Self-regulation: Employing a **Generative Adversarial Network** to Improve **Event Detection**

Yu Hong, Wenxuan Zhou, Jingli Zhang, Qiaoming Zhu and Guodong Zhou
Soochow University
tianxianer@gmail.com

Task Definition

Event detection is required to go through a sentence, so as to pick up a trigger and then determine the event type it evokes.

Generality – *taken home* <Transport>
Ambiguity 1 – *campaign in Iraq* <Attack>
Ambiguity 2 – *political campaign* <Elect>
Coreference – *Either its bad or good* <Marry>

(The underlined are triggers, <*> denotes an event type)

Challenges

The frequent utilization of common words, ambiguous words and pronouns in event mentions makes them harder to detect

Motivation (Leverage NN? Of cause, but …)

The neurons suffer from spurious features. Such features appear as semantically related information but actually they ARE NOT.

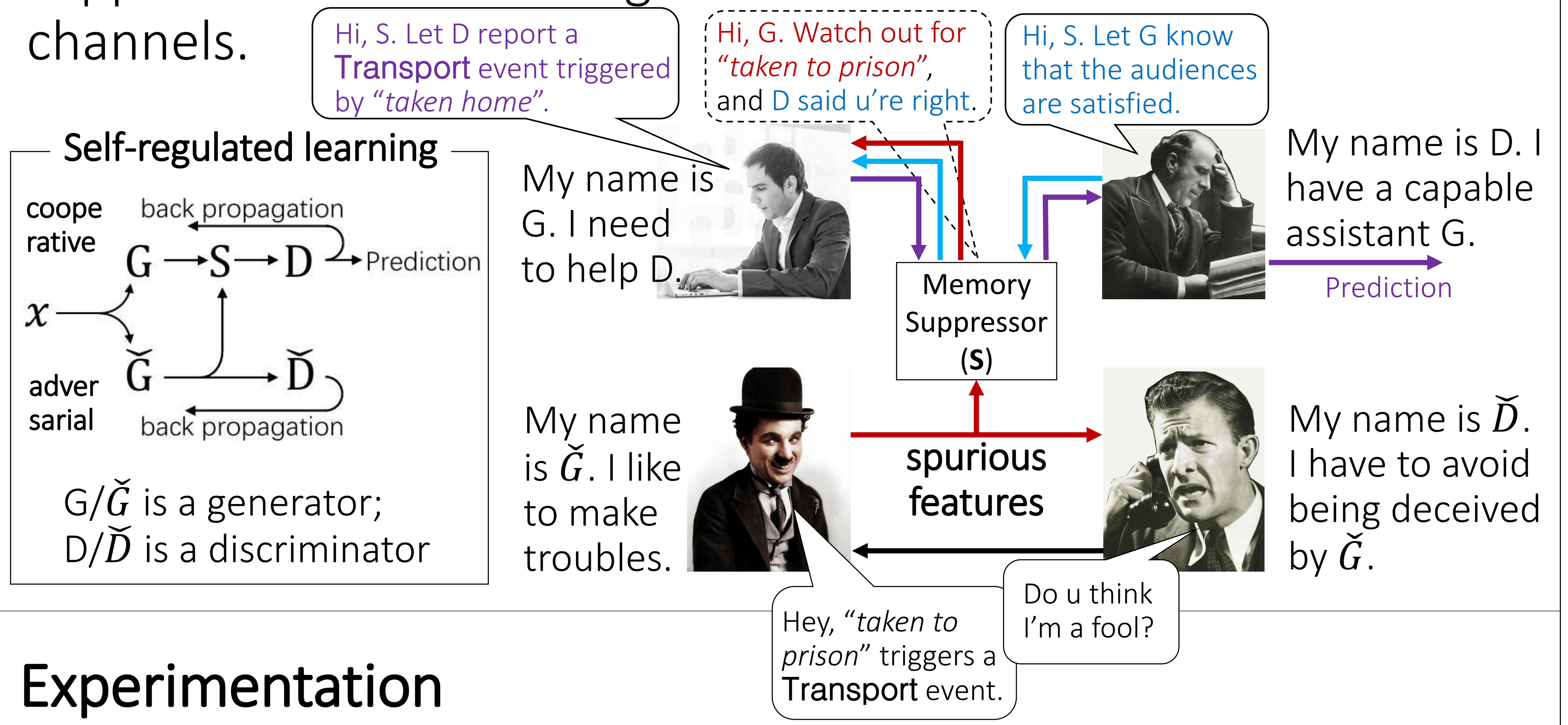
spurious reliable
Prison authorities have given the nod for An- war to be taken home later in the afternoon.
Trigger: *taken*. **Event Type**: Transport

taken **home**? or taken to **prison**?

The paper is motivated by the thought of purifying the latent feature space.

taken home -> Transport
taken to prison -> Arrest

Model (Self-regulation, abbr., SELF): SELF is a double-channel model, consisted of a cooperative network and a GAN. A suppressor **S** is used to regulate communication between the channels.



Experimentation

We run SELF on the ACE'05 corpus.
ACE'05 is annotated with single-token event triggers, and has 33 predefined event types, along with one "non-trigger" type.

Settings: The source code, along with the configuration document, has been made publicly available.
(https://github.com/JoeZhouWenxuan/Self-regulation-Employing-a-Generative-Adversarial-Network-to-Improve-Event-Detection/tree/master)

Performance: SELF outperforms all the state-of-the-art methods for trigger identification, and achieves comparable performance to the top 1 model (Hybrid).

Method	P (%)	R (%)	F (%)
Joint (Local+Global)	76.9	65.0	70.4
MSEP-EMD	75.6	69.8	72.6
DM-CNN	80.4	67.7	73.5
DM-CNN*	79.7	69.6	74.3
Bi-RNN	68.5	75.7	71.9
Hybrid: Bi-LSTM+CNN	80.8	71.5	75.9
SELF: Bi-LSTM+GAN	**75.3**	**78.8**	**77.0**

Table 1: Trigger identification performance

Methods	P (%)	R (%)	F (%)
MSEP-EMD	70.4	65.0	67.6
Cross-Event	68.8	68.9	68.8
Cross-Entity	72.9	64.3	68.3
Joint (Local+Global)	73.7	62.3	67.5
CNN	71.8	66.4	69.0
DM-CNN	75.6	63.6	69.1
NC-CNN	-	-	71.3
FB-RNN (GRU)	66.8	68.0	67.4
Bi-RNN (GRU)	66.0	73.0	69.3
ANNs (ACE+FN)	77.6	65.2	70.7
DM-CNN*(ACE+Wiki)	75.7	66.0	70.5
ANN-S2 (ACE+FN)	76.8	67.5	71.9
Hybrid: Bi-LSTM+**CNN**	**84.6**	64.9	**73.4**
SELF: Bi-LSTM+**GAN**	71.3	**74.7**	**73.0**

Table 2: Type classification performance

Advantage 1: SELF recalls more instances (see Table 1&2).
Advantage 2: SELF balances recall and precision (see Figure 1).
Advantage 3: SELF is domain-adaptable (see Section 5.5 in pp)

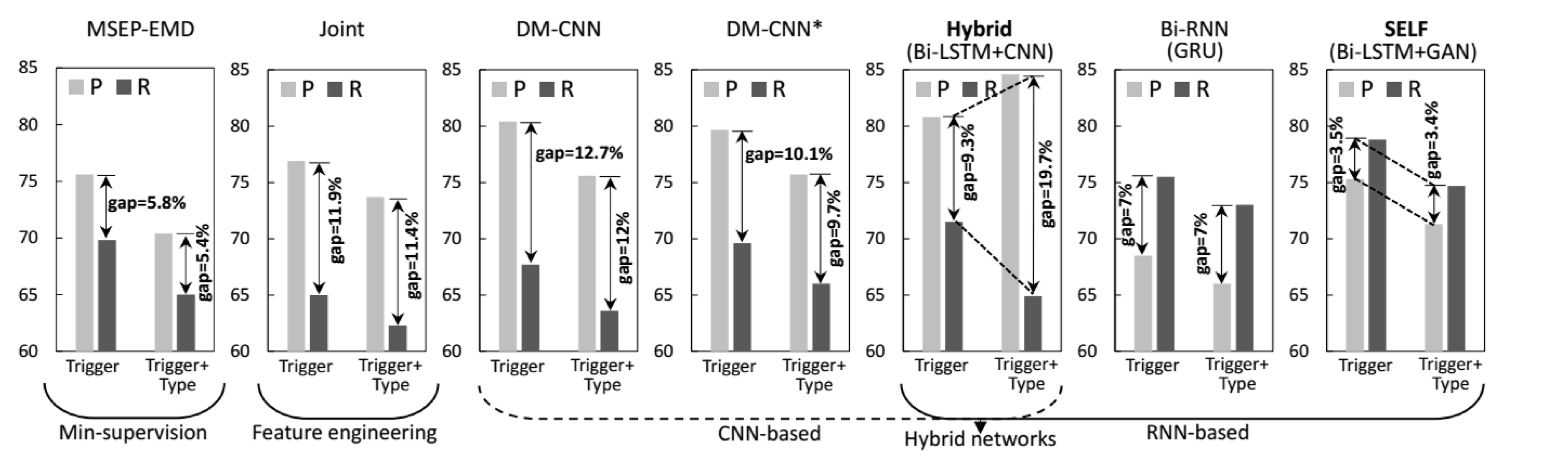
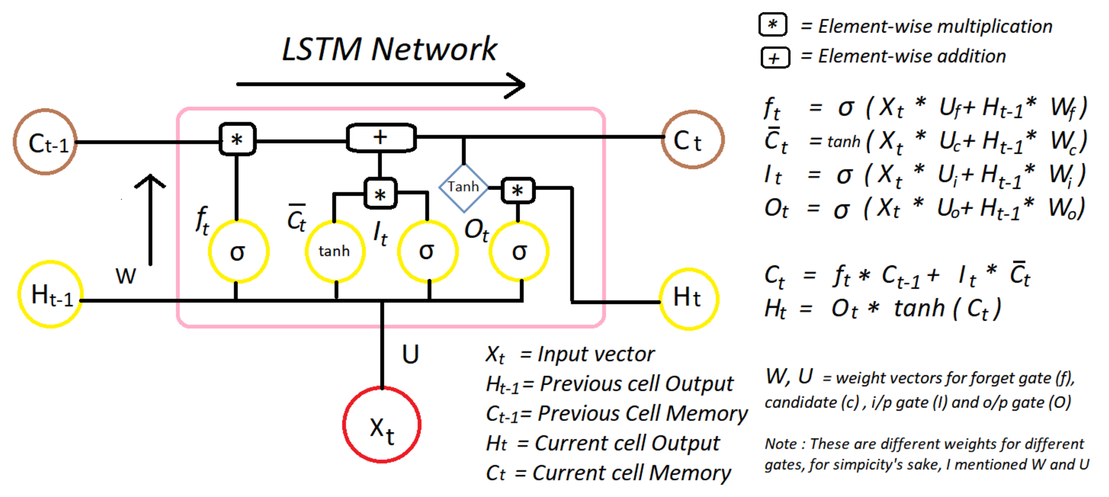


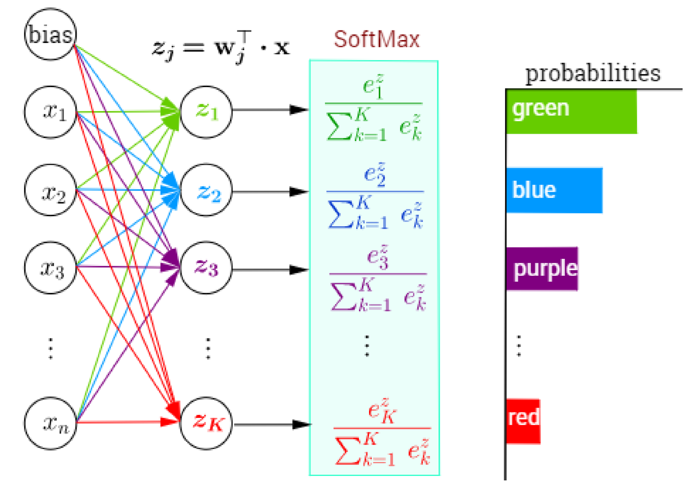
Figure 1: Gaps between precision and recall in the tasks of trigger identification and event classification

Implementation (by LSTM)

LSTM based RNN is used to construct the generators.



A discriminator is specified as the softmax layer following a full-connected layer.



Multi-class classification is performed. Given a word in a sentence:

$$\hat{y} = softmax(\hat{W} \cdot o_t + \hat{b})$$

$\hat{y}$ denotes the prediction for event type: it is either a predefined type or non-trigger

Training

For the adversarial channel:

$$\mathcal{L}(\hat{y}_{\breve{g}}, y) = -\sum_{i=1}^{N}\sum_{j=1}^{C} y_i^j log(\hat{y}^j_{\breve{g},i})$$

$$\theta_{\breve{g}} = argmax\ \mathcal{L}(\hat{y}_{\breve{g}}, y)$$

$$\theta_{\breve{d}} = argmin\ \mathcal{L}(\hat{y}_{\breve{g}}, y)$$

y denotes the ground-truth event type; $\theta_{\breve{g}}$ represents all the parameters of LSTM in $\breve{g}$, while $\theta_{\breve{d}}$ represents the parameters of the full-connected layer in $\breve{d}$

For the cooperative channel:

$$\mathcal{L}(\hat{y}_g, y) = -\sum_{i=1}^{N}\sum_{j=1}^{C} y_i^j log(\hat{y}^j_{g,i})$$

$$\theta_g = argmin\ (\mathcal{L}\hat{y}_g, y)$$

$$\theta_d = argmin\ (\mathcal{L}(\hat{y}_g, y) + \lambda \cdot \mathcal{L}_{SELF})$$

θ_g represents the parameters in g, while θ_d in d. L_{SELF} is the self-regulation loss: $\mathcal{L}_{SELF} = \sum_{i=1}^{N'} \mathcal{L}_{diff}(O_g, O_{\breve{g}})$

$$\mathcal{L}_{diff}(O_g, O_{\breve{g}}) = \left(\sum_{i=1}^{l}\sum_{j=1}^{l} |o_{g,i} o_{\breve{g},j}|^2\right)^{\frac{1}{2}}$$

This is the job of the suppressor S.