Self-Adaptive Hierarchical Sentence Model

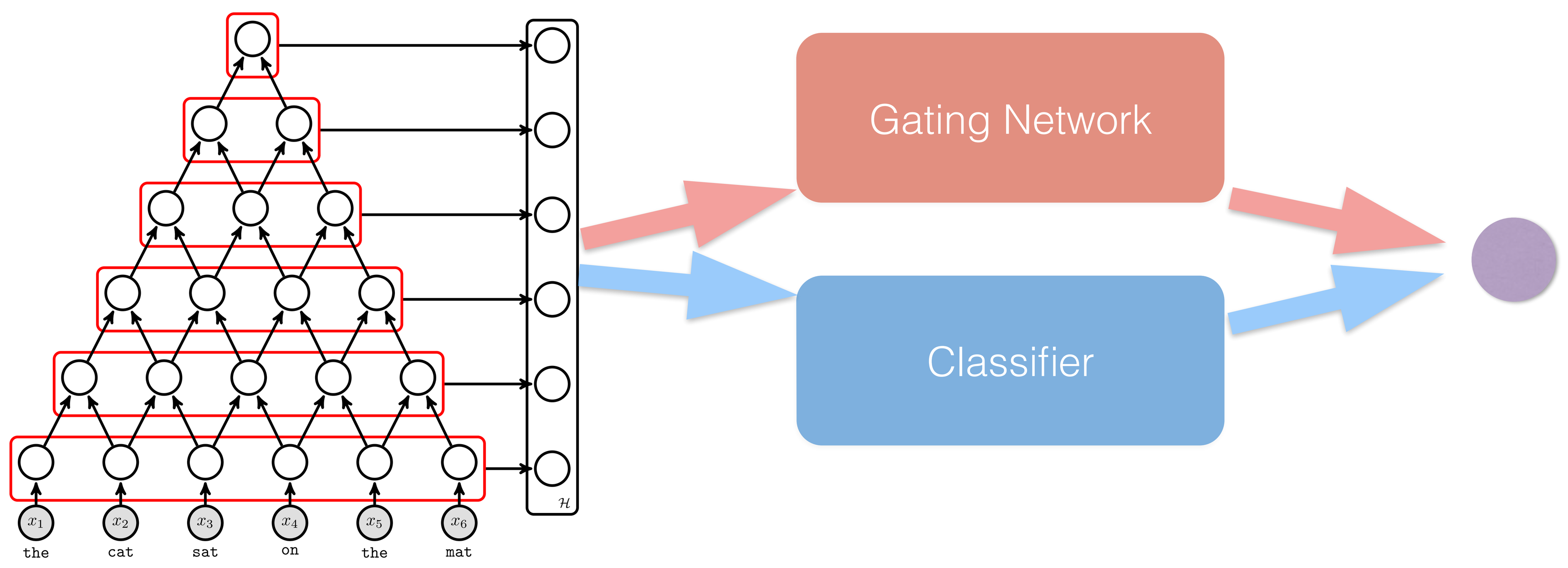
Han Zhao†, Zhengdong Lu‡ and Pascal Poupart†

†{han.zhao, ppoupart}@uwaterloo.ca, ‡lu.zhengdong@huawei.com

†University of Waterloo, ‡Noah's Ark Lab, Huawei Technologies

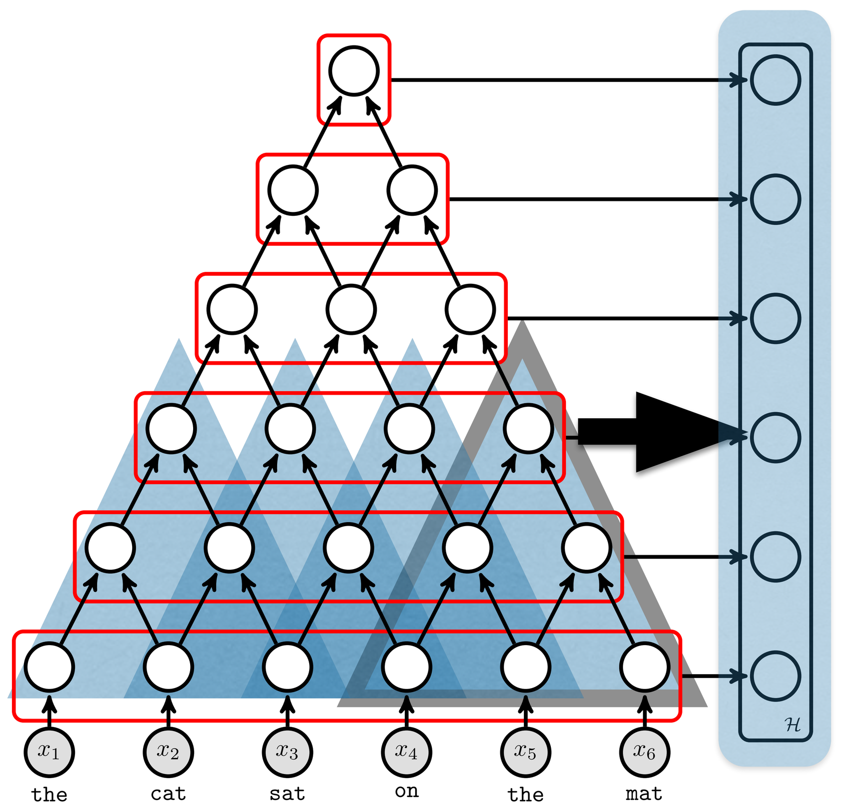
Introduction

- We propose a self-adaptive hierarchical sentence model (AdaSent) to represent phrases/short sentences in a hierarchy.
- We apply the mixture-of-experts framework to summarize representations of different granularities to make a final consensus.
- AdaSent is able to automatically learn the representation that is suitable for the task at hand through proper training.
- Empirical studies on 5 benchmark data sets show the superiority of AdaSent over previous approaches.



Architecture

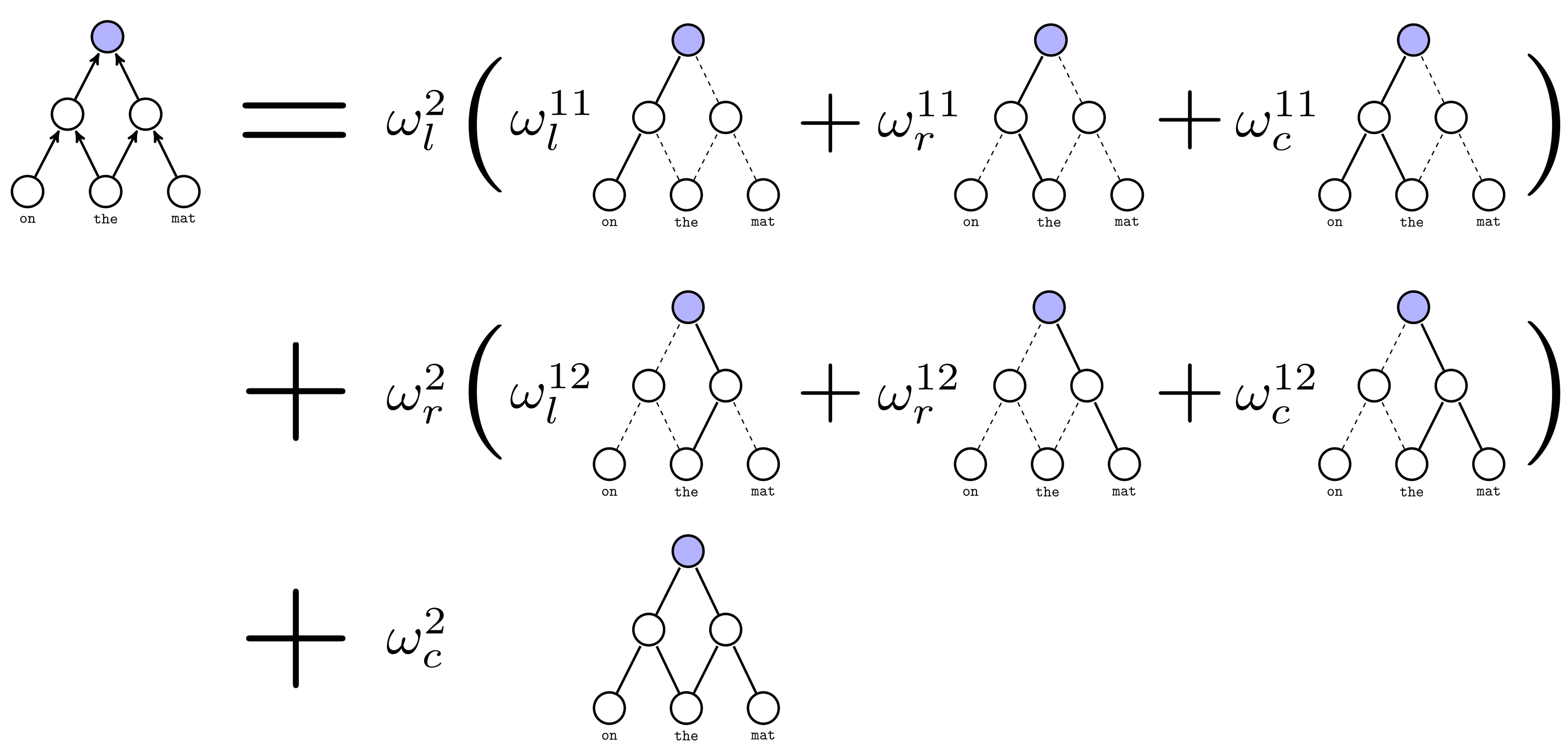
Structure:



- For an input sentence with length T, AdaSent builds a directed acyclic graph with T levels.
- Word embeddings are mapped from $\mathbb{R}^d \mapsto \mathbb{R}^D$ at the bottom, where $d \leq D$.
- Hidden units on t-th level contain intermediate representation for phrases of length t.
- Unit on the top is the global representation for the sentence.

Local Composition:

$$\begin{cases} h_j^t = \omega_l h_j^{t-1} + \omega_r h_{j+1}^{t-1} + \omega_c \tilde{h}_j^t \\ \tilde{h}_j^t = f(W_L h_j^{t-1} + W_R h_{j+1}^{t-1} + b_W) \end{cases}, \begin{pmatrix} \omega_l \\ \omega_r \\ \omega_c \end{pmatrix} = softmax(G_L h_j^{t-1} + G_R h_{j+1}^{t-1} + b_G)$$



Level Pooling:

Average Pooling:

$$\bar{h}^t = \frac{1}{T-t+1} \sum_{j=1}^{T-t+1} h_j^t$$

Max Pooling:

$$\bar{h}_i^t = \max_{j \in 1:T-t+1} h_{ji}^t, \forall i \in 1:D$$

Gating Network: A gating network takes $\bar{h}^t \in \mathbb{R}^D, t = 1:T$ as input and outputs a belief score $0 \leq \gamma_t \leq 1$ that depicts how confident the t-th level summarization in the hierarchy is suitable to be used as a proper representation of the current input instance for the task at hand. We require $\sum_{t=1}^T \gamma_t = 1$.

Decision Consensus:

$$p(C=c|\mathbf{x}_{1:T}) = \sum_{t=1}^{T} p(C=c|\mathcal{H}_\mathbf{x}=t) \cdot p(\mathcal{H}_\mathbf{x}=t|\mathbf{x}_{1:T}) = \sum_{t=1}^{T} g_c(\bar{h}^t) \cdot w(\bar{h}^t)$$

$g(\cdot)$ is the classification function and $\omega(\cdot)$ is the gating network.

Learning:

$$\text{minimize} \quad \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}(\mathbf{x}_i, y_i) + \lambda \left(||W_L||_F^2 + ||W_R||_F^2 \right)$$

where $\mathcal{L}(\cdot,\cdot)$ is the negative class conditional log-likelihood function. We use mini-batch AdaGrad to optimize the objective. Compute partial derivatives using back-propagation through structure:

$$\frac{\partial \mathcal{L}}{\partial W_L} = \sum_{t=1}^{T} \sum_{j=1}^{T-t+1} \frac{\partial \mathcal{L}}{\partial h_j^t} \frac{\partial h_j^t}{\partial W_L}, \frac{\partial \mathcal{L}}{\partial W_R} = \sum_{t=1}^{T} \sum_{j=1}^{T-t+1} \frac{\partial \mathcal{L}}{\partial h_j^t} \frac{\partial h_j^t}{\partial W_R}$$

where

$$\frac{\partial \mathcal{L}}{\partial h_j^t} = \frac{\partial \mathcal{L}}{\partial h_j^{t+1}} \frac{\partial h_j^{t+1}}{\partial h_j^t} + \frac{\partial \mathcal{L}}{\partial h_{j-1}^{t+1}} \frac{\partial h_{j-1}^{t+1}}{\partial h_j^t}$$

$$\frac{\partial h_{j-1}^{t+1}}{\partial h_j^t} = \omega_r I + \omega_c \cdot \text{diag}(f') W_R, \frac{\partial h_j^{t+1}}{\partial h_j^t} = \omega_l I + \omega_c \cdot \text{diag}(f') W_L$$

Experiments

Data Sets:

Data	MR	CR	SUBJ	MPQA	TREC
N	10662	3788	10000	10099	5952
$\lvert C \rvert$	2	2	2	2	6

Classification Accuracy:

Model	MR	CR	SUBJ	MPQA	TREC
NB-SVM	79.4	81.8	93.2	86.3	-
MNB	79.0	80.0	93.6	86.3	-
RAE	77.7	-	-	86.4	-
MV-RecNN	79.0	-	-	-	-
CNN	81.5	85.0	93.4	89.6	**93.6**
DCNN	-	-	-	-	93.0
P.V.	74.8	78.1	90.5	74.2	91.8
cBoW	77.2	79.9	91.3	86.4	87.3
RNN	77.2	82.3	93.7	90.1	90.2
BRNN	82.3	82.6	94.2	90.3	91.0
GrConv	76.3	81.3	89.5	84.5	88.4
AdaSent	**83.1**	**86.3**	**95.5**	**93.3**	92.4

Representations:

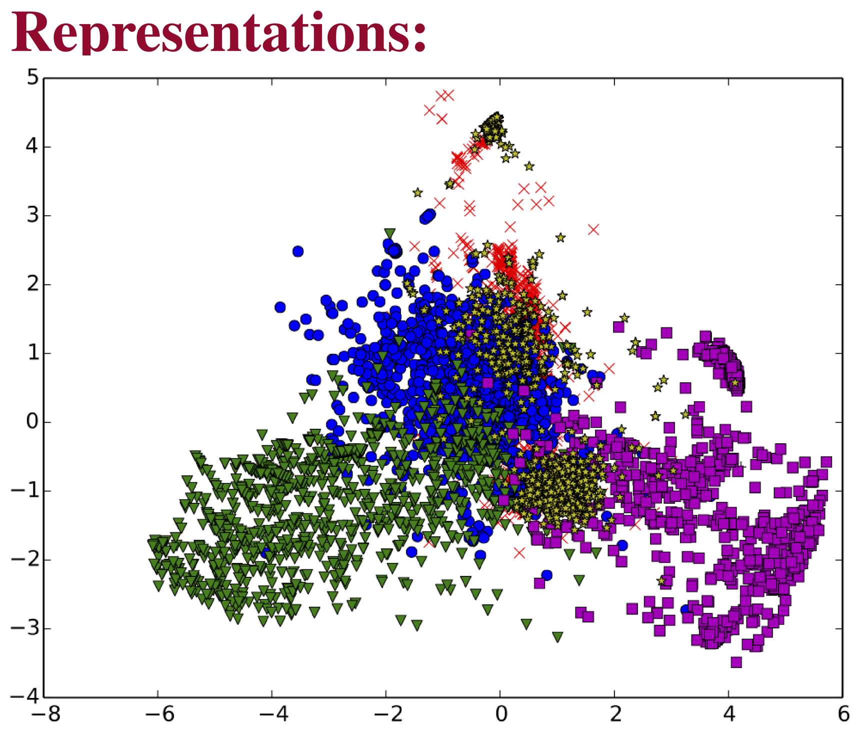


Figure 1: AdaSent on TREC

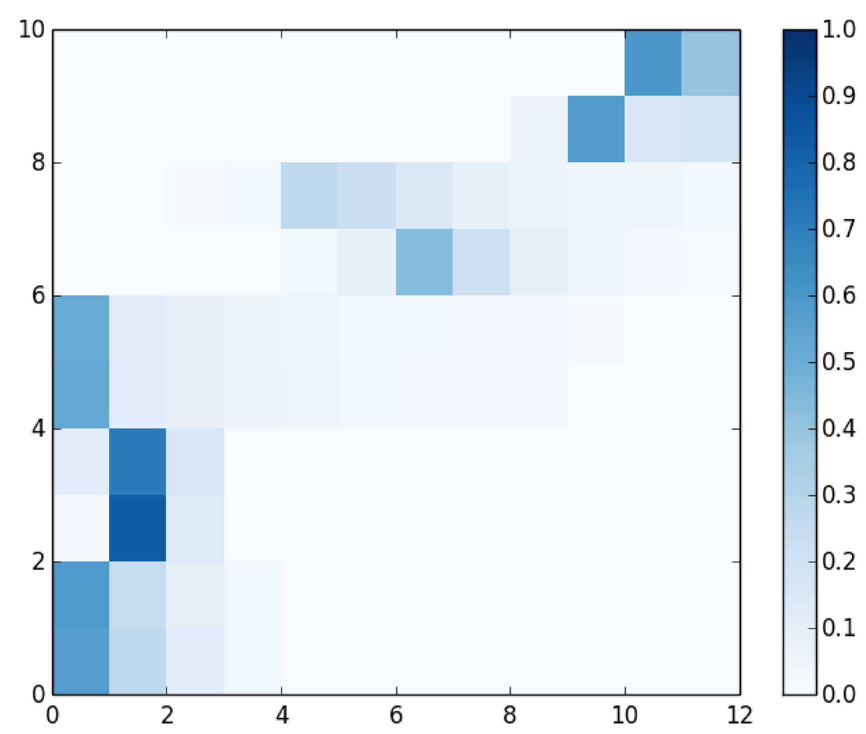


Figure 2: Belief score distribution

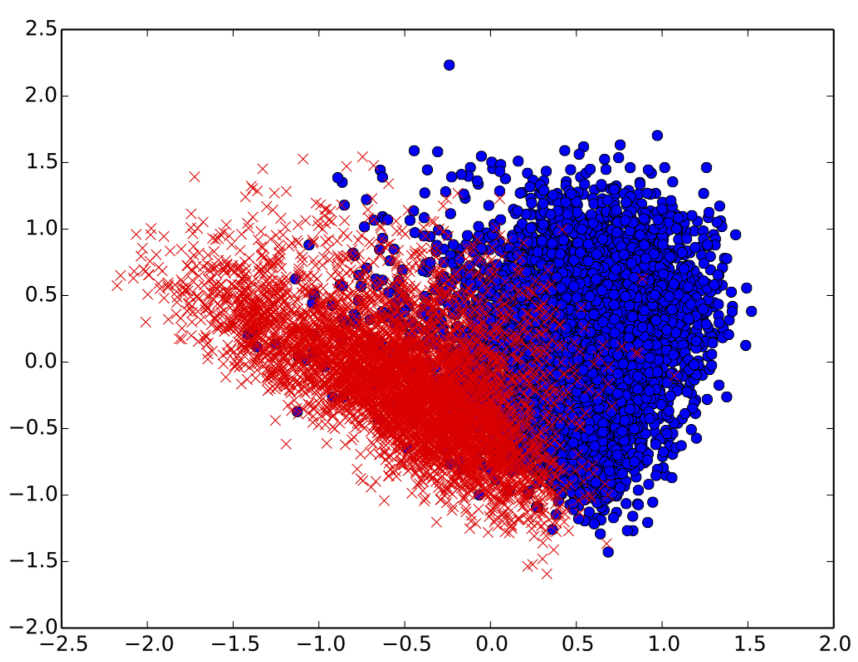


Figure 3: AdaSent on SUBJ

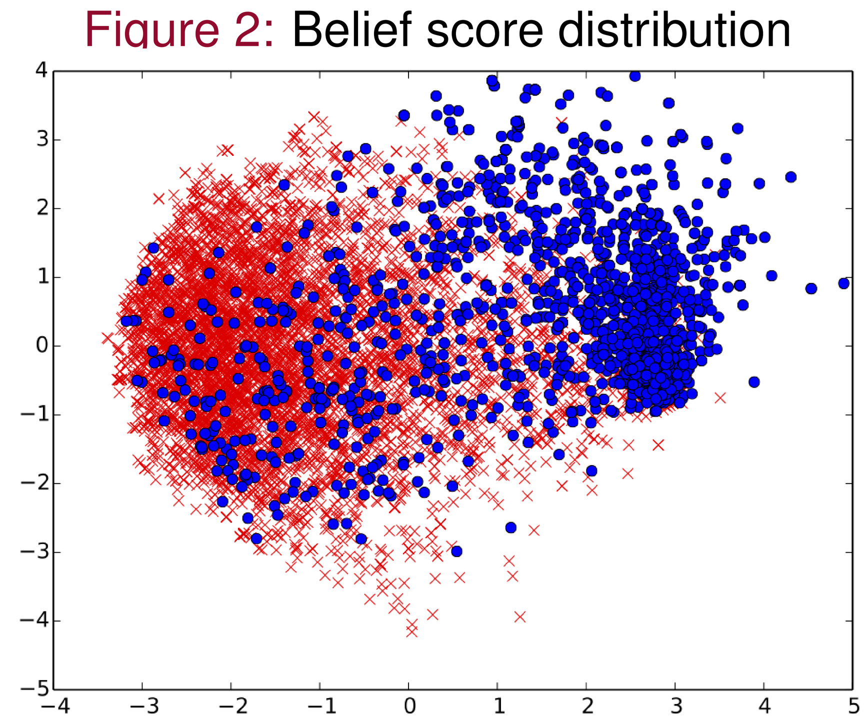


Figure 4: AdaSent on MPQA