Replicate, Walk, and Stop on Syntax: an Effective Neural Network Model for Aspect-Level Sentiment Classification

Yaowei Zheng[1,2], Richong Zhang[1,2*], Samuel Mensah[1,2], Yongyi Mao[3]

[1]SKLSDE, School of Computer Science and Engineering, Beihang University, China

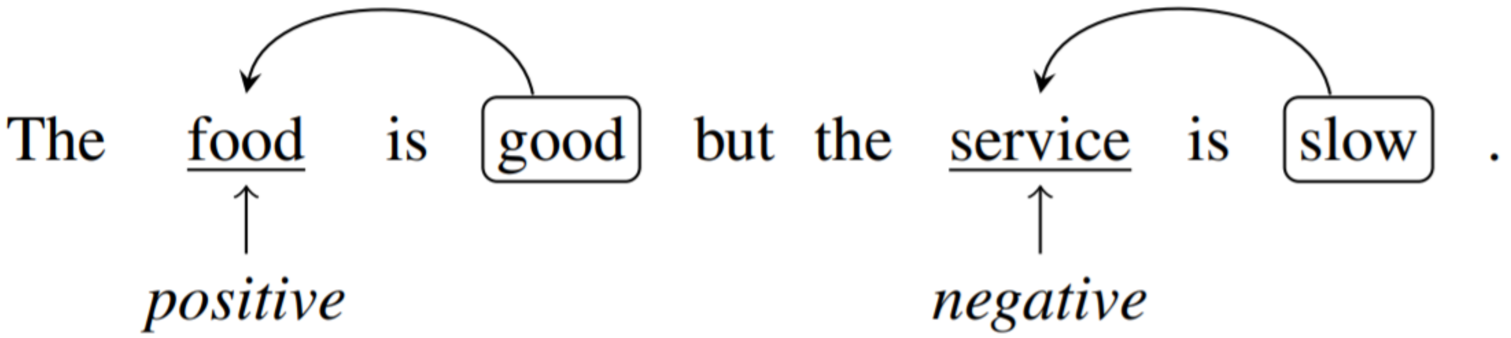
[2]Beijing Advanced Institution on Big Data and Brain Computing, Beihang University, China

[3]School of Electrical Engineering and Computer Science, University of Ottawa, Canada

December 12 2019, Qiu Lin Lecture Hall, Peking University

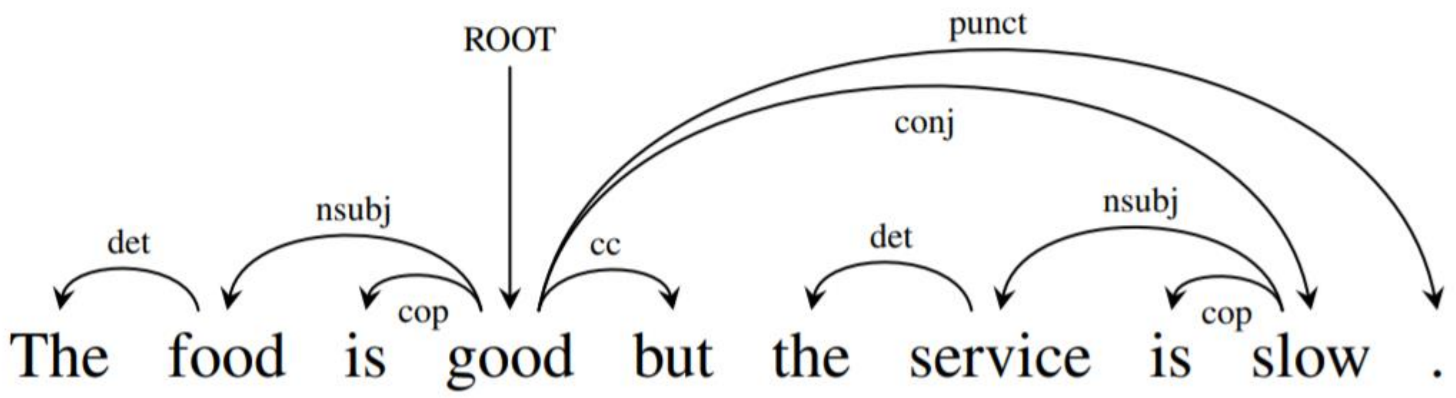
Task definition

Aspect-level sentiment classification (ALSC) aims at predicting the sentiment polarity of a sentence toward a specific aspect term. There may be multiple aspects in one sentence.



Innovation

- Leverage the syntactic information to enhance the power of neural network models.
- As explored in several works [1,2], but these methods cannot effectively incorporating the syntactic structure.



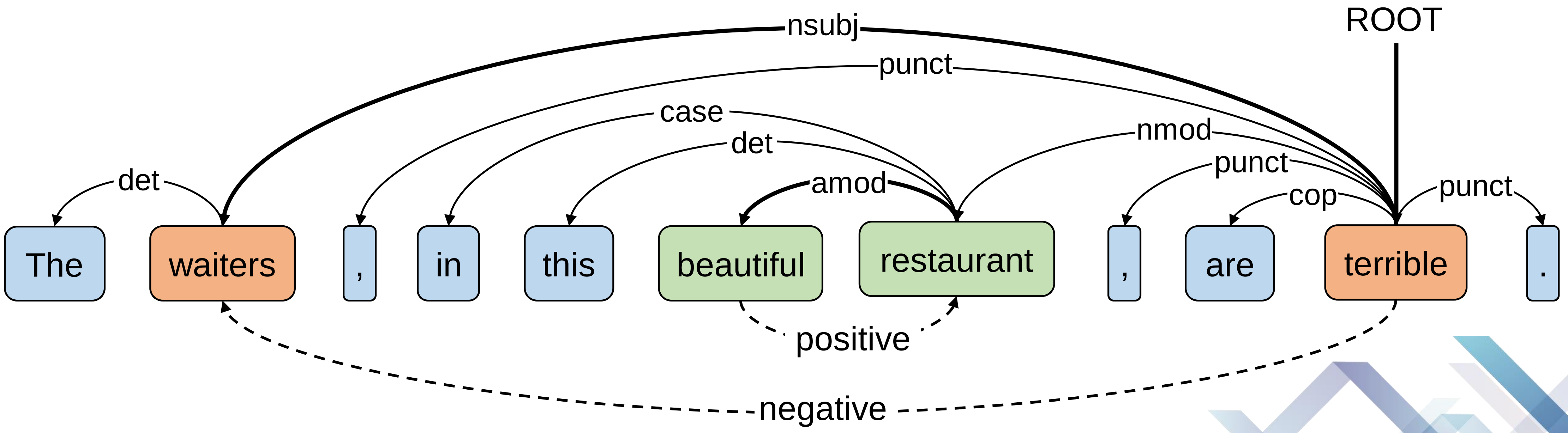
Dependency tree:

- Connected
- Acyclic
- Single-head

[1] Dong et al. Adaptive recursive neural network for target-dependent twitter sentiment classification. ACL'14.
[2] He et al. Effective attention modeling for aspect-level sentiment classification. COLING'18.

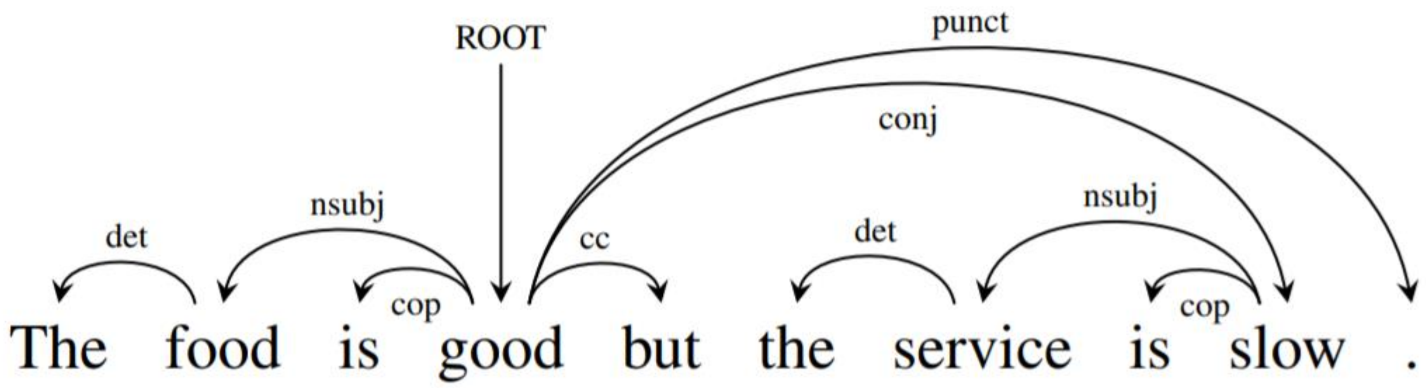
Observation

- The crucial words on the subtree of the dependency tree may contribute significantly to identify the aspect-level sentiment.
- Notably, traditional sequential models are limited to the distance between the aspect term and the opinion words.



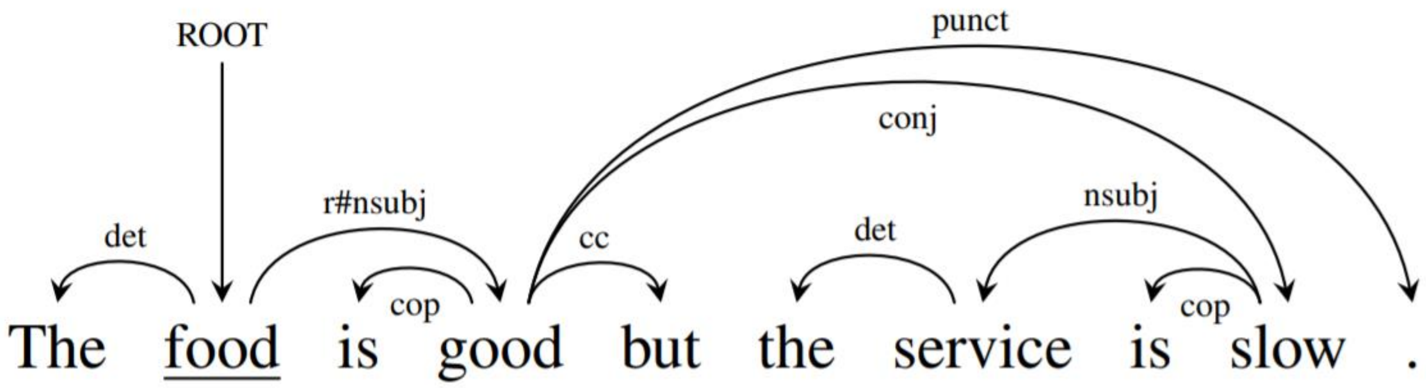
Problem

How can we effectively find the crucial words in the dependency tree?



Solution

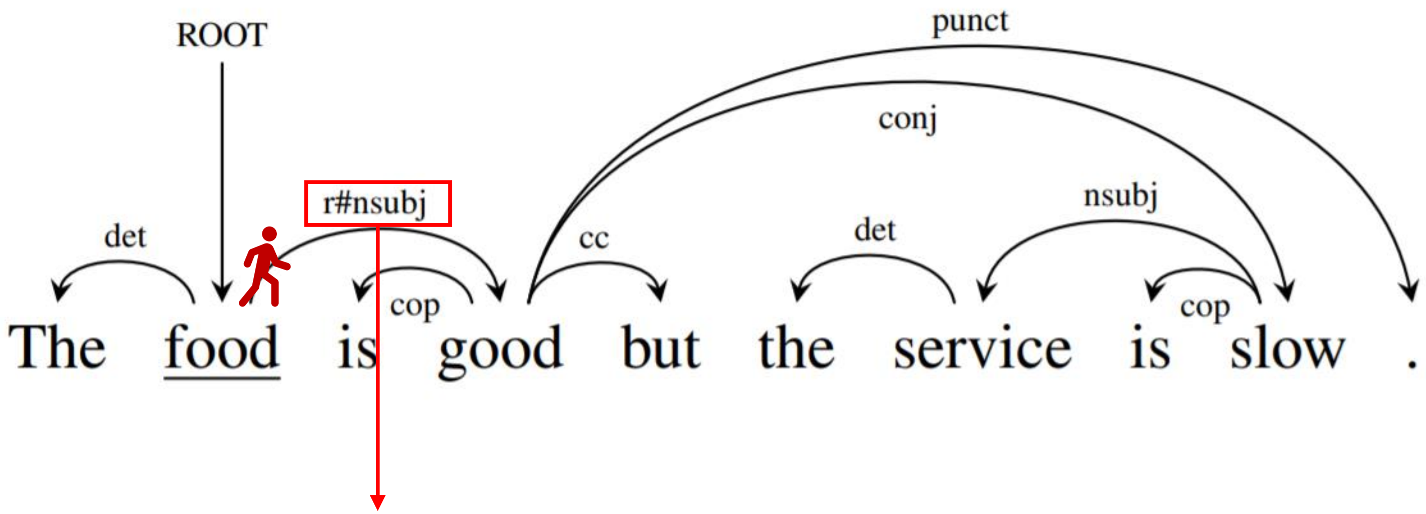
1. Let the aspect term be the starting point.



aspect-rooted dependency tree

Solution

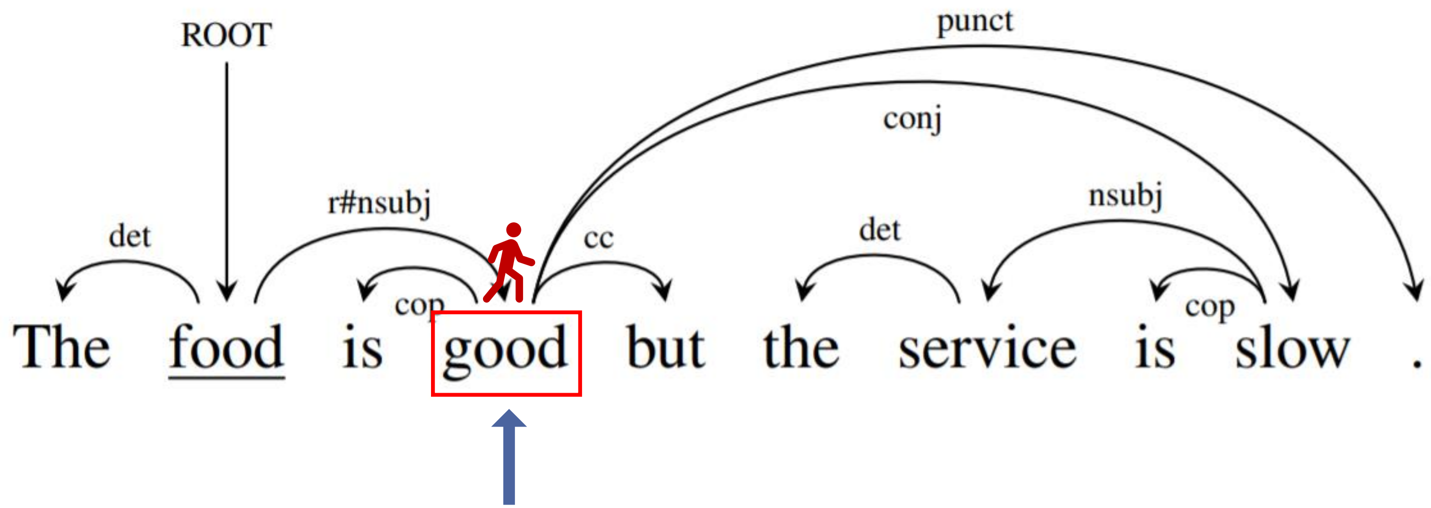
2. Find the crucial words along the edges of the tree.



food $\xrightarrow{r\#subj}$ good is probably connected to the opinion word

Solution

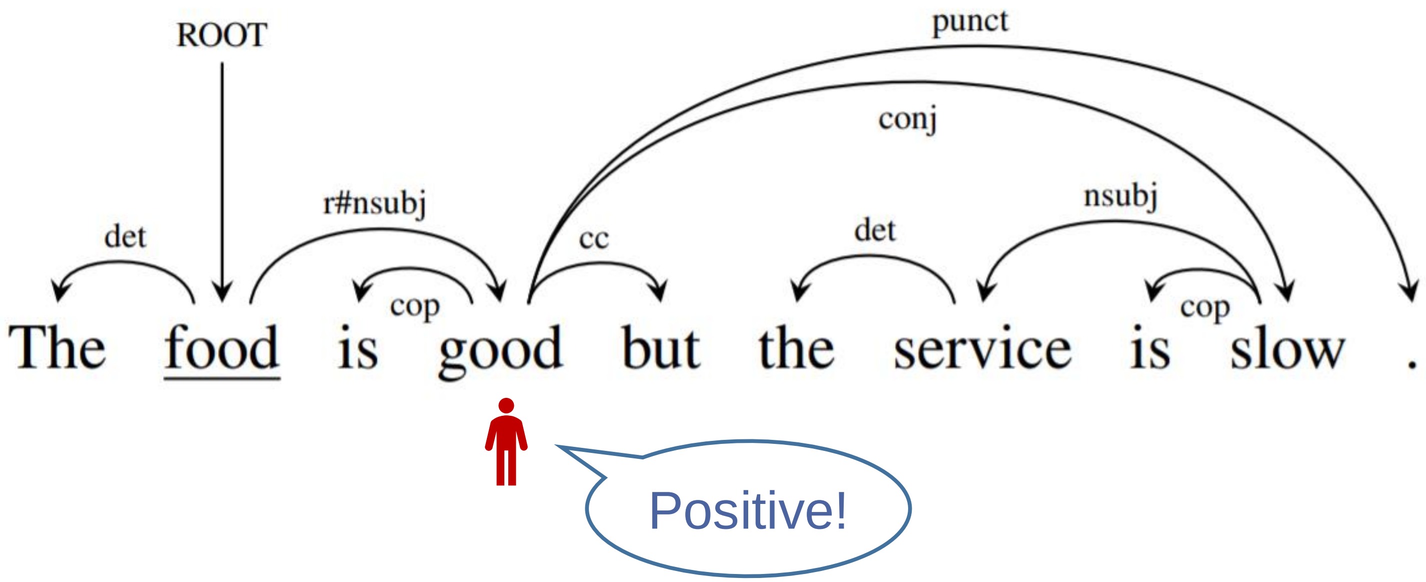
2. Find the crucial words along the edges of the tree.



“good” is probably a opinion word w.r.t. aspect term “food”

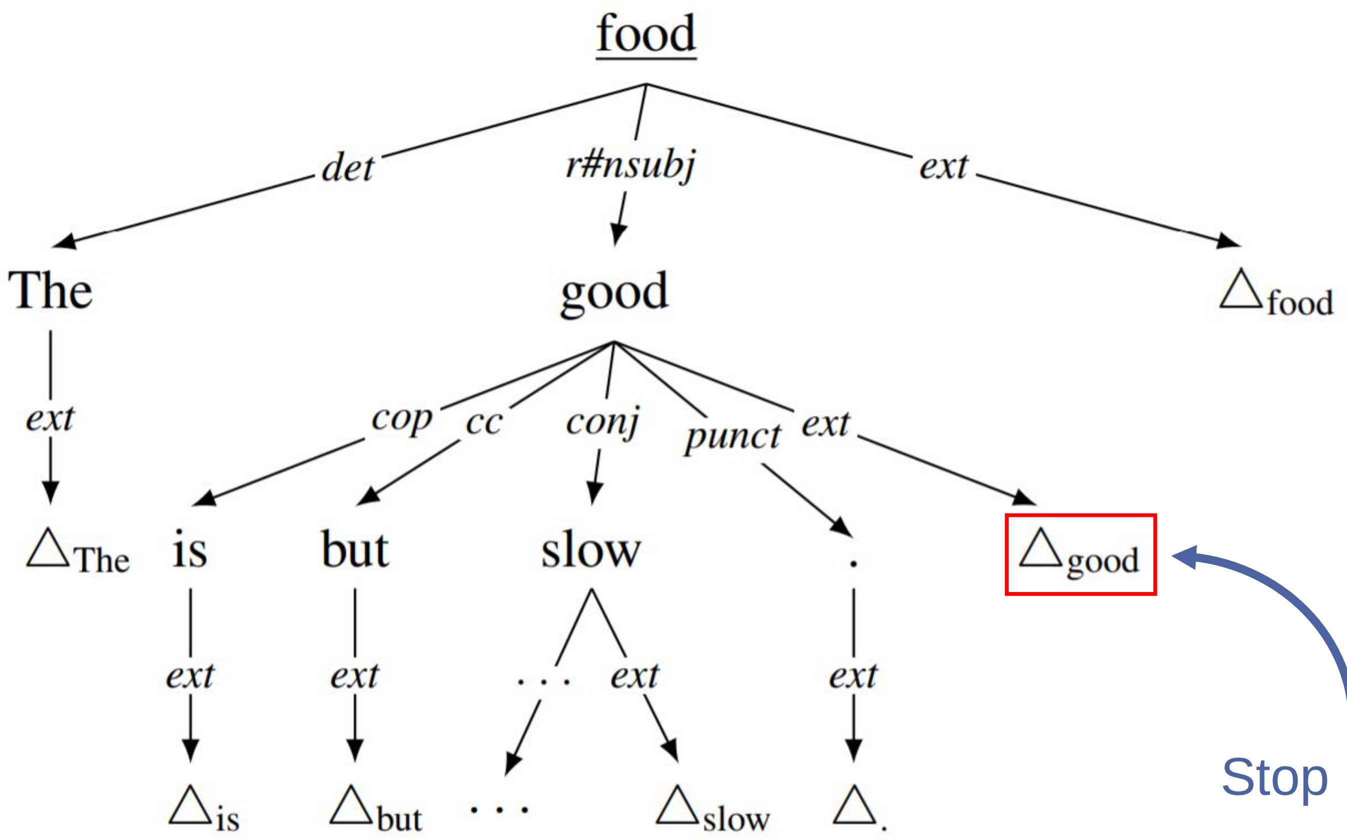
Solution

3. Stop at the opinion word.



Syntax Graph

Induce a **syntax graph** from the aspect-rooted dependency tree.



Stop node

Syntax graph

=

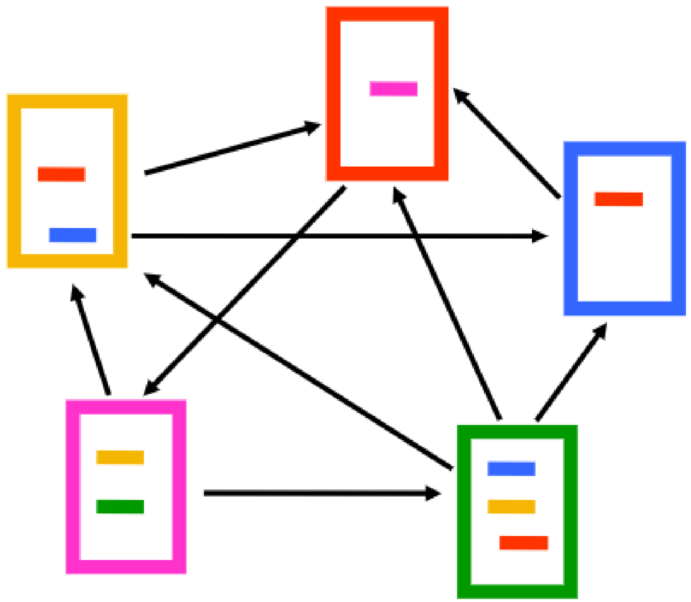
Aspect-rooted dependency tree

+

Stop nodes attaching to each word

Random Walk

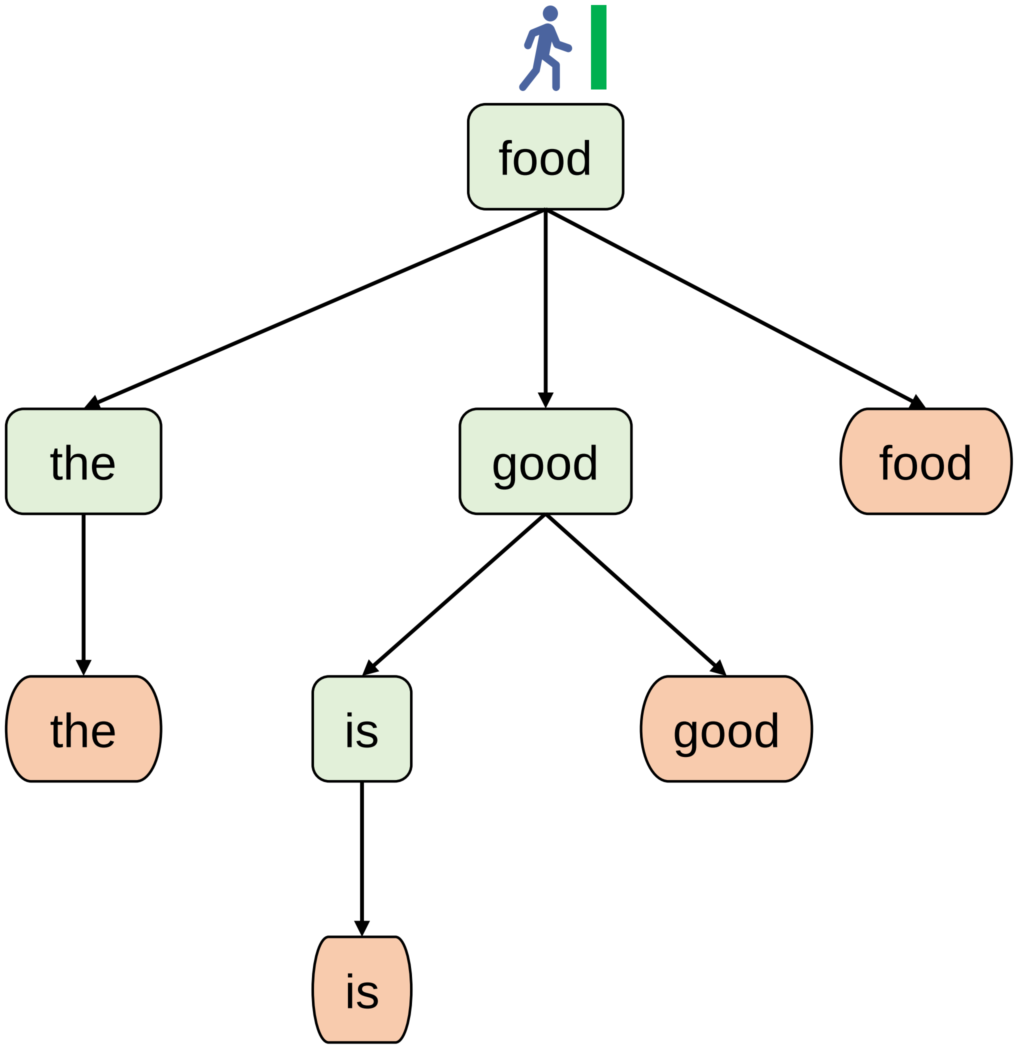
- Random walk learns representations for graphs or networks, as explored in [1][2].
- A famous application known as “PageRank”.



$$P = \begin{bmatrix} 0 & 1/2 & 1/2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 1/3 & 1/3 & 1/3 & 0 & 0 \\ 1/2 & 0 & 0 & 1/2 & 0 \end{bmatrix}$$

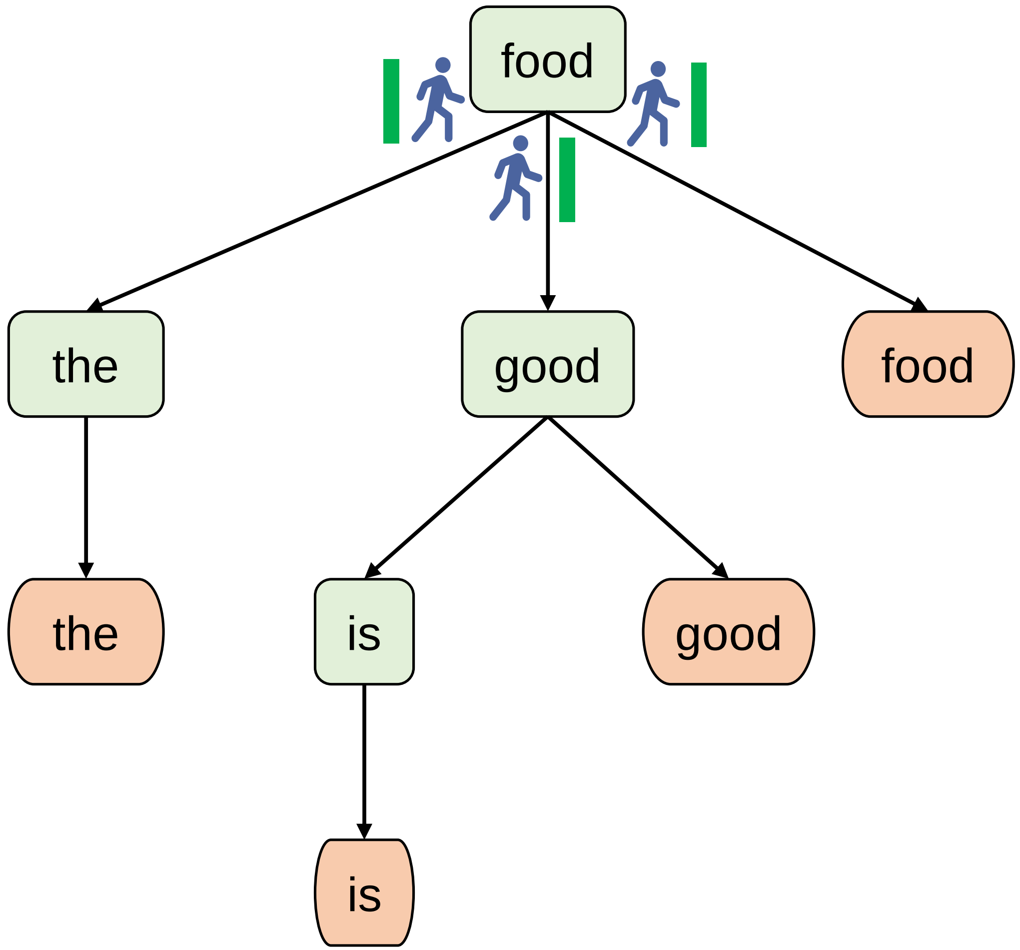
[1] Perozzi et al. Deepwalk: Online learning of social representations. SIGKDD’14.
[2] Li et al. Discriminative deep random walk for network classification. ACL’16.

Replicated Random Walk



0. Start with the first word of the aspect term.

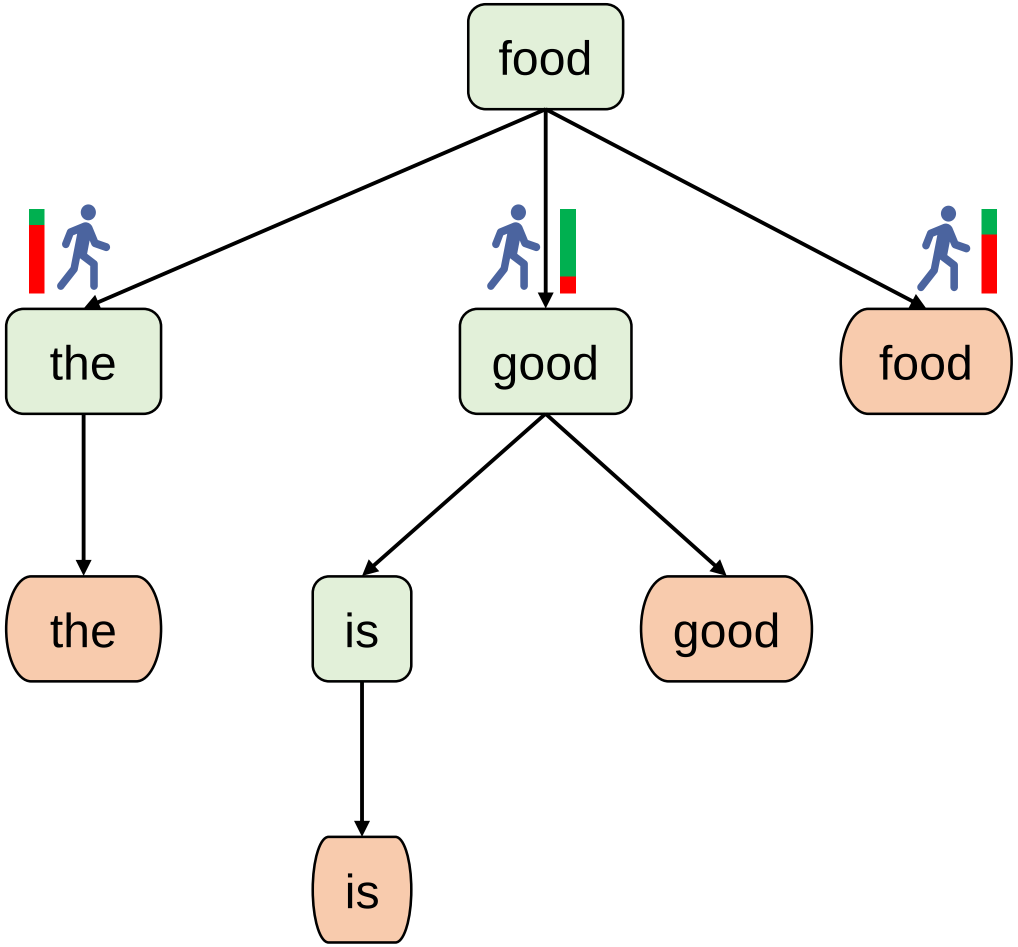
Replicated Random Walk



1. **Replicate** itself to d copies at each node.

Value d equals to the number of downstream edges.

Replicated Random Walk

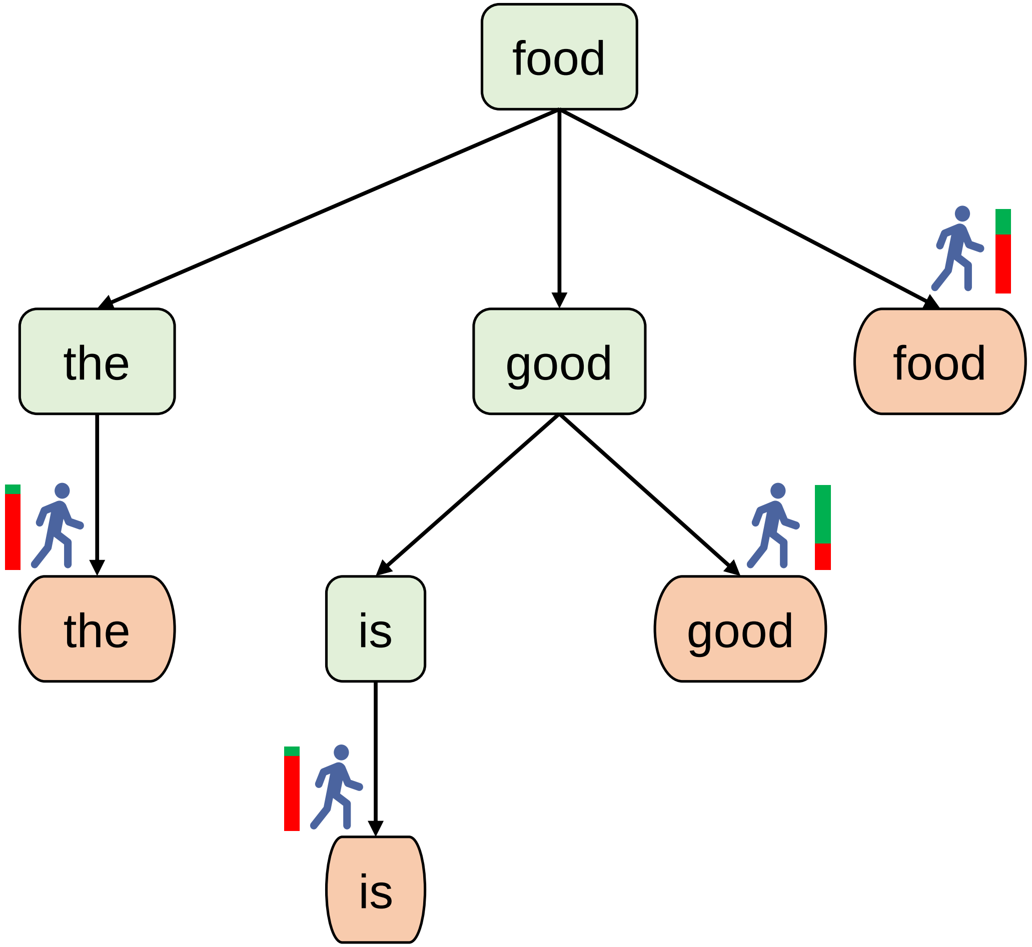


2. **Walk** along the downstream edges from the node.
Each replica arrives at next node if the edge is activated or dies.

Edge $e = u \xrightarrow{r} v$ is activated with a probability of p, where

$$p(e) = \sigma\left(\begin{bmatrix} u \\ v \end{bmatrix}^T W_p \theta_r + b_p\right)$$

Replicated Random Walk

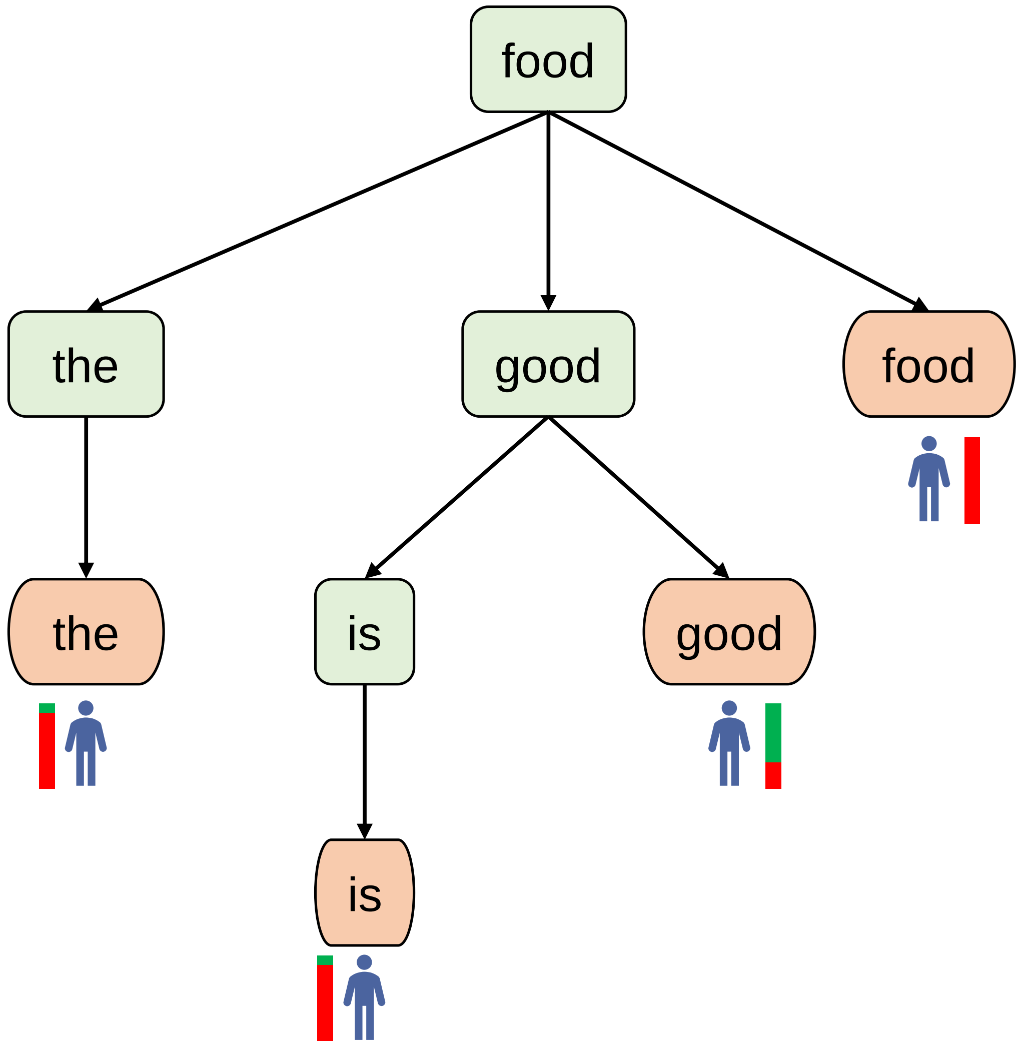


2. **Walk** along the downstream edges from the node.
Each replica arrives at next node if the edge is activated or dies.

Edge $e = u \xrightarrow{r} v$ is activated with a probability of p, where

$$p(e) = \sigma\left(\begin{bmatrix} u \\ v \end{bmatrix}^T W_p \theta_r + b_p\right)$$

Replicated Random Walk

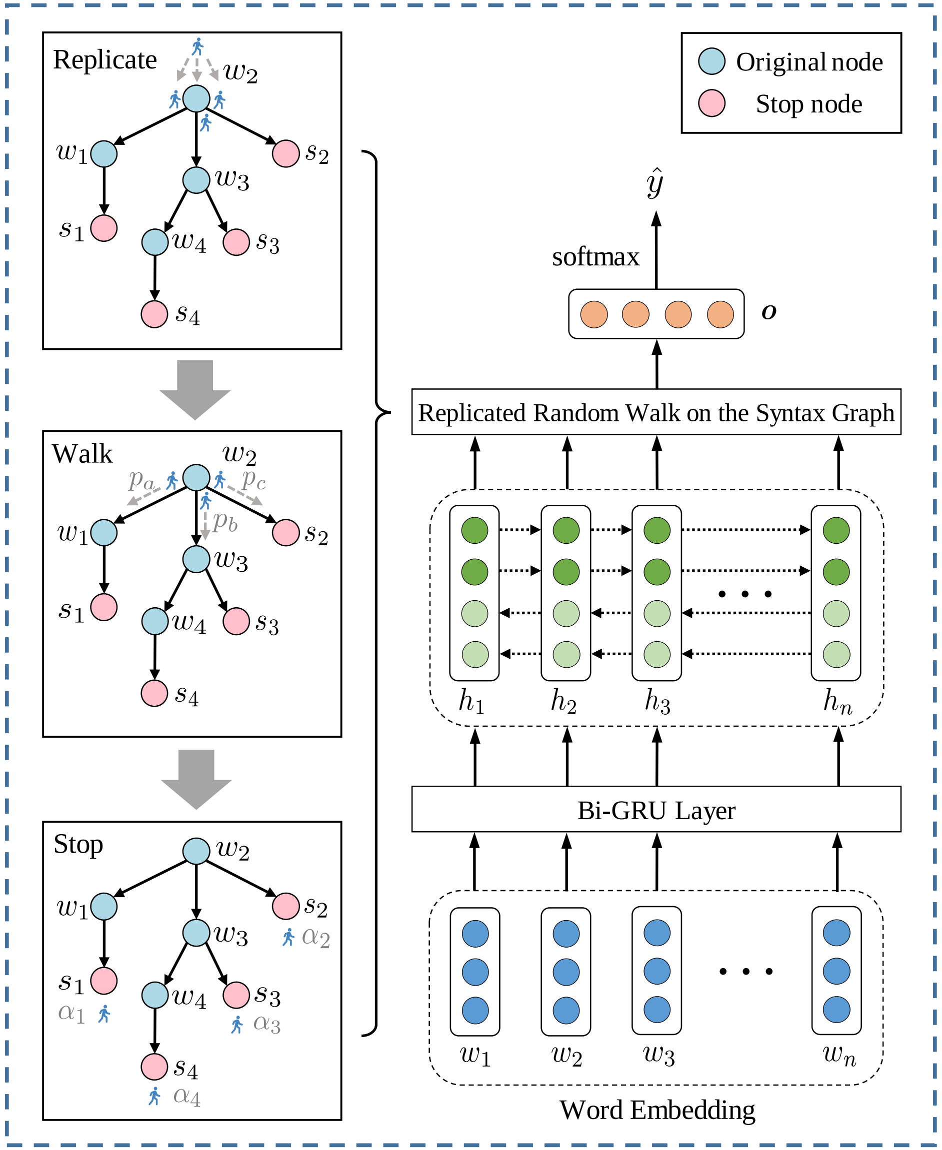


3. **Stop** at the stop nodes with a probability α or die halfway.

Arrive stop node s_i through the edges $\mathcal{E}_{s_i}$ with a probability of α_i, where

$$a_i = \begin{cases} 0, if\ w_i\ belongs\ the\ aspect \\ \prod_{e \in \mathcal{E}_{s_i}} p(e), otherwise \end{cases}$$

Model



1. Employ Bi-GRU networks to obtain contextual features.
2. Perform a replicated random work on the syntax graph, each replica arrive at the stop nodes with a probability of α_i.
3. Compute sentence representation:

$$o = \sum_{i=1}^{n} \alpha_i h_i$$

4. Predict the sentiment:

$$\hat{y} = softmax(W_o^T o + b_o)$$

Loss function

- Apply an L1 penalty on the weights of words to promote sparsity among the weights so that the model only selects a small number of words which really matter for the classification.

$$\mathcal{L}_w = \sum_{k=1}^{K} \left\| \alpha^k \right\|_1^2$$

- Use cross-entropy loss to optimize the model.

$$\mathcal{L}(\hat{y}, y) = -\sum_{i=1}^{K} \sum_{j=1}^{C} y_i^j \log\left(\hat{y}_i^j\right) + \beta \mathcal{L}_w + \lambda ||\Theta||_2^2$$

Settings

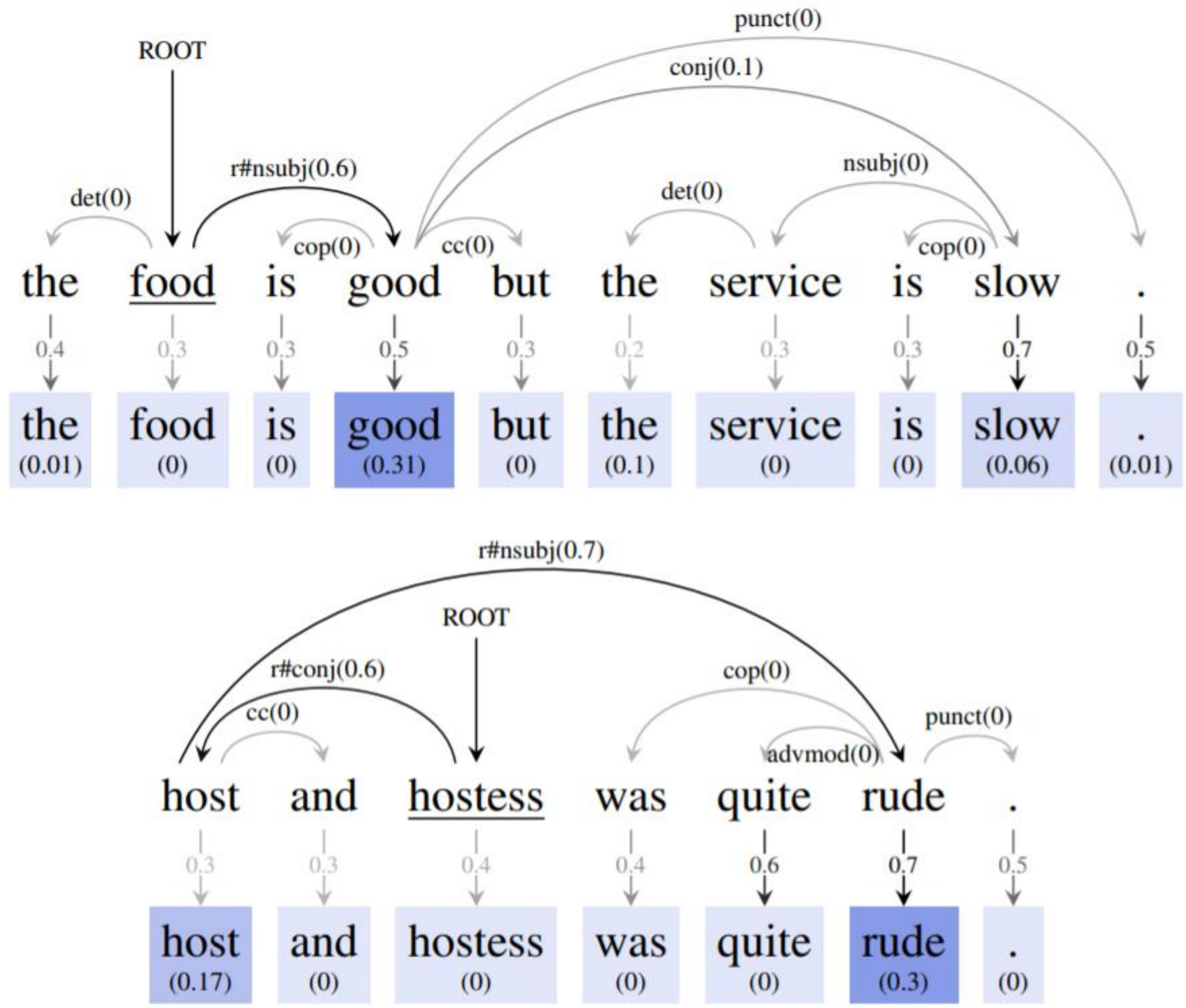
- Datasets
  - *Rest14*, *Laptop*: datasets from SemEval 2014 Task 4, containing the user reviews from *restaurant* and *laptop* domain respectively.
  - *Twitter*: a dataset built by (Dong et al., 2014), containing *twitter* posts and the opinion targets are annotated.
  - *Rest16*: a dataset from SemEval 2016 Task 5, containing the user reviews from *restaurant* domain which is similar to *Rest14*.
- Compared Methods
  - Rule-based methods: SVM-feature
  - Semantic-based methods: LSTM, ATAE-LSTM, MemNet, TNet, …
  - Syntactic-based methods: AdaRNN, LSTM+SynATT+TarRep, …
- Code available at *https://github.com/hiyouga/RepWalk*.

Results

Method	Rest14		Laptop		Twitter		Rest16	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1
SVM-feature (Kiritchenko et al. 2014)	80.2#	-	70.5#	-	63.4#	63.3#	-	-
AdaRNN (Dong et al. 2014)	-	-	-	-	66.3	65.9	-	-
LSTM (Tang et al. 2016)	74.3#	63.0#	66.5#	60.1#	66.5#	64.7#	81.9*	58.1*
TD-LSTM (Tang et al. 2016)	75.6#	64.5#	68.1#	63.9#	66.6#	64.0#	82.2*	54.2*
ATAE-LSTM (Wang et al. 2016b)	77.2	-	68.7	-	-	-	83.8*	61.7*
MemNet (Tang, Qin, and Liu 2016)	81.0	-	72.2	-	-	-	83.0*	57.9*
RAM (Chen et al. 2017)	80.2	70.8	74.5	71.4	69.4	67.3	83.9*	62.1*
IAN (Ma et al. 2017)	78.6	-	72.1	-	-	-	-	-
SA-LSTM-P (Wang and Lu 2018)	81.6	-	75.1	-	69.0	-	88.7	-
PRET+MULT (He et al. 2018b)	79.1	69.7	71.2	67.5	-	-	85.6	69.8
LSTM+SynATT+TarRep (He et al. 2018a)	80.6	71.3	71.9	69.2	-	-	84.6	67.5
MGAN (Fan, Feng, and Zhao 2018)	81.3	71.9	75.4	72.5	72.5	70.8	84.4♮	63.2♮
TNet (Li et al. 2018)	80.7	71.3	76.5	71.8	75.0	73.6	86.2♮	65.2♮
HSCN (Lei et al. 2019)	77.8	70.2	76.1	72.5	69.6	66.1	-	-
MGAN (Li et al. 2019)	81.5	71.5	76.2	71.4	74.6	73.5	-	-
PWCN (Zhang, Li, and Song 2019)	81.0	72.2	76.1	72.1	-	-	-	-
TransCap (Chen and Qian 2019)	79.3	70.9	73.9	70.1	-	-	-	-
TNet-ATT(+AS) (Tang et al. 2019)	81.5	72.9	77.6	73.8	**78.6**	**77.7**	-	-
RepWalk w/o pre-trained embedding	81.8	73.2	76.2	71.9	72.4	70.4	87.7	68.7
RepWalk w/o PoS tag embedding	81.7	73.0	75.4	71.7	72.5	70.7	87.8	66.8
RepWalk w/o dependency label	80.9	71.3	75.8	71.7	71.8	69.9	87.5	64.2
RepWalk w/o syntax graph	79.2	66.1	74.1	70.0	72.1	71.0	86.9	63.0
RepWalk w/o Bi-GRU	79.3	67.6	73.2	68.3	67.8	64.4	85.0	59.4
RepWalk	**83.8**	**76.9**	**78.2**	**74.3**	74.4	72.6	**89.6**	**71.2**

- Our model (RepWalk) achieves state-of-the-art results on three datasets.
- Ablation study shows the effectiveness of each component of our model.

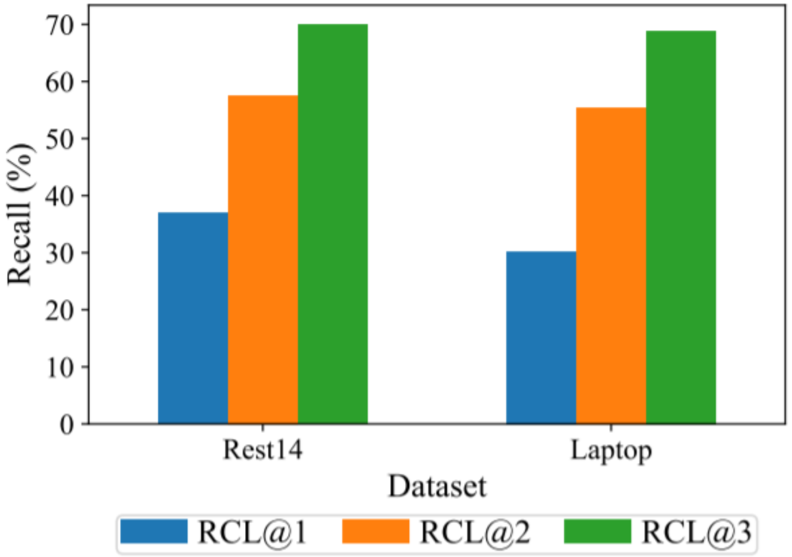
Case study



- We visualize the probabilities on the edges and the weights on the words.
- Our model successfully focuses on the crucial words in the subtree of the dependency tree by activating the edges of the syntax graph.

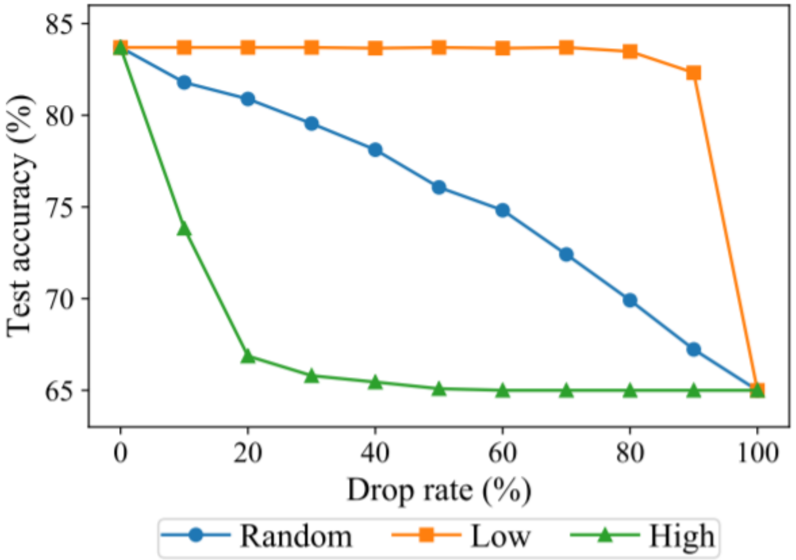
Analysis

"Recall"



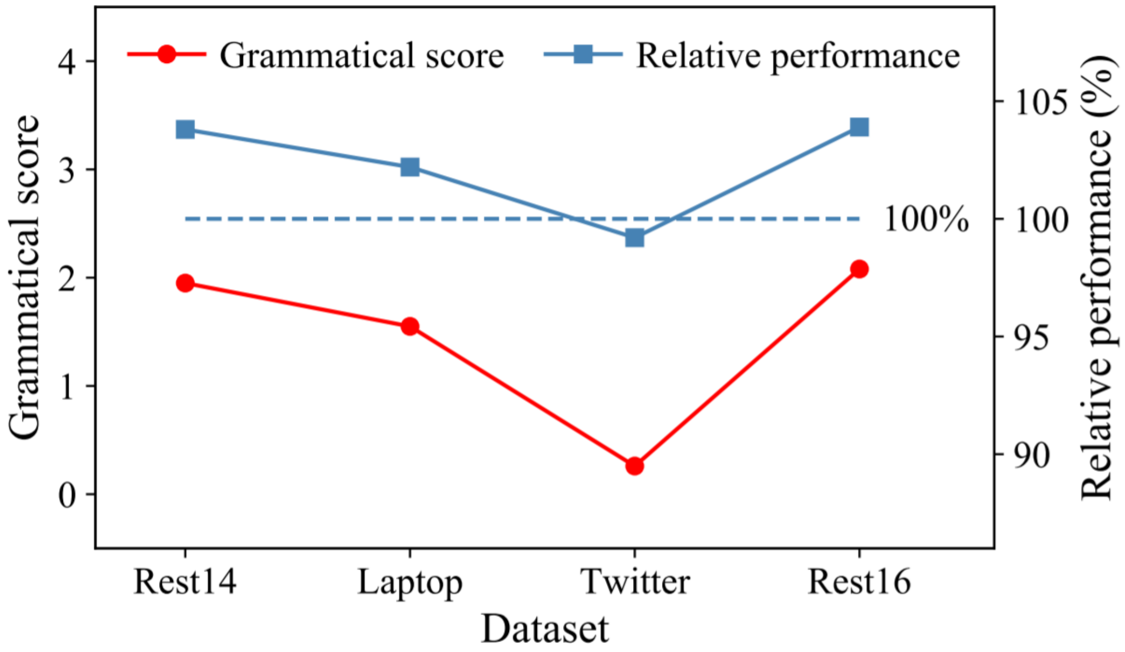
- Compare with the annotated opinion words.
- Among the set of opinion words, most of them are recalled by our model.

"Precision"



- Drop the features of words.
- Most of the words focused by our model are crucial for classification.

Analysis



- We use the grammar checker to judge the grammatical correctness of the datasets.
- The relative performance of our model compared with TNet shows the quality of sentences also affects to the effectiveness of our model, resulting in an unsatisfactory performance on the *Twitter* dataset.

Summary

- Our proposed model (RepWalk) can effectively leverage syntactic structures to improve sentence representations by performing a replicated random walk on the syntax graph induced from the dependency tree.
- The performance of the model is hinged on the ability to parse sentences into the correct dependency tree and hence limited to the grammatically correctness of the sentences.
- We can try BERT to see if this approach can further improve the performance over BERT.

Thank you for watching