KnowSemLM : A Knowledge Infused Semantic Language Model

Haoruo Peng[1], **Qiang Ning**[1], **Dan Roth**[1,2]
Department of Computer Science
[1]University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA
[2]University of Pennsylvania, Philadelphia, PA 19104, USA
{hpeng7,qning2}@illinois.edu, danroth@seas.upenn.edu

Abstract

Story understanding requires developing expectations of what events come next in text. Prior knowledge – both statistical and declarative – is essential in guiding such expectations. While existing semantic language models (SemLM) capture event co-occurrence information by modeling event sequences as semantic frames, entities, and other semantic units, this paper aims at augmenting them with causal knowledge (i.e., one event is likely to lead to another). Such knowledge is modeled at the frame and entity level, and can be obtained either statistically from text or stated declaratively. The proposed method, KnowSemLM[1], infuses this knowledge into a semantic LM by joint training and inference, and is shown to be effective on both the event cloze test and story/referent prediction tasks.

1 Introduction

Natural language understanding requires a coherent understanding of a series of events or actions in a story. In story comprehension, we need to understand not only what events have appeared in text, but also what is likely to happen *next*. While event extraction has been well studied (Ji and Grishman, 2008; Huang and Riloff, 2012; Li et al., 2013; Peng et al., 2016; Nguyen et al., 2016; Nguyen and Grishman, 2016), the task of predicting future events (Radinsky et al., 2012; Radinsky and Horvitz, 2013) has received less attention.

One perspective is to utilize the co-occurrence information between past and future events learned from a large corpus, which has been studied in *script learning* works (Chambers and Jurafsky, 2008; Pichotta and Mooney, 2014, 2016a; Peng and Roth, 2016; Peng et al., 2017). However, only considering co-occurrence information is not sufficient for modeling event sequences in natural language. Human decisions on the likelihood of a specific event depend on both *local context* – what has happened earlier in text – and *global context* – knowledge gained from human experience. This paper leverages both the *local* and *global* context information to model event sequences, and shows that it can lead to more accurate predictions of future events. For example, the following text snippet describes a scenario of someone taking a flight:

... I checked in at the counter, took my luggage to the security area, got cleared ten minutes in advance, and waited for my plane ...

This example consists of a series of events, i.e., "check in (a flight)", "be cleared (at the security)", "wait for (the plane)", etc., which humans who have traveled by plane are very familiar with. However, this event sequence appears infrequently in text.[2] Consequently, only relying on event co-occurrence in text is not sufficient – there is also a need to model some "common sense" information.

The local and global contexts in this example are illustrated in Figure 1. The existing event sequence is "(sub)check_in[flight]", "(sub)clear[security]" and "(sub)wait_for[plane]" (denoted by blue dots), where "sub" means subject. Language models (LM) for statistical co-occurrences of events can capture this local context and generate a distribution over all possible events, e.g., "(sub)purchase[food]" and "(sub)go_to[work]", as in the blue circle.

More importantly, global context is the knowledge of event causality learned from human experience in the form of "cause-effect" event pairs (i.e., one event leads to another). One such pair is represented as "(sub)wait_for[plane] ⇒

[1]Related resources refer to https://cogcomp.seas.upenn.edu/page/publication_view/886.

[2]The events "check in" and "be cleared" only co-occur twice in a same document in the 20-year New York Times corpus (1987-2007); we count with frame and entity level abstractions (see Section 2.1 for details).

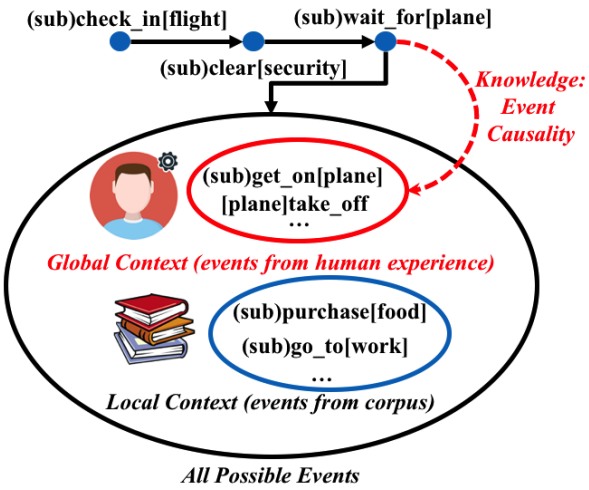

Figure 1: **Local and global context information when modeling event sequences.** The blue dots are events that are already described in text. The blue circle indicates local context, i.e., event sequences inferred from a large corpus via semantic LMs; the red circle represents global context, i.e., events learned from human experience via knowledge of event causality (which may overlap with local context). For event representations, we abstract over the surface forms of semantic frames and entities, where "sub" represents the shared common subject. The proposed KnowSemLM leverages both information to better predict future events.

(sub)get_on[plane]", which means that one has to wait for a plane before getting on it (red dashed arrow in Figure 1). Global context, as a result, helps generate a distribution over a focused set of expected events, as in the red circle. Note that the causality links have directions, and one event might lead to multiple possible events, e.g., one has to wait for the plane before it takes off "(sub)wait_for[plane] $\Rightarrow$ [plane]take_off". Such connections can be viewed as temporal relations. Here, we consider causality to include temporal orderings of events which align with common sense. More discussions are provided in Sec. 6.

Thus, we propose KnowSemLM, a knowledge infused semantic language model. It combines knowledge from external sources (in the form of event causality) with the basic semantic LM (Peng et al., 2017) trained on a given text corpus. Our model is a generative model of events, where each event is either generated based on a piece of knowledge or generated from the semantic LM. When predicting future events at inference time, we generate two distributions over events: one from the given knowledge, and the other from the semantic LM. We also learn a binary variable that selects the distribution from which we take the next event. In this way, the proposed KnowSemLM has the ability to generate event sequences based on both local and global context, and better imitate the story generation process.

This knowledge infused semantic LM operates on abstractions over the surface form – semantic frames and entities. We associate each semantic unit (frames and entities) with an embedding and construct a joint embedding space for each event. We train KnowSemLM on a large corpus and use the same embedding setting for events involved in the knowledge. The event causality knowledge is mined either statistically from the training corpus or declaratively for constrained domains (both in the form of event pairs). In the statistical way, we utilize a set of discourse connectives to identify "cause-effect" event pairs and filter them based on their counts; if provided with event templates for specific domains, we also manually write down such pairs based on human experience. In both ways, we further enrich the knowledge base by considering transitivity among event pairs.

We evaluate KnowSemLM on two tasks – event cloze test and story/referent predictions. In both cases, we model text as a sequence of events and apply trained KnowSemLM to calculate conditional probabilities of future events given text and knowledge. We show that KnowSemLM can outperform competitive results from models with no such knowledge. In addition, we demonstrate the language modeling ability of KnowSemLM through quantitative and qualitative analysis.

The main contributions can be summarized as follows: 1) formulation of knowledge used in story generation as event causality; 2) proposal of KnowSemLM to integrate such event causality knowledge into semantic language models; 3) demonstration of the effectiveness of KnowSemLM via multiple benchmark tests.

The rest of the paper is organized as follows. We define how we model events and event causality knowledge in Sec. 2, followed by the description of the knowledge infused KnowSemLM (Sec. 3). The training procedure of KnowSemLM is detailed in Sec. 4, followed by our experimental results and analysis (Sec. 5) and related work (Sec. 6). We conclude in Sec. 7.

2 Event and Knowledge Modeling

To better understand the proposed KnowSemLM, here we first introduce the event representation and

event causality model used in this paper.

2.1 Event Representation

To preserve the full semantic meaning of events, we need to consider multiple semantic aspects: semantic frames, entities, and sentiments. We adopt the event representation proposed in Peng et al. (2017), which is built upon abstractions of three basic semantic units: (disambiguated) semantic frames, subjects & objects in such semantic frames, and sentiments of the frame text.

In a nutshell, the event representation is a combination of the above three semantic elements.

... Steven Avery committed murder. He was arrested, charged and tried ...

For example, the event representations of the above text would be (four separate events):

PER[new]-commit.01-ARG[new](NEG)
ARG[new]-arrest.01-PER[old](NEU)
ARG[new]-charge.05-PER[old](NEU)
ARG[new]-try.01-PER[old](NEG)

Here, "commit.01", "arrest.01" and so on represent disambiguated predicates ("01" and "05" refer to the disambiguated senses in VerbNet). The arguments (subject and object) of a predicate are denoted with NER types ("PER, LOC, ORG, MISC") or "ARG" if unknown, along with a "[new/old]" label indicating if it is the first appearance in the sequence. Additionally, the sentiment of a frame is represented as positive (POS), neural (NEU), or negative (NEG).

We formally define such an explicit and abstracted event as e. Computationally, the vector representation of an event e^{vec} is built in a joint semantic space:

$$e^{\text{vec}} = W_f r_f + W_e r_e + W_s r_s.$$

During language model training, we learn frame embeddings W_f (r_f, r_e, r_s are one-hot vectors for each unique frame, entity and sentiment abstraction, respectively) as well as the transforming matrices W_e and W_s.

2.2 Knowledge: Causality between Events

We model the knowledge gained from human experience as pre-determined relationship between events. Since we are modeling event sequences, the knowledge of one event leads to another is very important, hence event causality. We formally define a piece of event knowledge as

$$e_x \Rightarrow e_y,$$

meaning that the *outcome* event e_y is a possible result of the *causal* event e_x. Note that event causality here is directional, and one event may lead to multiple different outcomes. We group all event knowledge pairs with the same causal event, thus event e_x can lead to a set of events:

$$e_x \Rightarrow \{e_{y_1}, e_{y_2}, e_{y_3}, \cdots, e_{y_m}\}.$$

We store all such event causality structures in a knowledge base KB_{EC}.

3 Knowledge Infused SemLM

With a proper modeling of events and event causality above, this section explains the proposed `KnowSemLM`, a method to inject causality knowledge into a semantic LM. Specifically, `KnowSemLM` is based on FES-RNNLM (*Frame-Entity-Sentiment infused Recurrent Neural Net Language Model*) proposed in Peng et al. (2017). We briefly review FES-RNNLM and describe how `KnowSemLM` adds knowledge on top of it.

3.1 FES-RNNLM

To model semantic sequences and train the joint event representations in Sec. 2.1, we build neural language models over such sequences. Peng et al. (2017) uses Log-Bilinear Language model (Mnih and Hinton, 2007), but since we require the use of event causality knowledge to be based on past events, we choose to implement an RNN language model (RNNLM) where the generation of future events is only dependent on past events.

For ease of explanation, we denote a semantic sequence of joint event representations as $[e_1, e_2, \cdots, e_t]$, with e_t being the t_{th} event in the sequence. Thus, we model the conditional probability of an event e_t given its context as

$$\begin{aligned} & p_{\text{lm}}(e_t | e_1, \cdots, e_{t-1}) \\ = & \text{softmax}(W_s h_t + b_s) \\ = & \frac{\exp(e_t^{\text{vec}}(W_s h_t + b_s))}{\sum_{e \in \mathcal{V}} \exp(e^{\text{vec}}(W_s h_t + b_s))}. \end{aligned}$$

Note that the softmax operation is carried out over the event vocabulary $\mathcal{V}$, i.e., all possible events in the language model. Moreover, the hidden layer h_t in RNN is computed as: $h_t = \phi(e_t^{\text{vec}} W_i + h_{t-1} W_h + b_h)$, where ϕ is the activation function. For language model training, we learn parameters W_s, b_s, W_i, W_h, and b_h, and maximize the sequence probability $\prod_{t=1}^{k} p_{\text{lm}}(e_t | e_1, e_2, \cdots, e_{t-1})$.

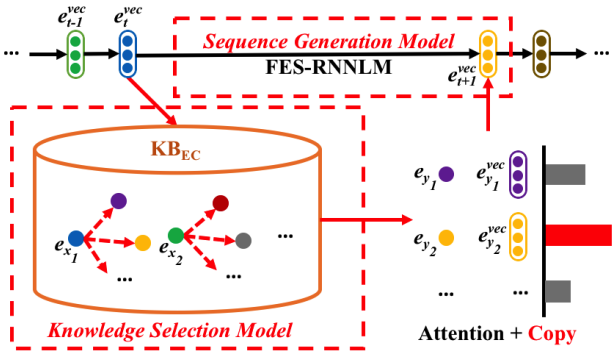

Figure 2: **Overview of the computational workflow for the proposed KnowSemLM.** There are two key components: 1) a knowledge selection model, which activates the use of knowledge based on probabilistically matching causal event and produce a distribution over outcome events via attention; 2) a sequence generation model, which takes input from both the knowledge selection model and the base semantic language model (FES-RNNLM) to generate future events via a copying mechanism. Note that the single dots indicate explicit event representations while three consecutive dots stand for event vectors.

3.2 KnowSemLM

In Figure 2, we show the computational workflow of the proposed KnowSemLM. There are two key components: 1) a knowledge selection model, which activates the use of knowledge based on probabilistically matching causal events and produces a distribution over outcome events; 2) a sequence generation model, which takes input from both the knowledge selection model and the base semantic language model (FES-RNNLM) to generate future events via a copying mechanism.[3]

Knowledge Selection Model

For an event in the sequence e_t, we first match it with possible causal events $\{e_x\}$ in the knowledge base KB_{EC} based on the bi-focal attention of previous events. Thus, from the knowledge base, we get a list of outcome events $\mathcal{V}_y \triangleq \{e_{y_1}, e_{y_2}, \cdots\}$.

Computationally, we model the conditional probability of matching with causal event e_x and outcome event e_y from knowledge base given the context of $e_1, e_2, \cdots, e_t$ as

$$p_{\text{kn}}(e_x \Rightarrow e_y | e_1, e_2, \cdots, e_t) = \frac{\exp(e_x^{\text{vec}} W_a h_t) \exp(e_y^{\text{vec}} W_b h_t)}{\sum_{e \in \mathcal{V}_x, e' \in \mathcal{V}_y} \exp(e^{\text{vec}} W_a h_t) \exp(e'^{\text{vec}} W_b h_t)}.$$

[3]The proposed computational framework of KnowSemLM is similar to DynoNet proposed in He et al. (2017). Compared to DynoNet, the knowledge base utilized here operates on event level representations rather than on tokens.

Here, we use the bi-focal attention mechanism (Nema et al., 2018) via attention parameters W_a, W_b, and apply it on the hidden layer h_t, which embeds information from all previous events in the sequence. Therefore, we produce a distribution over the set of possible outcome events $\mathcal{V}_y$.

Sequence Generation Model

The base semantic LM produces a distribution over events from the language model vocabulary, which represents *local* context, while the knowledge selection model generates a set of outcome events with a probability distribution, which represents *global* context of event causality knowledge. The sequence generation model then combines the local and global context for generating future events. Therefore, we model the conditional probability of event e_{t+1} given context $p(e_{t+1}|\text{Context}) = p(e_{t+1}|e_1, e_2, \cdots, e_t, \text{KB}_{\text{EC}})$. This overall distribution is computed via a copying mechanism (Jia and Liang, 2016), i.e., we either generate the next event (e_i) from the language model vocabulary ($\mathcal{V}$) or copy from the outcome event set (e_y) based on the following probabilities:

$$\begin{cases} p(e_{t+1} = e_i \in \mathcal{V} | \text{Context}) & = (1-\lambda) p_{\text{lm}}(e_i) \\ p(e_{t+1} = e_y \in \mathcal{V}_y | \text{Context}) & = \lambda p_{\text{kn}}(e_y). \end{cases}$$

Here, λ is a learned scaling parameter to choose between events from LM vocabulary $\mathcal{V}$ and events from event causality knowledge base KB_{EC}.

4 Construction of KnowSemLM

4.1 Dataset and Preprocessing

Dataset: We use the New York Times (NYT) Corpus[4] (from year 1987 to 2007) as the training corpus. It contains over 1.8M documents in total.

Preprocessing: We preprocess all training documents with Semantic Role Labeling and Part-of-Speech tagging. We also implement the explicit discourse connective identification module of a shallow discourse parser (Song et al., 2015). Additionally, we utilize within-document entity co-reference (Peng et al., 2015a) to produce co-reference chains and get the anaphoricity information. To obtain all annotations, we use the Illinois NLP tools (Khashabi et al., 2018).[5] Further, we obtain event representations from text with frame, entity and sentiment level abstractions by following procedures described in Peng et al. (2017).

[4]https://catalog.ldc.upenn.edu/LDC2008T19

[5]http://cogcomp.org/page/software/

4.2 Knowledge Mining

Statistical Way: Part of the human knowledge can be mined from text itself. Since discourse connectives are important for relating different text spans, we carefully select discourse connectives which can indicate a "cause-effect" situation. For example, "The police arrested Jack *because* he killed someone." In this sentence, readers can gain the knowledge of "the person who kills shall be arrested", which can be represented as "PER[*]-kill.01-*[*](*) $\Rightarrow$ *[*]-arrest.01-PER[old](*)" according to the abstractions specified in Sec. 2.

In practice, we choose 22 "cause-effect" connectives/phrases (such as "because", "due to", "in order to"). We then extract all event pairs connected by such connectives from the NYT training data, and abstract over their surface forms to get the event level representations. Finally, we filter cases where the direction of the event causality pairs is unclear from a statistical standpoint. Specifically, we calculate the ratio of counts of one direction over another, i.e. $\theta = \frac{\#(e_x \Rightarrow e_y)}{\#(e_y \Rightarrow e_x)}$. If $\theta > 2$, then we store $e_x \Rightarrow e_y$ as knowledge; while $\theta < 0.5$, we only keep $e_y \Rightarrow e_x$. In the case of $0.5 < \theta < 2$, we filter both event causality pairs since we are unsure of the knowledge statistically.

After the above filtering procedures, we automatically get 8,293 different pairs of event pairs (without human efforts). According to Sec. 2, we merge them if they have the same causal event, i.e. $e_x \Rightarrow e_y$ and $e_x \Rightarrow e_z$ becomes $e_x \Rightarrow \{e_y, e_x\}$. Thus, we get a total of 2,037 causal events (trees); and on average, each causal event has 4 possible outcome events. Furthermore, those event pairs of knowledge defined in this work are transitive, e.g., if $e_1 \Rightarrow e_2$ and $e_2 \Rightarrow e_3$, then we can have $e_1 \Rightarrow e_3$. Considering this *transitivity*, we iterate over all pairs twice, and derive more event causality pairs, achieving a total number of 9,022.[6]

Declarative Way: Besides mining knowledge automatically from text corpus, we also take full advantage of human input in some practical situations. For the InScript Corpus (Modi et al., 2017), it specifies 10 everyday scenarios, e.g., "Bath", "Flight", "Haircut". In each scenario, the corpus also provides event templates and the corresponding event template annotations for the text. Examples of such generated event causality knowledge can be referred back to Sec. 1, e.g., "(sub)wait_for[plane] $\Rightarrow$ (sub)get_on[plane]". In total, we manually generate 875 event causality pairs and group them with 121 causal events. Here, since during the manual generation process, we try to cover all event causality knowledge that makes sense; we do not further apply the transitive property and expand.

[6]We do not further carry out the transitivity expansion process, since empirically the noise it introduces outweighs the benefits it brings (see Sec. 5.4 for details).

Method	Accuracy
Granroth-Wilding and Clark (2016)	49.57%
Wang et al. (2017)	55.12%
KnowSemLM w/o knowledge	39.23%
KnowSemLM w/o transit. & fine-tuning	43.56%
KnowSemLM w/o fine-tuning	45.28%
KnowSemLM	**56.27%**

Table 1: **Accuracy results for the event cloze task.** KnowSemLM outperforms previously reported results and we show the ablation study results for model without the use of knowledge (w/o knowledge), without the use of knowledge transitivity as described in Sec 4.2 (w/o transit.) and without fine-tuning on the dev data (w/o fine-tuning), resepctively.

4.3 Model Training

Based on the formulation in Sec. 3, we apply the overall sequence probability as the objective: $\prod_{t=1}^{k} p(e_t|e_1, e_2, \cdots, e_{t-1}, \text{KB}_{\text{EC}})$. where k is the sequence length. For the sequence generation model, we implement the Long Short-Term Memory (LSTM) network with a layer of 64 hidden units while the dimension of the input event vector representation is 200. Because we carry out the same event-level abstractions as in Peng et al. (2017), the event vocabulary is the same, with the size of $\sim$4M different events.[7]

5 Experiments

We show that KnowSemLM can achieve better performance for the event cloze test and story/referent prediction tasks compared to models without the use of knowledge. We also evaluate the language modeling ability of KnowSemLM through quantitative and qualitative analysis.

5.1 Application for Event Cloze Test

Task Description and Setting: We utilize the MCNC task and dataset proposed in Granroth-Wilding and Clark (2016) as the benchmark evaluation. For each test instance, the goal is to recover the event (defined as predicate with associated entities) from an event chain given multiple choices.

[7]Please see Table 2 in Peng et al. (2017) for details.

Since the event definition in this task is compatible with our representation defined in Sec 2.1[8], we can directly convert event chains into our semantic event sequences. In this application task, we train KnowSemLM on the NYT portion of the Gigaword[9] corpus, and also fine-tune on the development set specified in this task[10].

Application of KnowSemLM: For each test case (i.e., an event chain inside a document), we first construct the event level representation as described in Sec. 2 for each event in the chain. We then apply KnowSemLM to obtain the overall sequence probability by replacing the missing event with each candidate choice. The final decision is made by choosing the event with the highest probability. Note that the event causality knowledge here for both training and testing is generated automatically from NYT corpus specified in Sec. 4.2 (the Statistical Way). To efficiently calculate the sequence probability, we limit the context window size surrounding the missing event to be 10.

Results: The accuracy results are shown in Table 1. We compare KnowSemLM with previous reported results on this event cloze test (Granroth-Wilding and Clark, 2016; Wang et al., 2017). KnowSemLM outperforms both baselines and we further carry out the ablation study to measure the impact of knowledge, transitivity of knowledge, and fine-tuning. We can see that it is important for the semantic LM to consider knowledge and also learn the process of applying such knowledge in event sequences, i.e., the fine-tuning step.

5.2 Application for Story Prediction

Task Description and Setting: We use the benchmark ROCStories dataset (Mostafazadeh et al., 2017), and follow the test setting in Peng et al. (2017). For each instance, we are given a four-sentence story and the system needs to predict the correct fifth sentence from two choices; with the incorrect ending being semantically unreasonable, or un-related. Instead of treating the task as a supervised binary classification problem with a development set to tune, we evaluate KnowSemLM in an unsupervised fashion where we directly evaluate on the test set. In such a way, we can directly compare with the FES-LM model proposed in Peng et al. (2017), which is base model of KnowSemLM without the use of knowledge. Similar to the training of FES-LM, we fine tune KnowSemLM on the in-domain short story training data, with the model trained on NYT corpus as initialization.[11]

Baselines	Accuracy	
Seq2Seq	58.0%	
Mostafazadeh et al. (2016)	58.5%	
Seq2Seq with attention	59.1%	
Model w/o Knowledge	S.	M.V.
FES-LM (Peng et al., 2017)	62.3%	61.6%
Knowledge Model	S.	M.V.
KnowSemLM	**66.5%**	63.1%

Table 2: **Accuracy results for story cloze test in the unsupervised setting.** "S." represents the inference method with the single most informative feature while "M.V." means majority voting.

Application of KnowSemLM: For each test story, we generate a set of conditional probability features from KnowSemLM. We first construct the event level representation as described in Sec. 2. We then utilize the conditional probability of the fifth sentence given previous context sentences and the knowledge base KB_{EC} as features. Here KB_{EC} is generated automatically from NYT corpus specified in Sec. 4.2 without human efforts. We get multiple features depending on how long we go back in the context in terms of events. In practice, we get at most 12 events as context since one sentence can contain multiple events. Thus, for each story, we generate at most 12 pairs of conditional probability features from two given choices. Every pair of such features can yield a decision on which ending is more probable. Here, we test two different inference methods: a single most informative feature (where we go with the decision made by the pair of features which have the highest ratio) or majority voting based on the decision made jointly by all feature pairs.

Results: The accuracy results are shown in Table 2. We compare KnowSemLM with Seq2Seq baselines (Sutskever et al., 2014) and Seq2Seq with attention mechanism (Bahdanau et al., 2014). We also include the DSSM system

[8] Our event representation is abstracted on a higher level. Thus, we process the original NYT documents, where event chains come from, for abstraction purposes; and then match it to the event chains in the test data.

[9] https://catalog.ldc.upenn.edu/LDC2011T07

[10] https://mark.granroth-wilding.co.uk/papers/what_happens_next/

[11] We iterate over the NYT corpus until it converges on the perplexity metric for the development set, and then the model is further trained on ROC-Stories training set for 5 epochs.

Method	Accuracy
Base (Modi et al., 2017)	62.65%
EntityNLM (Ji et al., 2017)	74.23%
Base*	60.58%
Base* w/ FES-RNNLM	63.79%
Base* w/ KnowSemLM	**76.15%**

Table 3: **Accuracy results for the referent prediction task on InScript Corpus.** We re-implement the base model (Modi et al., 2017) as "Base*", and apply KnowSemLM to add additional features. "Base* w/ FES-RNNLM" is the ablation study where no event causality knowledge is used. Even though "Base*" model performs not as good as the original base model, we achieve the best performance with added KnowSemLM features.

Perplexity	
FES-RNNLM	121.8
KnowSemLM w/o transitivity	120.7
KnowSemLM	120.4
Narrative Cloze Test (Recall@30)	
FES-RNNLM	47.9
KnowSemLM w/o transitivity	49.3
KnowSemLM	49.6

Table 4: **Results for perplexity and narrative cloze test.** Both studies are conducted on the NYT hold-out data. "FES-RNNLM" represents the semantic LM without the use of knowledge. The numbers show that KnowSemLM has lower perplexity and higher recall on narrative cloze test, which demonstrates the contribution of the infused knowledge.

	Match/Event	Activation/Event	λ
NYT	0.13	0.03	0.36
InScript	0.82	0.28	0.46

Table 5: **Statistics for the use of event causality knowledge in KnowSemLM.** We gather the statistics for both NYT and InScript Corpus. "Match/Event" represents average number of times a causal event match is found in the event causality knowledge base per event; while "Activation/Event" stands for the average number of times we actually generate event predictions from the outcome events of the knowledge base. In addition, we believe the ratio of "Activation/Event" over "Match/Event" co-relates with the scaling parameter λ.

from Mostafazadeh et al. (2016) as the original reported result. KnowSemLM outperforms both baselines and the base model without the use of knowledge, i.e., FES-LM. The best performance achieved by KnowSemLM uses single most informative feature, with the feature being the conditional probability depending on only the nearest preceding event and event causality knowledge).

5.3 Application for Referent Prediction

Task Description and Setting: For referent prediction task, we follow the setting in Modi et al. (2017), where the system predicts the referent of an entity (or a new entity) given the preceding text. The task is evaluated on the InScript Corpus, which contains a group of documents where events are manually annotated according to predefined event templates. Each document contains one entity which needs to be resolved. The InScript Corpus can be divided into 10 situations and is split into standard training, development, and testing sets. We fine-tune KnowSemLM on the InScript Corpus training set, with the model trained on NYT corpus as initialization.

Application of KnowSemLM: For each test case (i.e., an entity inside a document), each candidate choice will be represented as a different event representation. Note that the event representation here comes from the event templates defined in the InScript Corpus. In the meantime, we can extract the event sequence from the preceding context. Thus, we can apply KnowSemLM to compute the conditional probability of the candidate event e_{t+1} given the event sequence and the event causality knowledge: $p_k(e_{t+1}|e_{t-k}, e_{t-k+1}, \cdots, e_t, \text{KB}_{\text{EC}})$. Here, knowledge in KB_{EC} is generated manually from event templates specified in Sec. 4.2. Moreover, index k decides how far back we consider the preceding event sequence. We then add this set of conditional probabilities as additional features in a base model (re-implementation of the linear model proposed in Modi et al. (2017), namely "Base*") to train a classifier to predict the right referent.

Results: The accuracy results are shown in Table 3. We compare with the original base model as well as the EntityNLM proposed in Ji et al. (2017) as baselines. Our re-implemented base model ("Re-base") does not perform as good as the original model. However, with the help of additional features from FES-RNNLM, we outperform the base model. More importantly, with additional features from KnowSemLM, we achieve the best performance and beat the EntityNLM system. This demonstrates the importance of the manually added event causality knowledge, and the ability of KnowSemLM to successfully capture it.

5.4 Analysis of KnowSemLM

First, to evaluate the language modeling ability of KnowSemLM, we report perplexity and narrative cloze test results. We employ the same experimental setting as detailed in Peng and Roth (2016) on the NYT hold-out data. Results are shown in Table 4. Here, "FES-RNNLM" serves as the semantic LM without the use of knowledge for the ablation study. The numbers shows that KnowSemLM has lower perplexity and higher recall on narrative cloze test; which demonstrates the contribution of the infused event causality knowledge. The results w.r.t. the transitivity evaluation shows that the expansion through knowledge transitivity improves the model quality.

We also gather the statistics to analyze the usage of event causality knowledge in KnowSemLM. We compute two key values: 1) average number of times a causal event match is found in the event causality knowledge base per event (so that we can potentially use the outcome events to predict), i.e. "Match/Event"; 2) average number of times we actually generate event predictions from the outcome events of the knowledge base (result of the final probability distribution), i.e. "Activation/Event". We get the statistics on both NYT and InScript Corpus, and associate the numbers with the scaling parameter λ in Table 5. The frequency of event matches and event activations from knowledge are both much lower in NYT than in InScript. Moreover, we can compute the chance of an outcome event being used as the prediction when it participates in the probability distribution. On NYT, it is $0.03/0.13 = 23\%$; while on InScript, it is $0.28/0.82 = 34\%$. We believe such chance co-relates with the scaling parameter λ.

For qualitative analysis, we provide a comparative example between KnowSemLM and FES-RNNLM in practice. The system is fed into the following input:

... Jane wanted to buy a new car. She had to borrow some money from her father. ...

So, on an event level, we abstract the text as "PER[new]-want.01-buy.01-ARG[new](NEU), PER[old]-have.04-borrow.01-ARG[new](NEU)". For FES-RNNLM, the system predicts the next event as "PER[old]-sell.01-ARG[new](NEU)" since in training data, there are many co-occurrences between the "borrow" event and "sell" event (coming from financial news articles in NYT). In contrast, for KnowSemLM, since we have the knowledge "PER[*]-borrow.01-ARG[*](*) $\Rightarrow$ PER[old]-return.01-ARG[old](*)", meaning that something borrowed by someone is likely to be returned, the predicted event would be "PER[old]-return.01-ARG[old](NEU)". This is closer to the real text semantically: ... *She promised to return the money once she got a job* ... Such an example shows that KnowSemLM works in situations where 1) the required knowledge is stored in the event causality knowledge base, and 2) the training data contains scenarios where required knowledge is put into use.

6 Related Work

Our work is built upon the previous works for semantic language models (Peng and Roth, 2016; Peng et al., 2017; Chaturvedi et al., 2017). This line of work is in general inspired by script learning. Early works (Schank and Abelson, 1977; Mooney and DeJong, 1985) tried to learn scripts via construction of knowledge bases from text. More recently, researchers focused on utilizing statistical models to extract high-quality scripts from large amounts of data (Chambers and Jurafsky, 2008; Bejan, 2008; Jans et al., 2012; Pichotta and Mooney, 2014; Granroth-Wilding and Clark, 2016; Rudinger et al., 2015; Pichotta and Mooney, 2016a,b). Other works aimed at learning a collection of structured events (Chambers, 2013; Cheung et al., 2013; Balasubramanian et al., 2013; Bamman and Smith, 2014; Nguyen et al., 2015; Inoue et al., 2016). In particular, Ferraro and Durme (2016) presented a unified probabilistic model of syntactic and semantic frames while also demonstrating improved coherence. Several works have employed neural embeddings (Modi and Titov, 2014a,b; Frermann et al., 2014; Titov and Khoddam, 2015). Some prior works have used scripts-related ideas to help improve NLP tasks (Irwin et al., 2011; Rahman and Ng, 2011; Peng et al., 2015b).

Several recent works focus on narrative/story telling (Rishes et al., 2013), as well as studying event structures (Brown et al., 2017). Most recently, Mostafazadeh et al. (2016, 2017) proposed story cloze test as a standard way to test a system's ability to model semantics. They released ROCStories dataset, and organized a shared task for LSDSem'17; which yields many interesting works on this task. Cai et al. (2017) developed a model that uses hierarchical recurrent networks with at-

tention to encode sentences and produced a strong baseline. Lee and Goldwasser (2019) considered the problem of learning relation aware event embeddings for commonsense inference, which can account for different relations between events, beyond simple event similarity. We differ from them because the basic semantic unit we model is event level abstractions instead of word tokens.

The definition of event causality knowledge in this work includes temporal ordering relationships. Much progress has been made in identifying and modeling such relations. In early works (Mani et al., 2006; Chambers et al., 2007; Bethard et al., 2007; Verhagen and Pustejovsky, 2008), the problem was formulated as a classification problem for determining the pair-wise event temporal relations; while recent works (Do et al., 2012; Mirza and Tonelli, 2016; Ning et al., 2017, 2018) took advantage of utilizing structural constraints such as transitive properties of temporal relationships via ILP to achieve better results. Comparatively, the concept of event causality knowledge here is broader and more flexible. Any event causality relation gained from human experience could be represented and utilized in KnowSemLM; as shown in Sec. 4.2 that such knowledge can be both mined from corpus and written down declaratively.

Since we formulate the semantic sequence modeling problem as a language modeling issue, we also review recent neural language modeling literature. Bengio et al. (2003) introduced a model that learns word vector representations as part of a simple neural network architecture for language modeling. Collobert and Weston (2008) decoupled the word vector training from the downstream training objectives, which paved the way for Collobert et al. (2011) to use the full context of a word for learning the word representations. The skip-gram and continuous bag-of-words (CBOW) models of Mikolov et al. (2013) propose a simple single-layer architecture based on the inner product between two word vectors. Mnih and Kavukcuoglu (2013) also proposed closely-related vector log-bilinear models, vLBL and ivLBL, and Levy and Goldberg (2014) proposed explicit word embeddings based on a PPMI metric. Additionally, researcher have been attempting to infuse knowledge into the language modeling process (Ahn et al., 2016; Yang et al., 2016; Ji et al., 2017; He et al., 2017; Clark et al., 2018).

Most recently, pre-trained language models such as BERT (Devlin et al., 2019), GPT (Radford et al., 2018), and XLNET (Yang et al., 2019) have achieved much success for language modeling and generation tasks. Our proposed knowledge infused semantic language model can not be directly applied upon such word-level pre-trained language models. However, as future works, we are interested in exploring the possibility of pre-training a semantic language model with frame and entity abstractions on a large corpus with event causality knowledge, and fine-tune it on application tasks.

7 Conclusion

This paper proposes KnowSemLM, a knowledge infused semantic LM. It utilizes both local context (i.e., what has been described in text) and global context (i.e., causality knowledge about events) to predict future events. We show that such event causality knowledge can be obtained statistically from a corpus or declaratively in specific scenarios. Similar to previous works, KnowSemLM takes advantage of event-level abstractions to achieve generalization. Evaluations demonstrate that the knowledge awareness of the proposed KnowSemLM helps improve results on tasks such as the event cloze test and story/referent prediction.

Acknowledgments

We thank the anonymous reviewers for their insightful comments. This work was supported by the IBM-ILLINOIS Center for Cognitive Computing Systems Research (C3SR) - a research collaboration as part of the IBM AI Horizons Network, as well as by contracts HR0011-15-C-0113 and HR0011-18-2-0052 with the US Defense Advanced Research Projects Agency (DARPA). Approved for Public Release, Distribution Unlimited. The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

References

Sungjin Ahn, Heeyoul Choi, Tanel Pärnamaa, and Yoshua Bengio. 2016. A neural knowledge language model. *arXiv preprint arXiv:1608.00318*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Niranjan Balasubramanian, Stephen Soderland, Mausam, and Oren Etzioni. 2013. Generating coherent event schemas at scale. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

David Bamman and Noah A. Smith. 2014. Unsupervised discovery of biographical structure from text. *Transactions of the Association for Computational Linguistics (TACL)*.

Cosmin Adrian Bejan. 2008. Unsupervised discovery of event scenarios from texts. In *Proceedings of the 21st International Florida Artificial Intelligence Research Society Conference*.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of Machine Learning Research (JMLR)*.

Steven Bethard, James H Martin, and Sara Klingenstein. 2007. Timelines from text: Identification of syntactic temporal relations. In *Proceedings of the International Conference on Semantic Computing (ICSC)*.

Susan Brown, Claire Bonial, Leo Obrst, and Martha Palmer. 2017. The rich event ontology. In *Proceedings of the Events and Stories in the News Workshop*.

Zheng Cai, Lifu Tu, and Kevin Gimpel. 2017. Pay attention to the ending: Strong neural baselines for the ROC story cloze task. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Nathanael Chambers. 2013. Event schema induction with a probabilistic entity-driven model. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Nathanael Chambers and Daniel Jurafsky. 2008. Unsupervised Learning of Narrative Event Chains. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Nathanael Chambers, Shan Wang, and Dan Jurafsky. 2007. Classifying temporal relations between events. In *Proceedings of the Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*.

Snigdha Chaturvedi, Haoruo Peng, and Dan Roth. 2017. Story comprehension for predicting what happens next. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Jackie Chi Kit Cheung, Hoifung Poon, and Lucy Vanderwende. 2013. Probabilistic Frame Induction. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Elizabeth Clark, Yangfeng Ji, and Noah A Smith. 2018. Neural text generation in stories using entity representations as context. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: deep neural networks with multitask learning. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel P. Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research (JMLR)*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Quang Do, Wei Lu, and Dan Roth. 2012. Joint inference for event timeline construction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Francis Ferraro and Benjamin Van Durme. 2016. A unified bayesian model of scripts, frames and language. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*.

Lea Frermann, Ivan Titov, and Manfred Pinkal. 2014. A hierarchical bayesian model for unsupervised induction of script knowledge. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics (EACL)*.

Mark Granroth-Wilding and Stephen Clark. 2016. What Happens Next? Event Prediction Using a Compositional Neural Network Model. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*.

He He, Anusha Balakrishnan, Mihail Eric, and Percy Liang. 2017. Learning symmetric collaborative dialogue agents with dynamic knowledge graph embeddings. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Ruihong Huang and Ellen Riloff. 2012. Modeling textual cohesion for event extraction. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*.

Naoya Inoue, Yuichiroh Matsubayashi, Masayuki Ono, Naoaki Okazaki, and Kentaro Inui. 2016. Modeling context-sensitive selectional preference with distributed representations. In *Proceedings of the International Conference on Computational Linguistics (COLING)*.

Joseph Irwin, Mamoru Komachi, and Yuji Matsumoto. 2011. Narrative schema as world knowledge for coreference resolution. In *Proceedings of the SIGNLL Conference on Computational Natural Language Learning (CoNLL)*.

Bram Jans, Steven Bethard, Ivan Vulic, and Marie-Francine Moens. 2012. Skip n-grams and ranking functions for predicting script events. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics (EACL)*.

Heng Ji and Ralph Grishman. 2008. Refining event extraction through cross-document inference. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Yangfeng Ji, Chenhao Tan, Sebastian Martschat, Yejin Choi, and Noah A Smith. 2017. Dynamic entity representations in neural language models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Robin Jia and Percy Liang. 2016. Data recombination for neural semantic parsing. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Daniel Khashabi, Mark Sammons, Ben Zhou, Tom Redman, Christos Christodoulopoulos, Vivek Srikumar, Nicholas Rizzolo, Lev Ratinov, Guanheng Luo, Quang Do, Chen-Tse Tsai, Subhro Roy, Stephen Mayhew, Zhili Feng, John Wieting, Xiaodong Yu, Yangqiu Song, Shashank Gupta, Shyam Upadhyay, Naveen Arivazhagan, Qiang Ning, Shaoshi Ling, and Dan Roth. 2018. CogCompNLP: Your Swiss Army Knife for NLP. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*.

I-Ta Lee and Dan Goldwasser. 2019. Multi-relational script learning for discourse relations. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Omer Levy and Yoav Goldberg. 2014. Linguistic regularities in sparse and explicit word representations. In *Proceedings of the SIGNLL Conference on Computational Natural Language Learning (CoNLL)*.

Qi Li, Heng Ji, and Liang Huang. 2013. Joint event extraction via structured prediction with global features. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Inderjeet Mani, Marc Verhagen, Ben Wellner, Chong Min Lee, and James Pustejovsky. 2006. Machine learning of temporal relations. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *Proceedings of Workshop at ICLR*.

Paramita Mirza and Sara Tonelli. 2016. CATENA: Causal and temporal relation extraction from natural language texts. In *Proceedings of the International Conference on Computational Linguistics (COLING)*.

Andriy Mnih and Geoffrey Hinton. 2007. Three new graphical models for statistical language modelling. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Andriy Mnih and Koray Kavukcuoglu. 2013. Learning word embeddings efficiently with noise-contrastive estimation. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*.

Ashutosh Modi and Ivan Titov. 2014a. Inducing neural models of script knowledge. In *Proceedings of the SIGNLL Conference on Computational Natural Language Learning (CoNLL)*.

Ashutosh Modi and Ivan Titov. 2014b. Learning semantic script knowledge with event embeddings. In *ICLR Workshop*.

Ashutosh Modi, Ivan Titov, Vera Demberg, Asad Sayeed, and Manfred Pinkal. 2017. Modeling semantic expectation: Using script knowledge for referent prediction. *Transactions of the Association for Computational Linguistics (TACL)*.

Raymond J. Mooney and Gerald DeJong. 1985. Learning schemata for natural language processing. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James F. Allen. 2016. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Nasrin Mostafazadeh, Michael Roth, Annie Louis, Nathanael Chambers, and James Allen. 2017. LSDSem 2017 Shared Task: The Story Cloze Test. In *Proceedings of the 2nd Workshop on Linking Models of Lexical, Sentential and Discourse-level Semantics*.

Preksha Nema, Shreyas Shetty, Parag Jain, Anirban Laha, Karthik Sankaranarayanan, and Mitesh M.

Khapra. 2018. Generating descriptions from structured data using a bifocal attention mechanism and gated orthogonalization. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Kiem-Hieu Nguyen, Xavier Tannier, Olivier Ferret, and Romaric Besançon. 2015. Generative event schema induction with entity disambiguation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Thien Huu Nguyen, Kyunghyun Cho, and Ralph Grishman. 2016. Joint event extraction via recurrent neural networks. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Thien Huu Nguyen and Ralph Grishman. 2016. Modeling skip-grams for event detection with convolutional neural networks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Qiang Ning, Zhili Feng, and Dan Roth. 2017. A structured learning approach to temporal relation extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Qiang Ning, Hao Wu, Haoruo Peng, and Dan Roth. 2018. Improving Temporal Relation Extraction with a Globally Acquired Statistical Resource. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Haoruo Peng, Kai-Wei Chang, and Dan Roth. 2015a. A joint framework for coreference resolution and mention head detection. In *Proceedings of the SIGNLL Conference on Computational Natural Language Learning (CoNLL)*.

Haoruo Peng, Snigdha Chaturvedi, and Dan Roth. 2017. A joint model for semantic sequences: Frames, entities, sentiments. In *Proceedings of the SIGNLL Conference on Computational Natural Language Learning (CoNLL)*.

Haoruo Peng, Daniel Khashabi, and Dan Roth. 2015b. Solving hard coreference problems. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Haoruo Peng and Dan Roth. 2016. Two discourse driven language models for semantics. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Haoruo Peng, Yangqiu Song, and Dan Roth. 2016. Event detection and co-reference with minimal supervision. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Karl Pichotta and Raymond J. Mooney. 2014. Statistical script learning with multi-argument events. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics (EACL)*.

Karl Pichotta and Raymond J. Mooney. 2016a. Learning statistical scripts with LSTM recurrent neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*.

Karl Pichotta and Raymond J Mooney. 2016b. Using sentence-level lstm language models for script inference. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf*.

Kira Radinsky, Sagie Davidovich, and Shaul Markovitch. 2012. Learning causality for news events prediction. In *Proceedings of the International World Wide Web Conferences (WWW)*.

Kira Radinsky and Eric Horvitz. 2013. Mining the web to predict future events. In *Proceedings of the ACM International Conference on Web Search and Data Mining (WSDM)*.

Altaf Rahman and Vincent Ng. 2011. Coreference resolution with world knowledge. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Elena Rishes, Stephanie M Lukin, David K Elson, and Marilyn A Walker. 2013. Generating different story tellings from semantic representations of narrative. In *International Conference on Interactive Digital Storytelling*, pages 192–204. Springer.

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

Roger C. Schank and Robert P. Abelson. 1977. *Scripts, plans, goals, and understanding: An inquiry into human knowledge structures*. Oxford, England: Lawrence Erlbaum.

Yangqiu Song, Haoruo Peng, Parisa Kordjamshidi, Mark Sammons, and Dan Roth. 2015. Improving a pipeline architecture for shallow discourse parsing. In *Proceedings of the SIGNLL Conference on Computational Natural Language Learning (CoNLL)*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*.

Ivan Titov and Ehsan Khoddam. 2015. Unsupervised induction of semantic roles within a reconstruction-error minimization framework. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Marc Verhagen and James Pustejovsky. 2008. Temporal processing with the TARSQI toolkit. In *Proceedings of the International Conference on Computational Linguistics (COLING)*.

Zhongqing Wang, Yue Zhang, and Ching-Yun Chang. 2017. Integrating order information and event relation for script event prediction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. 2019. XLNet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*.

Zichao Yang, Phil Blunsom, Chris Dyer, and Wang Ling. 2016. Reference-aware language models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.