Learning to Select Important Context Words for Event Detection

Nghia Trung Ngo[1], Tuan Ngo Nguyen[2], Thien Huu Nguyen[2,3]

[1]Hanoi University of Science and Technology, Hanoi, Vietnam

[2]Department of Computer and Information Science, University of Oregon, USA

[3]VinAI Research, Hanoi, Vietnam

Event Detection

- Event detection seeks to identify **event triggers** and and classify them into some predefined types of interests
  - Event triggers are the most important words or phrases that are responsible for evoking the events

S1 (*Attack* event)

The police **fired** tear gas and water cannons in street battles with activists.

Key ED feature: context words

- Some context words in the sentences are necessary to correctly classify the trigger words

S2 (*Transport* event)

She needs to go home to meet a friend at 5, so she has to **leave** the company early.

S3 (*End-Position* event)

She decided to **leave** the company for a better job.

- Therefore, event detection models should be able to learn and select context words in accompany with event triggers

Prior strategies for selecting context words

- Two typical strategies
  - entity mention strategy: selecting the words that correspond to entity mentions and the surrounding words
  - dependency strategy: selecting the words connected to the trigger words in the dependency parse trees of the sentences
- An issue: these pre-determined strategies are not flexible for all possible sentences
- We propose a method to automatically learn to select important context words for event detection models

Model

- Encoding sentence: concatenating pre-trained word embeddings, position embeddings and entity embeddings
- LearnToSelect: a squence of word selections where the most important word is selected at each step i
  - computing relevant scores
  - selecting the most important word with Gumbel-Softmax distribution
- Feeding the triggers and selected context words into fully-connected layer and perform prediction

LearnToSelect: Notations

- At selection step i, we denote v^i to be the accumulated representation vector up to the current step
- Let $G_i = \{w_1^i, w_2^i, ..., w_{N_i}^i\}$ be the list of non-selected words in the input sentence W
- Let $U_i = \{u_1^i, u_2^i, ..., u_{N_i}^i\}$ be the corresponding list of contextualized vectors for words in G_i

LearnToSelect: computing relevance scores

- At step $i > 1$, there are $N - i$ words that have not been selected in the previous $i - 1$ steps
- We compute a relevance score for each of the non-selected words

$$r(w_j^i) = q^T \odot v^{i+1} = q^T \odot m(u_j^i, v^i, c^i)$$

  - q is a query vector to be learned during training
  - cm is a function that incorporate info in u_j^i into v^i and forms new representation vector v^{i+1}
- $cm(u_j^i, v^i, c^i)$ function is motivated by the LSTM units

$$\begin{bmatrix} f \\ i \\ o \\ g \end{bmatrix} = \begin{bmatrix} \sigma \\ \sigma \\ \sigma \\ tanh \end{bmatrix} \left(\mathbf{W}_{com} \begin{bmatrix} u_j^i \\ v^i \end{bmatrix} + \mathbf{b}_{com} \right)$$

$$m_j^{i+1} = f \odot c^i + i \odot g$$

$$cm(u_j^i, v^i, c^i) = o \odot tanh(m_j^{i+1})$$

LearnToSelect: selecting the most important word with Gumbel-Softmax

- Simply taking word with the highest score via $d_i = \arg\max_j w^i_{d_j}$ would cause a problem to train the whole model.
  - d_i is discrete and non-differential, blocking back-propagation during training
- Applying Straight-Through Gumbel-Softmax (STGS) estimator (Ji et al, 2018)
  - an efficient method to sample from the categorical distribution over G_i
  - allowing the model to be trained with discrete sample from G_i

Experiments

- Two benchmark datasets
  - ACE 2005 dataset
  - TAC KBP 2015 dataset
- Ablation studies
  - LSTM-Compose function cm vs. a simple non-gating RNN-Compose
  - LearnToSelect method vs. other pre-determined selection strategies

Evaluation on the ACE 2005 dataset

Model	P	R	F
CNN	71.8	66.4	69.0
DM-CNN	75.6	63.6	69.1
DM-CNN+ ‡	75.7	66.0	70.5
JRNN	66.0	73.0	69.3
CNN-LSTM	84.6	64.9	73.4
FrameNet ‡	77.6	65.2	70.7
DepTensor	-	-	69.6
SELF-GAN	71.3	74.7	73.0
NCNN	-	-	71.3
GCNN-ENT	77.9	68.8	73.1
SupervisedAtt ‡	76.8	67.5	71.9
MultiAtt ‡	78.9	66.9	72.4
HBT †	77.9	69.1	73.3
DEEB-RNN	72.3	75.8	74.0
Transformer	73.4	69.5	71.4
LearnToSelectED	75.4	75.0	**75.2**

Evaluation on the TAC KBP 2015 dataset

Model	P	R	F
TAC TOP	75.2	47.7	58.4
TAC SECOND	74.0	46.6	57.2
TAC THIRD	73.7	44.9	55.8
SSED	69.9	48.8	57.5
MSEP	69.2	47.8	56.6
GCNN-ANC	67.3	50.8	57.9
GCNN-ENT	70.3	50.6	58.8
LearnToSelectED	63.7	58.7	**61.1**

Ablation study: LSTM-Compose and LearnToSelect

Model	P	R	F
LSTM-Compose + LearnSelection (a.k.a *LearnToSelectED*)	74.6	71.5	73.0
RNN-Compose + LearnSelection	73.6	65.8	69.5
LSTM-Compose + All	73.3	63.6	68.1
LSTM-Compose + Entity	74.4	67.8	71.0
LSTM-Compose + Syntax	75.7	66.7	70.9
LSTM-Compose + Entity-Syntax	74.9	69.5	72.1
LSTM-Compose + Window	76.2	67.6	71.6

Conclusion

- In this work: A novel method to automatically learn to select important context words for event detection models.
- In future work: extend and evaluate the proposed method to the related tasks e.g., relation extraction.

Thank you !