EMNLP 2021

GNN for NLP papers

Content is fully borrowed from slides of original presentation

Document Graph for Neural Machine Translation

Mingzhou Xu[1], Liangyou Li[2], Derek F. Wong[1], Qun Liu[2],Lidia S. Chao[1],
[1]NLP2CT Lab, University of Macau
[2]Huawei Noah's Ark Lab nlp2ct.mzxu@gmail.com, {derekfw,lidiasc}@um.edu.com {liliangyou,qun.liu}@huawei.com

澳門大學
UNIVERSIDADE DE MACAU
UNIVERSITY OF MACAU

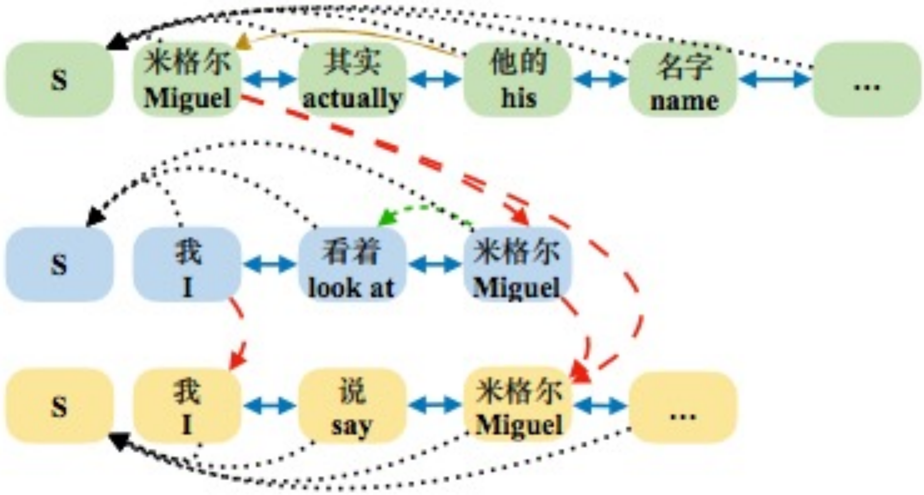
HUAWEI
Noah's Ark Lab

Background

- Problems:
  - How to use Long-distance contexts?
  - Not all the words in a document are beneficial to context integration(Kim et al. 2019)
- Motivation
  - It is essential for each word to focus on its own relevant context.
  - A graph allows each word to connect to those words which have a direct influence on its translation.

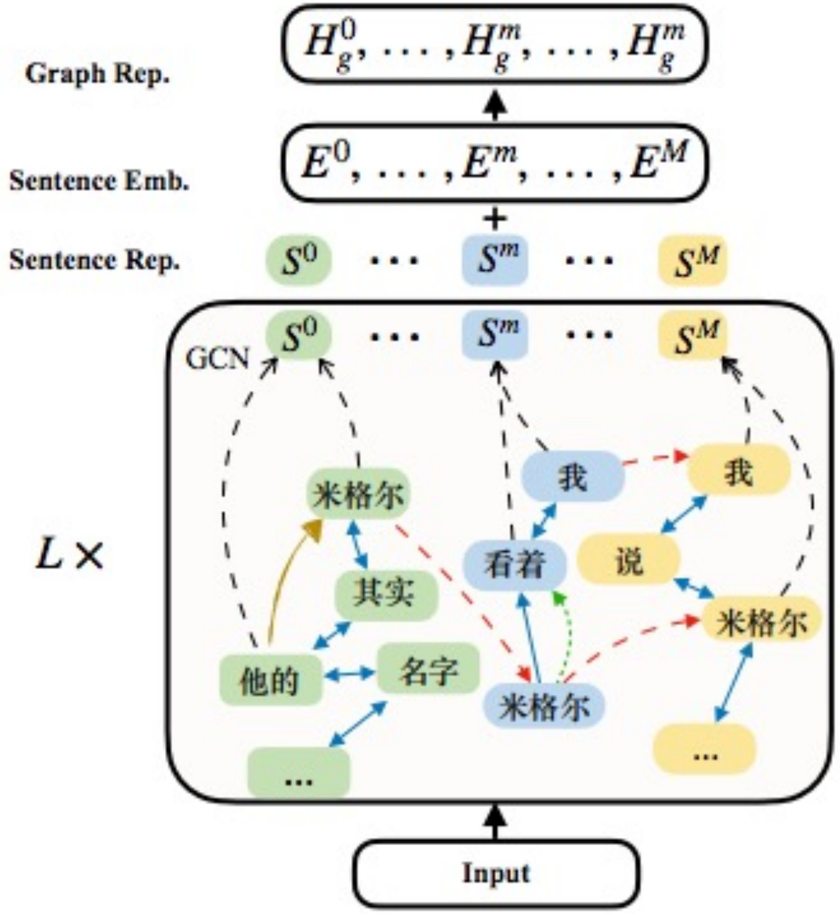
Graph Construction

- Sentence-Level Nodes
  - Fully connection
- Word-Level Nodes
  - Intra-sentential Relation:
    - Adjacency: provides a local lexicalized context.
    - Dependency: directly models syntactic and semantic relations between two words.
  - Inter-sentential Relation:
    - Lexical consistency: considers repeated and similar words across sentences in the document.
    - Coreference: helps understand the logic and structure of the document and resolve the ambiguities.

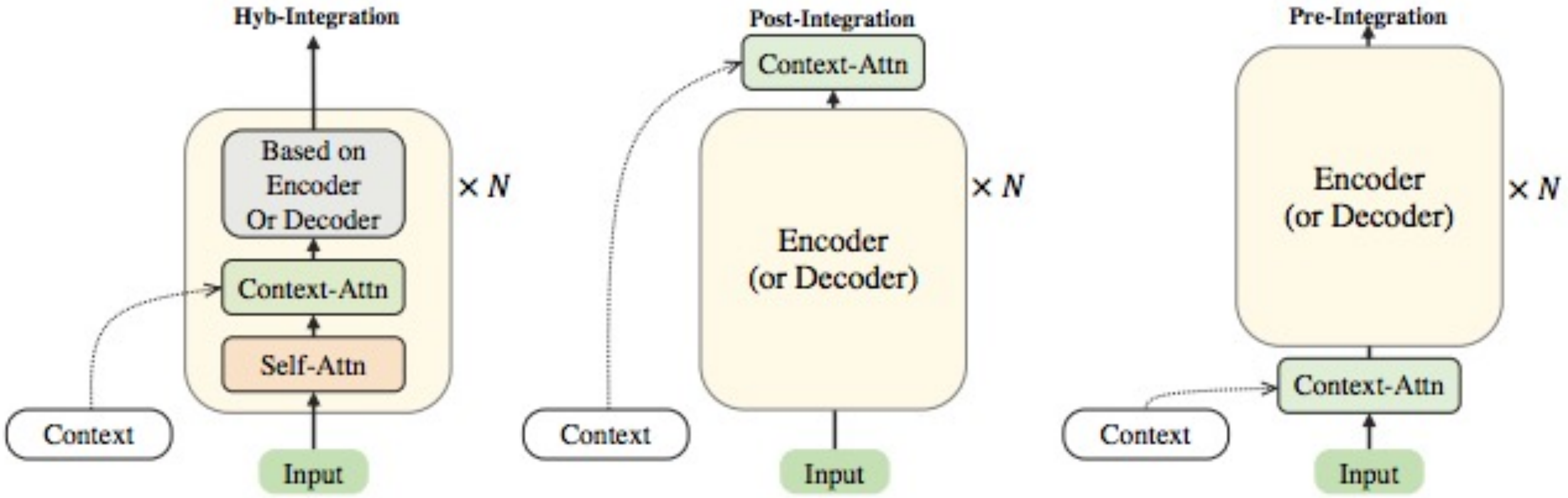


Doc-Graph Encoder

- **Graph Encoder:**
  - $H^{l+1} = \sigma\left(D^{-\frac{1}{2}}AD^{-\frac{1}{2}}\left(W^{l+1}H^{l} + B^{l+1}\right)\right)$
  - Where A, D is the adjacency- and degree-matrix
- **Directional GCN:**
  - To fully use direction information in the graph
  - $\hat{H}_t^{l+1} = \sigma\left(\hat{D}_t^{-\frac{1}{2}}\hat{A}_t\hat{D}_t^{-\frac{1}{2}}\left(\hat{W}_t^{l+1}H^{l} + B_t^{l+1}\right)\right)$
- **Type-Attention:**
  - $H^{l+1} = \sum_t \alpha_t \hat{H}_t^{l+1}$
  - $\alpha_t = \text{Softmax}\left(\frac{H^{l}\hat{H}_t^{l+1}}{\sqrt{d}}\right)$
  - where the α_t are attention weights given by a multi-head attention algorithm (Vaswani et al., 2017).



Integration of Context



- **Hyb-integration**: integrates the contextual information with an additional Context-Attn layer inside each encoder layer (Zhang et al., 2018).

- **Post-integration**: aggregates the contextual information by adding a Context-Attn layer after the encoder (Tan et al., 2019; Miculicich et al., 2018; Maruf et al., 2019).

- **Pre-integration**: interpolates the context representation before the encoder, which can be considered as the hierarchical embedded (Ma et al., 2020).

Hierarchical Heterogeneous Graph Representation Learning for Short Text Classification

Yaqing Wang[1]

Joint work with Song Wang[1,2], Quanming Yao[3], Dejing Dou[1]

EMNLP 2021 Online & in the Dominican Republic
Conference on Empirical Methods in Natural Language Processing

1 Baidu 百度

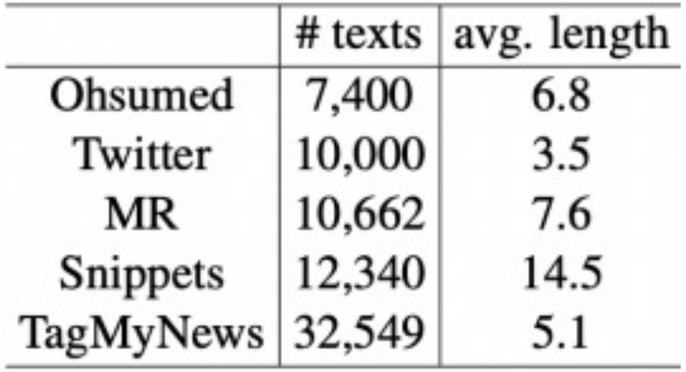
2 University of Virginia

3 清華大學 Tsinghua University

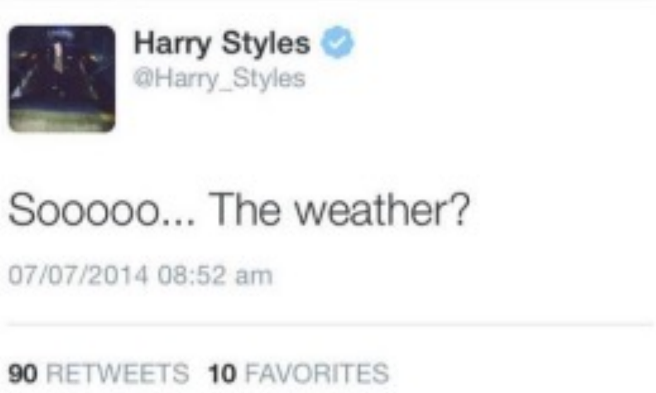
STC is Challenging

- Short texts only contain one or a few sentences whose overall length is small
  - Lack enough context information
  - May not obey strict syntactic structure

Hard to understand



	# texts	avg. length
Ohsumed	7,400	6.8
Twitter	10,000	3.5
MR	10,662	7.6
Snippets	12,340	14.5
TagMyNews	32,549	5.1



Harry Styles
@Harry_Styles

Sooooo... The weather?

07/07/2014 08:52 am

90 RETWEETS 10 FAVORITES

STC is Challenging

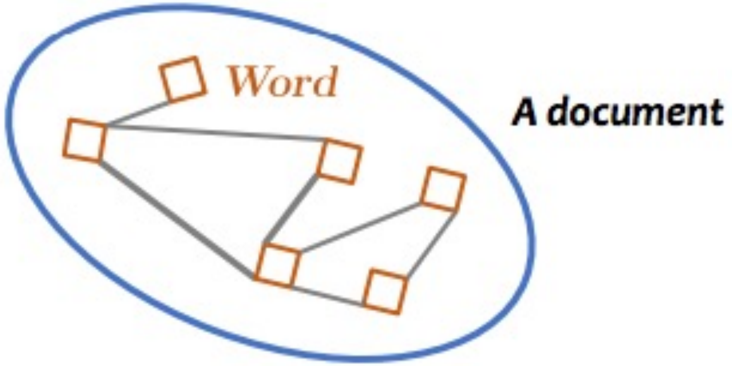
Requires auxiliary knowledge to help understand the short texts

- **Concepts** of common sense knowledge graphs
- **Latent topics** extracted from the corpus
- **Entities** residing in knowledge bases

In addition, real STC tasks usually only have a limited number of labeled data in comparison to the abundant unlabeled short texts emerging everyday

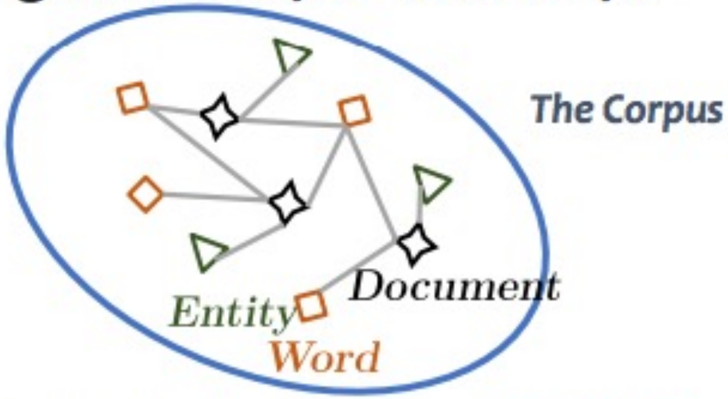
GNNs for Text Classification

Document-level Graph



- Model each document as a graph of word nodes
- Conduct graph classification
- Establish word-word edges differently
- Cannot work well when labeled graphs are scarce

Heterogenous Corpus-level Graph

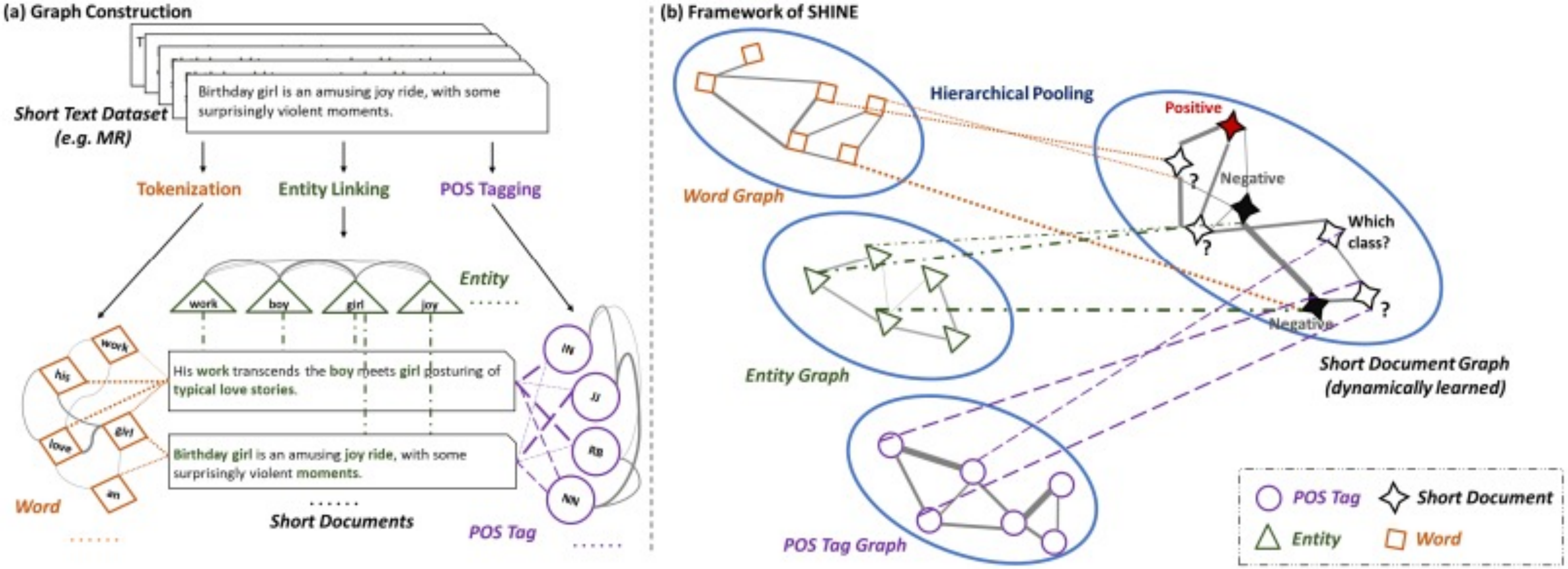


- Operate on a heterogeneous corpus-level graph with mixed nodes of different types
- Classify unlabeled texts by node classification
- Cannot fully exploit interactions between nodes of the same type

We Present SHINE

SHINE: a HIerarchical heterogeNEous graph representation learning method for STC

- Fully exploit interactions between nodes of the same types
- Capture similarity between short documents during learning



Birthday girl is an amusing joy ride, with some surprisingly violent moments.

His work transcends the boy meets girl posturing of typical love stories.

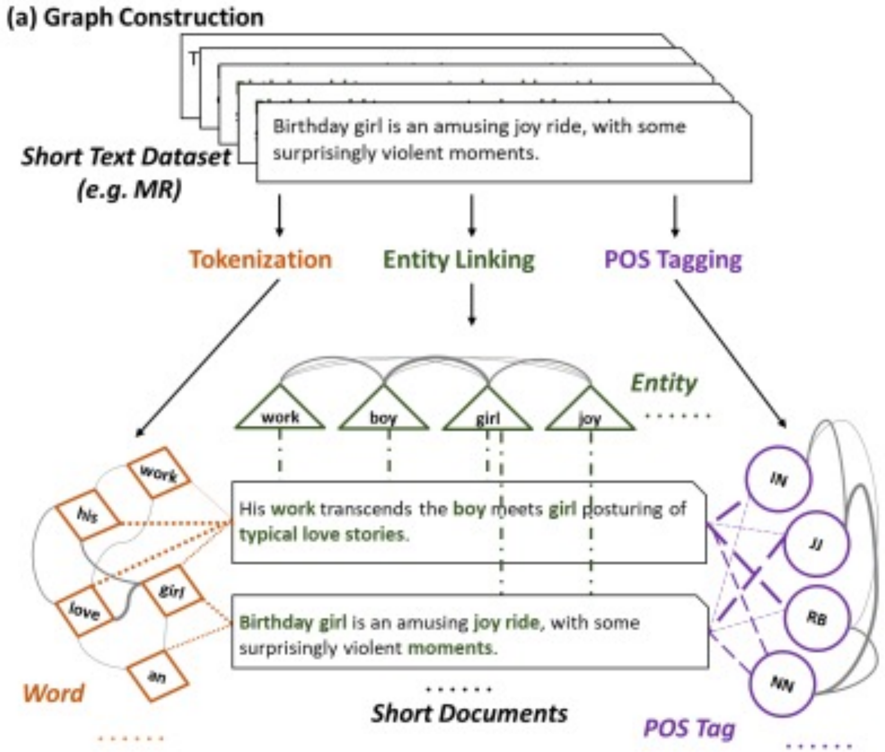
Word-Level Component Graphs

To bring in more syntactic and semantic information, we leverage various word-level components

- **word** (w) makes up short documents and carries semantic meaning
- **POS tag** (p) marks the syntactic role such as noun and verb of each word
- **entity** (e) corresponds to words that can be found in auxiliary knowledge bases

They are well-known, easy to obtain at a low cost

SHINE can be easily extended with other components such as topics and concepts


(a) Graph Construction
Birthday girl is an amusing joy ride, with some surprisingly violent moments.
Short Text Dataset (e.g. MR)
Tokenization
Entity Linking
POS Tagging
Entity
work
boy
girl
joy
......
work
his
love
girl
an
Word
......
His work transcends the boy meets girl posturing of typical love stories.
Birthday girl is an amusing joy ride, with some surprisingly violent moments.
......
Short Documents
IN
JJ
RB
NN
POS Tag
......

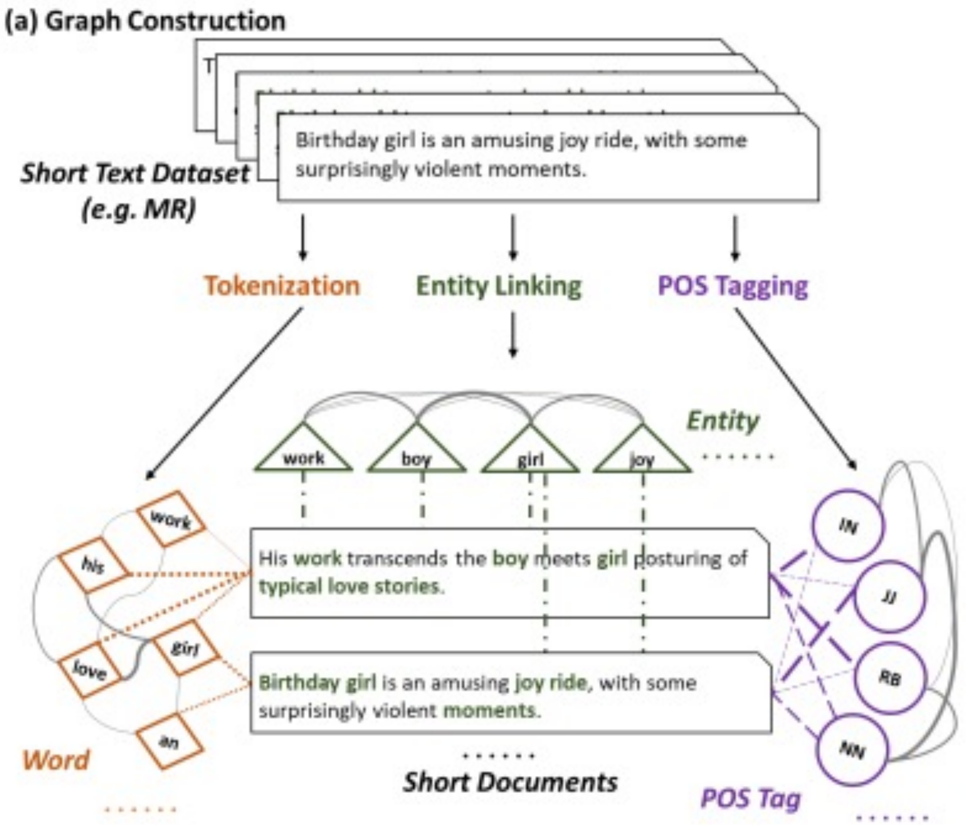
Word-Level Component Graphs

To bring in more syntactic and semantic information, we leverage various word-level components

- **word** (w) makes up short documents and carries semantic meaning
- **POS tag** (p) marks the syntactic role such as noun and verb of each word
- **entity** (e) corresponds to words that can be found in auxiliary knowledge bases

They are well-known, easy to obtain at a low cost

SHINE can be easily extended with other components such as topics and concepts



(a) Graph Construction
Birthday girl is an amusing joy ride, with some surprisingly violent moments.
Short Text Dataset
(e.g. MR)
Tokenization
Entity Linking
POS Tagging
Entity
work
boy
girl
joy
......
work
his
girl
love
an
Word
......
His work transcends the boy meets girl posturing of typical love stories.
Birthday girl is an amusing joy ride, with some surprisingly violent moments.
......
Short Documents
IN
JJ
RB
NN
POS Tag
......

Component Graph $G_\tau = \{V_\tau, A_\tau\}$ Construction

- $\tau = w \ or \ p$

A_τ models local co-occurrence statistics between components by point-wise mutual information (PMI)

x_τ is initialized as one-hot feature

- $\tau = e$

A_e models similarity between entities using entity embeddings pretrained from auxiliary knowledge bases

x_e is initialized as pretrained entity embeddings

$$\underset{\text{node embeddings}}{H_\tau} = \underset{\text{Normalized } A_\tau}{\widetilde{A_\tau}} \cdot ReLu(\widetilde{A_\tau}\underset{\text{node features}}{X_\tau} W_\tau^1)\ \underset{\text{trainable parameters}}{W_\tau^2}$$

Component Graph $G_\tau = \{V_\tau, A_\tau\}$ Construction

Short Document Graph $G_s = \{V_s, A_s\}$ Learning

We dynamically learn G_s based on embeddings pooled over word-level component graphs

Step 1: Obtain node features

$$\widehat{x_\tau^i} = u(H_\tau^T s_\tau^i)$$

➢ $\tau = w \; or \; p$

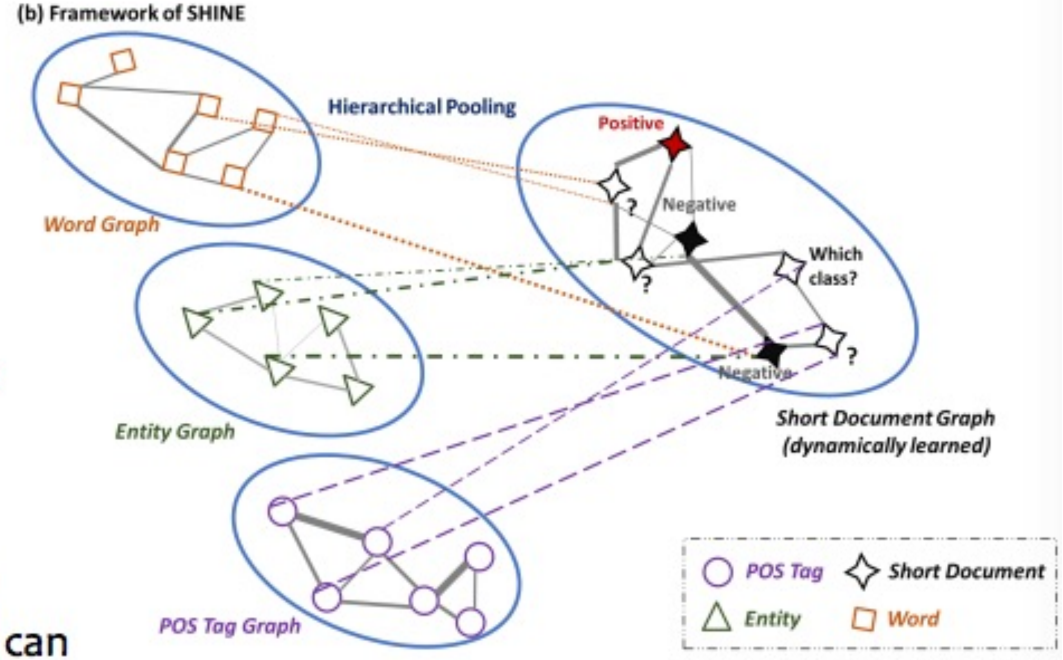
$$[s_\tau^i]_j = TF - IDF(v_\tau^j, v_s^i)$$

➢ $\tau = e$

$[s_e^i]_j = 1$ if v_e^j exists in v_s^i and 0 otherwise

$$x_s^i = \widehat{x_w^i} || \widehat{x_p^i} || \widehat{x_e^i}$$

- Explains each short document from the perspective of words, POS tags and entities
- Concatenation is just an instantiation, which can be replaced by more complex functions



Short Document Graph $G_s = \{V_S, A_S\}$ Learning

Step 2: Obtain adjacency matrix

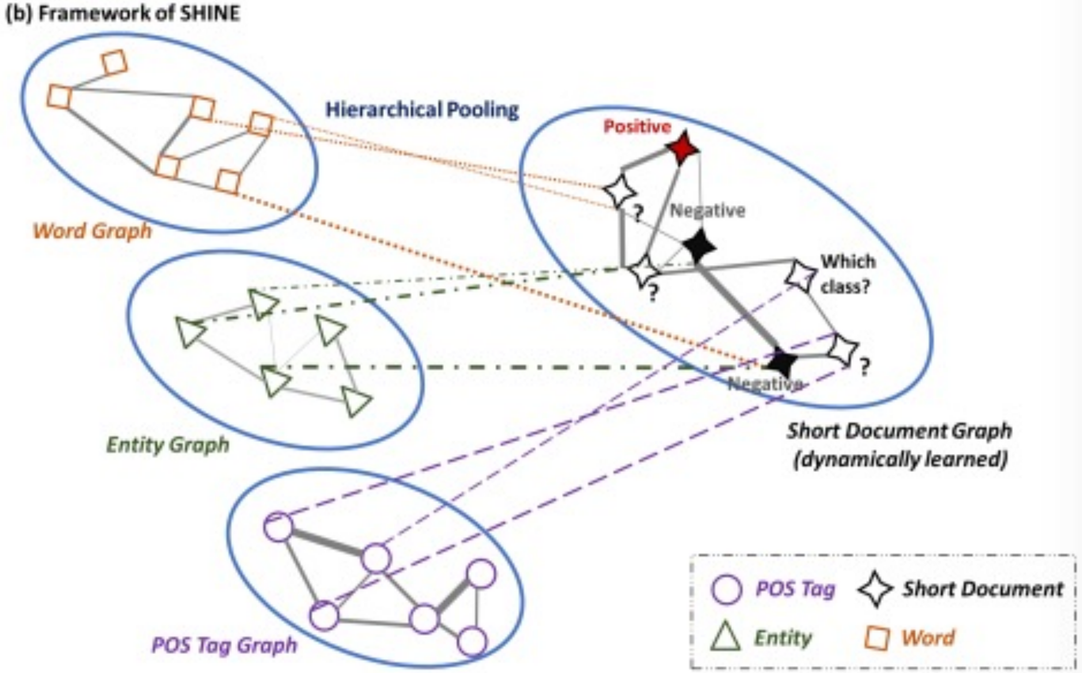
$$[A_S]_{ij} = \begin{cases} (x_S^i)^T x_S^j \; if \; (x_S^i)^T x_S^j > \delta_s \\ 0 \qquad otherwise \end{cases}$$

- Short documents are connected only if they are similar enough viewed from the perspective of $G_\tau s$
- G_s is dynamically changing along with the optimization process

Step 3: Obtain class predictions

$$\underbrace{\hat{Y}_S}_{\text{class prediction}} = softmax(A_s \cdot ReLu(A_s X_s W_s^1)\, W_s^2)$$

- softmax is applied for each row

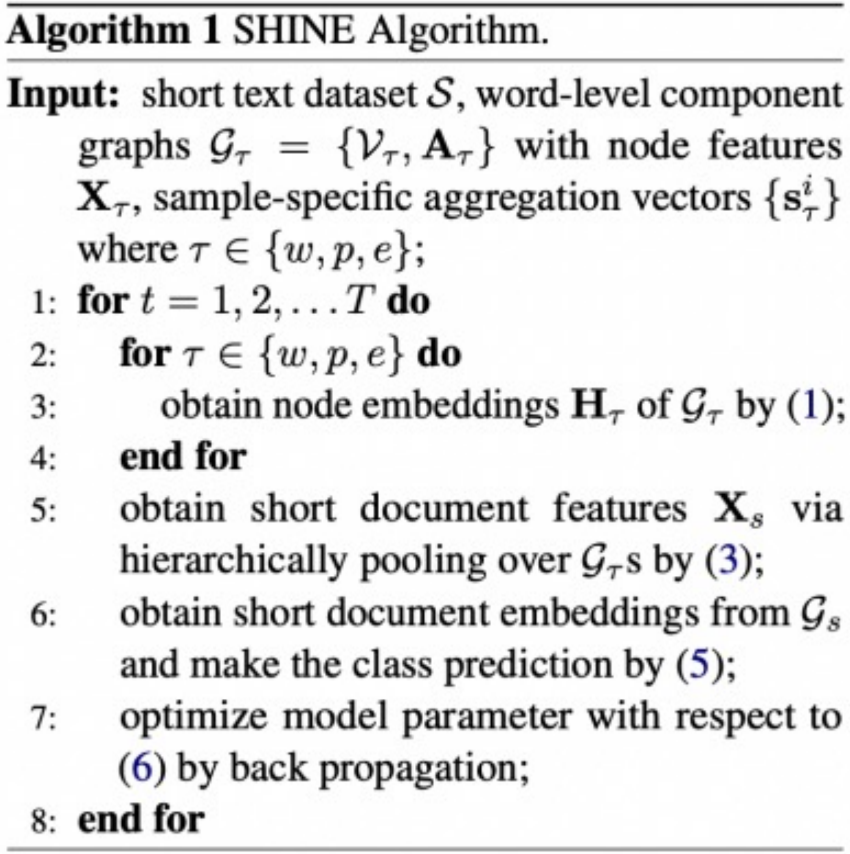


Optimization Algorithm

We train the complete model by optimizing the cross-entropy loss function in an end-to-end manner

$$L = -\sum_{\underbrace{i \in I_l}_{\text{Indices of labeled documents}}} \underbrace{(y_s^i)^T}_{\text{ground truth}} \log(\underbrace{\hat{y}_s^i}_{\text{class prediction}})$$

- Different types of graphs can influence each other
- During learning, node embeddings of $G_\tau s$ for all $\tau \epsilon \{$w,p,e,s$\}$ and A_s are all updated



Algorithm 1 SHINE Algorithm.

```
Input: short text dataset $\mathcal{S}$, word-level component graphs $\mathcal{G}_\tau = \{\mathcal{V}_\tau, \mathbf{A}_\tau\}$ with node features $\mathbf{X}_\tau$, sample-specific aggregation vectors $\{\mathbf{s}_\tau^i\}$ where $\tau \in \{w, p, e\}$;
for $t = 1, 2, \ldots T$ do
    for $\tau \in \{w, p, e\}$ do
        obtain node embeddings $\mathbf{H}_\tau$ of $\mathcal{G}_\tau$ by (1);
    end for
    obtain short document features $\mathbf{X}_s$ via hierarchically pooling over $\mathcal{G}_\tau$s by (3);
    obtain short document embeddings from $\mathcal{G}_s$ and make the class prediction by (5);
    optimize model parameter with respect to (6) by back propagation;
end for
```