an absolute average improvement of 7% over the model of Regneri et al. on the task of event ordering.

For our unsupervised Bayesian model the limited size of this training set constitutes an additional challenge. In order to alleviate this problem, we use an informed prior on the word distributions. Instead of using Dirichlet priors which do not encode a-priori correlations between words, we incorporate a logistic normal distribution with the covariance matrix derived from WordNet. While we will show that prior knowledge as defined above enables the application of our model to small data sets, we emphasize that the model is generally widely applicable for two reasons. First, the data, collected using crowdsourcing, is comparatively easy and cheap to extend. Secondly, our model is domain independent and can be applied to scenario descriptions from any domain without any modification. Note that parameters were tuned on held-out scenarios, and no scenario-specific tuning was performed.

2 Related Work

In the 1970s, scripts were introduced as a way to equip AI systems with world knowledge (Schank and Abelson, 1975; Barr and Feigenbaum, 1986). Task-specific script databases were developed manually. FrameNet (Baker et al., 1998) follows a similar idea, in defining verb frames together with argument types that can fill the verbs' argument slots. Frames can then be combined into "scenario frames". Manual composition of such databases, is arguably expensive and does not scale well.

This paper follows a series of more recent work which aims to infer script knowledge automatically from data. Chambers and Jurafsky (2008) present a system which learns *narrative chains* from newswire texts. Relevant phrases are identified based on shared protagonists. The phrases are clustered into equivalence classes and temporally ordered using a pipeline of methods. We work with explicit event sequence descriptions of a specific scenario, arguing that large-scale common sense knowledge is hard to acquire from natural text, since it is often left implicit. Regneri et al. (2010) induce script knowledge from explicit ESDs using a graph-based method. Event types and ordering constraints are induced by aligning descriptions of equivalent events using WordNet-based semantic similarity. On this basis an abstract graph-representation (Temporal Script Graph; TSG) of the scenario is computed, using Multiple Sequence Alignment (MSA). Our work follows the work of Regneri et al. (2010), in that we use the same data and aim to focus on the same task. However, the two approaches described above employ a pipeline architecture and treat event learning and learning ordering constraints as separate problems. In contrast, we propose to learn both tasks *jointly*. We incorporate both tasks in a hierarchical Bayesian model, thus using one unified framework.

A related task, unsupervised frame induction, has also been considered in the past (Titov and Klementiev, 2011; Modi et al., 2012; O'Connor, 2012); the frame representations encode events and participants but ignore the temporal aspect of script knowledge.

We model temporal constraints on event type orderings with the Generalized Mallows Model (GMM; Mallows (1957); Fligner and Verducci (1986); Klementiev et al. (2008)), a statistical model over permutations. The GMM is a flexible model which can specify item-specific sensitivity to perturbation from the item's position in the canonical permutation. With the GMM we are thus able to model event type-specific temporal flexibility – a feature of scripts that MSA cannot capture.

The GMM has been successfully applied to modeling ordering constraints in NLP tasks. Chen et al. (2009) augment classical topic models with a GMM, under the assumption that topics in structured domains (e.g., biographies in Wikipedia) tend to follow an underlying canonical ordering, an assumption which matches well our data (the annotators were asked to follow the temporal or-

der of events in their descriptions (Regneri et al., 2010)). Chen et al. show that for these domains their approach significantly outperforms Markovian modeling of topics. This is expected as Markov models (MMs) are not very appropriate for representing linear structure with potentially missing topics (e.g., they cannot encode that every topic is assigned to at most one continuous fragment of text). Also GMMs are preferable for smaller collections such as ours, as the parameter number is linear in the number of topics (i.e., for us, event types) rather than quadratic as in Markov models. We are not aware of previous work on modeling events with GMMs. Conversely, MMs were considered in the very recent work of Cheung et al. (2013) in the context of script induction from news corpora where the Markovian assumption is much more natural.

There exists a body of work for learning participant types involved in scripts. Regneri et al. (2011) extend their work by inducing participant types on the basis of the TSG, using structural information about participant mentions in the TSG as well as WordNet similarity, which they then combine into an Integer Linear Program. Similarly, Chambers and Jurafsky (2009) extend their work on narrative chains, presenting a system with which they jointly learn event types and semantic roles of the participants involved, but do not consider event orderings. We include participant types as a latent feature in our model, assuming that participant mentions in an event description are a predictive feature for the corresponding event type.

One way of alleviating the problem of small data sets is incorporating informed prior knowledge. Raina et al. (2006) encode word correlations in a variance-covariance matrix of a multivariate normal distribution (MVN), and sample prior parameter vectors from it, thus introducing dependencies among the parameters. They induce the covariances from supervised learning tasks in the transfer learning set-up. We use the same idea, but obtain word covariances from WordNet relations. In a slightly different setting, covariance matrices of MVNs have been used in topic models to induce correlation between topics in documents (Blei and Lafferty, 2006).

3 Problem Formulation

Our input consists of a corpus of scenario-specific ESDs, and our goal is to label each event description in an ESD with one event type e. We specify the number of possible event types E a priori as a number exceeding the number of event types in all the scripts considered. The model will select an effective subset of those types.

Assume a scenario-specific corpus c, consisting of D ESDs, $c = \{d_1, ..., d_D\}$. Each ESD d_i consists of N_d event descriptions $d_i = \{d_{i,1}, ..., d_{i,N_i}\}$. Boundaries between descriptions of single events are marked in the data. For each event description $d_{i,n}$ a bag of participant descriptions is extracted. Each participant description corresponds to one noun phrase as identified automatically by a dependency parser (cf. Regneri et al. (2011)). We also associate participant types with participant descriptions, these types are latent and induced at the inference stage.

Given such a corpus of ESDs, our model assigns each event description $d_{i,n}$ in an ESD d_i one event type $z_{d_{i,n}} = e$, where $e \in \{1, ..., E\}$. Assuming that all ESDs are generated from the same underlying set of event types, our objective is to assign the same event type to equivalent event descriptions across all ESDs in the corpus.

We furthermore assume that there exists a canonical temporal ordering of event types for each scenario type, and that events in observed scenarios tend to follow this ordering, but allowing for some flexibility. The event labeling sequence $\mathbf{z}_{d_i}$ of an entire ESD should reflect this canonical ordering. This allows us to use global structural patterns of ESDs in the event type assignments, and thus introducing dependence between event types through their position in the sequence.

4 The Model

Before we describe our model, we briefly explain the Generalized Mallows Model (GMM) which we use to encode a preference for linear ordering of events in a script.

4.1 The (Generalized) Mallows Model

The Mallows Model (MM) is a statistical model over orderings (Mallows, 1957). It takes two parameters $\boldsymbol{\sigma}$, the canonical ordering, and $\rho > 0$, a dispersion parameter. The dispersion parameter is a penalty for the divergence $d(\boldsymbol{\pi}, \boldsymbol{\sigma})$ of an observed ordering $\boldsymbol{\pi}$ from the canonical ordering $\boldsymbol{\sigma}$. The divergence can be any distance metric but Kendall's tau distance ("bubble-sort" distance), a number of swaps needed to bring $\boldsymbol{\pi}$ in the order $\boldsymbol{\sigma}$,

is arguably the most common choice. The probability of an observed ordering $\boldsymbol{\pi}$ is defined as

$$P(\boldsymbol{\pi}|\rho, \boldsymbol{\sigma}) = \frac{e^{-\rho\, d(\boldsymbol{\pi},\boldsymbol{\sigma})}}{\psi(\rho)},$$

where $\psi(\rho)$ is a normalization factor. The distribution is centered around the canonical ordering (as $d(\boldsymbol{\sigma}, \boldsymbol{\sigma}) = 0$), and the probability decreases exponentially with an increasing distance. For our purposes, without loss of generality, we can assume that $\boldsymbol{\sigma}$ is the identity permutation, that is $\boldsymbol{\sigma} = [1, \ldots, n]$, where n is the number of items.

The Mallows model has been generalized to take as a parameter a vector of *item-specific* dispersion parameters $\boldsymbol{\rho}$ (Fligner and Verducci, 1986). In order to introduce this extension, we first need to reformulate Kendall's tau in a way that captures item-specific distance. An ordering $\boldsymbol{\pi}$ of n items can be equivalently represented by a vector of inversion counts $\boldsymbol{v}$ of length $n-1$, where each component v_i equals the number of items $j > i$ that occur before item i in $\boldsymbol{\pi}$. For example, for an observed ordering $\boldsymbol{\pi}$ = [2,1,0] the inversion vector $\boldsymbol{v} = (2, 1)$.[1] Then the generalized Mallows model (GMM) is defined as

$$GMM(\boldsymbol{\pi}|\boldsymbol{\rho}) \propto \prod_i e^{-\rho_i\, v_i}.$$

The GMM can be factorized into item-specific components, which allows for efficient inference:

$$GMM_i(v_i|\rho_i) \propto e^{-\rho_i\, v_i}. \quad (1)$$

Intuitively, we will be able to induce event type-specific penalty parameters, and will thus be able to model individual degrees of temporal flexibility among the event types.

Since the GMM is member of the exponential family, a conjugate prior can be defined, which allows for efficient learning of the parameters $\boldsymbol{\rho}$ (Fligner and Verducci, 1990). Like the GMM, its prior distribution GMM_0 can be factorized into independent components for each item i:

$$GMM_0(\rho_i|v_{i,0}, \nu_0) \propto e^{-\rho_i v_{i,0} - log(\psi_i(\rho_i))\nu_0}. \quad (2)$$

The parameters $v_{i,0}$ and ν_0 represent our prior beliefs about flexibility for each item i, and the strength of these beliefs, respectively.

[1]Trivially, the inversion count for the last element in the canonical ordering is always 0.

4.2 The Generative Story

Our model encodes two fundamental assumptions, based on characteristics observed in the data: (1) We assume that each event type can occur at most once per ESD; (2) Each participant type is assumed to occur at most once per event type.

The formalized generative story is given in Figure 1. For each document (ESD) d, we decide independently for each event type e whether to realize it or not by drawing from $Binomial(\theta_e)$.[2] We obtain a binary event vector $\boldsymbol{t}$ where $t_e = 1$ if event type e is realized and $t_e = 0$ otherwise. We draw an event ordering $\boldsymbol{\pi}$ from $GMM(\boldsymbol{\rho})$, represented as a vector of inversion counts.

Now, we pass event types in the order defined by $\boldsymbol{\pi}$. For each realized event type i (i.e., $i : t_i = 1$), we first generate a word (normally a predicate) from the corresponding language model $Mult(\vartheta_i)$. Then we independently decide for each participant type p whether to realize it or not with the probability $Binomial(\varphi^i_p)$. If realized, the participant word (its syntactic head) is generated from the participant language model $Mult(\varpi_p)$.

Note that though the distribution controlling frequency of participant generation (φ^i_j) is event type-specific, the language model associated with the participant ($Mult(\varpi_j)$) is shared across events, thus, ensuring that participant types are defined across events.

The learnt binary realization parameters $\boldsymbol{\theta}$ and $\boldsymbol{\varphi}^e$ should ensure that an appropriate number of events and participants is generated (e.g. the realization probability for obligatory events, observed in almost every ESD for a particular scenario, should be close to 1).

Priors We draw the parameters for the binomial distributions from the Beta distribution, which allows us to model a global preference for using only few event types and only few participant types for each event type. We draw the parameters of the multinomials from the Dirichlet distribution, and can thus model a preference towards sparsity. The GMM parameter vector $\boldsymbol{\rho}$ is drawn from GMM_0 (c.f. Equation (2)).

4.3 Adding Prior Knowledge

Since we are faced with a limited amount of training data, we augment the model described above

[2]We slightly abuse the notation by dropping the superscript d for ESD-specific variables.

Generation of parameters

for event type $e = 1, \ldots, E$ **do**
  $\theta_e \sim Beta(\alpha^+, \alpha^-)$ [freq of event]
  $\vartheta_e \sim Dirichlet(\boldsymbol{\gamma})$ [event lang mod]
  for participant type $p = 1, \ldots, P$ **do**
    $\varphi_p^e \sim Beta(\beta^+, \beta^-)$ [freq of ptcpt]
for participant type $p = 1, \ldots, P$ **do**
  $\varpi_p \sim Dirichlet(\boldsymbol{\delta})$ [ptcpt lang mod]
for event type $e = 1, \ldots, E-1$ **do**
  $\rho_e \sim GMM_0(\boldsymbol{\rho}_0, \boldsymbol{\nu}_0)$ [ordering params]

Generation of ESD d

for event type $e = 1, \ldots, E$ **do**
  $t_e \sim Binomial(\theta^e)$ [realized events]
$\boldsymbol{\pi} \sim GMM(\boldsymbol{\rho}, \boldsymbol{\nu})$ [event ordering]
for event i from $\boldsymbol{\pi}$ s.th. t_i=1 **do**
  $w_i \sim Mult(\vartheta_i)$ [event lexical unit]
  for participant type $p = 1, \ldots, P$ **do**
    $u_p \sim Binomial(\varphi_p^e)$ [realized ptcpts]
    if $u_p = 1$ **then**
      $w_p \sim Mult(\varpi_p)$ [ptcpt lexical unit]

Figure 1: The generative story of the basic model.

to encode correlations between semantically similar words in the priors for language models. We describe our approach by first introducing the model extension allowing for injecting prior correlations between words, and then explaining how the word correlations are derived from WordNet (Fellbaum, 1998). Since the event vocabulary and the participant vocabulary are separate in our model, the following procedure is carried out separately, but equivalently, for the two vocabularies.

4.3.1 Modeling Word Correlation

Dirichlet distributions do not provide a way to encode correlations between words. To tackle this problem we add another level in the model hierarchy: instead of specifying priors $Dirichlet(\boldsymbol{\gamma})$ and $Dirichlet(\boldsymbol{\delta})$ directly, we generate them for each event type e and participant type p using multivariate normal distributions.

The modification for the generative story is shown in Figure 2. In this extension, each event type e and participant type p has a different associated (nonsymmetric) Dirichlet prior, $\boldsymbol{\gamma}^e$ and $\boldsymbol{\delta}^p$, respectively. The generative story for choosing $\boldsymbol{\gamma}^e$ is the following: A vector $\boldsymbol{\eta}_e$ is drawn from the zero-mean normal distribution $N(\Sigma_\eta, \mathbf{0})$, where Σ_η is

Generation of parameters ϑ_e and ϖ_p

for event type $e = 1, \ldots, E$ **do**
  $\boldsymbol{\eta}^e \sim N(\Sigma_\eta, 0)$
  for all words w **do**
    $\gamma_w^e = \exp(\eta_w^e) / \sum_{w'} \exp(\eta_{w'}^e)$ [Dir prior]
  $\vartheta_e \sim Dirichlet(\boldsymbol{\gamma}^e)$ [event lang mod]
for participant type $p = 1, \ldots, P$ **do**
  $\boldsymbol{\xi}^p \sim N(\Sigma_\xi, 0)$
  for all words w **do**
    $\delta_w^p = \exp(\xi_w^p) / \sum_{w'} \exp(\xi_{w'}^p)$ [Dir prior]
  $\varpi_p \sim Dirichlet(\boldsymbol{\delta}^p)$ [ptcpt lang mod]

Figure 2: The modified parameter generation procedure for ϑ_e and ϖ_p to encode word correlations.

the covariance matrix encoding the semantic relatedness of words (see Section 4.3.2). The vector's dimensionality corresponds to size of the vocabulary of event words. Then, the vector is exponentiated and normalized to yield $\boldsymbol{\gamma}^e$.[3] The same procedure is used to choose $\boldsymbol{\delta}^p$ as shown in Figure 2.

4.3.2 Defining Semantic Similarity

We use WordNet to obtain semantic similarity scores for each pair of words in our vocabulary. Since we work on limited domains, we define a subset of WordNet as all synsets that any word in our vocabulary is a member of, plus the hypernym sets of all these synsets. We then create a feature vector for each word $f(w_i)$ as follows:

$$f(w_i)_n = \begin{cases} 1 & \text{any sense of } w_i \in \text{synset } n \\ 0 & \text{otherwise} \end{cases}$$

The similarity of two words w_i and w_j is defined as the dot product $f(w_i) \cdot f(w_j)$. We use this similarity to define the covariance matrices Σ_η and Σ_ξ. Each component (i, j) stores the similarity between words w_i and w_j as defined above. Note that the matrices are guaranteed to be valid covariance matrices, as they are positive semidefinite by construction.

5 Inference

Our goal is to infer the set of labelings $\boldsymbol{z}$ of our corpus of ESDs. A labeling $\boldsymbol{z}$ consists of event

[3] In fact, Dirichlet concentration parameters do not need to sum to one. We experimented with normalizing them to yield a different constant, thus regulating the influence of the prior, but have not observed much of improvement from this extension.

types $\boldsymbol{t}$, participant types $\boldsymbol{u}$ and event ordering $\boldsymbol{\pi}$. Additionally, we induce parameters of our model: ordering dispersion parameters ($\boldsymbol{\rho}$) and the language model parameters $\boldsymbol{\eta}$ and $\boldsymbol{\xi}$. We induce these variables conditioned on all the observable words in the data set $\boldsymbol{w}$. Since direct joint sampling from the posterior distributions is intractable, we use Gibbs sampling for approximate inference. Since we chose conjugate prior distributions over the parameter distributions, we can "collapse" the Gibbs sampler by integrating out all parameters (Griffiths and Steyvers, 2004), except for the ones listed above. The unnormalized posterior can be written as the following product of terms:

$$P(\boldsymbol{z}, \boldsymbol{\rho}, \boldsymbol{\eta}, \boldsymbol{\xi} | \boldsymbol{w}) \propto \prod_e DCM_e \prod_p DCM_p \prod_e BBM_e \prod_p BBM_{ep} \prod_e GMM_e \quad MN_e \prod_p MN_p.$$

The terms DCM_e and DCM_p are Dirichlet compound multinomials associated with event-specific and participant-specific language models:

$$DCM_e = \frac{\Gamma(\sum_v \gamma_v^e)}{\Gamma(\sum_v N_v^e + \gamma_v^e)} \prod_v \frac{\Gamma(N_v^e + \gamma_v^e)}{\Gamma(\gamma_v^e)}$$

$$DCM_p = \frac{\Gamma(\sum_v \delta_v^p)}{\Gamma(\sum_v N_v^p + \delta_v^p)} \prod_v \frac{\Gamma(N_v^p + \delta_v^p)}{\Gamma(\delta_v^p)},$$

where N_v^e and N_v^p is the number of times word type v is assigned to event e and participant p, respectively. The terms BBM_e and BBM_{ep} are the Beta-Binomial distributions associated with generating event types and generating participant types for each event type (i.e. encoding optionality of events and participants):

$$BBM_e \propto \frac{\Gamma(N_e^+ + \alpha^+)\Gamma(N_e^- + \alpha^-)}{\Gamma(N_e^+ + N_e^- + \alpha^+ + \alpha^-)}$$

$$BBM_{ep} \propto \prod_e \prod_p \frac{\Gamma(N_{ep}^+ + \beta^+)\Gamma(N_{ep}^- + \beta^-)}{\Gamma(N_{ep}^+ + N_{ep}^- + \beta^+ + \beta^-)},$$

where N_e^+ and N_e^- is the number of ESDs where event type is generated and the number of ESD where it is not generated, respectively. N_{ep}^+ and N_{ep}^- are analogously defined for participant types (for each event type e). The term GMM_e is associated with the inversion count distribution for event type e and has the form

$$GMM_e \propto GMM_0(\rho_e; \frac{\sum_d v_e^d + v_{e,0}\nu_0}{N + \nu_0}, N + \nu_0),$$

where GMM_0 is defined in expression (2) and v_e^d is the inversion count for event e in ESD d. N is the cumulative number of event occurrences in the data set.

Finally, MN_e and MN_p correspond to the probability of drawing $\boldsymbol{\eta}^e$ and $\boldsymbol{\xi}^p$ from the corresponding normal distributions, as discussed in Section 4.3.1.

Though, at each step of Gibbs sampling, components of $\boldsymbol{z}$ could potentially be sampled by considering the full unnormalized posterior, this clearly can be made much more efficient by observing that only a fraction of terms affect the corresponding conditional probability. For example, when sampling an event type for a given event in a ESD d, only the terms DCM_e, BBM_{ep} and BBM_e for all e and p are affected. For DCMs it can be simplified further as only a few word types are affected. Due to space constraints, we cannot describe the entire sampling algorithms but it naturally follows from the above equations and is similar to the one described in Chen et al. (2009).

For sampling the other parameters of our model, ranking dispersion parameters $\boldsymbol{\rho}$ and the language model parameters $\boldsymbol{\eta}$ and $\boldsymbol{\xi}$, we use slice sampling (MacKay, 2002). For each event type e we draw its dispersion parameter ρ_e independently from the slice sampler.

After every n^{th} iteration we resample $\boldsymbol{\eta}$ and $\boldsymbol{\xi}$ for all language models to capture the correlations. However, to improve mixing time, we also resample components η_i^k and η_i^l when word i has changed event membership from type k to type l. In addition we define classes of closely related words (heuristically based on the covariance matrix) by classifying words as related when their similarity exceeds an empirically determined threshold. We also resample all components η_j^k and η_j^l for each word j that related to word i. We re-normalize $\boldsymbol{\eta}^m$ and $\boldsymbol{\eta}^n$ after resampling to update the Dirichlet concentration parameters. The same procedure is used for participant language models (parameters $\boldsymbol{\xi}$).

6 Evaluation

In our evaluation, we evaluate the quality of the event clusters induced by the model and the extent to which the clusters capture the global event ordering underlying the script, as well as the benefit of the GMM and the informed prior knowledge. We start by describing data and evaluation metrics.

Scenario Name	♯ESDs	Avg len
OMICS corpus		
Cook in microwave	59	5.03
Answer the telephone	55	4.47
Buy from vending machine	32	4.53
Make coffee	38	5.00
R10 corpus		
Iron clothes	19	8.79
Make scrambled eggs	20	10.3
Eat in fast food restaurant	15	8.93
Return food (in a restaurant)	15	5.93
Take a shower	21	11.29
Take the bus	19	8.53

Table 1: Test scenarios used in experiments (left), the size of the corresponding corpus (middle), and the average length of an ESD in events (right).

	Event Paraphrase			Evt. Ordering		
	P	R	F	P	R	F
Ret. Food	0.92	0.52	**0.67**	0.87	0.72	**0.79**
-GMM	0.70	0.30	0.42	0.46	0.44	0.45
-COVAR	0.92	0.52	**0.67**	0.77	0.67	0.71
Vending	0.76	0.78	0.77	0.90	0.74	**0.81**
-GMM	0.74	0.39	0.51	0.64	0.47	0.54
-COVAR	0.74	0.87	**0.80**	0.85	0.73	0.78
Shower	0.68	0.67	**0.67**	0.85	0.84	**0.85**
-GMM	0.36	0.17	0.23	0.42	0.38	0.40
-COVAR	0.64	0.44	0.52	0.77	0.73	0.75
Microwave	0.85	0.80	0.82	0.91	0.74	0.82
-GMM	0.88	0.30	0.45	0.67	0.62	0.64
-COVAR	0.89	0.81	**0.85**	0.92	0.82	**0.87**

Table 2: Comparison of model variants: For each scenario: The full model (top), a version without the GMM (−GMM), and a version with a uniform Dirichlet prior over language models (−COVAR).

6.1 Data

We use the data sets presented in Regneri et al. (2010) (henceforth R10) for development and testing. The data is comprised of ESDs from two corpora. R10 collected a corpus, consisting of sets of ESDs for a variety of scenarios, via a web experiment from non-expert annotators. In addition we use ESDs from the OMICS corpus[4] (Kochenderfer and Gupta, 2003), which consists of instantiations of descriptions of several 'stories', but is restricted to indoor activities. The details of our data are displayed in Table 1. For each event description we extract all noun phrases, as automatically identified by Regneri et al. (2011), separating participant descriptions from action descriptions. We remove articles and pronouns, and reduce NPs to their head words.

6.2 Gold Standard and Evaluation Metrics

We follow R10 in evaluating induced event types and orderings in a binary classification setting. R10 collected a gold standard by classifying pairs of event descriptions w.r.t. whether or not they are paraphrases. Our model classifies two event descriptions as equivalent whenever $z_{e_1} = z_{e_2}$.

Equivalently, R10 classify ordered pairs of event descriptions as to whether they are presented in their natural order. Assuming the identity ordering as canonical ordering in the Generalized Mallows Model, event types tending to occur earlier in the script should be assigned lower cluster IDs than event types occurring later. Thus, whenever $z_{e_1} < z_{e_2}$, our the model predicts that two event descriptions occur in their natural order.

[4]http://csc.media.mit.edu/

We evaluate the output of our model against the described gold standard, using Precision, Recall and F1 as evaluation metrics, so that our results are directly comparable to R10. We tune our parameters on a development set of 5 scenarios which are not used in testing.

6.3 Results

Table 3 presents the results of our two evaluation tasks. While on the event paraphrase task the R10 system performs slightly better, our model outperforms the R10 system on the event ordering task by a substantial margin of 7 points average F-score. While both systems perform similarly on the task of event type induction, we induce a *joint* model for both objectives. The results show that, despite the limited amount of data, and the more complex learning objective, our model succeeds in inducing event types and ordering constraints.

In order to demonstrate the benefit of the GMM, we compare the performance of our model to a variant which excludes this component (−GMM), cf. Table 2. The results confirm our expectation that biasing the model towards encouraging a linear ordering on the event types provides a strong cue for event cluster inference.

As an example of a clustering learnt by our model, consider the following event chain:

{get} → {open,take} → {put,place} → {close} → {set,select,enter,turn} → {start} → {wait} → {remove,take,open} → {push,press,turn}

We display the most frequent words in the clusters

Scenario	Event Paraphrase Task						Event Ordering Task					
	Precision		Recall		F1		Precision		Recall		F1	
	R10	BS	R10	BS	R10	BS	R10	BS	R10	BS	R10	BS
Coffee	0.50	0.47	0.94	0.58	**0.65**	0.52	0.70	0.68	0.78	0.57	**0.74**	0.62
Telephone	0.93	0.92	0.85	0.72	**0.89**	0.81	0.83	0.92	0.86	0.87	0.84	**0.89**
Bus	0.65	0.52	0.87	0.43	**0.74**	0.47	0.80	0.76	0.80	0.76	**0.80**	0.76
Iron	0.52	0.65	0.94	0.56	**0.67**	0.60	0.78	0.87	0.72	0.69	0.75	**0.77**
Scr. Eggs	0.58	0.92	0.86	0.65	0.69	**0.76**	0.67	0.77	0.64	0.59	0.66	**0.67**
Vending	0.59	0.76	0.83	0.78	0.69	**0.77**	0.84	0.90	0.85	0.74	**0.84**	0.81
Microwave•	0.75	0.85	0.75	0.80	0.75	**0.82**	0.47	0.91	0.83	0.74	0.60	**0.82**
Shower•	0.70	0.68	0.88	0.67	**0.78**	0.67	0.48	0.85	0.82	0.84	0.61	**0.85**
Fastfood•	0.50	0.74	0.73	0.87	0.59	**0.80**	0.53	0.97	0.81	0.65	0.64	**0.78**
Ret. Food•	0.73	0.92	0.68	0.52	**0.71**	0.67	0.48	0.87	0.75	0.72	0.58	**0.79**
Average	0.645	**0.743**	**0.833**	0.658	**0.716**	0.689	0.658	**0.850**	**0.786**	0.717	0.706	**0.776**

Table 3: Results of our model for the event paraphrase task (left) and event type ordering task (right). Our system (BS) is compared to the system in Regneri et al. (2010) (R10). We were able to obtain the R10 system from the authors and evaluate on additional scenarios for which no results are reported in the paper. These additional scenarios are marked with a dot (•).

inferred for the "Microwave" scenario. Clusters are sorted by event type ID. Note that the word 'open' is assigned to two event types in the sequence, which is intuitively reasonable. This illustrates why assuming a deterministic mapping from predicates to events (as in Chambers and Jurafsky (2008)) is limiting for our dataset.

We finally examined the influence of the informed prior component, comparing to a model variant which uses uniform Dirichlet parameters (`-COVAR`; see Table 2). As expected, using an informed prior component leads to improved performance on scenario types with fewer training ESDs available ('Take a shower' and 'Return food'; cf. Table 1). For scenarios with a larger set of training documents no reliable benefit from the informed prior is observable. We did not optimize this component, e.g. by testing more sophisticated methods for construction of the covariance matrix, but expect to be able to improve its reliability.

7 Discussion

The evaluation shows that our model is able to create meaningful event type clusters, which resemble the underlying event ordering imposed by the scenario. We achieve an absolute average improvement of 7% over a state-of-the-art model. In contrast to previous approaches to script induction, our model does not include specifically customized components, and is thus flexibly applicable without additional engineering effort.

Our model provides a clean, statistical formulation of the problem of jointly inducing event types and their ordering. Using a Bayesian model allows for flexible enhancement of the model. One straightforward next step would be to explore the influence of participants, and try to jointly infer them with our current set of latent variables.

Statistical models highly rely on a sufficient amount of training data in order to be able to induce latent structures. The limited amount of training data in our case is a bottleneck for the performance. The model performs best on the two scenarios with the most training data ('Telephone' and 'Microwave'), which supports this assumption. We showed, however, that our model can be applied to small data sets through incorporation of informed prior knowledge without supervision.

8 Conclusion

We presented a hierarchical Bayesian model for joint induction of event clusters and constraints on their orderings from sets of ESDs. We incorporate the Generalized Mallows Model over orderings. The evaluation shows that our model successfully induces event clusters and ordering constraints.

We compare our joint, statistical model to a pipeline based model using MSA for event clustering. Our system outperforms the system on the task of event ordering induction by a substantial margin, while achieving comparable results in the event induction task. We could further explicitly show the benefit of modeling global ESD structure, using the GMM.

In future work we plan to apply our model to larger data sets, and to examine the role of participants in our model, exploring the potential of inferring them jointly with our current objectives.

Acknowledgments

We thank Michaela Regneri for substantial support with the script data, and Mirella Lapata for helpful comments.

References

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The berkeley framenet project. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, pages 86–90.

A. Barr and E.A. Feigenbaum. 1986. *The handbook of artificial intelligence. 1 (1981)*. The Handbook of Artificial Intelligence. Addison-Wesley.

David Blei and John Lafferty. 2006. Correlated topic models. In *Advances in Neural Information Processing Systems 18*, pages 147–154.

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 47th Annual Meeting of the Association for Computational Linguistics*, pages 602–610.

H. Chen, S. R. K. Branavan, R. Barzilay, and D. R. Karger. 2009. Content modeling using latent permutations. *Journal of Artificial Intelligence Research*, 36(1):129–163.

Christiane Fellbaum, editor. 1998. *WordNet: an electronic lexical database*. MIT Press.

M. Fligner and J. Verducci. 1986. Distance based ranking models. *Journal of the Royal Statistical Society, Series B*, 48:359–369.

M. Fligner and J. Verducci. 1990. Posterior probabilities for a consensus ordering. *Psychometrika*, 55:53–63.

T. L. Griffiths and M. Steyvers. 2004. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101(Suppl. 1):5228–5235.

Alexandre Klementiev, Dan Roth, and Kevin Small. 2008. Unsupervised rank aggregation with distance-based models. In *Proceedings of the 25th International Conference on Machine Learning*, pages 472–479.

Mykel J. Kochenderfer and Rakesh Gupta. 2003. Common sense data acquisition for indoor mobile robots. In *Proceedings of the Nineteenth National Conference on Artificial Intelligence (AAAI-04)*, pages 605–610.

D. J. C. MacKay. 2002. *Information Theory, Inference & Learning Algorithms*. Cambridge University Press, New York, NY, USA.

C. L. Mallows. 1957. Non-null ranking models. *Biometrika*, 44:114–130.

Risto Miikkulainen. 1995. Script-based inference and memory retrieval in subsymbolic story processing. *Applied Intelligence*, pages 137–163.

Ashutosh Modi, Ivan Titov, and Alexandre Klementiev. 2012. Unsupervised induction of frame-semantic representations. In *Proceedings of the NAACL-HLT Workshop on the Induction of Linguistic Structure*, pages 1–7.

Erik T. Mueller. 2004. Understanding script-based stories using commonsense reasoning. *Cognitive Systems Research*, 5(4):307–340.

Brendan O'Connor. 2012. Bayesian unsupervised frame learning from text. Technical report, Carnegie Mellon University.

Rajat Raina, Andrew Y. Ng, and Daphne Koller. 2006. Constructing informative priors using transfer learning. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 713–720.

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.

Michaela Regneri, Alexander Koller, Josef Ruppenhofer, and Manfred Pinkal. 2011. Learning script participants from unlabeled data. In *Proceedings of RANLP 2011*, pages 463–470.

Roger C. Schank and Robert P. Abelson. 1975. Scripts, plans, and knowledge. In *Proceedings of the 4th International Joint Conference on Artificial Intelligence*, IJCAI'75, pages 151–157.

Ivan Titov and Alexandre Klementiev. 2011. A bayesian model for unsupervised semantic parsing. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1445–1455.