Inducing Script Structure from Crowdsourced Event Descriptions via Semi-Supervised Clustering

Lilian D. A. Wanzare **Alessandra Zarcone** **Stefan Thater** **Manfred Pinkal**
Universität des Saarlandes
Saarland, 66123, Germany
{wanzare,zarcone,stth,pinkal}coli.uni-saarland.de

Abstract

We present a semi-supervised clustering approach to induce script structure from crowdsourced descriptions of event sequences by grouping event descriptions into paraphrase sets (representing event types) and inducing their temporal order. Our model exploits semantic and positional similarity and allows for flexible event order, thus overcoming the rigidity of previous approaches. We incorporate crowdsourced alignments as prior knowledge and show that exploiting a small number of alignments results in a substantial improvement in cluster quality over state-of-the-art models and provides an appropriate basis for the induction of temporal order. We also show a coverage study to demonstrate the scalability of our approach.

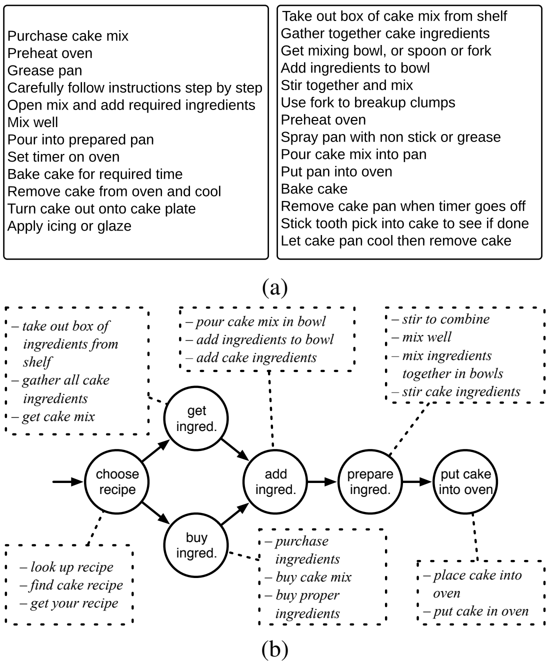
1 Introduction

During their daily social interactions, people make seamless use of knowledge about standardized event sequences (*scripts*) describing types of everyday activities, or *scenarios*, such as GOING TO THE RESTAURANT or BAKING A CAKE (Schank and Abelson, 1977; Barr and Feigenbaum, 1981). Script knowledge is often triggered by the broader discourse context and guides expectations in text understanding and makes missing events and referents in a discourse accessible. For example, if we hear someone say “I baked a cake on Sunday. I decorated it with buttercream icing!”, our script knowledge allows us to infer that the speaker must have *mixed the ingredients, turned on the oven*, etc., even if these events are not explicitly mentioned. Script knowledge is relevant for the computational modeling of various kinds of cognitive abilities and has the potential to support NLP tasks such as anaphora resolution (Rahman and Ng, 2011), discourse relation detection, semantic role labeling, temporal order analysis, and applications such as text understanding (Cullingford, 1977; Mueller, 2004), information extraction (Rau et al., 1989), question answering (Hajishirzi and Mueller, 2012).

Several methods for the automatic acquisition of script knowledge have been proposed. Seminal work by Chambers and Jurafsky (2008; 2009) provided methods for the unsupervised wide-coverage extraction of script knowledge from large text corpora. However, texts typically only mention small parts of a script, banking on the reader’s ability to infer missing script-related events. The task is therefore challenging, and the results are quite noisy.

The work presented in this paper follows the approach proposed in Regneri et al. (2010) (henceforth “RKP”) who crowdsourced scenario descriptions by asking people how they typically carry out a particular activity. The collected event sequence descriptions provide generic descriptions of a given scenario (e.g. BAKING A CAKE) in concise telegram style (Fig. 1a). Based on these crowdsourced event sequence descriptions or ESDs, RKP extracted high-quality script knowledge for a variety of different scenarios, in the form of temporal script graphs (Fig. 1b). Temporal script graphs are partially ordered structures whose nodes are sets of alternative descriptions denoting the same event type, and whose edges express temporal precedence.

While RKP employ Multiple Sequence Alignment (MSA) (Durbin et al., 1998), we use a *semi-supervised clustering approach* for script structure induction. The choice of MSA was motivated by the effect of positional information on the detection of scenario-specific paraphrases: event de-



Purchase cake mix
Preheat oven
Grease pan
Carefully follow instructions step by step
Open mix and add required ingredients
Mix well
Pour into prepared pan
Set timer on oven
Bake cake for required time
Remove cake from oven and cool
Turn cake out onto cake plate
Apply icing or glaze

Take out box of cake mix from shelf
Gather together cake ingredients
Get mixing bowl, or spoon or fork
Add ingredients to bowl
Stir together and mix
Use fork to breakup clumps
Preheat oven
Spray pan with non stick or grease
Pour cake mix into pan
Put pan into oven
Bake cake
Remove cake pan when timer goes off
Stick tooth pick into cake to see if done
Let cake pan cool then remove cake

(a)

(b)

Figure 1: Example ESDs (a) and induced script structure (b) for the BAKING A CAKE scenario from Wanzare et al. (2016)

scriptions occurring in similar positions in ESDs tend to denote the same event type. However, MSA makes far too strong an assumption about the temporal ordering information in the ESDs. It does not allow for crossing edges and thus must assume a fixed and invariable order, while the ordering of events in a script is to some degree flexible (e.g., one can *preheat the oven* before or after *mixing ingredients*). We propose clustering as an alternative method to overcome the rigidity of the MSA approach, and use a distance measure based on both semantic similarity and positional similarity information, making our clustering algorithm sensitive to ordering information, while allowing for order variation in the scripts.

Clustering accuracy depends on the reliability of similarity estimates, but scenario-specific paraphrase relations are often based on scenario-specific functional equivalence, which cannot be easily determined using semantic similarity, even if complemented with positional information. For example in the FLYING IN AN AIRPLANE scenario, it is challenging for any semantic similarity measure to predict that *walk up the ramp* and *board plane* refer to the same event, as the broader discourse context would suggest. To address this issue, we propose a semi-supervised approach, capitalizing on previous work by Klein et al. (2002). Semi-supervised approaches to clustering have shown that performance can be enhanced by incorporating prior knowledge in the form of a small number of instance-level constraints. We automatically identify event descriptions that are likely to cause alignment problems (called *outliers*), crowdsource alignments for these items and incorporate them as instance-level relational seeds into the clustering process.

Lastly, a main concern with the approach in RKP is scalability: temporal script graphs are created scenario-wise in a bottom-up fashion. They represent only fragments of the rich amount of script knowledge people use in everyday communication. In this paper we address this concern with the first assessment of the coverage of existing script resources, and an estimate of the concrete costs for their extension.

2 Data

We will now introduce the resources used in our study, namely the datasets of ESDs, the gold standards and the crowdsourced alignments between event descriptions.

Datasets and gold standards. Three large crowdsourced collections of activity descriptions in terms of ESDs are available: the OMICS corpus (Gupta and Kochenderfer, 2004), the SMILE corpus (Regneri et al., 2010) and DeScript corpus (Wanzare et al., 2016). Sections 3-4 of this paper focus on a subset of ESDs for 14 scenarios from SMILE and OMICS, with on average 29.9 ESDs per scenario. In RKP, in the follow-up studies by Frermann et al. (2014) and Modi and Titov (2014) as well as in the present study, 4 of these scenarios were used as development set and 10 as test set.

RKP provided two gold standard datasets for this subset: the *RKP paraphrase dataset* contains judgments for 60 event description pairs per scenario, the *RKP temporal order dataset* contains 60 event description pairs that are separately annotated in both directions, for a total of 120 datapoints per scenario. In order to directly evaluate our models for clustering quality, we also created a *clustering gold standard* for the RKP test set, adopting the experimental setup in Wanzare et al. (2016): we asked three trained students of computational linguistics to annotate the scenarios with gold standard alignments between event descriptions in different ESDs referring to the same event[1]. Every ESD was fully aligned with every other ESD in the same scenario. Based on the alignments, we derived gold clusters by grouping the event descriptions into gold paraphrase sets (17 clusters per scenario on average, ranging from 10 to 23).

In addition, we used a subset of 10 scenarios with 25 ESDs each from DeScript, for which Wanzare et al. (2016) provided gold clusters, to evaluate our models and to demonstrate that our method is independent of the specific choice of scenarios.

Crowdsourced alignments. To provide seed data for the semi-supervised clustering algorithm, we crowdsourced alignments between event descriptions, following the procedure in Wanzare et al. (2016). First, we identified challenging cases of event descriptions (called *outliers*), which we expected to be particularly informative and help improve clustering accuracy. To this purpose, an (unsupervised) clustering system (Affinity Propagation, see below) was run with varying parameter settings. Those event descriptions whose nearest neighbors changed clusters across different runs of the system were then identified as outliers (see Wanzare et al. (2016) for more details). A complementary type of seed data was obtained by selecting event descriptions that did not change cluster membership at all (called *stable cases*).

In a second step, groups of selected descriptions (outliers and stable cases) in their original ESD were presented to workers in a Mechanical Turk experiment, paired with a target ESD. The workers were asked to select a description in the target ESD denoting the same script event (e.g. for BAKING A CAKE: *pour into prepared pan* → *pour cake mix into pan*). We aimed at collecting two sets of high-quality seeds based on outliers and stable cases, respectively, each summing up to 3% of the links required for a total alignment between all pairs of scenario-specific ESDs (6% links in total). To guarantee high quality, we accepted only items where three (out of up to four) annotators agree. We checked the annotators' reliability by comparing their aligments for stable cases against the gold standard and rejected the work on 3% of the annotators.

We collected alignments for 20 scenarios: for the test scenarios of the SMILE+OMICS dataset, and for those in the clustering gold standard of DeScript. For the latter, a collection of alignment data was already available, but considerably differed in size between scenarios and was in general to small for our purposes.

3 Model

We first present a semi-supervised clustering method to induce script events from ESDs using crowdsourced alignments as seeds (Section 3.1). In Section 3.2, we describe how we calculate the underlying distance matrix based on semantic and positional similarity information. In Section 3.3, we describe the induction of temporal order for the script events, which turns the set of script events into a temporal script graph (TSG).

3.1 Semi-supervised Clustering

We use the crowdsourced alignments between event descriptions as instance-level relational seeds for clustering, more specifically as *must-link constraints*, requiring that the linked items should go into one cluster. We incorporate the constraints into the clustering process following the method in Klein et al. (2002): that is adapting the input distance matrix in a pre-processing step, rather than directly integrating the constraints into the clustering algorithm. This makes it possible to try different adaptation strategies, independently of the specific clustering algorithm, and the adapted matrices can be straightforwardly combined with the clustering algorithm of choice. Klein et al. (2002) handle must-link constraints by modifying the input matrix D in the following way: if two instances i and j are linked by a must-link constraint, then the corresponding entry $D_{i,j}$ is set to zero, which forces i and j to be grouped into the same cluster by the underlying clustering algorithm. In addition, distance scores for instances in the neighborhood of i or j are affected: if the distance is reduced for one pair of instances, triangle-inequality may be violated. An all-pairs-shortest-path algorithm propagates must-link constraints to other instances in D that restores triangle inequality.

We use a modified version of this approach. First, as the crowdsourced information may not be completely reliable, the clustering algorithm should be able to override it. We thus do not set $D_{i,j}$ to zero but rather to a small constant value d, that is the smallest non-identity distance value occurring in the matrix. Second, we exploit the

[1] This annotation was rather costly, so each scenario was aligned by one annotator only.

inherent transitivity of paraphrase judgments to derive additional constraints: if (*i*, *j*) and (*j*, *k*) are must links, we assume the pair (*i*, *k*) to be a must link as well, and set the distance to d. After the additional constraints are derived, the all-pairs-shortest-path algorithm is applied to the input matrix as in Klein et al. (2002).

We experimented with various state-of-art clustering algorithms including Spectral Clustering and Affinity Propagation (AP). The results presented in section 4 are based on AP, which proved to be most stable and provided the best results.

Determining the number of event-clusters. AP uses a parameter p, which influences the cluster granularity without determining the exact number of clusters beforehand. There is considerable variation between the optimal number of clusters between scenarios, depending on how many event types are required to describe the respective activity patterns (see Section 2). We use an unsupervised method for estimating scenario-specific settings of p, using the mean Silhouette Coefficient (Rousseeuw, 1987). This measure balances optimal inter-cluster tightness and intra-cluster distance, making sure that the elements of each cluster are as similar as possible to each other, and as dissimilar as possible to the elements of all other clusters. We run the unsupervised AP algorithm for each scenario with different settings of p and select the number resulting in the highest total Silhouette Coefficient as the optimal value for p.

3.2 Similarity Features

We now describe how we combine semantic and positional similarity information to obtain the distance measure that captures the similarities between event descriptions.

3.2.1 Semantic Similarity

We inspect different models for word-level similarity, as well as methods of deriving phrase-level semantic similarity from word-level similarity. We use pre-trained Word2Vec (w2v) word vectors (Mikolov et al., 2013) and vector representations (rNN) by Tilk et al. (2016) to obtain word-level similarity information. The rNN vectors are obtained from a neural network trained on large amounts of automatically role-labeled text and capture different aspects of word-level similarity than the w2v representations. We also experimented with WordNet/Lin similarity (Lin, 1998), but an ablation test (see below) showed that it was not useful.

To derive phrase-level similarity from word-level similarity, we employ the following three different empirically informed methods:

Centroid-based similarity. This method derives a phrase-level vector for an event description by taking the centroid over the word vectors of all content words in the event description. Similarity is computed using cosine.

Alignment-based similarity. Following RKP, we compute a similarity score for a pair of event descriptions by a linear combination of (a) the similarity of the head verbs of the two event descriptions and (b) the total score of the alignments between all noun phrases in the two descriptions, as computed by the Hungarian algorithm (Papadimitriou and Steiglitz, 1982).

Vocabulary similarity. We use the approach in Fernando and Stevenson (2008) to detect paraphrases and calculate semantic similarities between two event descriptions p_1 and p_2 as:

$$sim_{vocab}(\vec{p_1}, \vec{p_2}) = \frac{\vec{p_1} W \vec{p_2}^T}{|\vec{p_1}|\,|\vec{p_2}|} \quad (1)$$

where W is an $n \times n$ matrix that holds the similarities between all the words (vocabulary) in the two event descriptions being compared, n being the length of the vocabulary, and $\vec{p_1}$ and $\vec{p_2}$ are binary vectors representing the presence or absence of the words in the vocabulary.

Combining these three methods with the three word-level similarity measures we obtained a total of 8 different features[2].

3.2.2 Positional Similarity Feature

In addition to the semantic similarity features described above, we also used information about the position in which an event description occurs in an ESD. The basic idea here is that similar event descriptions tend to occur in similar (relative) positions in the ESDs. We set:

$$sim_{pos}(n_1, n_2) = 1 - abs\left(\frac{n_1}{T_1} - \frac{n_2}{T_2}\right) \quad (2)$$

where $n_1$and n_2 are the positions of the two event description and T_1 and T_2 represent the total number of event descriptions in the respective ESDs.

[2] The *centroid* method can not be combined with Lin similarity.

3.2.3 Combination

We linearly combine our 9 similarity features into a single similarity score, where the weights of the individual features are determined using logistic regression trained (10-fold cross validation) on the paraphrases from the 4 scenarios in the RKP development set (see Section 2). We run an ablation test by considering all possible subsets of features and using the 10 scenarios in the RKP test set, and found that the combination of the following five features performed best:

- centroid-based, alignment-based and vocabulary similarity with w2v vectors
- centroid-based similarity with rNN vectors
- position similarity

3.3 Temporal Script Graphs

After clustering the event descriptions of a given scenario into sets representing the scenario-specific event-types, we build a Temporal Script Graph (TSG) by determining the prototypical order between them. The nodes of the graph are the event types (clusters); an edge from a cluster E to a cluster E' indicates that E typically precedes E'. We induce the edges as follows. We say that an ESD *supports* $E \rightarrow E'$ if there are event descriptions $e \in E$ and $e' \in E'$ such that e precedes e' in the ESD. In a first step, we add an edge $E \rightarrow E'$ to the graph if there are more ESDs that support $E \rightarrow E'$ than $E' \rightarrow E$. In a second step, we compute transitive closure, i.e. we infer an edge $E \rightarrow E'$ in cases where there are clusters E, E', E'' such that $E \rightarrow E''$ and $E'' \rightarrow E'$. Finally, we form "arbitrary order" equivalence classes from those pairs of event clusters which have an equal number of supporting ESDs in either direction and are not yet connected by a directed temporal precedence edge.

This is an extension of the concept of a temporal script graph used in RKP, in order to allow for the flexible event order assumed by our approach. For example, the event descrpitions *preheat the oven* and *mixing ingredients* from the BAKING A CAKE scenario are likely to occur in different clusters, which are members of the same equivalence class, expressing that the event descriptions are not paraphrases, but may occur in any order.

4 Evaluation

4.1 Experimental Setup

We applied different versions of our clustering algorithm to the SMILE+OMICS dataset. In particular, we explored the influence of positional similarity, of the number of seeds (from 0 to 3%), as well as the proportion of the two seed types (outlier vs. stable). As a baseline, we ran the unsupervised clustering algorithm based on semantic similarity only. We evaluated the models on the tasks of event-type induction, paraphrase detection, and temporal order prediction, using the respective gold standard datasets (see Section 2).

Cluster quality. First, we evaluated the quality of the induced event types (i.e. sets of event descriptions) against the SMILE+OMICS gold clusters. We used the B-Cubed metric (Bagga and Baldwin, 1998), which is calculated by averaging per-element precision and recall scores. Amigó et al. (2009) showed B-Cubed to be the metric that appropriately captures all aspects of measuring cluster quality.

Paraphrase detection. For direct comparison with previous work, we tested our model on RKP's binary paraphrase detection task. The model classifies two event descriptions as paraphrases if they end up in the same cluster. We computed standard precision, recall and F-score by checking our classification against the RKP paraphrase dataset.

Temporal order prediction. We tested the quality of the temporal-order relation of the induced TSG structures using the RKP temporal order dataset as follows. For a pair of event descriptions (e, e'), we assume that (1) e precedes e', but not the other way round, if $e \in E$ and $e' \in E'$ for two different clusters E and E' such that $E \rightarrow E'$. (2) e precedes e' *and* vice versa (that is, both event orderings are possible), if $e \in E$ and $e' \in E'$, and E and E' are different clusters, but part of the same equivalence set. In all other cases (i.e. if e and e' are members of the same cluster), we assume that precedence does not hold. We computed standard precision, recall and F-score by checking our classification against the RKP temporal order dataset.

4.2 Results

The main results of our evaluation are shown in Table 1. The last three rows show results for

	Clustering	Paraphrasing			Temporal Ordering		
Model	B-Cubed	Precision	Recall	F-score	Precision	Recall	F-score
Regneri et al. (2010)	–	0.645	**0.833**	0.716	0.658	0.786	0.706
Modi and Titov (2014)	–	–	–	0.645	0.839	**0.843**	**0.841**
Frermann et al. (2014)	–	0.743	0.658	0.689	0.85	0.717	0.776
Baseline: USC	0.525	0.738	0.593	0.646	0.736	0.712	0.722
USC+Position	0.531	0.76	0.623	0.675	0.789	0.766	0.775
SSC+Outlier	0.635	0.781	0.751	0.756	0.858	0.791	**0.822**
SSC+Mixed	**0.655**	**0.796**	0.756	**0.764**	**0.865**	0.784	**0.822**

Table 1: Results on the clustering, paraphrasing and temporal ordering tasks for state-of-the-art models, our unsupervised (USC) and semi-supervised clustering approaches (SSC)

three of our model variants: unsupervised clustering with both semantic and positional information (USC+Position), semi-supervised clustering with positional information and only outlier constraints (3%, SSC+Outlier) and with the best-performing ratio of constraint types (SSC+Mixed, with 2% outliers and 1% stable cases). Row 4 shows the results for our unsupervised clustering baseline with semantic similarity only (USC).

For comparison against previous work, we added the results on the paraphrase and temporal ordering tasks of the MSA model by RKP, the Hierarchical Bayesian model by Fremann et al. (2014) and the Event Embedding model by Modi and Titov (2014) (for details about the latter, see Section 6).

On all three tasks, our best-performing model is SSC with mixed seed data (SSC+Mixed). Our best model outperforms the unsupervised model in RKP by 4.8 points (F-score) on the paraphrasing and by 11.6 points on the temporal ordering task. Interestingly, the performance gain is exclusively due to an increase of precision in both tasks (15.1 and 20.7 points, respectively). Our system comes close, but does not beat Modi and Titov (2014) on their unsupervised state-of-the-art model for temporal ordering, but outperforms it on the paraphrase task by almost 12 points F-score. The use of both positional information and mixed seed data in the distance measure has substantial effects on the quality of the results, improving on the unsupervised clustering baseline and reaching state-of-the-art results.

4.3 Discussion

The largest and most consistent performance gain of our model is due to use of crowdsourced alignment information.

undress
take off all your clothes, take off clothes, take clothes off, remove clothes, get undressed, disrobe, peeloff dresses, remove the clothes, *place clothes in hanger*, take off close, remove clothing, undress, i take off my clothes

adjust temp.
adjust temperature to your liking, wait for hot water, set handle to correct temperature, check the temperature, *adjust the position of shower*, feel for good temperature, adjust water temperature, adjust for desired temperature

soap body
scrub soap over your body, *wash the rest of the body*, repeat soaping body, *wash the body*, *wet entire body*, wash body with soap, soap body with soap or gel, put bath soap on body, *scrub the body parts*, apply soap, clean body with them,wash body with soap soap body, *add soap to cloth as needed*, i soap up,

dress
put clothes back on, get dressed,apply clothes, put clothes on,put on clothes, dress in clean clothing

Figure 2: Example clusters output by our model for TAKING A SHOWER.

Fig. 2 shows example clusters with script-specific paraphrases captured by our best model for the TAKING A SHOWER scenario. The model was able to capture a wide variety of lexical realizations of `undress`, including *peel off clothes, disrobe, remove clothes* etc., and similarly for `dress`, where we get *get dressed, apply clothes, put on clothes*, while these ended up in different clusters in the baseline model (e.g. *get dressed* was clustered together with *shampoo hair* cluster). There are still some incorrect classifications (indicated with italics in Fig. 2); note that these are often near misses rather than blatant errors.

Positional information substantially contributes to the quality of the derived TSGs. While the model using semantic similarity features only put *peel off dresses* in the `dress` cluster, positional similarity helped placing it correctly in the undress cluster, as it appears in the initial segment of its ESD. Positional information sometimes also caused wrong clustering decisions: *place cloth in hanger* typically occurs directly after undressing, and thus ended up in the undress cluster.

As described above, we collected alignments for outliers and for stable cases and tried several outlier-to-stable ratios. Outliers were much more effective than stable cases, as they improved recall by adjusting cluster boundaries to include scenario-specific functional paraphrases that were semantically dissimilar. Interestingly, adding a small number of stable cases leads to a slight improvement, but adding more stable cases leads to a performance drop, and using only stable cases does not improve the unsupervised baseline at all. Fig. 3 shows how the model improves as more constraints are added.

We tried to reduce the amount of manual annotation in several ways. The decision to derive additional must-links using transitivity paid off: F-score consistently improves by about 1 point F-score. To further increase the set of seeds, we experimented with propagating the links to nearest neighbors of aligned event descriptions, but did not see an improvement. Also, we tried to use alignments obtained by majority vote, which however led to a performance drop, showing that using high quality seeds is crucial.

To make sure that our results are not dependent on the selection of a specific scenario set, we evaluated our model also on the DeScript gold clusters. The results were comparable: B-Cubed improved from 0.551 (RKP: 0.525) to 0.662 (RKP: 0.655). As the DeScript corpus provides 100 ESDs per scenario, we were also able to test whether an increased number of input ESDs also improves clustering performance. We observed no effect with 50 ESDs compared to our model using 25 ESDs, and only a slight (less than 1 point) improvement with the full 100 ESDs dataset.

A leading motivation to use clustering instead of MSA was the opportunity to model flexible event order in script structures. Our expectations were confirmed by the evaluation results. A closer look at the induces TSGs (as shown by the example TSG in Fig. 4), suggests that our system makes extensive use of the option of flexible event ordering.

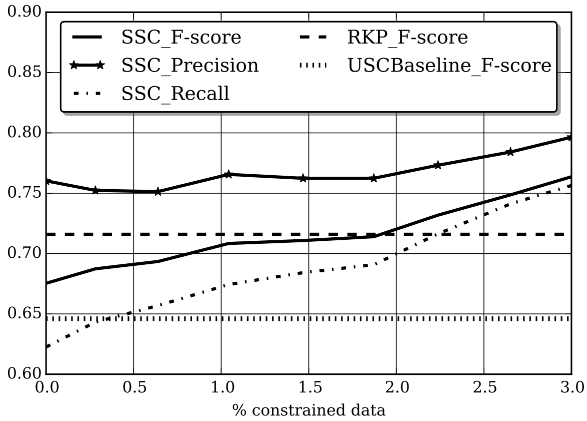


Figure 3: Paraphrase detection results for RKP, for our Unsupervised baseline (USC) and for our best Semi-supervised model (SSC+Mixed)

enter bathroom ⇒(turn on shower↔ undress)
⇒(adjust temp.↔ turn off water)⇒get in shower
⇒(soap body↔ close curtains)⇒shampoo hair
⇒(wash hair↔ wash body↔ shave)⇒rinse
⇒exit shower⇒dry off⇒dress

Figure 4: Example TSG for TAKING A SHOWER. The arrows stand for default temporal precedence, the parentheses enclose equivalence classes expressing arbitrary temporal order.

5 Costs and Coverage

We have demonstrated that semi-supervised clustering enables the extraction of script knowledge with substantially higher quality than existing methods. But how does the method scale? Can we expect to obtain a script knowledge database with sufficiently wide coverage at reasonable costs?

The process of script extraction requires crowdsourced data in terms of (1) ESDs and (2) seed alignments. To complete 3%+3% high-quality alignments for the 10 DeScript scenarios via Mechanical Turk (that is, 3% stable cases and 3% outliers), workers spent a total of 37.5 hours, with an average of 3.75 hrs per scenario, ranging from 2.5 (GOING GROCERY SHOPPING) to 7.52 hrs (BAKING A CAKE)[3]. It took on average 2.78 hrs to collect 25 scenario-specific ESDs, that is 6.53 hrs of data acquisition time per scenario.

The costs per scenario are moderate. But how many scenarios must be modeled to achieve suf-

[3] We used the DeScript scenarios as reference because they come with equal numbers of ESDs, while the ESD sets in the SMILE+OMICS corpus vary considerably in size.

Jessica needs milk. Jessica wakes up and wants to eat breakfast. She grabs the cereal and pours some into a bowl. She looks in the fridge for milk. There is no milk in the fridge so she can't eat her breakfast. She goes to the store to buy some milk comes home and eats breakfast.

MAKE BREAKFAST: **C**

GOING GROCERY SHOPPING: **P**

Figure 5: Example ROC-story with scenario annotation.

ficient coverage for the analysis of script knowledge in natural-language texts? Answering this question is not trivial, as scenarios vary considerably in granularity and it is not trivial that the type of script knowledge we model can capture all kinds of event structures, even in narrative texts. In order to provide a rough estimate of coverage for the currently existing script material, we carried out a simple annotation study on the recently published ROC-stories database (Mostafazadeh et al., 2016a). The database consists of 50,000 short narrative texts, collected via Mechanical Turk. Workers were asked to write a 5-sentence length story about an everyday commonsense event, and they were encouraged to write about "anything they have in mind" to guarantee wide distribution across topics.

For our annotation study, we merged the available datasets containing crowdsourced ESD collections (i.e. OMICS, SMILE, and DeScript), excluding two extremely general scenarios (GO OUTSIDE, CHILDHOOD), which gives us a total of 226 different scenarios.

We randomly selected 500 of the ROC-stories and asked annotators to determine for each story which scenario (if any) was centrally addressed and which scenarios were just referred to or partially addressed with at least one event mention, and to label them with "C" and "P", respectively. See an example story with its annotation in Fig. 5.

Each story was annotated by three students of computational linguistics. To facilitate annotation, the stories were presented alongside ten scenarios whose ESDs showed strongest lexical overlap with the story (calculated as tf-idf). However, annotators were expected to consider the full scenario list[4]. The three annotations were merged using majority vote. Cases without a clear majority vote containing one single "C" assignment were inspected and adjudicated by the authors of the paper.

26.4% of the stories were judged to centrally refer to one of the scenarios[5]. Although this percentage cannot be directly translated to coverage values, it indicates that the extraction method presented in this paper has the strong potential to provide a script knowledge resource with reasonable costs, which can substantially contribute to the task of text understanding.

6 Related Work

Following the seminal work of Chambers and Jurafsky (2008) and (2009) on the induction of script-like *narrative schemas* from large, unlabeled corpora of news articles, a series of models have been presented for improving the induction method or explore alternative data sources for script learning. Gordon (2010) mined commonsense knowledge from stories describing events in day-to-day life. Jans et al. (2012) studied different ways of selecting event chains and used skipgrams for computing event statistics. Pichotta and Mooney (2014) employed richer event representations, exploiting the interactions between multiple arguments to extract event sequences from a large corpus. Rahimtoroghi et al. (2016) learned contingency relations between events from a corpus of blog posts. All these approaches aim at high recall, resulting in a large amount of wide-coverage, but noisy schemas.

Abend et al. (2015) proposed an edge-factored model to determine the temporal order of events in cooking recipes, but their model is limited to scenarios with an underlying linear order of events. Bosselut et al. (2016) induce prototypical event structure in an unsupervised way from a large collection of photo albums with time-stamped images and captions. This method is however limited by the availability of albums for "special" events such as WEDDING or BARBECUE, in contrast to everyday, trivial activites such as MAKING COFFEE or

[4] We are aware that this setup may bias participants toward finding a scenario from our collection, leading to an increase in recall. However, they had the option to label stories where they felt a scenario was only partially addressed in a different way, thus setting these cases apart from those where the scenario was centrally addressed.

[5] While we take the judgment about the "C" class to be quite reliable (24.8% qualified by majority vote, only 1.6 % were added via adjudication), there was considerable confusion about the "P" label. So we decided not to use the "P" label at all.

GOING TO THE DENTIST. Mostafazadeh et al. (2016b) presented the ROC-stories, a dataset of c.a. 50.000 crowdsourced short commonsense everyday story. They propose to use it for the evaluation of script knowledge models, and it may also turn out to be a valuable resource for script learning, although to our knowledge this has not yet been attempted.

Closest to our approach is the work by RKP and subsequent work by Frermann et al. (2014) and Modi and Titov (2014). All these employ the same SMILE+OMICS dataset for evaluation, which we also used to allow for a direct comparison. Frermann et al. (2014) present a Bayesian generative model for joint learning of event types and ordering constraints. Their model promisingly shows that flexible event order in scripts can be suitably modelled. Modi and Titov (2014) focussed mainly on event ordering between script-related predicates, using distributed representations of predicates and arguments induced by a statistical model. They obtained paraphrase sets as a by-product, namely by creating an event timeline and grouping together event mentions corresponding to the same interval.

7 Conclusions

This paper presents a clustering-based approach to inducing script structure from crowdsourced descriptions of scenarios. We use semi-supervised clustering to group individual event descriptions into paraphrase sets representing event types, and induce a temporal order among them. Crowdsourced alignments between event descriptions proved highly effective as seed data. On a paraphrase task, our approach outperforms all previous proposals, while still performing very well on the task of temporal order prediction. A study on the ROC-stories suggests that a model of script knowledge created with our method can cover a large fraction of event structures occurring in topically unrestricted narrative text, thus demonstrating the scalability of our approach.

Acknowledgments

This research was funded by the German Research Foundation (DFG) as part of SFB 1102 'Information Density and Linguistic Encoding'.

References

Omri Abend, Shay B. Cohen, and Mark Steedman. 2015. Lexical event ordering with an edge-factored model. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1161–1171, Denver, Colorado, May–June. Association for Computational Linguistics.

Enrique Amigó, Julio Gonzalo, Javier Artiles, and Felisa Verdejo. 2009. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Information retrieval*, 12(4):461–486.

Amit Bagga and Breck Baldwin. 1998. Algorithms for scoring coreference chains. In *The First International Conference on Language Resources and Evaluation Workshop on Linguistics Coreference*, pages 563–566.

Avron Barr and Edward A. Feigenbaum. 1981. Frames and scripts. In *The Handbook of Artificial Intelligence*, volume 3, pages 216–222. Addison-Wesley, California.

Antoine Bosselut, Jianfu Chen, David Warren, Hannaneh Hajishirzi, and Yejin Choi. 2016. Learning prototypical event structure from photo albums. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1769–1779, Berlin, Germany, August. Association for Computational Linguistics.

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

Richard E. Cullingford. 1977. *Script Application: Computer Understanding of Newspaper Stories*. Ph.D. thesis, Yale University.

Richard Durbin, Sean Eddy, Anders Krogh, and Graeme Mitchison. 1998. *Biological sequence analysis: probabilistic models of proteins and nucleic acids*. Cambridge University Press, Cambridge, UK.

Samuel Fernando and Mark Stevenson. 2008. A semantic similarity approach to paraphrase detection. In *Proceedings of the 11th Annual Research Colloquium of the UK Special Interest Group for Computational Linguistics*.

Lea Frermann, Ivan Titov, and Manfred Pinkal. 2014. A hierarchical bayesian model for unsupervised induction of script knowledge. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 49–57, Gothenburg, Sweden, April. Association for Computational Linguistics.

Andrew S. Gordon. 2010. Mining commonsense knowledge from personal stories in internet weblogs. In *Proceedings of the First Workshop on Automated Knowledge Base Construction*, Grenoble, France.

Rakesh Gupta and Mykel J. Kochenderfer. 2004. Common sense data acquisition for indoor mobile robots. In *Proceedings of the 19th National Conference on Artificial intelligence*, pages 605–610. AAAI Press.

Hannaneh Hajishirzi and Erik T. Mueller. 2012. Question answering in natural language narratives using symbolic probabilistic reasoning. In *Proceedings of the Twenty-Fifth International Florida Artificial Intelligence Research Society Conference*, pages 38–43.

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, April. Association for Computational Linguistics.

Dan Klein, Sepandar D. Kamvar, and Christopher D. Manning. 2002. From instance-level constraints to space-level constraints: Making the most of prior knowledge in data clustering. In *Proceedings of the Nineteenth International Conference on Machine Learning*, pages 307–314.

Dekang Lin. 1998. An information-theoretic definition of similarity. In *In Proceedings of the 15th International Conference on Machine Learning*, pages 296–304.

Tomas Mikolov, Kai Chen, Greg S. Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *Proceedings of the International Conference on Learning Representations*.

Ashutosh Modi and Ivan Titov. 2014. Inducing neural models of script knowledge. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, pages 49–57, Ann Arbor, Michigan, June. Association for Computational Linguistics.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016a. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 839–849, San Diego, California, June. Association for Computational Linguistics.

Nasrin Mostafazadeh, Lucy Vanderwende, Wen-tau Yih, Pushmeet Kohli, and James Allen. 2016b. Story cloze evaluator: Vector space representation evaluation by predicting what happens next. In *Proceedings of the 1st Workshop on Evaluating Vector Space Representations for NLP*, pages 24–29.

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

Christos H. Papadimitriou and Kenneth Steiglitz. 1982. *Combinatorial Optimization: Algorithm und Complexity*. Dover Publications, Mineola, NY.

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, April. Association for Computational Linguistics.

Elahe Rahimtoroghi, Ernesto Hernandez, and Marilyn Walker. 2016. Learning fine-grained knowledge about contingent relations between everyday events. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 350–359.

Altaf Rahman and Vincent Ng. 2011. Narrowing the modeling gap: a cluster-ranking approach to coreference resolution. *Journal of Artificial Intelligence Research*, 40:469–521.

Lisa F. Rau, Paul S. Jacobs, and Uri Zernik. 1989. Information extraction and text summarization using linguistic knowledge acquisition. *Information Processing & Management*, 25(4):419–428.

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, Uppsala, Sweden, July. Association for Computational Linguistics.

Peter J. Rousseeuw. 1987. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53 – 65.

Roger C. Schank and Robert P. Abelson. 1977. *Scripts, Plans, Goals, and Understanding: An Inquiry into Human Knowledge Structures*. Erlbaum, Hillsdale, NJ.

Ottokar Tilk, Vera Demberg, Asad Sayeed, Dietrich Klakow, and Stefan Thater. 2016. Event participant modelling with neural networks. In *Proceedings of the 2016 Conference on Empirical Methods* *in Natural Language Processing*, pages 171–182, Austin, Texas, November. Association for Computational Linguistics.

Lilian D. A. Wanzare, Alessandra Zarcone, Stefan Thater, and Manfred Pinkal. 2016. A crowdsourced database of event sequence descriptions for the acquisition of high-quality script knowledge. In N. Calzolari (Conference Chair), K. Choukri, T. Declerck, S. Goggi, M. Grobelnik, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, and S. Piperidis, editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*. European Language Resources Association (ELRA).