Detecting Everyday Scenarios in Narrative Texts

Lilian D. A. Wanzare[1] **Michael Roth**[2] **Manfred Pinkal**[1]
Universität des Saarlandes[1] Universität Stuttgart[2]
{wanzare,pinkal}coli.uni-saarland.de rothml@ims.uni-stuttgart.de

Abstract

Script knowledge consists of detailed information on everyday activities. Such information is often taken for granted in text and needs to be inferred by readers. Therefore, script knowledge is a central component to language comprehension. Previous work on representing scripts is mostly based on extensive manual work or limited to scenarios that can be found with sufficient redundancy in large corpora. We introduce the task of *scenario detection*, in which we identify references to scripts. In this task, we address a wide range of different scripts (200 scenarios) and we attempt to identify all references to them in a collection of narrative texts. We present a first benchmark data set and a baseline model that tackles scenario detection using techniques from topic segmentation and text classification.

1 Introduction

According to Grice's (1975) theory of pragmatics, people tend to omit basic information when participating in a conversation (or writing a story) under the assumption that left out details are already known or can be inferred from commonsense knowledge by the hearer (or reader). Consider the following text fragment about `eating in a restaurant` from an online blog post:

Example 1.1 *(. . .) we drove to Sham Shui Po and looked for a place to eat. (. . .) [O]ne of the restaurants was fully seated [so we] chose another. We had 4 dishes—Cow tripe stir fried with shallots, ginger and chili. 1000-year-old-egg with watercress and omelet. Then another kind of tripe and egg—all crispy on the top and soft on the inside. Finally calamari stir fried with rock salt and chili. Washed down with beers and tea at the end. (. . .)*

The text in Example 1.1 obviously talks about a restaurant visit, but it omits many events that are involved while `eating in a restaurant`, such as *finding a table*, *sitting down*, *ordering food* etc., as well as participants such as *the waiter*, *the menu*,*the bill*. A human reader of the story will naturally assume that all these ingredients have their place in the reported event, based on their commonsense knowledge, although the text leaves them completely implicit. For text understanding machines that lack appropriate common-sense knowledge, the implicitness however poses a non-trivial challenge.

Writing and understanding of narrative texts makes particular use of a specific kind of commonsense knowledge, referred to as *script knowledge* (Schank and Abelson, 1977). Script knowledge is about prototypical everyday activity, called *scenarios*. Given a specific scenario, the associated script knowledge enables us to infer omitted events that happen before and after an explicitly mentioned event, as well as its associated participants. In other words, this knowledge can help us obtain more complete text representations, as required for many language comprehension tasks.

There has been some work on script parsing (Ostermann et al., 2017, 2018c), i.e., associating texts with script structure given a specific scenario. Unfortunately, only limited previous work exists on determining which scenarios are referred to in a text or text segment (see Section 2). To the best of our knowledge, this is the first dataset of narrative texts which have annotations at sentence level according to the scripts they instantiate.

In this paper, we describe first steps towards the automatic detection and labeling of scenario-specific text segments. Our contributions are as follows:

- We define the task of scenario detection and introduce a benchmark dataset of annotated narrative texts, with segments labeled according to the scripts they in-

Proceedings of the Second Storytelling Workshop, pages 90–106
Florence, Italy, August 1, 2019. ©2019 Association for Computational Linguistics

stantiate (Section 3). To the best of our knowledge, this is the first dataset of its kind. The corpus is publicly available for scientific research purposes at this http://www.sfb1102.uni-saarland.de/?page_id=2582.

- As a benchmark model for scenario detection, we present a two-stage model that combines established methods from topic segmentation and text classification (Section 4).
- Finally, we show that the proposed model achieves promising results but also reveals some of the difficulties underlying the task of scenario detection (Section 5).

2 Motivation and Background

A major line of research has focused on identifying specific events across documents, for example, as part of the Topic Detection and Tracking (TDT) initiative (Allan et al., 1998; Allan, 2012). The main subject of the TDT intiative are instances of world events such as *Cuban Riots in Panama*. In contrast, everyday scenarios and associated sequences of event types, as dealt with in this paper, have so far only been the subject of individual research efforts focusing either on acquiring script knowledge, constructing story corpora, or script-related downstream tasks. Below we describe significant previous work in these areas in more detail.

Script knowledge. Scripts are descriptions of prototypical everyday activities such as `eating in a restaurant` or `riding a bus` (Schank and Abelson, 1977). Different lines of research attempt to acquire script knowledge. Early researchers attempted to handcraft script knowledge (Mueller, 1999; Gordon, 2001). Another line of research focuses on the collection of scenario-specific script knowledge in form of event sequence descriptions (ESDs) via crowdsourcing, (Singh et al., 2002; Gupta and Kochenderfer, 2004; Li et al., 2012; Raisig et al., 2009; Regneri et al., 2010; Wanzare et al., 2016)). ESDs are sequences of short sentences, in bullet style, describing how a given scenario is typically realized. The top part of Table 1 summarizes various script knowledge-bases (ESDs). While datasets like OMICS seem large, they focus only on mundane indoor scenarios (e.g. `open door, switch off lights`). A third line of research tries to leverage existing large text corpora to induce script-like knowledge about the topics represented in these corpora. For instance, Chambers and Jurafsky (2008, 2009); Pichotta and Mooney (2014) leverage newswire texts, Manshadi et al. (2008); Gordon (2010); Rudinger et al. (2015); Tandon et al. (2014, 2017) leverage web articles while Ryu et al. (2010); Abend et al. (2015); Chu et al. (2017) leverage organized procedural knowledge (e.g. from eHow.com, wikiHow.com).

The top part of Table 1 summarizes various script knowledge-bases. Our work lies in between both lines of research and may help to connect them: we take an extended set of specific scenarios as a starting point and attempt to identify instances of those scenarios in a large-scale collection of narrative texts.

Textual resources. Previous work created script-related resources by crowdsourcing stories that instantiate script knowledge of specific scenarios. For example, Modi et al. (2016) and Ostermann et al. (2018a, 2019) asked crowd-workers to write stories that include mundane aspects of scripts "as if explaining to a child". The collected datasets, *InScript* and *MCScript*, are useful as training instances of narrative texts that refer to scripts. However, the texts are kind of unnatural and atypical because of their explicitness and the requirement to workers to tell a story that is related to one single scenario only. Gordon and Swanson (2009) employed statistical text classification in order to collect narrative texts about personal stories. The *Spinn3r*[1] dataset (Burton et al., 2009) contains about 1.5 Million stories. *Spinn3r* has been used to extract script information (Rahimtoroghi et al., 2016, see below). In this paper, we use the Spinn3r personal stories corpus as a source for our data collection and annotation. The bottom part of Table 1 summarizes various script-related resources. The large datasets come with no scenarios labels while the crowdsourced datasets only have scenario labels at story level. Our work provides a more fine grained scenario labeling at sentence level.

Script-related tasks. Several tasks have been proposed that require or test computational models of script knowledge. For example, Kasch and Oates (2010) and Rahimtoroghi et al. (2016) propose and evaluate a method that automatically creates event schemas, extracted from scenario-specific texts. Ostermann et al. (2017) attempt to iden-

[1] http://www.icwsm.org/data/

Scenario ESD collections	Scenarios	# ESDs
SMILE (Regneri et al., 2010)	22	386
Cooking (Regneri, 2013)	53	2500
OMICS (Singh et al., 2002)	175	9044
Raisig et al. (2009)	30	450
Li et al. (2012)	9	500
DeScript (Wanzare et al., 2016)	40	4000

Story Corpora	Scenarios	# stories	Classes	Segs.
Modi et al. (2016)	10	1000	✓	✗
Ostermann et al. (2019)	200	4000	✓	✗
Rahimtoroghi et al. (2016)	2	660	✓	✗
Mostafazadeh et al. (2016)	✗	~50000	✗	✗
Gordon and Swanson (2009)	✗	~1.5M	✗	✗
This work	200	504	✓	✓

Table 1: Top part shows scenario collections and number of associated event sequence descriptions (ESDs). Bottom part lists story corpora together with the number of stories and different scenarios covered. The last two columns indicate whether the stories are classified and segmented, respectively.

tify and label mentions of events from specific scenarios in corresponding texts. Finally, Ostermann et al. (2018b) present an end-to-end evaluation framework that assesses the performance of machine comprehension models using script knowledge. Scenario detection is a prerequisite for tackling such tasks, because the application of script knowledge requires awareness of the scenario a text segment is about.

3 Task and Data

We define scenario detection as the task of identifying segments of a text that are about a specific scenario and classifying these segments accordingly. For the purpose of this task, we view a segment as a consecutive part of text that consists of one or more sentences. Each segment can be assigned none, one, or multiple labels.

Scenario labels. As a set of target labels, we collected scenarios from all scenario lists available in the literature (see Table 1). During revision, we discarded scenarios that are too vague and general (e.g. `childhood`) or atomic (e.g. `switch on/off lights`), admitting only reasonably structured activities. Based on a sample annotation of Spinn3r stories, we further added 58 new scenarios, e.g. `attending a court hearing`, `going skiing`, to increase coverage. We deliberately included narrowly related scenarios that stand in the relation of specialisation (e.g. `going shopping` and `shopping for clothes`, or in a subscript relation (`flying in an airplane` and `checking in at the airport`). These cases are challenging to annotators because segments may refer to different scenarios at the same time.

Although our scenario list is incomplete, it is representative for the structural problems that can occur during annotation. We have scenarios that have varying degrees of complexity and cover a wide range of everyday activities. The complete list of scenarios[2] is provided in Appendix B.

Dataset. As a benchmark dataset, we annotated 504 texts from the Spinn3r corpus. To make sure that our dataset contains a sufficient number of relevant sentences, i.e., sentences that refer to scenarios from our collection, we selected texts that have a high affinity to at least one of these scenarios. We approximate this affinity using a logistic regression model fitted to texts from MCScript, based on LDA topics (Blei et al., 2003) as features to represent a document.

3.1 Annotation

We follow standard methodology for natural language annotation (Pustejovsky and Stubbs, 2012). Each text is independently annotated by two annotators, student assistants, who use an agreed upon set of guidelines that is built iteratively together with the annotators. For each text, the students had to identify segments referring to a scenario from the scenario list, and assign scenario labels. If a segment refers to more than one script, they were allowed to assign multiple labels. We worked with a total of four student assistants and used the Webanno[3] annotation tool (de Castilho et al., 2016).

The annotators labeled 504 documents, consisting of 10,754 sentences. On average, the annotated documents were 35.74 sentences long. A scenario label could be either one of our 200 scenarios or *None* to capture sentences that do not refer to any of our scenarios.

Guidelines. We developed a set of more detailed guidelines for handling different issues related to

[2] The scenario collection was jointly extended together with the authors of MCScript (Ostermann et al., 2018a, 2019). The same set was used in building MCScript 2.0 (Ostermann et al., 2019)

[3] https://webanno.github.io/webanno/

Annotators	2	3	4
1	0.57 (*0.65*)	0.63 (***0.72***)	**0.64** (*0.70*)
2		0.62 (*0.71*)	0.61 (*0.70*)
3			0.62 (*0.71*)

Table 2: Kappa (*and raw*) agreement between pairs of annotators on sentence-level scenario labels

Annotators	2	3	4
1	**78.8**	70.6	*59.3*
2		66.0	64.2
3			67.0

Table 3: Relative agreement on segment spans between annotated segments that overlap by at least one token and are assigned the same scenario label

the segmentation and classification, which is detailed in Appendix A. A major challenge when annotating segments is deciding when to count a sentence as referring to a particular scenario. For the task addressed here, we consider a segment only if it explicitly realizes aspects of script knowledge that go beyond an evoking expression (i.e., more than one event and participant need to be explicitly realized). Example 3.1 below shows a text segment with minimal scenario information for `going grocery shopping` with two events mentioned. In Example 3.2, only the evoking expression is mentioned, hence this example is not annotated.

Example 3.1 ✓`going grocery shopping`
...We also ***stopped at a small shop*** *near the hotel to* ***get some sandwiches*** *for dinner...*

Example 3.2 ✗`paying for gas`
... A customer was heading for the store to ***pay for gas*** *or whatever,...*

3.2 Statistics

Agreement. To measure agreement, we looked at sentence-wise label assignments for each double-annotated text. We counted agreement if the same scenario label is assigned to a sentence by both annotators. As an indication of chance-corrected agreement, we computed Kappa scores (Cohen, 1960). A kappa of 1 means that both annotators provided identical (sets of) scenario labels for each sentence. When calculating raw agreements, we counted agreement if there was at least one same scenario label assigned by both annotators. Table 2 shows the Kappa and raw (*in italics*) agreements for each pair of annotators. On average, the Kappa score was *0.61* ranging from 0.57 to 0.64. The average raw agreement score was *0.70* ranging from 0.65 to 0.72. The Kappa value indicates relatively consistent annotations across annotators even though the task was challenging.

We used fuzzy matching to calculate agreement in span between segments that overlap by at least one token. Table 3 shows pairwise % agreement scores between annotators. On average, the annotators achieve 67% agreement on segment spans. This shows considerable segment overlap when both annotators agreed that a particular scenario is referenced.

Analysis. Figure 1 shows to what extent the annotators agreed in the scenario labels. The *None* cases accounted for 32% of the sentences. Our scenario list is by far not complete. Although we selected stories with high affinity to our scenarios, other scenarios (not in our scenario list) may still occur in the stories. Sentences referring to other scenarios were annotated as *None* cases. The *None* label was also used to label sentences that described topics related to but not directly part of the script being referenced. For instance, sentences not part of the narration, but of a different discourse mode (e.g. argumentation, report) or sentences where no specific script events are mentioned[4]. About 20% of the sentences had *Single* annotations where only one annotator indicated that there was a scenario reference. 47% of the sentences were assigned some scenario label(s) by both annotators (*Identical, At least one, Different*). Less than 10% of the sentences had *Different* scenario labels for the case where both annotators assigned scenario labels to a sentence. This occurred frequently with scenarios that are closely related (e.g. `going to the shopping center`, `going shopping`) or scenarios in a sub-scenario relation (e.g. `flying in a plane`, `checking in at the airport`) that share script events and participants. In about 7% of the sentences, both annotators agreed on *At least one* scenario label. The remaining 30% of the sentences were assigned *Identical* (sets of) scenario labels by both annotators.

[4] See examples in Appendix A.

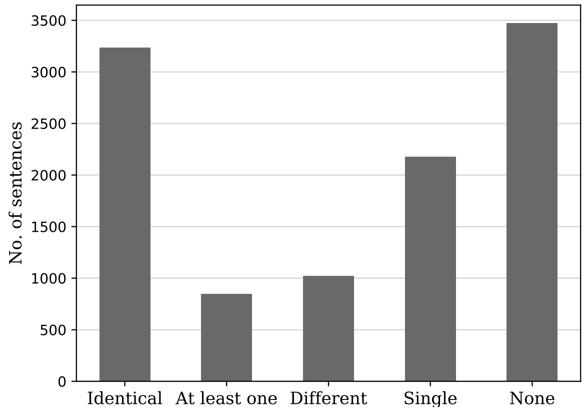

Figure 1: Absolute counts on sentence-level annotations that involve the same (*Identical*), overlapping (*At least one*) or disagreed (*Different*) labels; also shown are the number of sentences that received a label by only one annotator (*Single*) or no label at all (*None*).

3.3 Adjudication and Gold Standard

The annotation task is challenging, and so are gold standard creation and adjudication. We combined *automatic merging* and *manual adjudication* (by the main author of the paper) as two steps of gold-standard creation, to minimize manual post-processing of the dataset.

We automatically merged annotated segments that are identical or overlapping and have the same scenario label, thus maximizing segment length. Consider the two annotations shown in Example 3.3. One annotator labeled the whole text as `growing vegetables`, the other one identified the two bold-face sequences as `growing vegetables` instances, and left the middle part out. The result of the merger is the maximal `growing vegetables` chain, i.e., the full text. Taking the maximal chain ensures that all relevant information is included, although the annotators may not have agreed on what is script-relevant.

Example 3.3 `growing vegetables`
The tomato seedlings Mitch planted in the compost box have done really well and we noticed flowers on them today . Hopefully we will get a good *It has rained and rained here for the past month so that is doing the garden heaps of good . We bought some organic herbs seedlings recently and now have some thyme , parsley , oregano and mint growing in the garden .* ***We also planted some lettuce and a grape vine . We harvested our first crop of sweet potatoes a week or so ago (. . .)***

Scenario	# docs	# sents.	# segs.
eat in a restaurant	21	387	22
go on vacation	16	325	17
go shopping	34	276	35
take care of children	15	190	19
review movies	8	184	8
. . .			
taking a bath	3	34	6
borrow book from library	3	33	3
mow the lawn	3	33	3
drive a car	9	32	11
change a baby diaper	3	32	3
. . .			
replace a garbage bag	1	3	2
unclog the toilet	1	3	1
wash a cut	1	3	1
apply band aid	2	2	2
change batteries in alarm	1	2	1

Table 4: Distribution of scenario labels over documents (docs), sentences (sents) and segments (segs); the top and bottom parts show the ten most and least frequent labels, respectively. The middle part shows scenario labels that appear at an average frequency.

The adjudication guidelines were deliberately designed in a way that the adjudicator could not easily overrule the double-annotations. The segmentation could not be changed, and only the labels provided by the annotators were available for labeling. Since segment overlap is handled automatically, manual adjudication must only care about label disagreement: the two main cases are (1) a segment has been labeled by only one annotator and (2) a segment has been assigned different labels by its two annotators. In case (1), the adjudicator had to take a binary decision to accept the labeled segment, or to discard it. In case (2), the adjudicator had three options: to decide for one of the labels or to accept both of them.

Gold standard. The annotation process resulted in 2070 single segment annotations. 69% of the single segment annotations were automatically merged to create gold segments. The remaining segments were adjudicated, and relevant segments were added to the gold standard. Our final dataset consists of 7152 sentences (contained in 895 segments) with gold scenario labels. From the 7152 gold sentences, 1038 (15%) sentences have more than one scenario label. 181 scenarios (out of 200) occur as gold labels in our dataset, 179 of which are referred to in at least 2 sentences. Table 4 shows

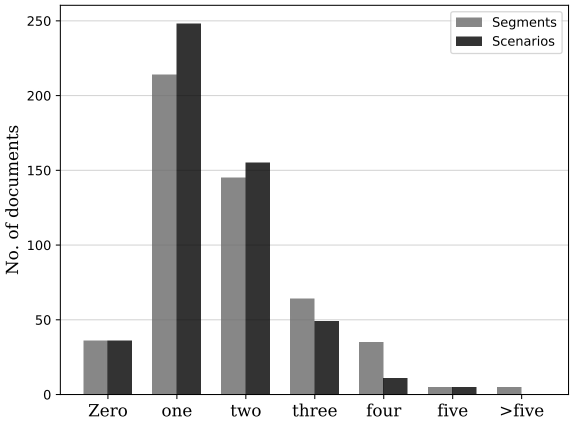

Figure 2: Segment and scenario distribution per text

example scenarios[5] and the distribution of scenario labels: the number of documents that refer to the given scenario, the number of gold sentences and segments referring to the given scenario, and the average segment length (in sentences) per scenario. 16 scenarios are referred to in more than 100 gold sentences, 105 scenarios in at least 20 gold sentences, 60 scenarios in less than 20 gold sentences. Figure 2 shows the distribution of segments and scenario references per text in the gold standard. On average, there are 1.8 segments per text and 44% of the texts refer to at least two scenarios.

4 Benchmark model

Texts typically consist of different passages that refer to different scenarios. When human hearers or readers come across an expression that evokes a particular script, they try to map verbs or clauses in the subsequent text to script events, until they face lexical material that is clearly unrelated to the script and may evoke a different scenario. Scenario identification, scenario segmentation, and script parsing are subtasks of story comprehension, which ideally work in close mutual interaction. In this section, we present a model for scenario identification, which is much simpler in several respects: we propose a two-step model consisting of a segmentation and a classification component. For segmentation, we assume that a change in scenario focus can be modeled by a shift in lexical cohesion. We identify segments that might be related to specific scripts or scenarios via topic segmentation, assuming that scenarios can be approximated as distributions over topics. After segmentation, a supervised classifier component is used to predict the scenario label(s) for each of the found segment. Our results show that the script segmentation problem can be solved in principle, and we propose our model as a benchmark model for future work.

Segmentation. The first component of our benchmark model reimplements a state-of-art unsupervised method for topic segmentation, called TopicTiling (Riedl and Biemann, 2012). TopicTiling (TT) uses latent topics inferred by a Latent Derichlet Allocation (LDA, Blei et al. (2003)) model to identify segments (i.e., sets of consecutive sentences) referring to similar topics.[6] The TT segmenter outputs topic boundaries between sentences where there are topic shifts. Boundaries are computed based on coherence scores. Coherence scores close to 1 indicate significant topic similarity while values close to 0 indicate minimal topic similarity. A window parameter is used to determine the block size i.e. the number of sentences to the left and right that should be considered when calculating coherence scores. To discover segment boundaries, all local minima in the coherence scores are identified using a depth score (Hearst, 1994). A threshold $\mu - \sigma/x$ is used to estimate the number of segments, where μ is the mean and σ is the standard deviation of the depth scores, and x is a weight parameter for setting the threshold.[7] Segment boundaries are placed at positions greater than the threshold.

Classification. We view the scenario classification subtask as a supervised multi-label classification problem. Specifically, we implement a multi-layer perceptron classifier in Keras (Chollet et al., 2015) with multiple layers: an input layer with 100 neurons and ReLU activation, followed by an intermediate layer with dropout (0.2), and finally an output layer with sigmoid activations. We optimize a cross-entropy loss using adam. Because multiple labels can be assigned to one segment, we train several one-vs-all classifiers, resulting in one classifier per scenario.

We also experimented with different features and feature combinations to represent text segments: term frequencies weighted by inverted document frequency (*tf.idf*, Salton and McGill (1986))[8] and topic features derived from LDA (see above), and

[5] The rest of the scenarios are listed in Appendix B

[6] We used the Gensim (Rehurek and Sojka, 2010) implementation of LDA.

[7] We experimentally set x to 0.1 using our held out development set.

[8] We use SciKit learn (Pedregosa et al., 2011) to build *tf.idf* representations

we tried to work with word embeddings. We found the performance with *tf.idf* features to be the best.

5 Experiments

The experiments and results presented in this section are based on our annotated dataset for scenario detection described in section 3.

5.1 Experimental setting

Preprocessing and model details. We represent each input to our model as a sequence of lemmatized content words, in particular nouns and verbs (including verb particles). This is achieved by preprocessing each text using Stanford CoreNLP (Chen and Manning, 2014).

Segmentation. Since the segmentation model is unsupervised, we can use all data from both MCScript and the Spinn3r personal stories corpora to build the LDA model. As input to the TopicTiling segmentor, each sentence is represented by a vector in which each component represents the (weight of a) topic from the LDA model (i.e. the value of the i^{th} component is the normalized weight of the words in the sentence whose most relevant topic is the i^{th} topic). For the segmentation model, we tune the number of topics (200) and the window size (2) based on an artificial development dataset, created by merging segments from multiple documents from MCScript.

Classification. We train the scenario classification model on the scenario labels provided in MCScript (one per text). For training and hyperparameter selection, we split MCScript dataset (see Section 2) into a training and development set, as indicated in Table 5. We additionally make use of 18 documents from our scenario detection data (Section 3) to tune a classification threshold. The remaining 486 documents are held out exclusively for testing (see Table 5). Since we train separate classifiers for each scenario (one-vs-all classifiers), we get a probability distribution of how likely a sentence refers to a scenario. We use entropy to measure the degree of scenario content in the sentences. Sentences with entropy values higher than the threshold are considered as not referencing any scenario (*None* cases), while sentences with lower entropy values reference some scenario.

Baselines. We experiment with three informed baselines: As a lower bound for the classification task, we compare our model against the baseline *sent_maj*, which assigns the majority label to all sentences. To assess the utility of segmentation, we compare against two baselines that use our proposed classifier but not the segmentation component: the baseline *sent_tf.idf* treats each sentence as a separate segment and *random_tf.idf* splits each document into random segments.

Dataset	**# train**	**# dev**	**# test**
MCScript	3492	408	-
Spinn3r (gold)	-	18	486

Table 5: Datasets (number of documents) used in the experiments

Model	**Precision**	**Recall**	**F$_1$-score**
sent_maj	0.08	0.05	0.06
sent_*tf.idf*	0.24	0.28	0.26
random_*tf.idf*	0.32	0.45	0.37
TT_*tf.idf* (F_1)	0.36	**0.54**	**0.43**
TT_*tf.idf* (Gold)	0.54	0.54	0.54

Table 6: Results for the scenario detection task

Evaluation. We evaluate scenario detection performance at the sentence level using micro-average precision, recall and F$_1$-score. We consider the top 1 predicted scenario for sentences with only one gold label (including the *None* label), and top *n* scenarios for sentences with *n* gold labels. For sentences with multiple scenario labels, we take into account partial matches and count each label proportionally. Assuming the gold labels are `washing ones hair` and `taking a bath`, and the classifier predicts `taking a bath` and `getting ready for bed`. `Taking a bath` is correctly predicted and accounts for 0.5 true positive (TP) while `washing ones hair` is incorrectly missed, thus accounts for 0.5 false negative (FN). `Getting ready for bed` is incorrectly predicted and accounts for 1 false positive (FP).

We additionally provide separate results of the segmentation component based on standard segmentation evaluation metrics.

5.2 Results

We present the micro-averaged results for scenario detection in Table 6. The *sent_maj* baseline achieves a F$_1$-score of only 6%, as the majority class forms only a small part of the dataset (4.7%). Our TT model with *tf.idf* features surpasses both

True label	Predicted label	# sents.	PMI
go_vacation	visit_sights	92	3.96
eat_restaurant	food_back	67	4.26
work_garden	grow_vegetables	57	4.45
attend_wedding	prepare_wedding	48	4.12
eat_restaurant	dinner_reservation	39	4.26
throw_party	go_party	36	4.09
shop_online	order_ on_phone	35	3.73
work_garden	planting a tree	33	4.81
shop_clothes	check_store_open	33	0.00
play_video_games	learn_board_game	32	0.00

Table 7: Top 10 misclassified scenario pairs (number of misclassified sentences (# sents.)) by our approach TT_*tf.idf* in relation to the PMI scores for each pair.

Scenario	# sents.	P	R	F_1
go to the dentist	47	0.90	0.96	0.93
have a barbecue	43	0.92	0.88	0.90
go to the sauna	28	0.80	0.89	0.84
make soup	60	0.81	0.87	0.84
bake a cake	69	0.71	0.97	0.82
go skiing	42	0.78	0.83	0.80
attend a court hearing	66	0.71	0.92	0.80
clean the floor	6	1.00	0.67	0.80
take a taxi	27	0.74	0.85	0.79
attend a church service	60	0.70	0.92	0.79

Table 8: Top 10 scenario-wise Precision (P), Recall (R) and F_1-score (F_1) results using our approach TT_*tf.idf* and the number of gold sentences (# sents.) for each scenario.

baselines that perform segmentation only naively (26% F_1) or randomly (37% F_1). This result shows that scenario detection works best when using predicted segments that are informative and topically consistent.

We estimated an upper bound for the classifier by taking into account the predicted segments from the segmentation step, but during evaluation, only considered those sentences with gold scenario labels (TT_*tf.idf* (Gold)), while ignoring the sentences with *None* label. We see an improvement in precision (54%), showing that the classifier correctly predicts the right scenario label for sentences with gold labels while also including other sentences that may be in topic but not directly referencing a given scenario.

To estimate the performance of the TT segmentor individually, we run TT on an artificial development set, created by merging segments from different scenarios from MCScript. We evaluate the performance of TT by using two standard topic segmentation evaluation metrics, P_k (Beeferman et al., 1999) and WindowDiff (WD, Pevzner and Hearst (2002)). Both metrics express the probability of segmentation error, thus lower values indicate better performance. We compute the average performance over several runs. TT attains P_k of 0.28 and WD of 0.28. The low segmentation errors suggest that TT segmentor does a good job in predicting the scenario boundaries.

5.3 Discussion

Even for a purpose-built model, scenario detection is a difficult task. This is partly to be expected as the task requires the assignment of one (or more) of 200 possible scenario labels, some of which are hard to distinguish. Many errors are due to misclassifications between scenarios that share script events as well as participants and that are usually mentioned in the same text: for example, `sending food back in a restaurant` requires and involves participants from `eating in a restaurant`. Table 7 shows the 10 most frequent misclassifications by our best model *TT_tf.idf* (F_1). These errors account for 16% of all incorrect label assignments (200 *by* 200 matrix). The 100 most frequent misclassifications account for 63% of all incorrect label assignments. In a quantitative analysis, we calculated the commonalities between scenarios in terms of the pointwise mutual information (PMI) between scenario labels in the associated stories. We calculated PMI using Equation (1). The probability of a scenario is given by the document frequency of the scenario divided by the number of documents.

$$PMI(S_1, S_2) = log\left(\frac{P(S_1 \wedge S_2)}{P(S_1) \cdot P(S_2)}\right) \quad (1)$$

Scenarios that tend to co-occur in texts have higher PMI scored. We observe that the scenario-wise recall and F_1-scores of our classifier are negatively correlated with PMI scores (Pearson correlation of -0.33 and -0.17, respectively). These correlations confirm a greater difficulty in distinguishing between scenarios that are highly related to other scenarios.

On the positive side, we observe that scenario-wise precision and F_1-score are positively correlated with the number of gold sentences annotated with the respective scenario label (Pearson correlation of 0.50 and 0.20, respectively). As one would

order pizza	laundry	gardening	barbecue
pizza	clothes	tree	invite
order	dryer	plant	guest
delivery	laundry	hole	grill
decide	washer	water	friend
place	wash	grow	everyone
deliver	dry	garden	beer
tip	white	dig	barbecue
phone	detergent	dirt	food
number	start	seed	serve
minute	washing	soil	season

Table 9: Example top 10 scenario-words

expect, our approach seems to perform better on scenarios that appear at higher frequency. Table 8 shows the 10 scenarios for which our approach achieves the best results.

Scenario approximation using topics. We performed an analyses to qualitatively examine in how far topic distributions, as used in our segmentation model, actually approximate scenarios. For this analysis, we computed a LDA topic model using only the MCScript dataset. We created *scenario-topics* by looking at all the prevalent topics in documents from a given scenario. Table 9 shows the top 10 words for each scenario extracted from the *scenario-topics*. As can be seen, the topics capture some of the most relevant words for different scenarios.

6 Summary

In this paper we introduced the task of scenario detection and curated a benchmark dataset for automatic scenario segmentation and identification. We proposed a benchmark model that automatically segments and identifies text fragments referring to a given scenario. While our model achieves promising first results, it also revealed some of the difficulties in detecting script references. Script detection is an important first step for large-scale data driven script induction for tasks that require the application of script knowledge. We are hopeful that our data and model will form a useful basis for future work.

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.

James Allan. 2012. *Topic detection and tracking: event-based information organization*, volume 12. Springer Science & Business Media.

James Allan, Jaime G Carbonell, George Doddington, Jonathan Yamron, and Yiming Yang. 1998. Topic detection and tracking pilot study final report.

Doug Beeferman, Adam Berger, and John Lafferty. 1999. Statistical models for text segmentation. *Machine learning*, 34(1-3):177–210.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

Kevin Burton, Akshay Java, and Ian Soboroff. 2009. The ICWSM 2009 Spinn3r Dataset. In *Third Annual Conference on Weblogs and Social Media (ICWSM 2009)*, San Jose, CA. AAAI.

Richard Eckart de Castilho, Eva Mujdricza-Maydt, Seid Muhie Yimam, Silvana Hartmann, Iryna Gurevych, Anette Frank, and Chris Biemann. 2016. A web-based tool for the integrated annotation of semantic and syntactic structures. In *Proceedings of the Workshop on Language Technology Resources and Tools for Digital Humanities (LT4DH)*, pages 76–84.

Nathanael Chambers and Daniel Jurafsky. 2008. Unsupervised learning of narrative event chains. In *ACL 2008, Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics, June 15-20, 2008, Columbus, Ohio, USA*, pages 789–797.

Nathanael Chambers and Daniel Jurafsky. 2009. Unsupervised learning of narrative schemas and their participants. In *Proceedings of 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP*, pages 602–610, Suntec, Singapore.

Danqi Chen and Christopher Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 740–750.

François Chollet et al. 2015. Keras. https://github.com/fchollet/keras.

Cuong Xuan Chu, Niket Tandon, and Gerhard Weikum. 2017. Distilling task knowledge from how-to communities. In *Proceedings of the 26th International Conference on World Wide Web*, WWW '17, pages 805–814, Republic and Canton of Geneva, Switzerland. International World Wide Web Conferences Steering Committee.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46.

Andrew Gordon and Reid Swanson. 2009. Identifying personal stories in millions of weblog entries. In *International Conference on Weblogs and Social Media, Data Challenge Workshop, May 20, San Jose, CA*.

Andrew S. Gordon. 2001. Browsing image collections with representations of common-sense activities. *Journal of the American Society for Information Science and Technology.*, 52:925.

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.

Herbert Paul Grice. 1975. Logic and conversation. In Peter Cole and Jerry L. Morgan, editors, *Syntax and Semantics: Vol. 3: Speech Acts*, pages 41–58. Academic Press, New York.

Rakesh Gupta and Mykel J. Kochenderfer. 2004. Common sense data acquisition for indoor mobile robots. In *Proceedings of the 19th National Conference on Artifical Intelligence*, AAAI'04, pages 605–610. AAAI Press.

Marti A Hearst. 1994. Multi-paragraph segmentation of expository text. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, pages 9–16. Association for Computational Linguistics.

Niels Kasch and Tim Oates. 2010. Mining script-like structures from the web. In *Proceedings of the NAACL HLT 2010 First International Workshop on Formalisms and Methodology for Learning by Reading*, FAM-LbR '10, pages 34–42, Stroudsburg, PA, USA. Association for Computational Linguistics.

Boyang Li, Stephen Lee-Urban, D. Scott Appling, and Mark O. Riedl. 2012. Crowdsourcing narrative intelligence. volume vol. 2. Advances in Cognitive Systems.

M Manshadi, R Swanson, and AS Gordon. 2008. Learning a probabilistic model of event sequences from internet weblog stories. FLAIRS Conference.

Ashutosh Modi, Tatjana Anikina, Simon Ostermann, and Manfred Pinkal. 2016. Inscript: Narrative texts annotated with script information. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 16)*, Portorož, Slovenia.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. 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. Association for Computational Linguistics.

Erik T. Mueller. 1999. A database and lexicon of scripts for thoughttreasure. *CoRR*, cs.AI/0003004.

Simon Ostermann, Ashutosh Modi, Michael Roth, Stefan Thater, and Manfred Pinkal. 2018a. MCScript: A Novel Dataset for Assessing Machine Comprehension Using Script Knowledge. In *Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan.

Simon Ostermann, Michael Roth, Ashutosh Modi, Stefan Thater, and Manfred Pinkal. 2018b. SemEval-2018 Task 11: Machine Comprehension using Commonsense Knowledge. In *Proceedings of International Workshop on Semantic Evaluation (SemEval-2018)*, New Orleans, LA, USA.

Simon Ostermann, Michael Roth, and Manfred Pinkal. 2019. MCScript2.0: A Machine Comprehension Corpus Focused on Script Events and Participants. *Proceedings of the 8th Joint Conference on Lexical and Computational Semantics (*SEM 2019)*.

Simon Ostermann, Michael Roth, Stefan Thater, and Manfred Pinkal. 2017. Aligning script events with narrative texts. *Proceedings of *SEM 2017*.

Simon Ostermann, Hannah Seitz, Stefan Thater, and Manfred Pinkal. 2018c. Mapping Texts to Scripts: An Entailment Study. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in Python. *Journal of machine learning research*, 12(Oct):2825–2830.

Lev Pevzner and Marti A Hearst. 2002. A critique and improvement of an evaluation metric for text segmentation. *Computational Linguistics*, 28(1):19–36.

Karl Pichotta and Raymond J. 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 (EACL 2014)*, pages 220–229, Gothenburg, Sweden.

James Pustejovsky and Amber Stubbs. 2012. *Natural Language Annotation for Machine Learning - a Guide to Corpus-Building for Applications*. O'Reilly.

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.

Susanne Raisig, Tinka Welke, Herbert Hagendorf, and Elke Van der Meer. 2009. Insights into knowledge representation: The influence of amodal and perceptual variables on event knowledge retrieval from memory. *Cognitive Science*, 33(7):1252–1266.

Michaela Regneri. 2013. *Event Structures in Knowledge, Pictures and Text*. Ph.D. thesis, Saarland University.

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.

Radim Rehurek and Petr Sojka. 2010. Software framework for topic modelling with large corpora. In *In Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*.

Martin Riedl and Chris Biemann. 2012. Topictiling: a text segmentation algorithm based on lda. In *Proceedings of ACL 2012 Student Research Workshop*, pages 37–42. Association for Computational Linguistics.

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

Jihee Ryu, Yuchul Jung, Kyung-min Kim, and Sung Hyon Myaeng. 2010. Automatic extraction of human activity knowledge from method-describing web articles. In *Proceedings of the 1st Workshop on Automated Knowledge Base Construction*, page 16. Citeseer.

Gerard Salton and Michael J McGill. 1986. Introduction to modern information retrieval.

Roger C. Schank and Robert P. Abelson. 1977. Scripts, plans, goals and understanding, an inquiry into human knowledge structures. *Hillsdale: Lawrence Erlbaum Associates,*, 3(2):211 – 217.

Push Singh, Thomas Lin, Erik T Mueller, Grace Lim, Travell Perkins, and Wan Li Zhu. 2002. Open mind common sense: Knowledge acquisition from the general public. In Robert Meersman and Zahir Tari, editors, *On the Move to Meaningful Internet Systems 2002: CoopIS, DOA, and ODBASE*, pages 1223–1237. Springer, Berlin / Heidelberg, Germany.

Niket Tandon, Gerard de Melo, Fabian M. Suchanek, and Gerhard Weikum. 2014. Webchild: harvesting and organizing commonsense knowledge from the web. In *Seventh ACM International Conference on Web Search and Data Mining, WSDM 2014, New York, NY, USA, February 24-28, 2014*, pages 523–532.

Niket Tandon, Gerard de Melo, and Gerhard Weikum. 2017. Webchild 2.0: Fine-grained commonsense knowledge distillation. *Proceedings of ACL 2017, System Demonstrations*, pages 115–120.

Lilian D. A. Wanzare, Alessandra Zarcone, Stefan Thater, and Manfred Pinkal. 2016. Descript: A crowdsourced corpus for the acquisition of high-quality script knowledge. In *The International Conference on Language Resources and Evaluation.*

Appendix

A Annotation guidelines

You are presented with several stories. Read each story carefully. You are required to highlight segments in the text where any of our scenarios is realized.

1. A segment can be a clause, a sentence, several sentences or any combination of sentences and clauses.
2. Usually segments will cover different parts of the text and be labeled with one scenario label each.
3. A text passage is highlighted as realizing a given scenario only if several scenario elements are addressed or referred to in the text, more than just the evoking expression but some more material e.g at least one event and a participant in that scenario is referred to in the text. (see examples (A.1 to A.5)).
4. A text passage referring to one scenario does not necessarily need to be contiguous i.e. the scenario could be referred to in different parts of the same text passage, so the scenario label can occur several times in the text. If the text passages are adjacent, mark the whole span as one segment. (see examples (A.6 to A.10))
5. One passage of text can be associated with more than one scenario label.
    - A passage of text associated with two or more related scenarios i.e. scenario that often coincide or occur together. (see example A.11).
    - A shorter passage of text referring to a given scenario is nested in a longer passage of text referring to a more general scenario. The nested text passage is therefore associated with both the general and specific scenarios. (see example A.12).
6. For a given text passage, if you do not find a full match from the scenario list, but a scenario that is related and similar in structure, you may annotate it. (see example A.13).

Rules of thumb for annotation

1. Do not annotate if no progress to events is made i.e. the text just mentions the scenario but no clear script events are addressed.

Example A.1 *short text with event and participants addressed*
✓ `feeding a child`
... Chloe loves to stand around babbling just generally keeping anyone amused as long as you bribe her with a piece of bread or cheese first.

✓ `going grocery shopping`
... but first stopped at a local shop to pick up some cheaper beer . We also stopped at a small shop near the hotel to get some sandwiches for dinner .

Example A.2 *scenario is just mentioned*
✗ `cooking pasta` *And a huge thanks to Megan & Andrew for a fantastic dinner, especially their first ever fresh pasta making effort of salmon filled ravioli - a big winner.*

✗ `riding on a bus`, ✗ `flying in a plane`
and then catch a bus down to Dublin for my 9:30AM flight the next morning.

We decide to stop at at Bob Evan's on the way home and feed the children.

Example A.3 *scenario is implied but no events are addressed*
✗ `answering the phone`
one of the citizens nodded and started talking on her cell phone. Several of the others were also on cell phones

✗ `taking a photograph`
Here are some before and after shots of Brandon . The first 3 were all taken this past May . I just took this one a few minutes ago.

Example A.4 *different discourse mode that is not narration e.g. information, argumentative, no specific events are mentioned*
✗ `writing a letter`
A long time ago, years before the Internet, I used to write to people from other countries. This people I met through a program called Pen Pal. I would send them my mail address, name, languages I could talk and preferences about my pen pals. Then I would receive a list of names and address and I could start sending them letters. ...

2. When a segment refers to more than one scenario, either related scenarios or scenarios where one is more general than the other, if there is only a weak reference to one of the scenarios, then annotate the text with the scenario having a stronger or more plausible reference.

   Example A.5 *one scenario is weakly referenced*
   ✓ `visiting a doctor`, ✗ `taking medicine`
   taking medicine is weakly referenced
   Now another week passes and I get a phone call and am told that the tests showed i had strep so i go in the next day and see the doc and he says that i don 't have strep . ugh what the hell . This time though they actually give me some antibiotic to help with a few different urinary track infections and other things while doing another blood test and urnine test on me .

   ✓ `taking a shower`, ✗ `washing ones hair`
   washing ones hair is weakly referenced
   I stand under the pressure of the shower , the water hitting my back in fierce beats . I stand and dip my hand back , exposing my delicate throat and neck . My hair gets soaked and detangles in the water as it flows through my hair , every bead of water putting back the moisture which day to day life rids my hair of . I run my hands through my hair shaking out the water as I bring my head back down to look down towards my feet . *The white marble base of the shower shines back at me from below . My feet covered in water , the water working its way up to my ankles but it never gets there . I find the soap and rub my body all over*

3. Sometimes there is a piece of text intervening two instances (or the same instance) of a scenario, that is not directly part of the scenario that is currently being talked about. We call this a separator. Leave out the separator if it is long or talks about something not related to the scenario being addressed. The separator can be included if it is short, argumentative or a comment, or somehow relates to the scenario being addressed. When there are multiple adjacent instances of a scenario, annotate them as a single unit.

   Example A.6 *two mentions of a scenario annotated as one segment*
   ✓ `writing a letter`
   I asked him about a month ago to write a letter of recommendation for me to help me get a library gig. After bugging him on and off for the past month, as mentioned above, he wrote me about a paragraph. I was sort of pissed as it was quite generic and short.

   I asked for advice, put it off myself for a week and finally wrote the letter of recommendation myself. I had both Evan and Adj. take a look at it- and they both liked my version.

   Example A.7 *a separator referring to topic related to the current scenario is included*
   ✓ `writing an exam`
   The Basic Science Exam (practice board exam) that took place on Friday April 18 was interesting to say the least. We had 4 hours to complete 200 questions, which will be the approximate time frame for the boards as well. I was completing questions at a good pace for the first 1/3 of the exam, slowed during the second 1/3 and had to rush myself during the last 20 or so questions to complete the exam in time.

   ✓ separator: Starting in May, I am going to start timing myself when I do practice questions so I can get use to pacing. There was a lot of information that was familiar to me on the exam (which is definitely a good thing) but it also showed me that I have a LOT of reviewing to do.

   Monday April 21 was the written exam for ECM. This exam was surprisingly challenging. For me, the most difficult part were reading and interpreting the EKGs. I felt like once I looked at them, everything I knew just fell out of my brain. Fortunately, it was a pass/fail exam and I passed.

   Example A.8 *a long separator is excluded*
   ✓ `going to the beach`
   Today , on the very last day of summer vacation , we finally made it to the beach . Oh , it 's not that we hadn 't been to a beach before . We were on a Lake Michigan beach just last

weekend . And we 've stuck our toes in the water at AJ 's and my lake a couple of times . But today , we actually planned to go . We wore our bathing suits and everything . We went with AJ 's friend D , his brother and his mom .

✗ separator: D and AJ became friends their very first year of preschool when they were two . They live in the next town over and we don 't see them as often as we would like . It 's not so much the distance , which isn 't far at all , but that the school and athletic schedules are constantly conflicting . But for the first time , they are both going back to school on the same day . So we decided to celebrate the end of summer together .

✓ `going to the beach`
It nearly looked too cold to go this morning ' the temperature didn 't reach 60 until after 9 :00. The lake water was chilly , too cool for me , but the kids didn 't mind . They splashed and shrieked with laughter and dug in the sand and pointed at the boat that looked like a hot dog and climbed onto the raft and jumped off and had races and splashed some more . D 's mom and I sat in the sun and talked about nothing in particular and waved off seagulls .

Example A.9 *a short separator is included*
✓ `throwing a party`
... My wife planned a surprise party for me at my place in the evening - I was told that we 'd go out and that I was supposed to meet her at Dhobi Ghaut exchange at 7 .

✓ separator: But I was getting bored in the office around 5 and thought I 'd go home - when I came home , I surprised her !

She was busy blowing balloons , decorating , etc with her friend . I guess I ruined it for her . But the fun part started here - She invited my sister and my cousin ...

✓ `visiting sights`
Before getting to the museum we swung by Notre Dame which was very beautiful . I tried taking some pictures inside Notre Dame but I dont think they turned out particularly well . After Notre Dame , Paul decided to show us the Crypte Archeologioue .
✓ separator: This is apparently French for parking garage there are some excellent pictures on Flickr of our trip there .

Also on the way to the museum we swung by Saint Chapelle which is another church . We didnt go inside this one because we hadnt bought a museum pass yet but we plan to return later on in the trip

4. Similarly to intervening text (separator), there may be text before or after that is a motivation, pre or post condition for the applications of the script currently being referred to. Leave out the text if it is long. The text can be included if it is short, or relates to the scenario being addressed.

Example A.10 *the first one or two sentences introduce the topic*
✓ `getting a haircut`
I AM , however , upset at the woman who cut his hair recently . *He had an appointment with my stylist (the one he normally goes to) but I FORGOT about it because I kept thinking that it was a different day than it was . When I called to reschedule , she couldn 't get him in until OCTOBER (?!?!?!) ...*

✓ `baking a cake`
I tried out this upside down cake from Bill Grangers , Simply Bill . As I have mentioned before , I love plums am always trying out new recipes featuring them when they are in season . *I didnt read the recipe properly so was surprised when I came to make it that it was actually cooked much in the same way as a tarte tartin , ie making a caramel with the fruit in a frying pan first , then pouring over the cake mixture baking in the frypan in the oven before turning out onto a serving plate , the difference being that it was a cake mixture not pastry ....*

5. If a text passage refers to several related scenarios, e.g. ”renovating a room” and ”painting a wall”, ”laying flooring in a room”, ”papering a room”; or ”working in the garden” and ”growing vegetables”, annotate all the related scenarios.

Example A.11 *segment referring to related scenarios*
✓ `growing vegetables,` ✓

`working in the garden`
The tomato seedlings Mitch planted in the compost box have done really well and we noticed flowers on them today. Hopefully we will get a good crop. It has rained and rained here for the past month so that is doing the garden heaps of good. We bought some organic herbs seedlings recently and now have some thyme, parsley, oregano and mint growing in the garden. We also planted some lettuce and a grape vine. ...

6. If part of a longer text passage refers to a scenario that is more specific than the scenario currently being talked about, annotate the nested text passage with all referred scenarios.

   Example A.12 *nested segment*
   ✓ `preparing dinner`
   I can remember the recipe, it's pretty adaptable and you can add or substitute the vegetables as you see fit!! One Pot Chicken Casserole 750g chicken thigh meat, cut into big cubes olive oil for frying 1

   ✓ `preparing dinner`, ✓ `chopping vegetables`
   large onion, chopped 3 potatoes, waxy is best 3 carrots 4 stalks of celery, chopped 2 cups of chicken stock 2 zucchini, sliced large handful of beans 300 ml cream 1 or 2 tablespoons of wholegrain mustard salt and pepper parsley, chopped

   ##42 The potatoes and carrots need to be cut into chunks,. I used chat potatoes which are smaller and cut them in half, but I would probably cut a normal potato into quarters. Heat the oil; in a large pan and then fry the chicken in batches until it is well browned...

7. If you do not find a full match for a text segment in the scenario list, but a scenario that is related and similar in its structure, you may annotate it.

   Example A.13 *topic similarity*

   - *Same structure in scenario e.g. going fishing for leisure or for work, share the same core events in going fishing*
   - *Baking something with flour (baking a cake, baking Blondies,)*

B List of Scenarios

	scenario	# docs	# sents.		scenario	# docs	# sents.
1	eating in a restaurant	21	387	101	receiving a letter	5	27
2	going on vacation	16	325	102	taking a shower	4	27
3	going shopping	34	276	103	taking a taxi	4	27
4	taking care of children	15	190	104	going to the playground	3	25
5	reviewing movies	8	184	105	taking a photograph	5	25
6	shopping for clothes	11	182	106	going on a date	3	24
7	working in the garden	13	179	107	making a bonfire	2	23
8	preparing dinner	14	155	108	renting a movie	3	23
9	playing a board game	8	129	109	buying a house	2	22
10	attend a wedding ceremony	9	125	110	designing t-shirts	2	22
11	playing video games	6	124	111	doing online banking	3	22
12	throwing a party	10	123	112	planting flowers	4	22
13	eat in a fast food restaurant	9	113	113	taking out the garbage	4	22
14	adopting a pet	7	111	114	brushing teeth	3	21
15	taking a child to bed	9	108	115	changing bed sheets	3	21
16	shopping online	7	102	116	going bowling	2	21
17	going on a bike tour	6	93	117	going for a walk	4	21
18	playing tennis	5	91	118	making coffee	2	21
19	renovating a room	9	87	119	serving a drink	5	20
20	growing vegetables	7	82	120	taking children to school	3	20
21	listening to music	8	81	121	taking the underground	2	20
22	sewing clothes	6	79	122	feeding a cat	4	19
23	training a dog	3	79	123	going to a party	5	19
24	moving into a new flat	8	78	124	ironing laundry	2	19
25	answering the phone	11	75	125	making tea	3	18
26	going to a concert	5	74	126	sending a fax	3	18
27	looking for a job	5	74	127	sending party invitations	3	18
28	visiting relatives	12	73	128	planting a tree	3	17
29	checking in at an airport	5	71	129	setting up presentation equipment	2	17
30	making a camping trip	5	71	130	visiting a museum	2	17
31	painting a wall	8	71	131	calling 911	2	16
32	planning a holiday trip	12	71	132	changing a light bulb	3	16
33	baking a cake	3	69	133	making toasted bread	1	16
34	going to the gym	6	69	134	playing a song	2	16
35	attending a court hearing	3	66	135	washing clothes	3	16
36	going to the theater	6	66	136	putting up a painting	2	15
37	going to a pub	4	65	137	serving a meal	5	15
38	playing football	3	65	138	washing dishes	3	15
39	going to a funeral	5	64	139	cooking pasta	2	14
40	visiting a doctor	7	64	140	moving furniture	4	14
41	paying with a credit card	6	63	141	put a poster on the wall	2	13
42	settling bank transactions	5	63	142	cleaning up toys	1	12
43	paying bills	6	62	143	preparing a picnic	2	12
44	taking a swimming class	3	62	144	repairing a bicycle	2	12
45	looking for a flat	6	61	145	cooking meat	4	11
46	attending a church service	3	60	146	drying clothes	3	11
47	making soup	3	60	147	give a medicine to someone	3	11
48	flying in a plane	5	57	148	feeding an infant	4	10
49	going grocery shopping	13	57	149	telling a story	2	10
50	walking a dog	5	57	150	unloading the dishwasher	1	10
51	going to the swimming pool	5	56	151	putting away groceries	3	9
52	preparing a wedding	3	56	152	deciding on a movie	1	7
53	writing a letter	5	54	153	going to a shopping centre	1	7
54	buy from a vending machine	3	53	154	loading the dishwasher	2	7
55	attending a job interview	3	52	155	making a bed	1	7
56	visiting sights	9	52	156	making a dinner reservation	1	7
57	attending a football match	4	51	157	making scrambled eggs	1	7
58	cleaning up a flat	6	51	158	playing piano	2	7
59	washing ones hair	6	49	159	wrapping a gift	1	7
60	writing an exam	5	49	160	chopping vegetables	3	6
61	watching a tennis match	3	48	161	cleaning the floor	1	6
62	going to the dentist	3	47	162	getting the newspaper	1	6
63	making a sandwich	4	47	163	making fresh orange juice	1	6
64	playing golf	3	47	164	checking if a store is open	2	5
65	taking a driving lesson	2	44	165	heating food on kitchen gas	1	4
66	going fishing	4	43	166	locking up the house	2	4

	scenario	# docs	# sents.		scenario	# docs	# sents.
67	having a barbecue	4	43	167	cleaning the bathroom	2	3
68	riding on a bus	6	43	168	mailing a letter	1	3
69	going on a train	4	42	169	making a hot dog	1	3
70	going skiing	2	42	170	playing a movie	1	3
71	packing a suitcase	5	42	171	remove and replace garbage bag	1	3
72	vacuuming the carpet	3	41	172	taking copies	2	3
73	order something on the phone	6	40	173	unclogging the toilet	1	3
74	ordering a pizza	3	39	174	washing a cut	1	3
75	going to work	3	38	175	applying band aid	2	2
76	doing laundry	4	37	176	change batteries in an alarm clock	1	2
77	cooking fish	3	36	177	cleaning a kitchen	1	2
78	learning a board game	1	36	178	feeding the fish	1	2
79	fueling a car	3	35	179	setting an alarm	1	2
80	going dancing	3	35	180	getting ready for bed	1	1
81	laying flooring in a room	4	35	181	setting the dining table	1	1
82	making breakfast	2	35	182	change batteries in a camera	0	0
83	paying for gas	3	34	183	buying a tree	0	0
84	taking a bath	3	34	184	papering a room	0	0
85	visiting the beach	4	34	185	cutting your own hair	0	0
86	borrow a book from the library	3	33	186	watering indoor plants	0	0
87	mowing the lawn	3	33	187	organize a board game evening	0	0
88	changing a baby diaper	3	32	188	cleaning the shower	0	0
89	driving a car	9	32	189	canceling a party	0	0
90	making omelette	3	32	190	cooking rice	0	0
91	play music in church	2	32	191	buying a DVD player	0	0
92	taking medicine	5	31	192	folding clothes	0	0
93	getting a haircut	3	30	193	buying a birthday present	0	0
94	heating food in a microwave	3	30	194	Answering the doorbell	0	0
95	making a mixed salad	3	30	195	cleaning the table	0	0
96	going jogging	2	28	196	boiling milk	0	0
97	going to the sauna	3	28	197	sewing a button	0	0
98	paying taxes	2	28	198	reading a story to a child	0	0
99	sending food back	2	28	199	making a shopping list	0	0
100	making a flight reservation	2	27	200	emptying the kitchen sink	0	0