Scripts
Seminar: Hot & Odd Topics in Semantics

DAMYANA GATEVA

SAARLAND UNIVERSITY

JUNE 4, 2018
Overview

1. Introduction
2. Introduction
3. Motivation
4. Resources for Script Knowledge
5. Tasks
   - Mapping Text to Scripts
   - Learning Event Chains
Outline

1. Introduction

2. Introduction

3. Motivation

4. Resources for Script Knowledge

5. Tasks
   - Mapping Text to Scripts
   - Learning Event Chains
“A script is a predetermined, stereotyped sequence of actions that define a well-known situation”

(Schank and Abelson, 1975).
1. Introduction

2. Introduction

3. Motivation

4. Resources for Script Knowledge

5. Tasks
   - Mapping Text to Scripts
   - Learning Event Chains
John went into the restaurant.
he ordered a hamburger and a coke.
He asked the waitress for the check and left.
John went into the restaurant.  
he ordered a hamburger and a coke.  
He asked the waitress for the check and left.

John went to a restaurant.  
He ordered a hamburger.  
It was cold when the waitress brought it.  
He left her a very small tip.
What about these?
What about these?

Harriet went to a birthday party. She put on a green paper hat. Just when they sat down to eat the cake, a piece of plaster fell from the ceiling onto the table. She was lucky, because the dust didn’t get all over her hair.
What about these?

Harriet went to a birthday party. She put on a green paper hat. Just when they sat down to eat the cake, a piece of plaster fell from the ceiling onto the table. She was lucky, because the dust didn’t get all over her hair.

Harriet went to Jack’s birthday party. The cake tasted awful. Harriet left Jack’s mother a very small tip.
Scripts:
- contain prototypical actions/activities and the involved participants
- describe stereotypical events in an appropriate temporal and causal order
He pushed against the door. The room was empty.
He pushed against the door. The room was empty.

He pushed against the door. THE DOOR OPENED. HE LOOKED INSIDE. HE SAW THAT The room was empty.

→ Scripts enable inference about not explicitly mentioned events
Jane was invited to Tom’s birthday party. She wondered if he would like a kite.
Jane was invited to Tom’s birthday party. She wondered if he would like a kite. She went to her room and shook her piggy bank. It made no sound.
Jane was invited to Tom’s birthday party. She wondered if he would like a kite.
Jane was invited to Tom’s birthday party. She wondered if he would like a kite. He already has a kite. He will make you take it back.
Outline

1. Introduction

2. Introduction

3. Motivation

4. Resources for Script Knowledge

5. Tasks
   - Mapping Text to Scripts
   - Learning Event Chains
Motivation

Script knowledge:

- guides text understanding
- support coreference resolution
- enable common-sense knowledge inference
- determine text-internal temporal order
Motivation

Applications:

John went to a restaurant. The hostess seated John. The hostess gave John a menu. The waiter came to the table. John ordered lobster. John was served quickly. John left a large tip. John left the restaurant.

SAM (Script Applier Mechanism) (Schank and Abelson, 1975)
Motivation

Applications:

- question answering

John went to a restaurant. The hostess seated John. The hostess gave John a menu. The waiter came to the table. John ordered lobster. John was served quickly. John left a large tip. John left the restaurant.

Q: Who gave John the lobster?
A: PROBABLY THE WAITER.

Q: Who paid the check?
A: PROBABLY JOHN.

SAM (Script Applier Mechanism) (Schank and Abelson, 1975)
Motivation

Applications:

- question answering
- paraphrase

John went to a restaurant. The hostess seated John. The hostess gave John a menu. The waiter came to the table. John ordered lobster. John was served quickly. John left a Large tip. John left the restaurant.

JOHN DECIDED HE WAS GOING TO GO TO A RESTAURANT. JOHN WENT TO A RESTAURANT. JOHN INDICATED TO THE HOST HE WOULD LIKE TO BE AT A TABLE …

SAM (Script Applier Mechanism) (Schank and Abelson, 1975)
Motivation

Applications:

• question answering
• paraphrase
• summary

John went to a restaurant. The hostess seated John. The hostess gave John a menu. The waiter came to the table. John ordered lobster. John was served quickly. John left a Large tip. John left the restaurant.

JOHN WENT TO A RESTAURANT AND ATE LOBSTER.

SAM (Script Applier Mechanism) (Schank and Abelson, 1975)
Outline

1. Introduction
2. Introduction
3. Motivation
4. Resources for Script Knowledge
5. Tasks
   - Mapping Text to Scripts
   - Learning Event Chains
Resources for Script Knowledge

- InScript (Modi, Anikina, Ostermann, Pinkal; 2016)
- DeScript (Wanzare, Zarcone, Thater, Pinkal; 2016)
InScript

Sample event and participant annotation for the TAKING A BATH script

→ Texts: 910 stories from 10 scenarios
→ Annotation: event type, participants, coreference
DeScript

Example Event Sequence Descriptions (ESDs) + Alignments

→ Texts: 40 scenarios, 100 ESDs per scenario
→ Annotation: Aligned ESDs for 10 scenarios
Outline

1. Introduction
2. Introduction
3. Motivation
4. Resources for Script Knowledge
5. Tasks
   - Mapping Text to Scripts
   - Learning Event Chains
Mapping Text to Scripts

- Aligning Script Events with Narrative Texts (Ostermann, Roth, Thater, Pinkal; 2017)
  - identifying script related verbs and determining their event label
Mapping Text to Scripts

- Task: label the texts from InScript using the training data from DeScript

- Steps:
  1) identify script related verbs (decision tree classifier)
  2) label the event types (sequence labeling classifier)
Determining Script Verb

1) Identifying script related verbs (decision tree classifier)
   • Features:
     ▶ **syntactic features:**
       - verbs that rarely denote script events (auxiliaries, verbs governing adverbial phrase, non-action verbs)
       - number of direct and indirect objects
     ▶ **script features:**
       - verb used in ESD? [y/n]
       - scenario specific tf-idf score
     ▶ **frame features:**
       - frames predicted by semantic role labeler based on FrameNet
Determining Script Verb

1) Identifying script related verbs (decision tree classifier)

<table>
<thead>
<tr>
<th></th>
<th>P</th>
<th>R</th>
<th>F₁</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lemma</td>
<td>0.365</td>
<td>0.949</td>
<td>0.526</td>
</tr>
<tr>
<td>Our model</td>
<td>0.628</td>
<td>0.817</td>
<td>0.709</td>
</tr>
<tr>
<td>Our model (scen. indep.)</td>
<td>0.513</td>
<td>0.877</td>
<td>0.645</td>
</tr>
</tbody>
</table>

- Baseline Lemma: always assigns the event class if the verb lemma is mentioned in DeScript
- use 10-fold cross validation within all texts of one scenario
- for scenario independent: using 10-fold cross validation within the 10 scenarios as one fold and excluding script features
Determining Script Verb Label

2) Labeling the event types (sequence labeling classifier, CRF)
   • Features:
     ▶ sequential feature:
       - co-occurrence of two event type labels in a sequence [y/n]
     ▶ meaning representation features:
       - precomputed word embeddings: distributional information of the verb and its arguments - the lemma of the verb and its arguments
2) Labeling the event types (sequence labeling classifier, CRF)

<table>
<thead>
<tr>
<th></th>
<th>P</th>
<th>R</th>
<th>F₁</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lemma</td>
<td>0.343</td>
<td>0.416</td>
<td>0.374</td>
</tr>
<tr>
<td>Word2vec</td>
<td>0.356</td>
<td>0.448</td>
<td>0.395</td>
</tr>
<tr>
<td>CRF model</td>
<td>0.608</td>
<td>0.496</td>
<td>0.545</td>
</tr>
<tr>
<td>CRF, no seq.</td>
<td>0.599</td>
<td>0.487</td>
<td>0.536</td>
</tr>
</tbody>
</table>

- Baseline Lemma: always assigns the event class if the verb lemma is mentioned in DeScript
- Baseline word2vec: assign label based on cosine similarity between test verb and dependents and all ESDs in DeScript
Determining Script Verb & Label

- Full Text-to-Script Mapping Task

<table>
<thead>
<tr>
<th>Model</th>
<th>P</th>
<th>R</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ident. model + Lemma</td>
<td>0.253</td>
<td>0.451</td>
<td>0.323</td>
</tr>
<tr>
<td>Ident. model + Word2vec</td>
<td>0.255</td>
<td>0.477</td>
<td>0.331</td>
</tr>
<tr>
<td>Ident. model + CRF model</td>
<td>0.445</td>
<td>0.520</td>
<td>0.479</td>
</tr>
</tbody>
</table>

- Baseline Lemma: always assigns the event class if the verb lemma is mentioned in DeScript
- Baseline word2vec: assign label based on cosine similarity between test verb and dependents and all ESDs in DeScript
Results and Discussion

- Overall good results given the size of the corpus
- Problems:
  - lexical coverage
  - misleading ordering information
  - similar wrong label
- Future work: paraphrase identification, event prediction
Learning Event Chains

- Unsupervised Learning of Narrative Event Chains (Chambers & Jurafsky, 2008)
  → building narrative chains:
  - extracting relevant events that occur with the same protagonist
  - assign temporal order for these events
The Narrative Chain Model

- Protagonist and narrative coherence: Verbs sharing coreferring arguments are probably related

Tom pushed Jerry. Jerry fell.
Protagonist Tom: (push, Subj)
Protagonist Jerry: (push, Obj), (fall, Subj)
The Narrative Chain Model

- Narrative chain:
  - a set of related narrative events:
    \[ \{ e_1: (\text{push, Obj}), e_2: (\text{fall, Subj}) \} \]
  - partially ordered by a temporal relation:
    before: \( B(e_1, e_2) \)
Learning Event Chains

- Find verbs that have co-referential argument
- Extract verbs as pairwise relations: (event, dependency) + their counts

Tom denied pushing Jerry. (protagonist: Tom)  
(deny, Subj): 1

- Use these counts to compute the PMI between two narrative events
Learning Event Chains

Task: Guessing the next event given preceding events

- calculate \( \text{PMI}(e_1, e_x) \) for all pairs of chain events and candidate events
- sum over the values for each chain event
- get a list of most probable events that maximize PMI

<table>
<thead>
<tr>
<th>Known events:</th>
<th>(pleaded subj), (admits subj), (convicted obj)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Likely Events:</td>
<td>---------------------------------------------</td>
</tr>
<tr>
<td>sentenced obj</td>
<td>0.89</td>
</tr>
<tr>
<td>paroled obj</td>
<td>0.76</td>
</tr>
<tr>
<td>fired obj</td>
<td>0.75</td>
</tr>
<tr>
<td>indicted obj</td>
<td>0.74</td>
</tr>
<tr>
<td>fined obj</td>
<td>0.73</td>
</tr>
<tr>
<td>denied subj</td>
<td>0.73</td>
</tr>
</tbody>
</table>
Evaluation: Narrative Cloze Task

- Remove one event from a narrative chain
- Get a rank list of candidates for the missing event
- Models
  - Baseline: learns relatedness based upon verb co-occurrence
  - Protagonist model: ignores typed dependencies
  - Typed dependencies model
Temporal Ordering

Determining if a temporal chain is ordered

- Labeling temporal attributes of events: tense, grammatical aspect, aspectual class
- Classifying the temporal relationship between two events that share arguments
- Choose the correct ordering among 300 random orderings

<table>
<thead>
<tr>
<th></th>
<th>All</th>
<th>≥ 6</th>
<th>≥ 10</th>
</tr>
</thead>
<tbody>
<tr>
<td>correct</td>
<td>8086</td>
<td>7603</td>
<td>6307</td>
</tr>
<tr>
<td>incorrect</td>
<td>1738</td>
<td>1493</td>
<td>619</td>
</tr>
<tr>
<td>tie</td>
<td>931</td>
<td>627</td>
<td>160</td>
</tr>
</tbody>
</table>
Results and Discussion

Achievements:

- extracting list of related events using shared argument (protagonist)
- unsupervised learning of narrative chains
- temporal classification of narrative events
Summary

• Scripts can be helpful for many different tasks such as:
  - coreference resolution
  - text understanding and summarization
  - recognizing paraphrases and entailment
References


