# Dependency Parsing for Weibo: An Efficient Probabilistic Logic Programming Approach

William Yang Wang, Lingpeng Kong, Kathryn Mazaitis, William W. Cohen
Carnegie Mellon University, Pittsburgh, PA, USA.

(problems & methods)

## 1 dependency parsing for weibo

- **motivation:** weibo attracts 30% of internet users, but NLP techniques for analyzing weibo are not well-studied.
- **question:** can we find efficient inference and learning methods for dependency arc prediction on weibo?
- **goals:**
  - develop a *new chinese weibo treebank*
  - make parser *programmable* via theory engineering
  - *efficient* non-linear first-order probabilistic logic learning
  - *effective* and *efficient* inference of the dependency structure

## 2 parsing via ProPPR

**•programming with personalized pagerank (ProPPR)**
**(Wang et al., 2013 CIKM, 2014 CIKM, 2015 MLJ)**

ProPPR is a new, scalable probabilistic first-order logic, where we assign feature vectors for each clause, and perform supervised personalized pagerank learning to upweight weights on edges (features) that lead to positive solutions, and perform an approximate inference by performing locally-groundable search with personalized pagerank.

### strategies

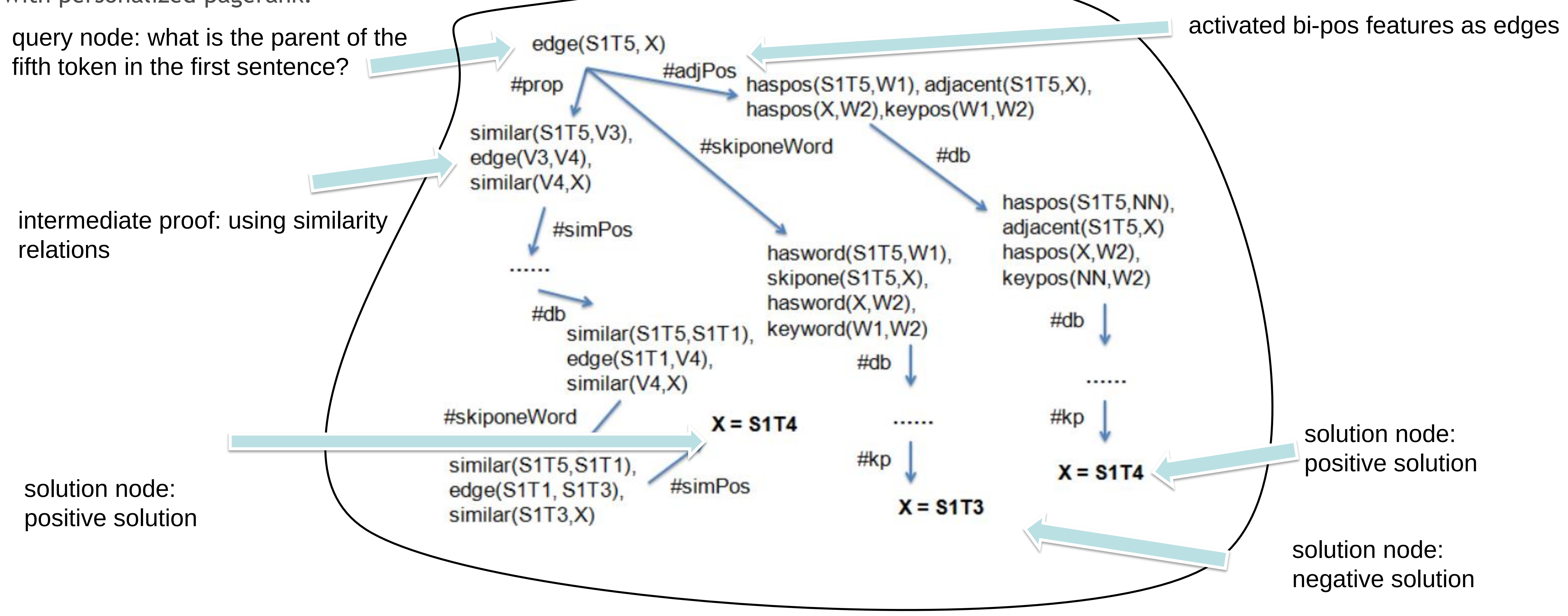
- **ProPPR inference with first-order rules:**

```
edge(V1,V2) :-
  adjacent(V1,V2),hasword(V1,W1),
  hasword(V2,W2),keyword(W1,W2) #adjWord.

edge(V1,V2) :-
  adjacent(V1,V2),haspos(V1,W1),
  haspos(V2,W2),keypos(W1,W2) #adjPos.

keyword(W1,W2) :- # kw(W1,W2).
keypos(W1,W2) :- # kp(W1,W2).
```

An example dependency arc prediction ProPPR program


query node: what is the parent of the fifth token in the first sentence?
activated bi-pos features as edges
edge(S1T5, X)
#prop
#adjPos
haspos(S1T5,W1), adjacent(S1T5,X), haspos(X,W2),keypos(W1,W2)
similar(S1T5,V3), edge(V3,V4), similar(V4,X)
#skiponeWord
#db
intermediate proof: using similarity relations
#simPos
haspos(S1T5,NN), adjacent(S1T5,X) haspos(X,W2), keypos(NN,W2)
hasword(S1T5,W1), skipone(S1T5,X), hasword(X,W2), keyword(W1,W2)
......
#db
similar(S1T5,S1T1), edge(S1T1,V4), similar(V4,X)
#db
#db
......
#skiponeWord
X = S1T4
......
#kp
#kp
X = S1T4
similar(S1T5,S1T1), edge(S1T1, S1T3), similar(S1T3,X)
#simPos
X = S1T3
solution node: positive solution
solution node: positive solution
solution node: negative solution

An example of grounded ProPPR program for dependency arc prediction

## 3 efficient non-linear weight learning and inference

- **learning**: a parallel stochastic gradient descent algorithm to optimize the log loss via supervised personalized pagerank:

$$\Pr_{\mathbf{W}}(v|u) = \frac{1}{Z} f(\mathbf{w}, \phi(\mathbf{x}_{u,v}))$$

$$-\left(\sum_{k=1}^{I} \log \mathbf{p}_{v_0}[u_+^k] + \sum_{k=1}^{J} \log(1 - \mathbf{p}_{v_0}[u_-^k])\right) + \mu||\mathbf{w}||_2^2$$

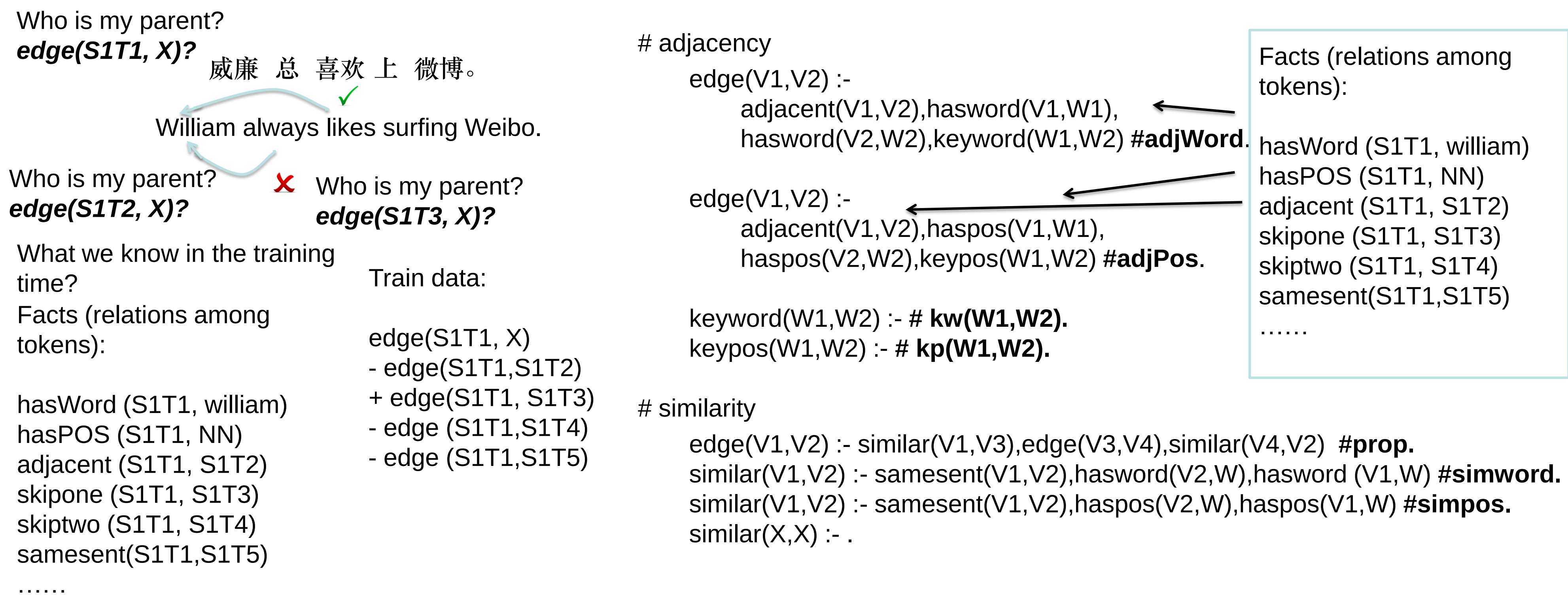
- **inference**: efficient approximate personalized pagerank with provable bound $\frac{1}{\alpha'\varepsilon}$ .

### non-linear edge strength functions

- Rectified Linear Unit (ReLU) (Nair and Hinton, 2010): $max(0, x)$;
- The Hyperbolic Function (Glorot and Bengio, 2010): $tanh(x)$.

( algorithm & results )

## 4 a dependency arc prediction algorithm



What we know in the training time?
Facts (relations among tokens):

hasWord (S1T1, william)
hasPOS (S1T1, NN)
adjacent (S1T1, S1T2)
skipone (S1T1, S1T3)
skiptwo (S1T1, S1T4)
samesent(S1T1,S1T5)
……

Train data:

edge(S1T1, X)
- edge(S1T1,S1T2)
+ edge(S1T1, S1T3)
- edge (S1T1,S1T4)
- edge (S1T1,S1T5)

```
# adjacency
    edge(V1,V2) :-
        adjacent(V1,V2),hasword(V1,W1),
        hasword(V2,W2),keyword(W1,W2) #adjWord.

    edge(V1,V2) :-
        adjacent(V1,V2),haspos(V1,W1),
        haspos(V2,W2),keypos(W1,W2) #adjPos.

    keyword(W1,W2) :- # kw(W1,W2).
    keypos(W1,W2) :- # kp(W1,W2).

# similarity
    edge(V1,V2) :- similar(V1,V3),edge(V3,V4),similar(V4,V2)  #prop.
    similar(V1,V2) :- samesent(V1,V2),hasword(V2,W),hasword (V1,W) #simword.
    similar(V1,V2) :- samesent(V1,V2),haspos(V2,W),haspos(V1,W) #simpos.
    similar(X,X) :- .
```

Facts (relations among tokens):

hasWord (S1T1, william)
hasPOS (S1T1, NN)
adjacent (S1T1, S1T2)
skipone (S1T1, S1T3)
skiptwo (S1T1, S1T4)
samesent(S1T1,S1T5)
……

**Algorithm 1** A Dependency Arc Inference Algorithm for Parsing Weibo

```
Given:
(1) a sentence with tokens $T_i$, where $i$ is the index, and $L$ is the length;
(2) a database $D$ of token relations from the corpus;
(3) first-order logic inference rule set $R$.

for $i = 1 \to L$ tokens do
  $\mathbb{S} \leftarrow ConstructSearchSpace(T_i, R, D)$;
  $\vec{P}_i \leftarrow InferParentUsingProPPR(T_i, \mathbb{S})$;
end for

Greedy Global Inference
for $i = 1 \to L$ tokens do
  $Y_i = \arg\max \vec{P}_i$;
end for
```

## 5 a new chinese weibo dependency treebank

- **freely available at:** http://www.cs.cmu.edu/~yww/data/WeiboTreebank.zip
- **annotation method:** FUDG (Schneider et al., 2013) and GFL annotation tool (Mordowanec et al., 2014).
- **training set #tokens:** 14,774.
- **development set #tokens:** 1,846.
- **test set #tokens:** 1,857
- **two stage annotations:** first, we run the stanford chinese word segmenter and pos tagger on the weibo data, and two graduate students with strong linguistic background annotate the weibo posts. second, the annotators frequently discuss the tricky cases, and try to reach agreements. finally, a second pass over the data corrects the auto-segment errors, and proofread the annotation.
- **inter-annotator agreement:** 82.31%.
- **annotation style:** influenced by the stanford chinese dependencies.

## 6 experimental results

| Method | Dev. | Test |
|---|---|---|
| Stanford Parser (Xinhua) | 0.507 | 0.489 |
| Stanford Parser (Chinese) | 0.597 | 0.581 |
| MaltParser (Full) | **0.669** | 0.654 |
| Our methods — ProPPR | | |
| ReLU (Bi-POS) | 0.506 | 0.517 |
| ReLU (Bilexical) | 0.635 | 0.616 |
| ReLU (Full) | 0.668 | 0.666 |
| Truncated *tanh* (Bi-POS) | 0.601 | 0.594 |
| Truncated *tanh* (Bilexical) | 0.650 | 0.634 |
| Truncated *tanh* (Full) | 0.667 | **0.675*** |

## 7 conclusions

- **a new chinese weibo dependency treebank**: we provide a freely available chinese weibo dependency treebank.
- **programmable dependency parsing on weibo**: we show that it is easy to use language-specific parsing theory for weibo parsing in ProPPR.
- **promising results**: we show that with language and genre specific first-order theory, our performance is better than an off-the-shelf stanford parser and a state-of-the-art maltparser that is trained on the same in domain data.