Enhanced LSTM for Natural Language Inference

Qian Chen[1], Xiaodan Zhu[23], Zhen-Hua Ling[1],
Si Wei[4], Hui Jiang[5], Diana Inkpen[6]
[1]University of Science and Technology of China
[2]National Research Council Canada [3]Queen's University
[4]iFLYTEK Research [5]York University [6]University of Ottawa

Contributions

- ★ Propose a hybrid neural network model for natural language inference.
  - ⋆ The model achieves the best results on the SNLI dataset.
- ★ Our first component, **E**nhanced **S**equential **I**nference **M**odel (**ESIM**), has outperformed the previous best results.
- ★ Further using tree-LSTM [Zhu, ICML-2015, Tai, ACL-2015, Le, *SEM-2015] to encode syntactic parses can improve the performance additionally.

Source code available!!!

https://github.com/lukecq1231/nli

Our implementation uses python and is based on the **Theano** library.

An example

Natural language inference (NLI) aims to determine whether a natural-language **hypothesis** H can be inferred from a **premise** P.

- **Premise**: A woman wearing a black dress and green sweater is walking down the street looking at her cellphone.
- **Hypothesis 1**: A woman is holding her cell phone. (Entailment)
- **Hypothesis 2**: A woman is looking at a text on her cell phone. (Neutral)
- **Hypothesis 3**: A woman has her cell phone up to her ear. (Contradiction)

Analysis

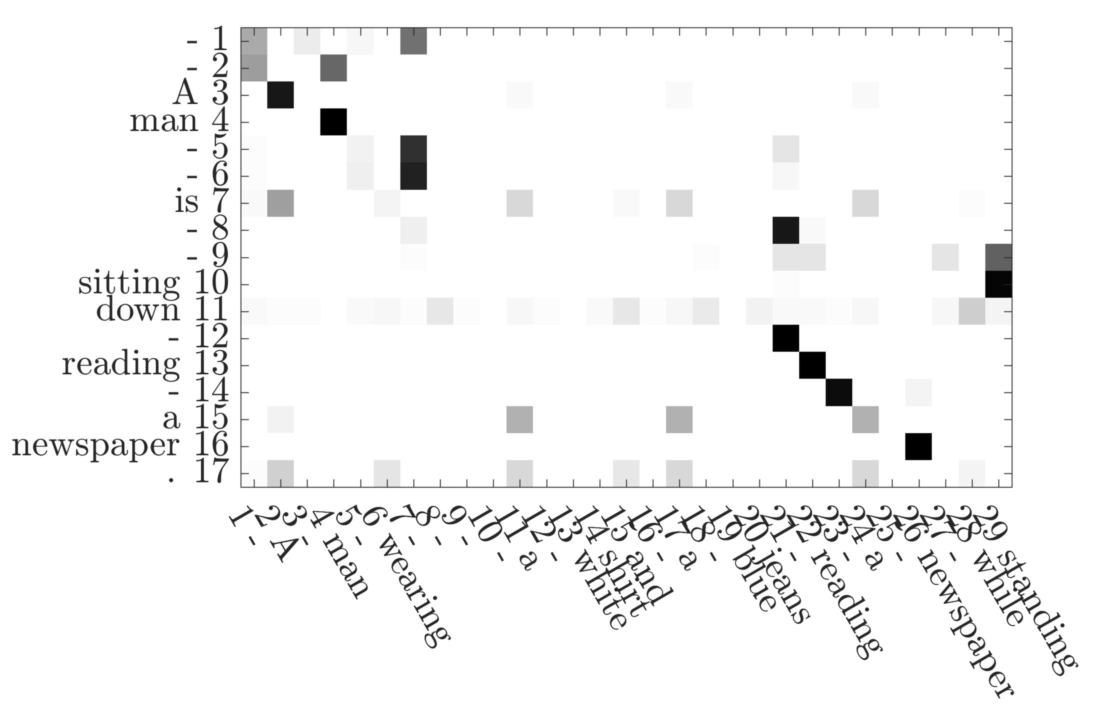

Figure 1: Attention visualization of stand-alone syntactic tree-LSTM model

Hybrid Neural Inference Model

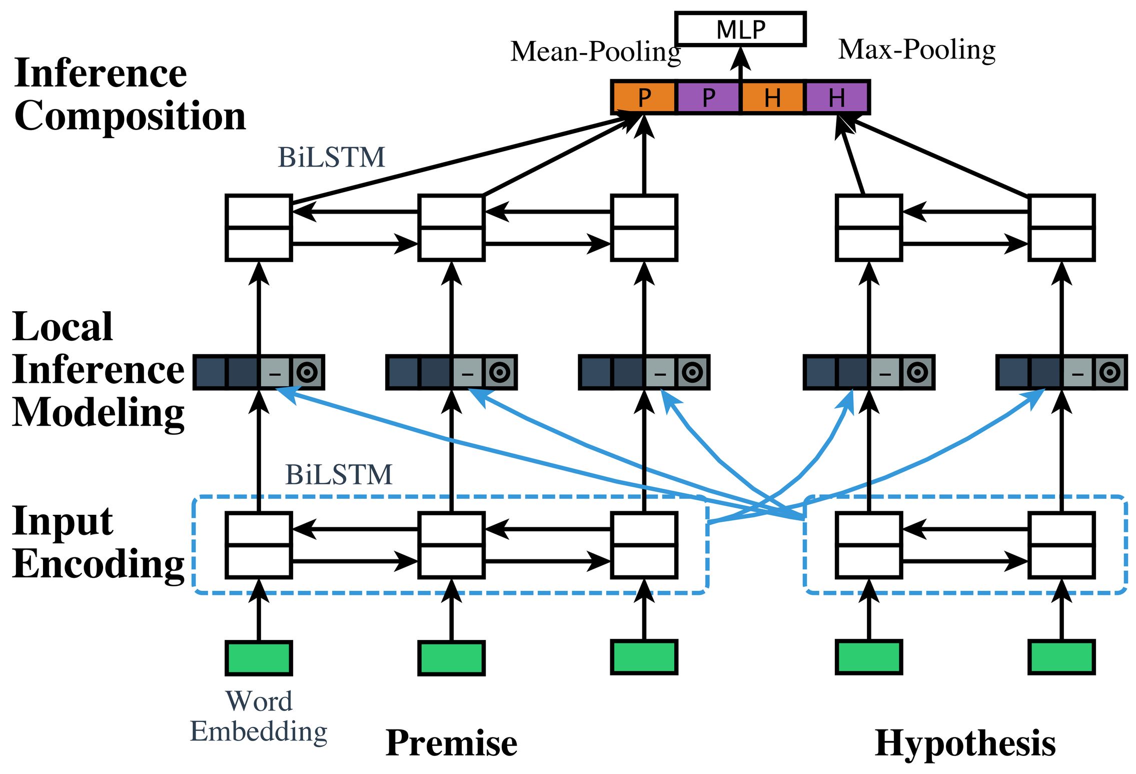

Figure 2: A high-level view of our **E**nhanced **S**equential **I**nference **M**odel (**ESIM**)

1. Input Encoding

Premise: $x_1^p, x_2^p, \dots, x_N^p$
Hypothesis: $x_1^h, x_2^h, \dots, x_M^h$
Embedding matrix: $E \in \mathbb{R}^{V\times D_m}$

$$h^p = \text{Enc}(E(x_1^p), \dots, E(x_N^p)) \in \mathbb{R}^{N\times D_e} \quad (1)$$

$$h^h = \text{Enc}(E(x_1^h), \dots, E(x_M^h)) \in \mathbb{R}^{M\times D_e} \quad (2)$$

where *Enc* is BiLSTM or Tree-LSTM [ZSG15, TSM15]. Here *Enc* learns to represent a word (or phrase) and its context.

2. Local Inference Modeling

- Local inference collected

$$e_{ij} = (h_i^p)^T h_j^h, e \in \mathbb{R}^{N\times M} \quad (3)$$

$$\tilde{h}_i^p = \sum_j \frac{\exp(e_{ij})}{\sum_k \exp(e_{ik})} h_j^h, \tilde{h}^p \in \mathbb{R}^{N\times D_e} \quad (4)$$

$$\tilde{h}_j^h = \sum_i \frac{\exp(e_{ij})}{\sum_k \exp(e_{kj})} h_i^p, \tilde{h}^h \in \mathbb{R}^{M\times D_e} \quad (5)$$

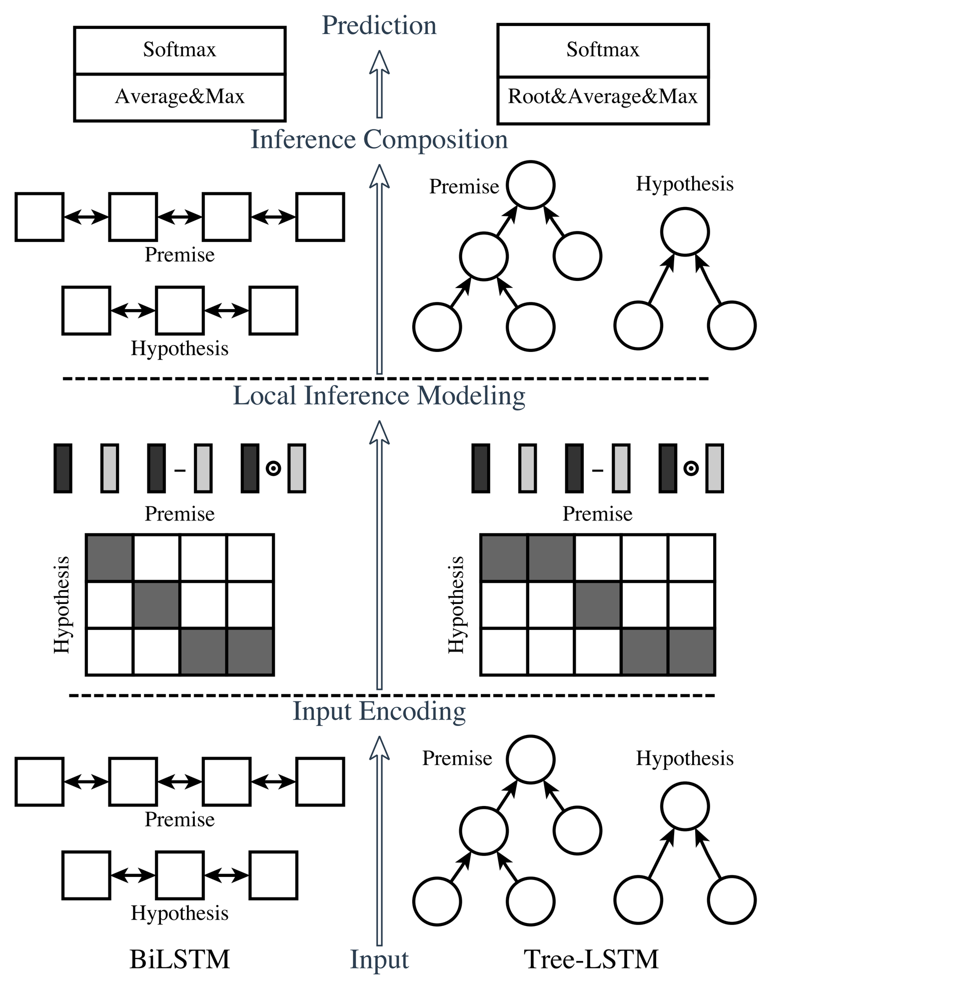

Figure 3: A high-level view of our **H**ybrid **I**nference **M**odel (**HIM**)

Intuitively, the content in h^h that is relevant to h_i^p will be selected and represented as $\tilde{h}_i^p$, and vice versa.

- Enhancement of local inference information

$$m^p = [h^p; \tilde{h}^p; h^p - \tilde{h}^p; h^p \odot \tilde{h}^p] \in \mathbb{R}^{N\times 4D_e} \quad (6)$$

$$m^h = [h^h; \tilde{h}^h; h^h - \tilde{h}^h; h^h \odot \tilde{h}^h] \in \mathbb{R}^{M\times 4D_e} \quad (7)$$

3. Inference Composition

$$v^p = \text{Cmp}(m_1^p, \dots, m_N^p) \in \mathbb{R}^{N\times D_c} \quad (8)$$

$$v^h = \text{Cmp}(m_1^h, \dots, m_M^g) \in \mathbb{R}^{M\times D_c} \quad (9)$$

$$v = [max(v^p); ave(v^p); max(v^h); ave(v^h)] \in \mathbb{R}^{4D_c} \quad (10)$$

where *Cmp* is BiLSTM or Tree-LSTM. Finally, we put v into a final MLP classifier.

Results

- **Data: S**tanford **N**atural **L**anguage **I**nference (SNLI) (Training: 550k sentence pairs, held-out: 10k, testing: 10k)

Table 1: Accuracies of the models on SNLI

Model	Test
(1) Handcrafted features [BAPM15]	78.2
(2) LSTM [BGR+16]	80.6
(3) GRU [VKFU15]	81.4
(4) Tree CNN [MML+16]	82.1
(5) SPINN-PI [BGR+16]	83.2
(6) BiLSTM intra-Att [LSLW16]	84.2
(7) NSE [MY16a]	84.6
(8) Att-LSTM [RGH+15]	83.5
(9) mLSTM [WJ16]	86.1
(10) LSTMN [CDL16]	86.3
(11) Decomposable Att [PTDU16]	86.3
(12) Intra-sent Att+(11) [PTDU16]	86.8
(13) NTI-SLSTM-LSTM [MY16b]	87.3
(14) Re-read LSTM [SCSL16]	87.5
(15) btree-LSTM [PAD+16]	87.6
(16) ESIM	88.0
(17) HIM (ESIM+Syn.tree-LSTM)	**88.6**

- **E**nhanced **S**equential **I**nference **M**odel (**ESIM**) achieves an accuracy of 88.0%, which has already outperformed all the previous models.
- **H**ybrid **I**nference **M**odel (**HIM**), which ensembles our ESIM model with syntactic tree-LSTMs [ZSG15] based on syntactic parse trees, achieve additional improvement.

Table 2: Ablation performance of the models

Model	Test
(17) HIM (ESIM + syn.tree)	88.6
(18) ESIM + tree	88.2
(16) ESIM	88.0
(19) ESIM - ave./max	87.1
(20) ESIM - diff./prod.	87.0
(21) ESIM - inference BiLSTM	87.3
(22) ESIM - encoding BiLSTM	86.3
(23) ESIM - P-based attention	87.2
(24) ESIM - H-based attention	86.5
(25) syn.tree	87.8

Training Speed: tree-LSTM takes about 40 hours on Nvidia-Tesla K40M and ESIM takes about 6 hours.