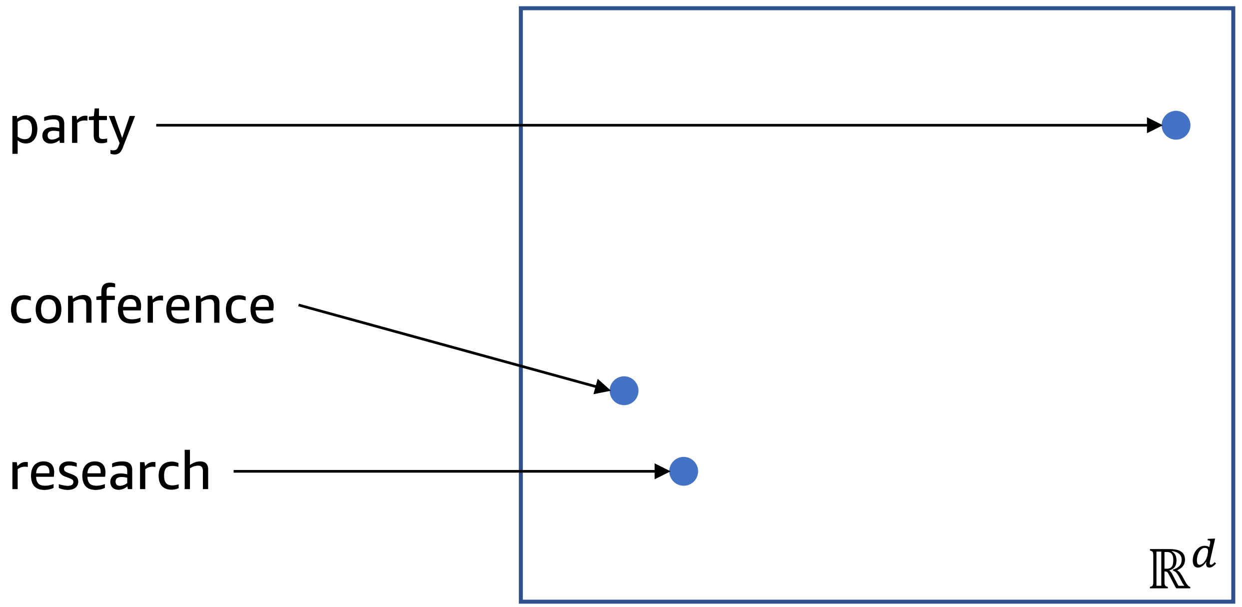
Multiplicative Tree-Structured LSTMs

Weiwei Cheng

Amazon

joint work with Nam Khanh Tran

Distributed Word Representations



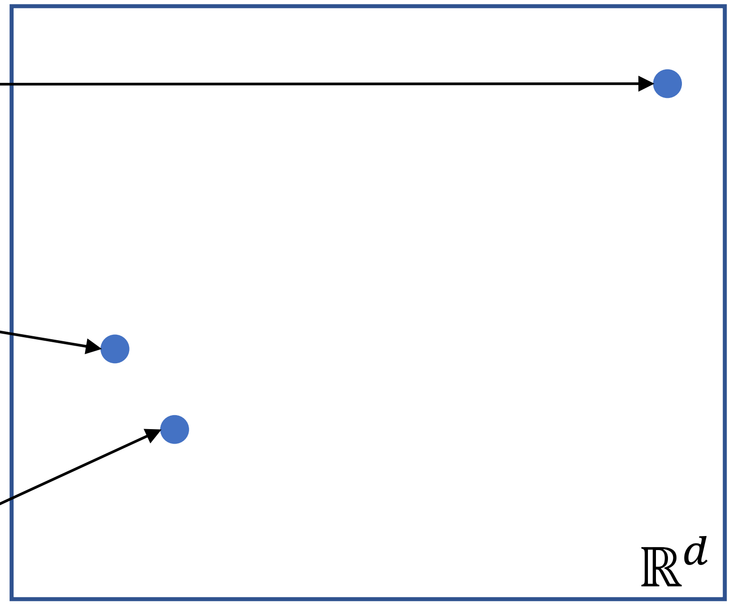
- Words are represented as real-valued vectors
- *E.g.*, Skip-Gram, GloVe

Distributed Sentence Representations

A lone biker is jumping in the air

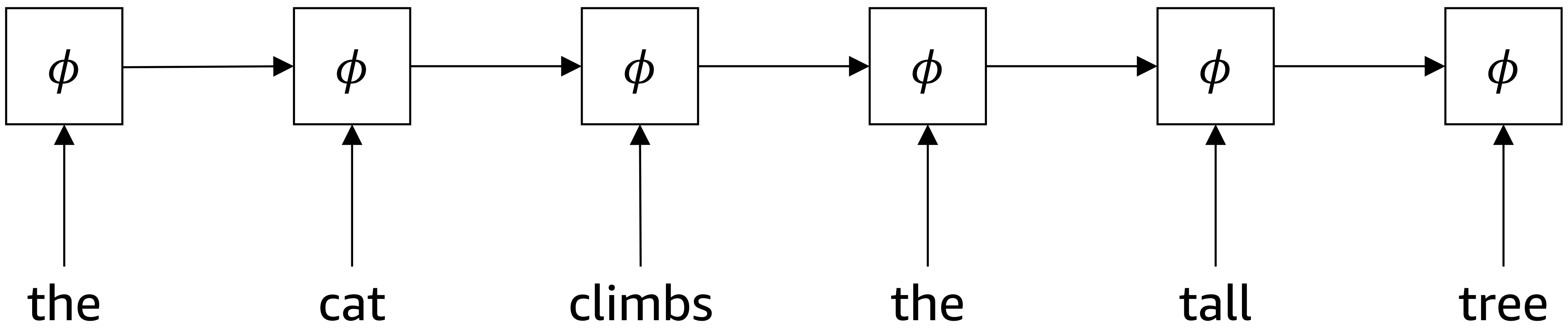
The kids are playing outdoors near
a man with a smile

The young boys are playing outdoors and
the man is smiling nearby



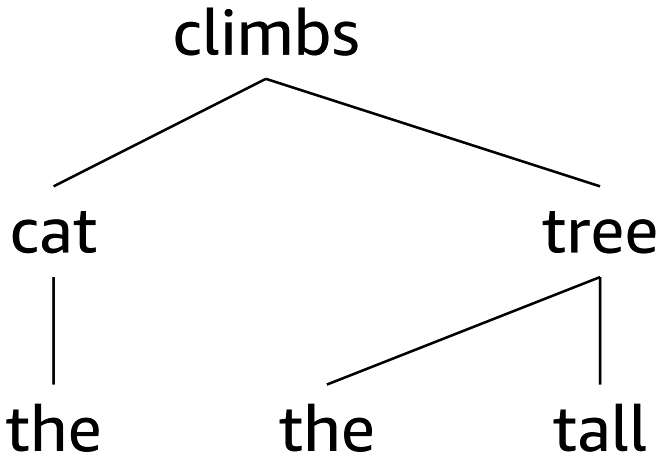
- Sentences are represented as real-valued vectors
- Useful in, *e.g.*, sentence classification, natural language understanding

Sequence Models (RNN)

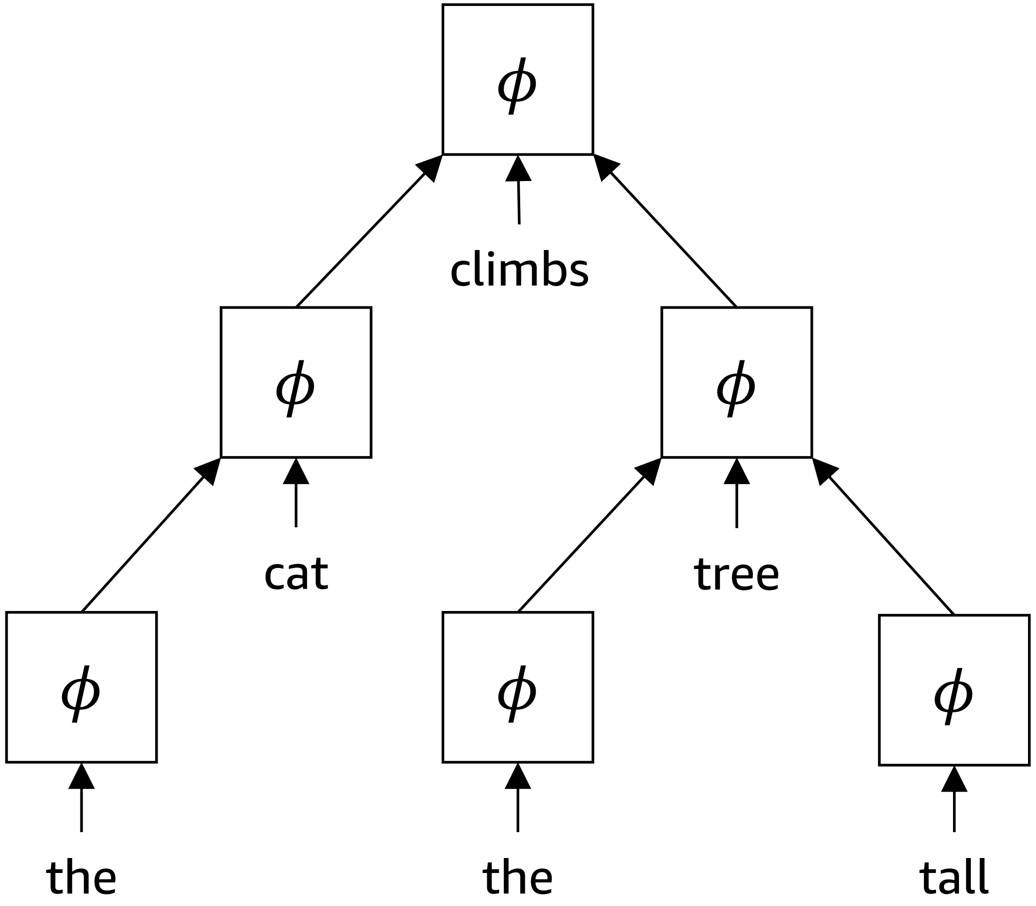


- Process sentence from left-to-right
- Input at each step is a word vector and previous hidden state
- Rightmost output is the representation of the sentence
- Common parameterization: RNN, LSTM

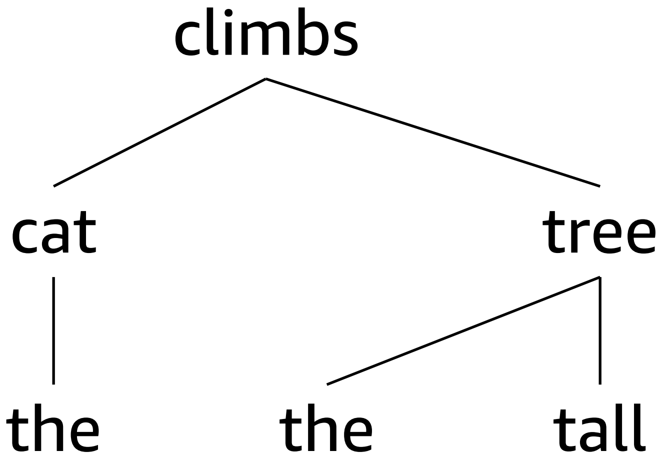
Recursive Tree Model



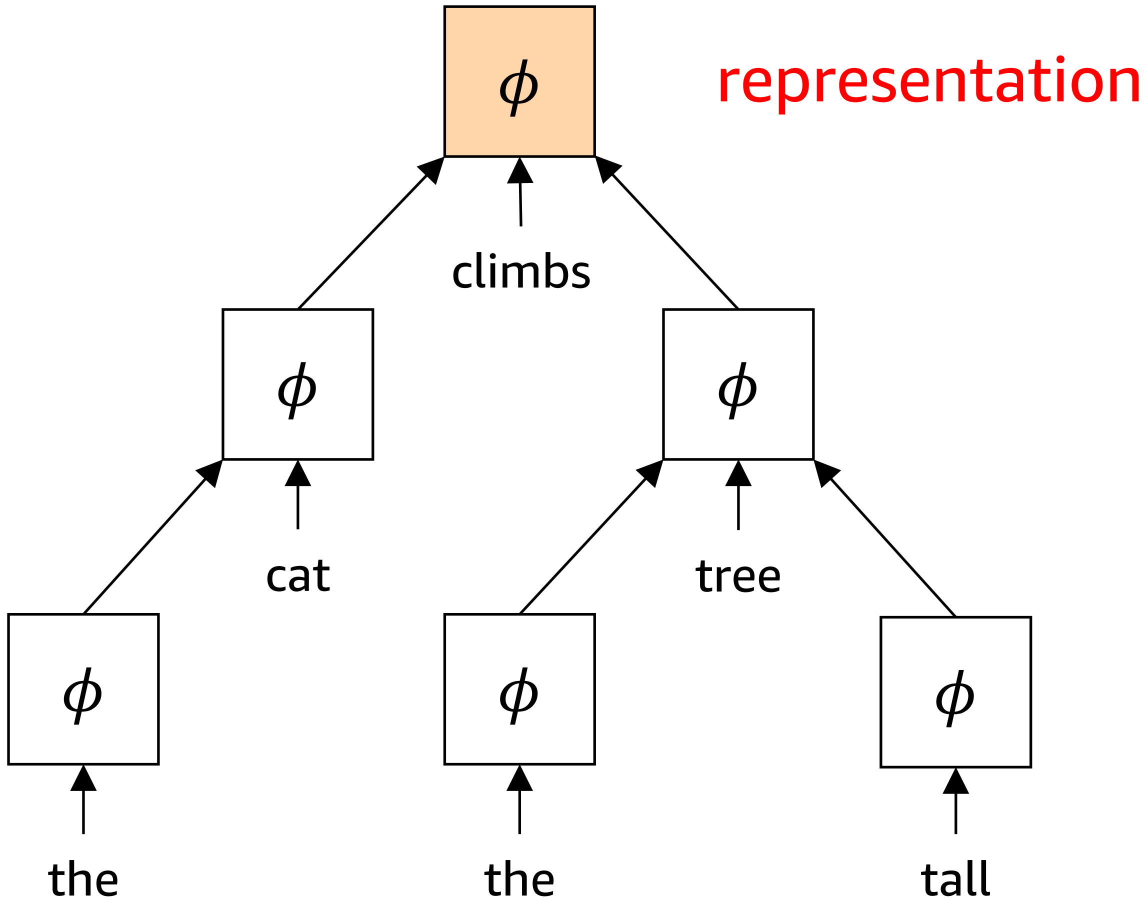
dependency tree



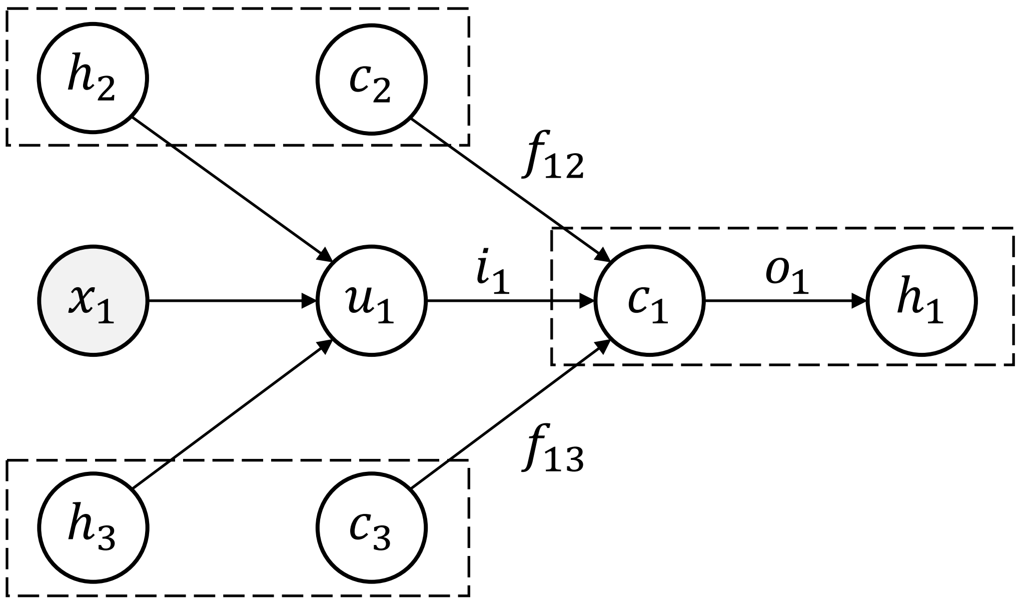
Recursive Tree Model



dependency tree



TreeLSTMs [Tai et al. 2015]



$$\tilde{h}_j = \sum_{k \in C(j)} h_k$$

$$i_j = \sigma\left(W^{(i)} x_j + U^{(i)} \tilde{h}_j + b^{(i)}\right)$$

$$o_j = \sigma\left(W^{(o)} x_j + U^{(o)} \tilde{h}_j + b^{(o)}\right)$$

$$f_{jk} = \sigma\left(W^{(f)} x_j + U^{(f)} h_k + b^{(f)}\right)$$

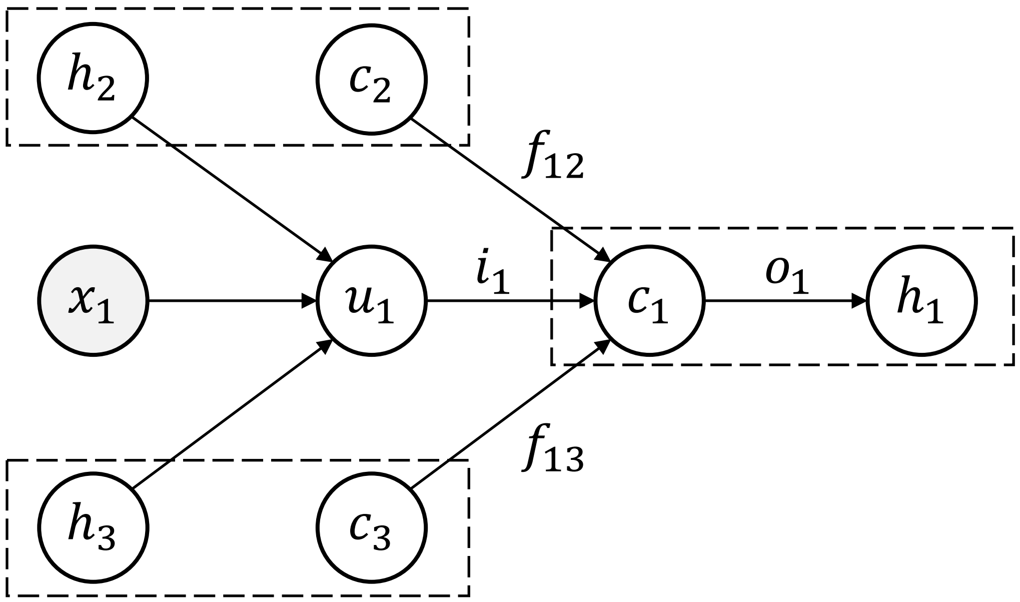
$$u_j = \tanh\left(W^{(u)} x_j + U^{(u)} \tilde{h}_j + b^{(u)}\right)$$

$$c_j = i_j \odot u_j + \sum_{k \in C(j)} f_{jk} \odot c_k$$

$$h_j = o_j \odot \tanh(c_j)$$

K. Tai, R. Socher, C. Manning. Improved semantic representations from tree-structured LSTMs. ACL 2015.

TreeLSTMs



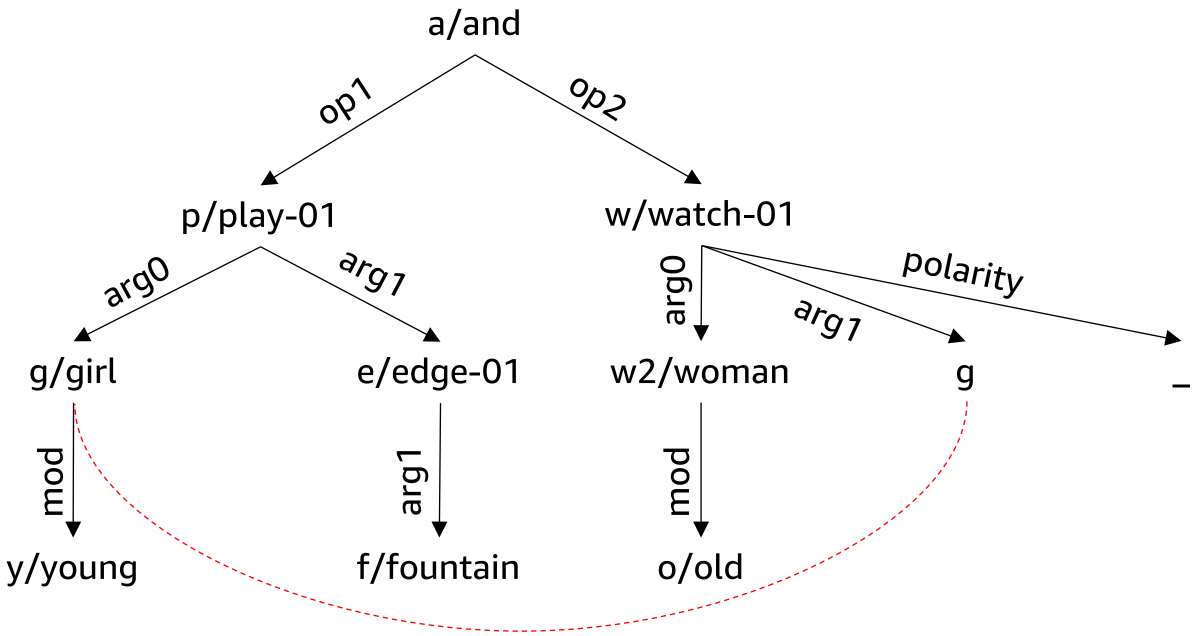
- Generalization of the sequential LSTM composition function
- A separate forget gate for each child
- Selectively preserve information from each child

This Work

- We propose multiplicative TreeLSTM that utilizes not only the lexical information of words, but also the relation information between the words.

- We investigate the use of lexical semantic information induced by Abstract Meaning Representation in tree structured models.

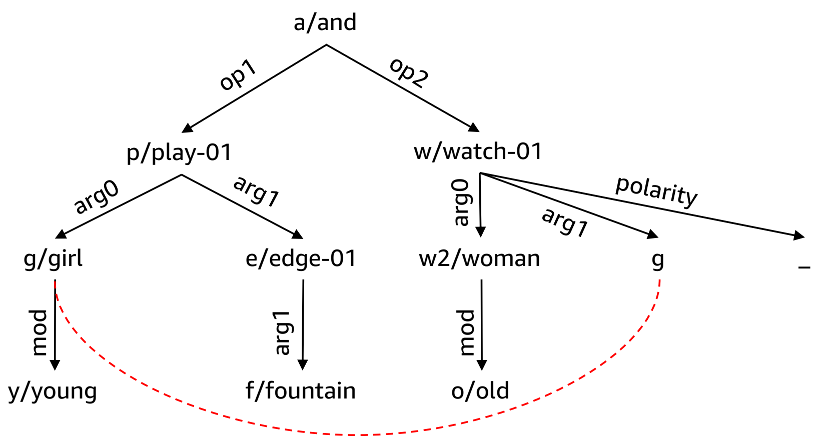
Abstract Meaning Representation (AMR)




```
(a / and
      :op1 (p / play-01
            :ARG0 (g / girl
                  :mod (y / young))
            :ARG1 (e / edge-01
                  :ARG1 (f / fountain)))
      :op2 (w / watch-01
            :ARG0 (w2 / woman
                  :mod (o / old))
            :ARG1 g
            :polarity -))
```

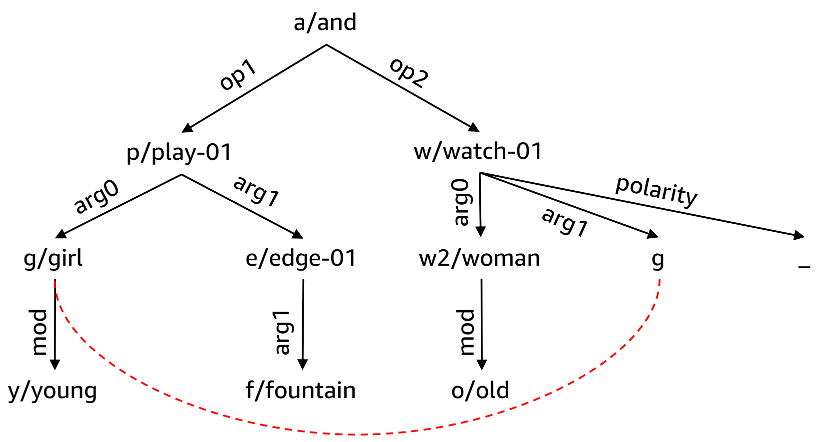
A young girl is playing on the edge of a fountain and an older woman is not watching her

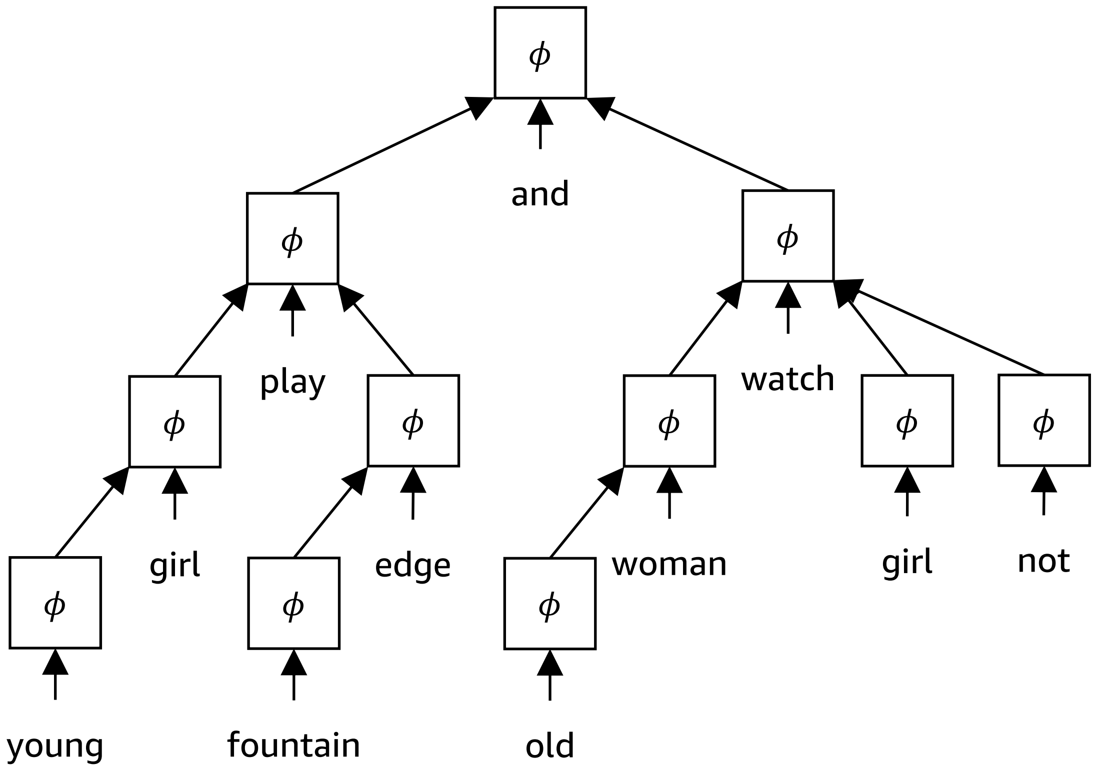
Abstract Meaning Representation (AMR)



- AMR is a semantic formalism where the meaning of a sentence is encoded as a single rooted, directed, acyclic graph
- **AMR concepts**: predicate senses, named entity annotations, and lemmas
- **AMR relations**: semantic roles (Propbank), semantic relations defined specifically for AMR

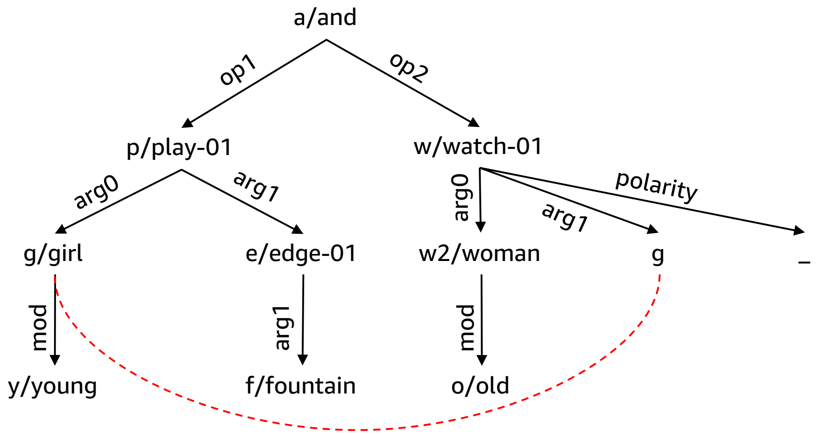
Tree-Structured LSTMs with AMR

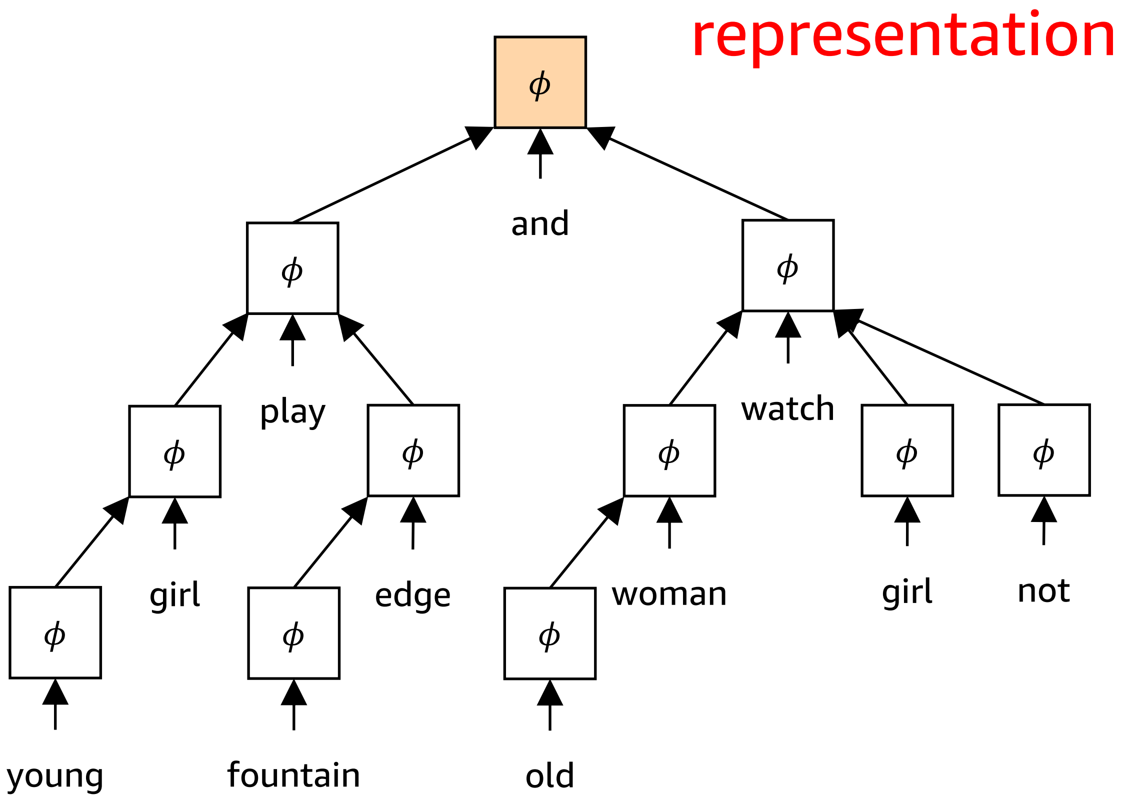




Recursively apply ϕ on the AMR structure

Tree-Structured LSTMs with AMR

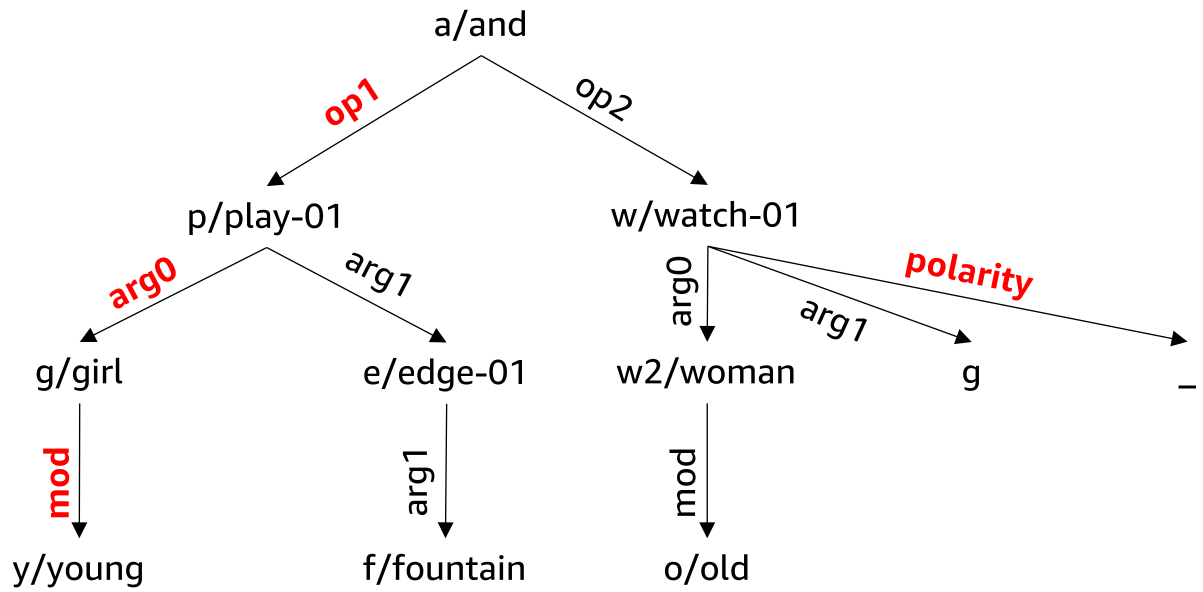




Recursively apply ϕ on the AMR structure

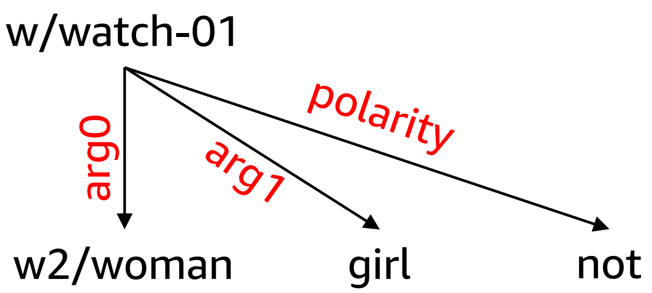
Tree-Structured LSTMs with AMR

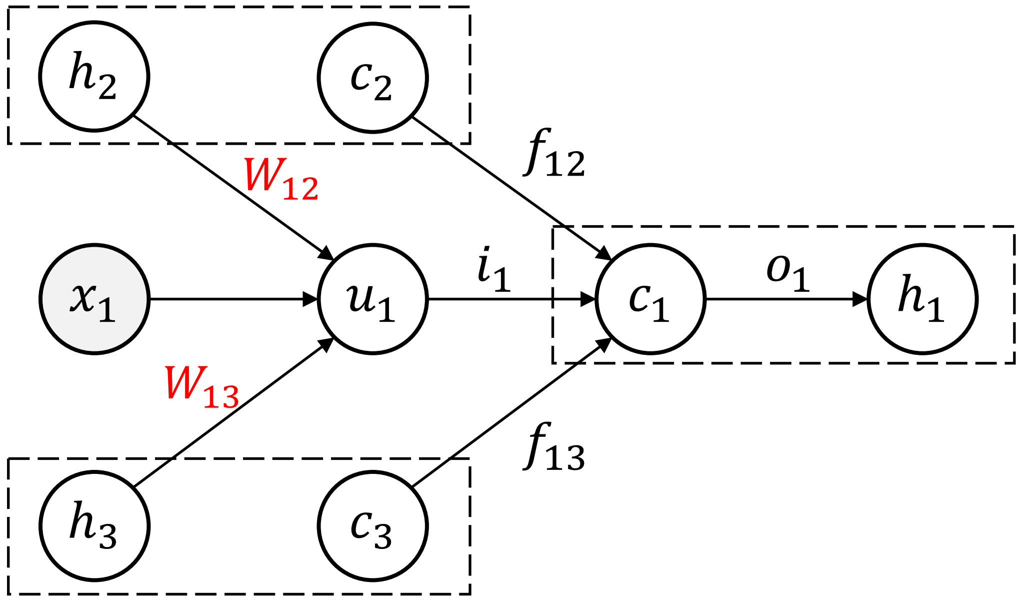
- **Observation**: Distinct edges / relations between nodes → Possibility for flexible parametrization
- **Proposal**: Multiplicative TreeLSTMs for modeling relations



Our Contribution – Multiplicative TreeLSTM

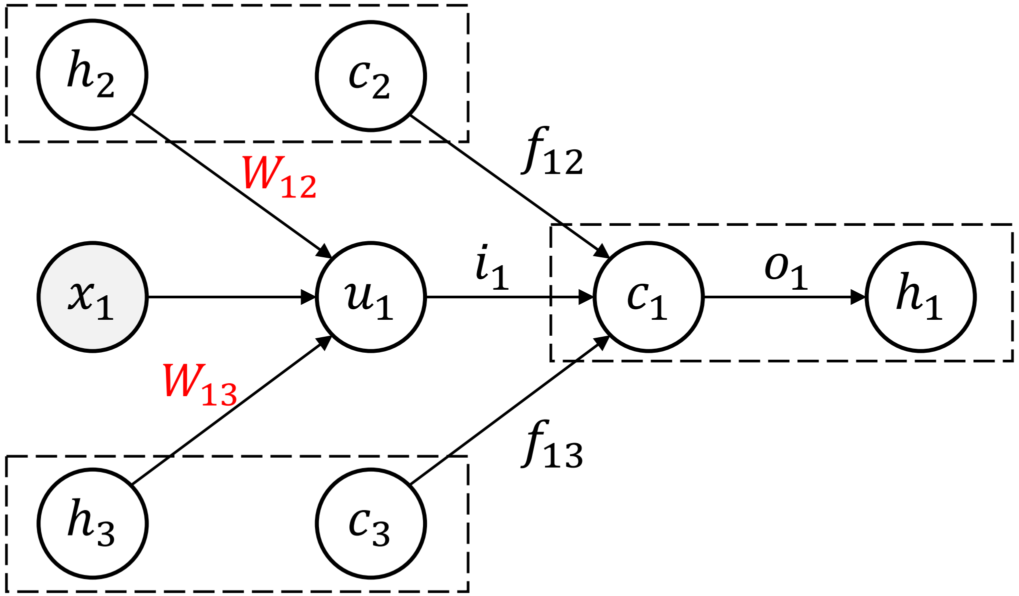
- **Main idea**: Introduce fine-grained parameters based on the edge types, *i.e.*, separate transition matrix for each edge type





Our Contribution – Multiplicative TreeLSTM

- **Main idea**: Introduce fine-grained parameters based on the edge types



$$\tilde{h}_j = \sum_{k \in C(j)} W_{hh}^{r(j,k)} h_k$$

$$i_j = \sigma\left(W^{(i)} x_j + U^{(i)} \tilde{h}_j + b^{(i)}\right)$$

$$o_j = \sigma\left(W^{(o)} x_j + U^{(o)} \tilde{h}_j + b^{(o)}\right)$$

$$f_{jk} = \sigma\left(W^{(f)} x_j + U^{(f)} h_k + b^{(f)}\right)$$

$$u_j = \tanh\left(W^{(u)} x_j + U^{(u)} \tilde{h}_j + b^{(u)}\right)$$

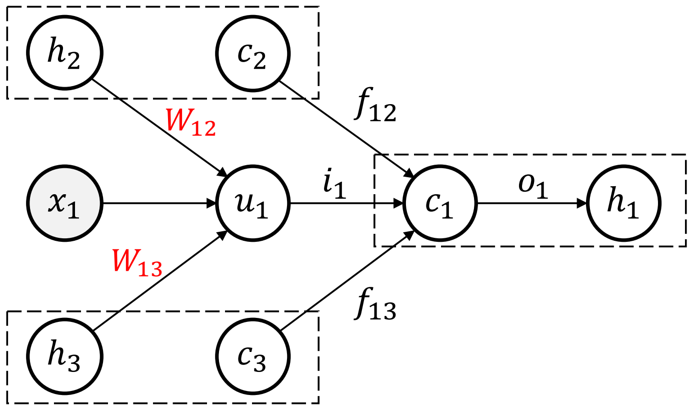
$$c_j = i_j \odot u_j + \sum_{k \in C(j)} f_{jk} \odot c_k$$

$$h_j = o_j \odot \tanh(c_j)$$

Our Contribution – Multiplicative TreeLSTM

- **Main idea**: Introduce fine-grained parameters based on the edge types

- Use semantic relations to combine hidden states from children nodes, via W_{hh}
  - But, 3-way tensors require large number of parameters → overfitting



$$\tilde{h}_j = \sum_{k \in C(j)} W_{hh}^{r(j,k)} h_k$$

Our Contribution – Multiplicative TreeLSTM

- **Solution**: Factorize $W_{hh}^{r(j,k)}$ by using the product of two dense matrices shared across edge types, with an diagonal matrix that is edge-type dependent

$$W_{hh}^{r(j,k)} = W_{hm}\,\mathrm{diag}\left(W_{mr} e_{jk}\right) W_{mh}$$

- The mapping $\tilde{h}_j = \sum_{k \in C(j)} W_{hh}^{r(j,k)} h_k$ is then given by

$$m_{jk} = \left(W_{mr} e_{jk}\right) \odot \left(W_{mh} h_k\right)$$

$$\tilde{h}_j = \sum_{k \in C(j)} W_{hm} m_{jk}$$

Our Contribution – Multiplicative TreeLSTM

- **mTreeLSTM** can be applied to any tree, where connection types between nodes are given, *e.g.*, dependency trees, AMR trees
    - Fewer number of parameters
    - Leverage potential correlation among fine-grained edge types
- In the experiments, we apply mTreeLSTM on a wide range of NLP tasks, including **sentiment classification**, **sentence relatedness**, and **natural language inference**, and investigate the usefulness of AMR.

Experiment 1: Sentiment Classification

- **Task**: Predict the sentiment of review sentences
  - Binary subtask: *positive* vs. *negative*
  - 5-class subtask: *strongly positive*, *positive*, *neutral*, *negative*, *strongly negative*
- **Data**: Stanford Sentiment Treebank [Socher et al. 2013]
- **Model**:
  - TreeLSTMs on given parse trees
  - Softmax classifier at root node

R. Socher, A. Perelygin, J. Wu, J. Chuang, C. Manning, A. Ng, C. Potts. Recursive deep models for semantic compositionality over a sentiment treebank. EMNLP 2013.

Experiment 1: Results

model	fine-grained	binary
LSTM	45.6	85.6
TreeLSTM (C)	46.3	**85.8**
TreeLSTM (D)	46.0	85.0
TreeLSTM (A)	44.4	82.9
mTreeLSTM (A)	45.2	83.2
mTreeLSTM (D)	**46.7**	85.7

Whenever a tree structure is applicable to both mTreeLSTM and TreeLSTM, the performance of mTreeLSTM with that tree structure is better.

Experiment 2: Sentence Relatedness

- **Task**: Predict how related two sentences are
- **Data**:
  - SICK from SemEval 2014 Task 1 [Marelli et al. 2014]
  - Manually annotated relatedness scores from 1 to 5
- **Model**:
  - TreeLSTMs on given parse trees
  - Relatedness scores predicted by an additional feedforward layer on top

M. Marelli, L. Bentivogli, M. Baroni, R. Bernardi, S. Menini, R. Zamparelli. SemEval-2014 Task 1. SemEval 2014.

Experiment 2: Results

model	Pearson	Spearman	MSE
LSTM	0.8409	0.7782	0.3035
TreeLSTM (C)	0.8497	0.7904	0.2861
TreeLSTM (D)	0.8631	0.8034	0.2600
TreeLSTM (A)	0.8415	0.7742	0.2986
mTreeLSTM (A)	0.8527	0.7884	0.2788
mTreeLSTM (D)	**0.8717**	**0.8141**	**0.2443**

Dependency trees and mTreeLSTM work the best.

Experiment 3: Natural Language Inference

- **Task**: Predict relation of two sentences – *entailment*, *contradiction*, *neutral*

- **Data**:
  - SICK from SemEval 2014 Task 1
  - Stanford NLI dataset [Bowman et al. 2015]

- **Model**:
  - TreeLSTMs on given parse trees
  - Classes predicted by an additional feedforward layer on top

S. Bowman, G. Angeli, C. Potts, C. Manning. A large annotated corpus for learning natural language inference. EMNLP 2015.

Experiment 3: SICK

model	all	long sentence	negation
LSTM	77.3	74.6	77.5
TreeLSTM (C)	79.0	78.1	85.3
TreeLSTM (D)	82.9	81.0	84.3
TreeLSTM (A)	82.6	84.0	88.2
mTreeLSTM (A)	83.3	**85.3**	**88.5**
mTreeLSTM (D)	**84.0**	81.6	87.8

Experiment 3: Model Size

model	# parameters	accuracy (%)
TreeLSTM (D)	301K	82.9
addTreeLSTM (D)	361K	83.4
fullTreeLSTM (D)	1.1M	83.5
mTreeLSTM (D)	361K	84.0

Experiment 3: SNLI

model	**accuracy (%)**
LSTM [Bowman et al., 2015]	77.6
Syntax TreeLSTM [Yogatama et al., 2017]	80.5
CYK TreeLSTM [Maillard et al., 2017]	81.6
Gumbel TreeLSTM [Choi et al., 2018]	81.8
Gumbel TreeLSTM + leaf LSTM [Choi et al., 2018]	82.6
TreeLSTM (D)	81.0
mTreeLSTM (D)	81.9

Conclusions

- We present **multiplicative TreeLSTM** for distributed sentence representation that utilizes not only the lexical information of words, but also the **relation information** between the words.
- Modeling relation information is helpful: **mTreeLSTMs outperform TreeLSTMs** on the same parse trees.
- With **AMR** as backbone, tree structured models can effectively handle long-range and complex dependencies.

Multiplicative Tree-Structured LSTMs

Weiwei Cheng

Amazon

joint work with Nam Khanh Tran