Improving event prediction by representing script participants

Simon Ahrendt and **Vera Demberg**
Saarland University
66123 Saarbrücken
Germany
{simona,vera}@coli.uni-saarland.de

Abstract

Automatically learning script knowledge has proved difficult, with previous work not or just barely beating a most-frequent baseline. Script knowledge is a type of world knowledge which can however be useful for various task in NLP and psycholinguistic modelling. We here propose a model that includes participant information (i.e., knowledge about which participants are relevant for a script) and show, on the Dinners from Hell corpus as well as the InScript corpus, that this knowledge helps us to significantly improve prediction performance on the narrative cloze task.

1 Introduction

Scripts represent knowledge about typical event sequences (Schank and Abelson, 1977), for example the sequence of events happening when eating at a restaurant. Script knowledge thereby includes events like *order, bring* and *eat* as well as participants of those events, e.g., *menu, waiter, food, guest*. Script knowledge is a form of structured world knowledge that is useful in NLP applications for natural language understanding tasks (e.g., ambiguity resolution Rahman and Ng, 2012), as well as for psycholinguistic models of human language processing, which need to represent event knowledge to model human expectations (Zwaan et al., 1995; Schütz-Bosbach and Prinz, 2007) of upcoming referents and utterances.

One recent line of research has tried to learn scripts in an unsupervised way from large text collections. The core idea in Chambers and Jurafsky (2008, 2009); Jans et al. (2012) is to use coreference chains to identify events involving the same entity, with the intuition that these events would, if observed in many texts, be likely to represent a prototypical event sequence. Rudinger et al. (2015) show that this method is also applicable for learning specific targeted scripts from a domain-specific corpus, shown at the example of "Dinners From Hell" stories and the restaurant script.

Pichotta and Mooney (2014) (P&M) have demonstrated that using richer event representations containing multiple arguments improves prediciton accuracy on the narrative cloze task over the simpler models by Chambers and Jurafsky (2008). While they represent a script event as a pair of a verb and a dependency (an example of an event chain would be <call,obj>; <bring,subj>; <take,subj>), which is problematic for weak verbs and verb ambiguity, P&M represent events using a multi-argument event representation, e.g., *call(guest,waiter,*); bring(waiter,menu,*); take(waiter,order,*)*.

This richer event representation however still has some shortcomings. As the representation is based on coreference chains, the model runs into difficulties for entities that are in a chain of length one. Entities in a chain are internally mapped onto variables, but all single entities are mapped onto a common category *Other*. This means that all information about such referents is lost, e.g. $enjoy(customer, fish, *)$ can not be distinguished from $enjoy(customer, silence, *)$ when neither fish or silence have appeared before in the text.

The coreference chains provide a good approxi-

Proceedings of NAACL-HLT 2016, pages 546–551,
San Diego, California, June 12-17, 2016.

mation for identifying events that involve the same participants. But would performance improve substantially if we could represent event participants? This specifically addresses the problem of unlinked coreference chains (e.g., "food", "it", "steak") not appearing in the same coreference chain even though they represent the same role within the script, and the problem of mapping referents which are not part of a chain onto a single "other" representation.

Kampmann et al. (2015) show that referring expressions in a script can be automatically categorized in terms of the role they play within the script by using coreference chains, as well as information from WordNet (telling us e.g., that a *steak* is a kind of *food*).

In this paper, we extend the existing approach by P&M and demonstrate that explicitly labelling participants (instead of using coreference chains) leads to improved event prediction performance. We furthermore provide a systematic evaluation of the effect of automatically-annotated coreference chains vs. gold coreference chains, and automatically-annotated script participants vs. gold participant annotation. We evaluate our approach on the Dinners from Hell corpus (Rudinger et al., 2015), as well as the newly available InScript corpus (Modi et al., 2016).

Following earlier work, we evaluate the quality of script models using the so-called narrative cloze task, where the model has to predict a missing event given surrounding events in the text.

2 Methods

2.1 Participant-labeled Events

In order to capture script-relevant information conveyed by arguments we represent texts as chains of **participant-labeled events (PLEs)**. A PLE is a verb accompanied with the participant labels of its arguments.

The general form of a PLE is $verb(p_{subj}, p_{dobj}, p_{iobj})$, where p_{subj}, p_{dobj} and p_{iobj} are the participant labels of the subject, direct object and indirect object, respectively. For example, in the sentence *The waitress brought us some water*, the corresponding PLE would be $bring(waiter, drink, customer)$.

To automatically create PLEs from our training data, we first extract syntactic relations between verbs and their arguments as well as coreference information using Stanford CoreNLP (Manning et al. (2014)). We then use the max-hypernym heuristic described in Kampmann et al. (2015) to label the arguments with participant roles. This approach assigns to token w the participant label with the highest hyponym-similarity score between the wordnet-synsets associated with the label and one of the synsets of any word present in the coreference chain connected to w.

Where an argument slot of the event is not filled syntactically or the argument is not a participant of the script, a dummy participant O serves as a placeholder to indicate the absence of a labeled argument. Every extracted event that contains at least one participant is included into the chain.

Knowledge about the participants provides a much richer represenation of events. With this representation we are able to generalize from a specific word or entity to its overall role in the script. This way the model can also learn from cases where multiple entities fill one participant role or where a participant occurs in the text only once.

2.2 Predictive Model

Our script model is an adapted version of the bigram model in Jans et al. (2012) with an extension of the skip-gram option to skip all possible intervening events. This means we rank an event e to belong to a given ordered event sequence c at insertion point m according to its score as defined by:

$$Score(e) = \sum_{k=0}^{m} logP(e|c_k) + \sum_{k=m+1}^{n} logP(c_k|e), \quad (1)$$

where c_k denotes the k^{th} event in the chain and the conditional probabilties are estimated by skip-all bigram counts:

$$P(e_2|e_1) = \frac{freq(e_1, e_2)}{\sum_{e'} freq(e_1, e')} \quad (2)$$

with $freq(e_1, e_2)$ being the the number of times e_1 has been encountered prior to e_2 with an arbitrary number of events between them. This counting method might be noisy in case of long documents or a high number of unrelated events but it significantly

reduces data sparsity. In our specialized and rather small data sets of between 87 and 133 stories per scenario, sparsity is more of an issue than unrelatedness, so skip-all performs well, but other counting techniques might prove more suitable for different corpora.

In case our model assigns the same score to several events, we backoff to the simple unigram model described in section 4.2.

3 Data

The Dinners from Hell corpus (Rudinger et al., 2015) contains stories from an internet blog about terrible restaurant experiences. The corpus contains 143 stories (out of which 10 are reserved as a development set), which all have to do with the script of going to a restaurant. All non-copula verbs in this corpus are annotated as to whether they are relevant to the restaurant script.

The InScript corpus is a novel resource (Modi et al., 2016), which contains a total of 910 short stories containing on average 12 sentences each. The stories were collected via Mechanical Turk, instructing workers to describe a specific instance of an activity, as if explaining it to a child. The corpus contains 10 different scenario types, for which there are about 90 stories each. This corpus also contains annotation for whether a verb is script-relevant, coreference annotation and participant type information.

4 Evaluation

4.1 Narrative Cloze Task

The evaluation on the narrative cloze task (originally suggested by Chambers and Jurafsky, 2008) expresses how well a model can predict a missing event in a sequence of events. In order to make the methods comparable, all predictions of our model are mapped onto the encoding used by the simple pair event model, e.g. <order,obj>, as follows:

We map an PLE $v(p_{subj}, p_{dobj}, p_{iobj})$ to a verb-dependency pair $< v, d >$ relating to a participant p if p fills the PLE slot of dependency d. We subsequently define the score of a pair event as the maximum score of all PLEs which are mapped to this pair event. When evaluating on pair events, we rank events according to this redefined score.

4.2 Systems

Unigram Model This baseline is a simple model that ranks any event (whether it is a participant-labeled event or a pair event) by its overall frequency in the training data. It was first used in Pichotta and Mooney (2014) and has proven to be a very competitive baseline on the task.

Verb-dependecy Pair Event Model This is a bigram model over verb-dependency pair events as introduced by Jans et al. (2012) and following the general idea of Chambers and Jurafsky (2008). It has been slightly modified to model not only subjects and objects, but also indirect objects. We use the setting Rudinger et al. (2015) has shown performs best: Skip-all as a counting method, a count threshold of 1, a document threshold of 5 and absolute discounting. Note that their results on the same data set differ from ours as we do not include syntactic relations other than 'subj', 'dobj' and 'iobj' into training and evaluation.

Pichotta and Mooney We re-implemented the approach by Pichotta and Mooney (2014) with the exception that we use $v(e_{subj}, e_{dobj}, e_{iobj})$ instead of $v(e_{subj}, e_{obj}, e_{prep})$ to represent events. That is, we do not model prepositional arguments of an event but discriminate between direct and indirect objects of verbs.

Participant-based model Our model, as described in section 2.2.

4.3 Automatic labels vs. gold standard

Automatic Coreference Chains We evaluate how much the results are effected by the quality of the automatic coreference chains produced by the Stanford Parser vs. annotated gold-chains on our data.

Automatic Participant Labelling We furthermore investigate to what extent our approach suffers from imperfect participant labeling, i.e. how good our model could have been if the labeling process was 100%-accurate. Kampmann et al. (2015) report a 0.59 micro F-score on the DinnersFromHell data, leaving an arguably large room for improvement (although they have a ceiling of 0.84 in terms of micro F-score because of a mismatch between their participant label set and the gold-standard labels they eval-

uate on). To compare against a perfect participant labeler, we use the participant annotation described in the same paper.

4.4 Testing

Following Rudinger et al. (2015), we perform leave-one-out testing at the document level, i.e., we use 133 folds for Dinners from Hell, and between 87 and 97 for InScript scenarios). We use the annotation provided for the corpora to construct a test set for every document as follows: Every verb that has been annotated as script-relevant is regarded as a test case if it takes an argument in any coreference chain of at least length two, and the dependency between the verb and the argument is either 'subj', 'dobj' or 'iobj'. The test then consists of inferring the verb and dependency, given the model's representation of the remaining events in the document after this held-out event is removed.

5 Results

Methods that encode events in a more complex way have a higher risk of running into sparsity issues, i.e. cases where the model has not encountered any of the events in the current context. Table 1 shows Recall@10 as a measure of prediction precision. We can see that our model beats the baseline models by a large margin on this measure. The core advantage of our model are its participant representations, which allow it to make more correct generalizations, and generate less noisy predictions. This is also reflected in its lower coverage: our model does not predict events when the context (including participant labels) has not been observed, while other models may predict based on non-matching participant types, and hence generalize incorrectly.

We also report the Recall@10 with respect to predicting the entire PLE in Table 1 (shown as

Model	Coverage	R@10	R@10full
this	0.757	0.41	0.18
P&M	0.957	0.26	
Jans	0.809	0.25	
MostFreq	0.936	0.27	0.13

Table 1: Performance for our model is reported with both automatic coreference chains and participant labels; R@10full refers to the evaluation on PLE's instead of pair events.

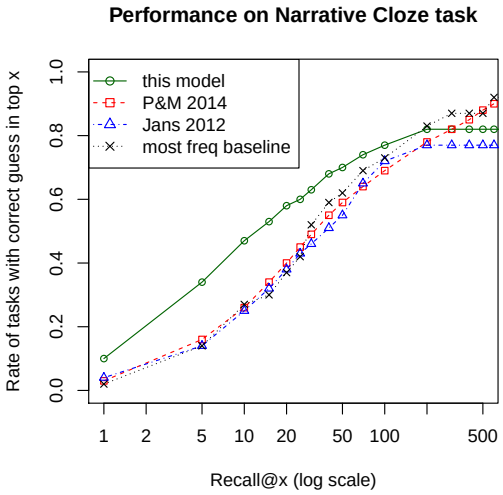

Figure 1: The participant-based method outperforms the other models and most frequent baseline. Performance shown for automatic chains and automatic participants.

R@10full), as we believe that inferring more structured events makes for a qualitative improvement on script modelling, we here provide a baseline for later work.

Figure 1 shows that the participant model succeeds in ranking the correct event high up more frequently than the other models. If the model cannot make any prediction due to coverage problems, it has to guess from unigram frequencies. This is reflected in our model's lower performance for Recall in sets larger than the top 500. We would however argue that performance at small Recall@x values is much more relevant for most applications, as it may matter little for most tasks where exactly in the low ranks 500-1000 a model manages to rank the correct solution. For future work, this lack of coverage could be compensated for by backing off to the P&M model.

Next, we'd like to see how the automatic versions of the models compare to a setting where the models have access to gold coreference chains and participants given by the annotation. Figure 2 shows that using automatic or gold corefernce chains makes no significant difference, but that there is quite a bit of scope for performance improvements if one can improve on the automatic participant labelling task.

Finally, we evaluated all models also on the 10 scenarios of the InScript dataset, to check whether the good performance of our model generalizes also to other datasets. We find that our model consis-

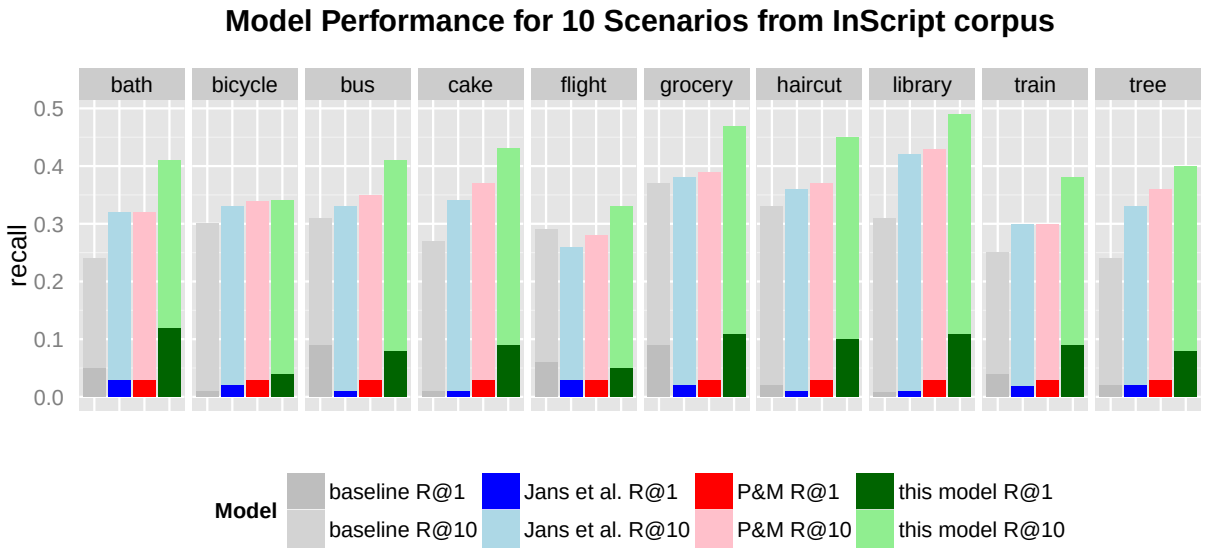

Figure 3: Performance of the different models with automatic participant labelling and automatic coreference chain annotation, on the stories for InScript dataset, showing results by scenario.

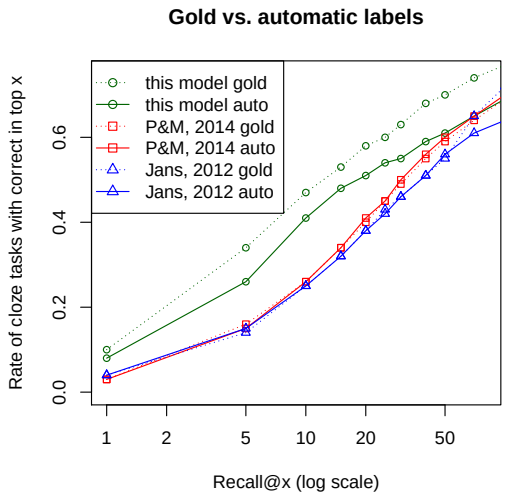

Figure 2: Gold models employ gold coreference chains and participants. There is little to no difference between using gold or automatic corefernce chains, but improving on participant labeling would help to further improve the model.

tently outperforms prior work also on this dataset, in particular with respect to succeeding to rank the correct event very high up on the list in the narrative cloze task. Figure 3 shows the Recall@1 and the Recall@10 measure separately for each of the scenarios from the InScript corpus.

6 Discussion and Conclusions

We have shown that the participant-based model can make much more accurate predictions in the narrative cloze task than previous models which do not have access to participant information; this even holds for automatic participant labelling, where we use a simple WordNet based method suggested in Kampmann et al. (2015). Our evaluation showed that the participant-based model substantially outperforms the state-of-the-art on the narrative cloze task, and that this performance holds for a set of naturalistic texts from blogs as well as for a corpus of narratives collected via crowd-sourcing. The present results hence represent an important step towards automatic inferencing for domains where knowledge of event sequences is relevant.

The automatic participant labeller takes as input a set of script participants, which can for example be acquired using the method of Regneri et al. (2010). The current approach therefore represents a way of combining the existing Mturk-based script acquisition methods by Regneri et al. (2010) with the unsupervised methods suggested in Chambers and Jurafsky (2008); Jans et al. (2012); Pichotta and Mooney (2014). Future work should further develop automated methods for participant labelling.

Acknowledgments

This research was funded by the German Research Foundation (DFG) as part of SFB 1102 'Information Density and Linguistic Encoding' and the Cluster of Excellence 'Multimodal Computing and Interaction' (EXC 284).

References

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

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

Jans, B., Bethard, S., Vulić, I., and Moens, M.-F. (2012). Skip n-grams and ranking functions for predicting script events. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 336–344, Avignon, France. Association for Computational Linguistics.

Kampmann, A., Thater, S., and Pinkal, M. (2015). A case-study of automatic participant labeling. In *Proceedings of the International Conference of the German Society for Computational Linguistics and Language Technology*, pages 97–105.

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S. J., and McClosky, D. (2014). The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60.

Modi, A., Anikina, T., Ostermann, S., and Pinkal, M. (2016). Inscript: Narrative texts annotated with script information. In *Proceedings of the 10th edition of the Language Resources and Evaluation Conference*.

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

Rahman, A. and Ng, V. (2012). Resolving complex cases of definite pronouns: the winograd schema challenge. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 777–789. Association for Computational Linguistics.

Regneri, M., Koller, A., and Pinkal, M. (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. Association for Computational Linguistics.

Rudinger, R., Demberg, V., Modi, A., Van Durme, B., and Pinkal, M. (2015). Learning to predict script events from domain-specific text. *Lexical and Computational Semantics (* SEM 2015)*, pages 205–210.

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

Schütz-Bosbach, S. and Prinz, W. (2007). Prospective coding in event representation. *Cognitive processing*, 8(2):93–102.

Zwaan, R. A., Langston, M. C., and Graesser, A. C. (1995). The construction of situation models in narrative comprehension: An event-indexing model. *Psychological science*, pages 292–297.