Generative Adversarial Networks in Neural Machine Translation

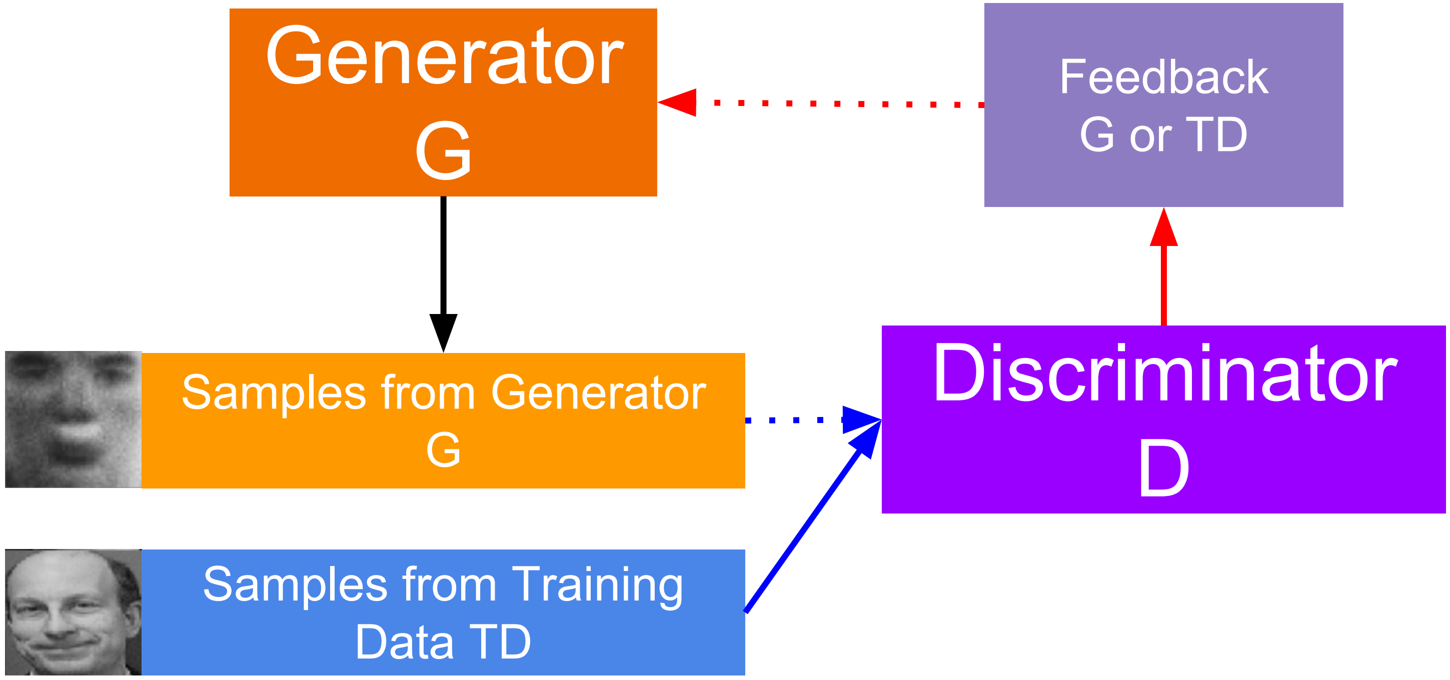
NGO HO Anh Khoa

Outline

- Main idea
- Model of Neural Machine Translation
- Algorithm
- Comparison with Dual Learning
- Conclusion
- Ongoing work

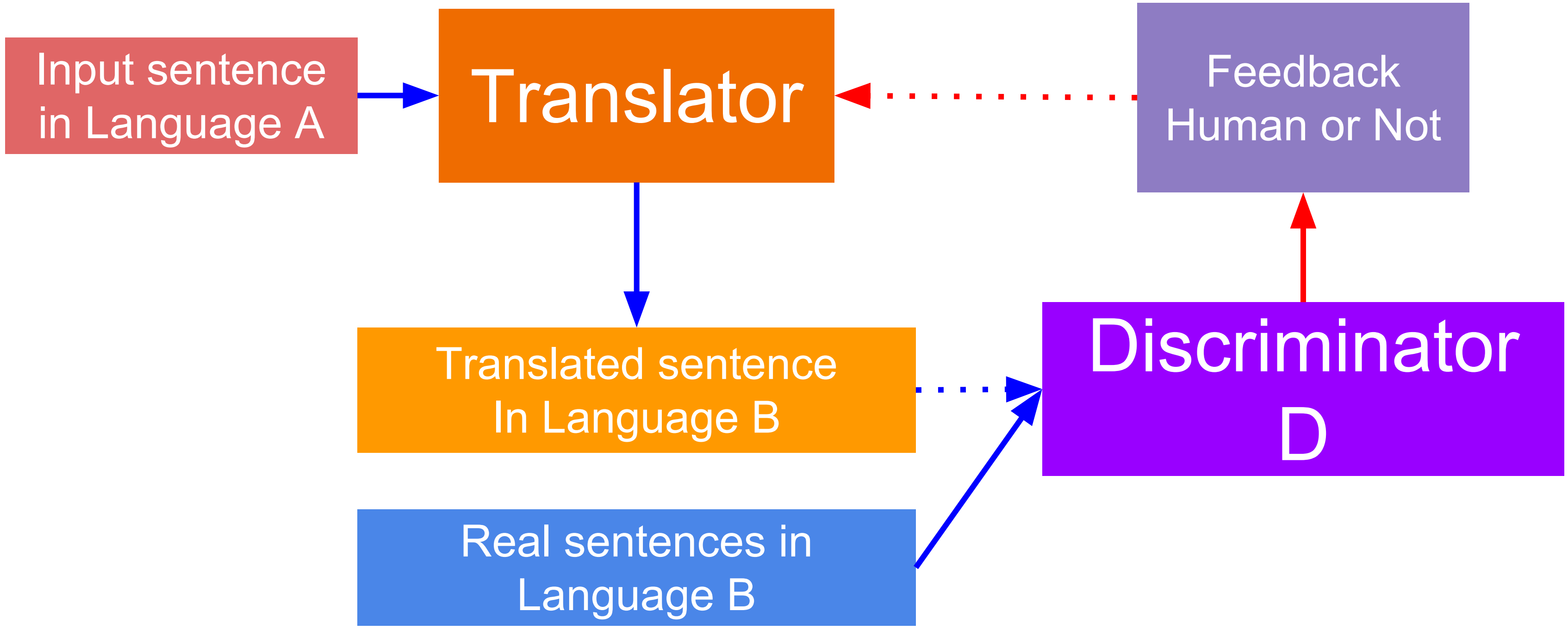
Main idea

- Applying **Generative Adversarial Nets** to Neural Machine Translation



Main idea

- Applying Generative Adversarial Nets to **Neural Machine Translation**



Model of NMT

- **Generator**: *Attention-based NMT model*
  - Objective function: Maximize the expected end reward of a sequence

$$J(\theta) = \sum_{y_t} G_\theta(y_t | y_{1:t-1}, x) \cdot R_D^{G_\theta}((y_{1:t-1}, x), y_t) \quad (13)$$

- **Discriminator**: *Binary classifier discriminates the machine-translated sentence from the human-translated sentence.*
  - **CNN-based** or RNN-based model
  - Return Probability of a *full sentence* being human-generated

$$D(x, y_{1:T})$$

  - Objective function: Minimize the cross entropy

$$-\mathbb{E}_{x,y \in p_{data}}[\log D(x,y)] - \mathbb{E}_{x,y \in G}[\log(1 - D(x,y))] \quad (17)$$

Model of NMT

- **Generator**: *Attention-based NMT model*
  - Objective function: Maximize the expected end reward of a sequence

$$J(\theta) = \sum_{y_t} G_\theta(y_t | y_{1:t-1}, x) \cdot \boxed{R_D^{G_\theta}((y_{1:t-1}, x), y_t)} \quad (13)$$

  - Problem: **Reward of partially generated sentence**
    - Reward for a full sentence

$$R_D^{G_\theta}((y_{1:T-1}, x), y_T) = \boxed{D(x, y_{1:T})} - b(x, y_{1:T}) \quad (14)$$

    - Solution: Sample the rest of a sentence with N-time Monte Carlo search by using the newest trained generator.

Model of NMT

- **Generator**: *Attention-based NMT model*
  - Objective function: Maximize the expected end reward of a sequence

$$J(\theta) = \sum_{y_t} G_\theta(y_t | y_{1:t-1}, x) \cdot \boxed{R_D^{G_\theta}((y_{1:t-1}, x), y_t)} \quad (13)$$

  - Problem: **Reward of partially generated sentence**
    - Solution: Sample the rest of a sentence with N-time Monte Carlo search by using the newest trained generator

$$R_D^{G_\theta}((y_{1:t-1}, x), y_t) = \qquad (16)$$
$$\begin{cases} \frac{1}{N} \sum_{n=1}^{N} D(x, y_{1:L_n}^n) - b(x, y_{1:L_n}^n), \boxed{y_{1:L_n}^n \in MC^{G_\theta}((y_{1:t}, x), N)} & \text{for } t < L \\ D(x, y_{1:t}) - b(x, y_{1:t}) & \text{for } t = L \end{cases}$$

Model of NMT

- **Generator**: *Attention-based NMT model*
  - Objective function: Maximize the expected end reward of a sequence

$$J(\theta) = \sum_{y_t} G_\theta(y_t|y_{1:t-1}, x) \cdot \boxed{R_D^{G_\theta}((y_{1:t-1}, x), y_t)} \quad (13)$$

  - Problem: **Problem of a strong discriminator**
    - Solution: Professor Forcing pushes Generator to generate the good-true sequence by using human-translated sentence

$$J(\theta) = \sum_{y_t} G_\theta(y_t|y_{1:t-1}, x) \cdot \boxed{\mathbf{1}}$$

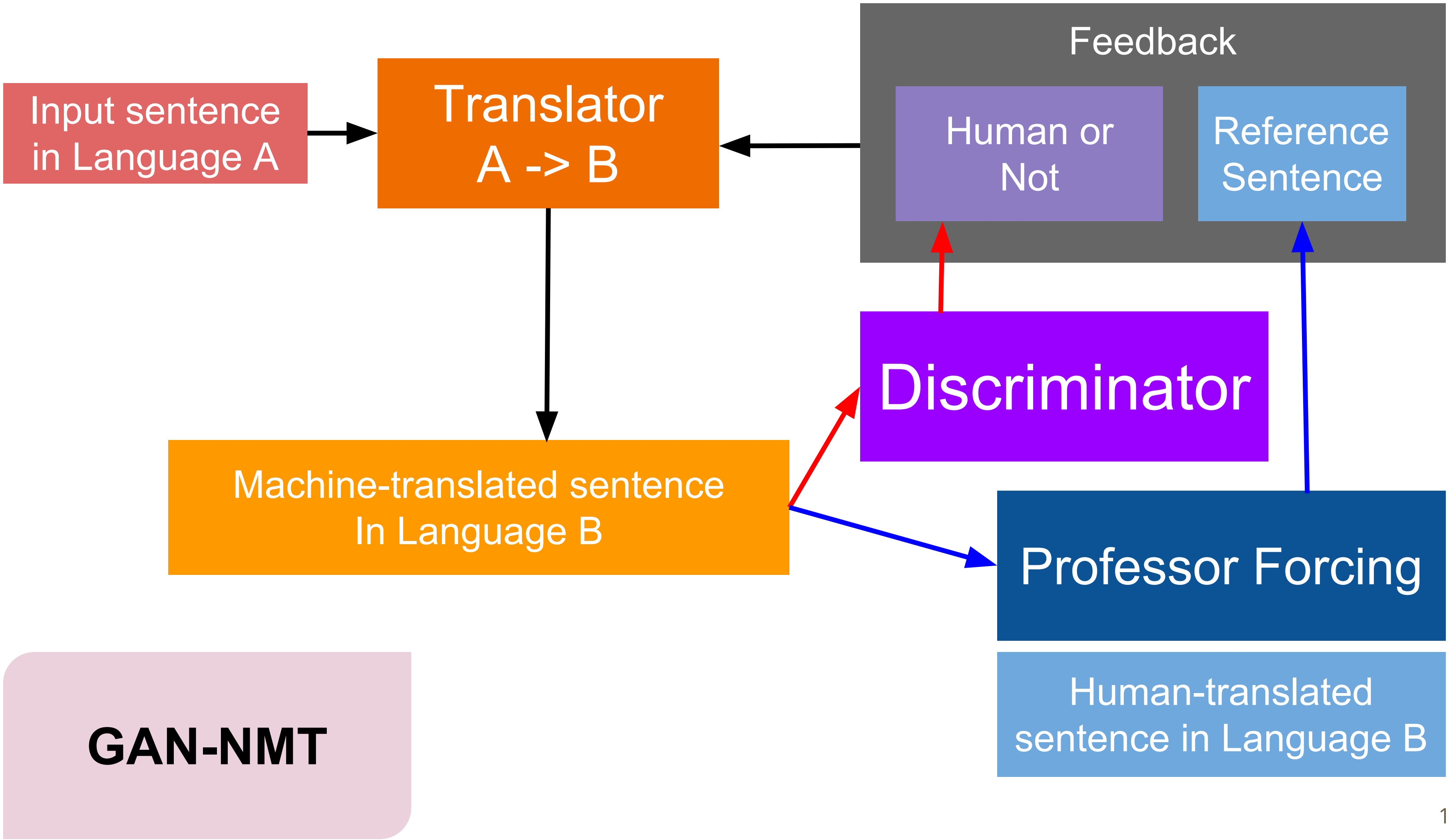
    - Result: **A modified objective function**

$$J(\theta) = \underbrace{\boxed{\sum_{y_t} G_\theta(y_t|y_{1:t-1}, x) \cdot R_D^{G_\theta}((y_{1:t-1}, x), y_t)}}_{\text{Machine-translated}} + \underbrace{\boxed{\sum_{y_t} G_\theta(y_t|y_{1:t-1}, x) \cdot \mathbf{1}}}_{\text{Human-translated}}$$

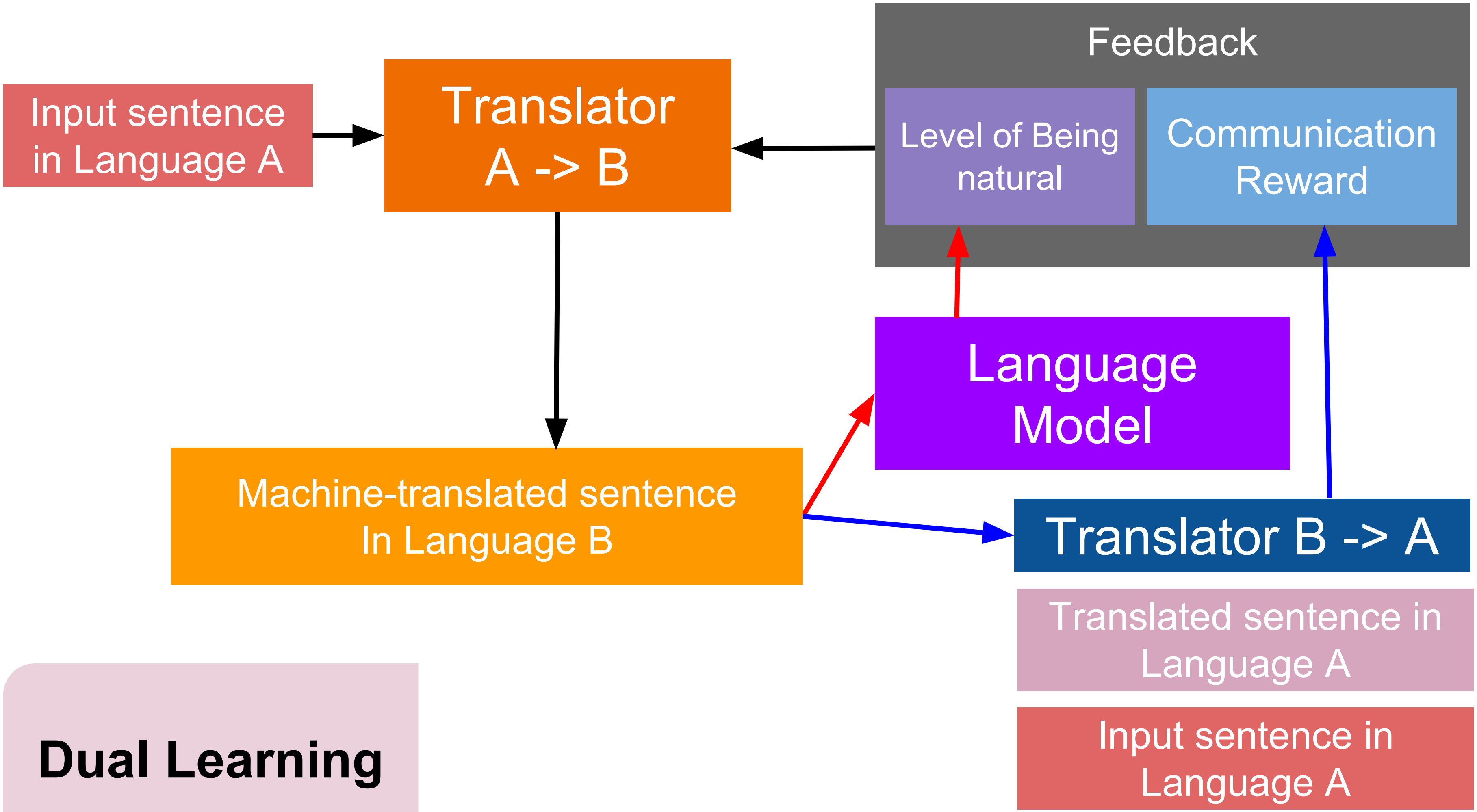
Algorithm

while *not convergence* **do**
  for *g-steps* **do**
    Sample randomly a batch of input sentence and human-translated sentence $(X, hY)_{1:S}$
    for $s \in 1{:}S$ **do**
      Generator translates X_s into $mY_{1:T} = (my_1, ..., my_T)$
      for $t \in 1{:}T$ **do**
        Compute $R_D^G(my_{1:t-1}, my_t)$ with N-time Monte Carlo search
      end
      Update Generator:
      $\bigtriangledown J(\theta) = \sum_{my_t} R_D^G(my_{1:t-1}, my_t) \bigtriangledown_\theta logP(my_t|my_{1:t-1}, X_s)$
      $\theta \longleftarrow \theta + \mu \bigtriangledown J(\theta)$
      Professor Forcing: Update Generator with hY_s
      $\bigtriangledown J(\theta) = \sum_{hy_t} 1 * \bigtriangledown_\theta logP(hy_t|hy_{1:t-1}, X_s)$
      $\theta \longleftarrow \theta + \mu \bigtriangledown J(\theta)$
    end
    for *d-steps* **do**
      Generator generates η sentences as Negative examples
      Sample randomly human-generates sentences as Positive examples
      Update Discriminator
    end
  end
end

Comparison with Dual Learning

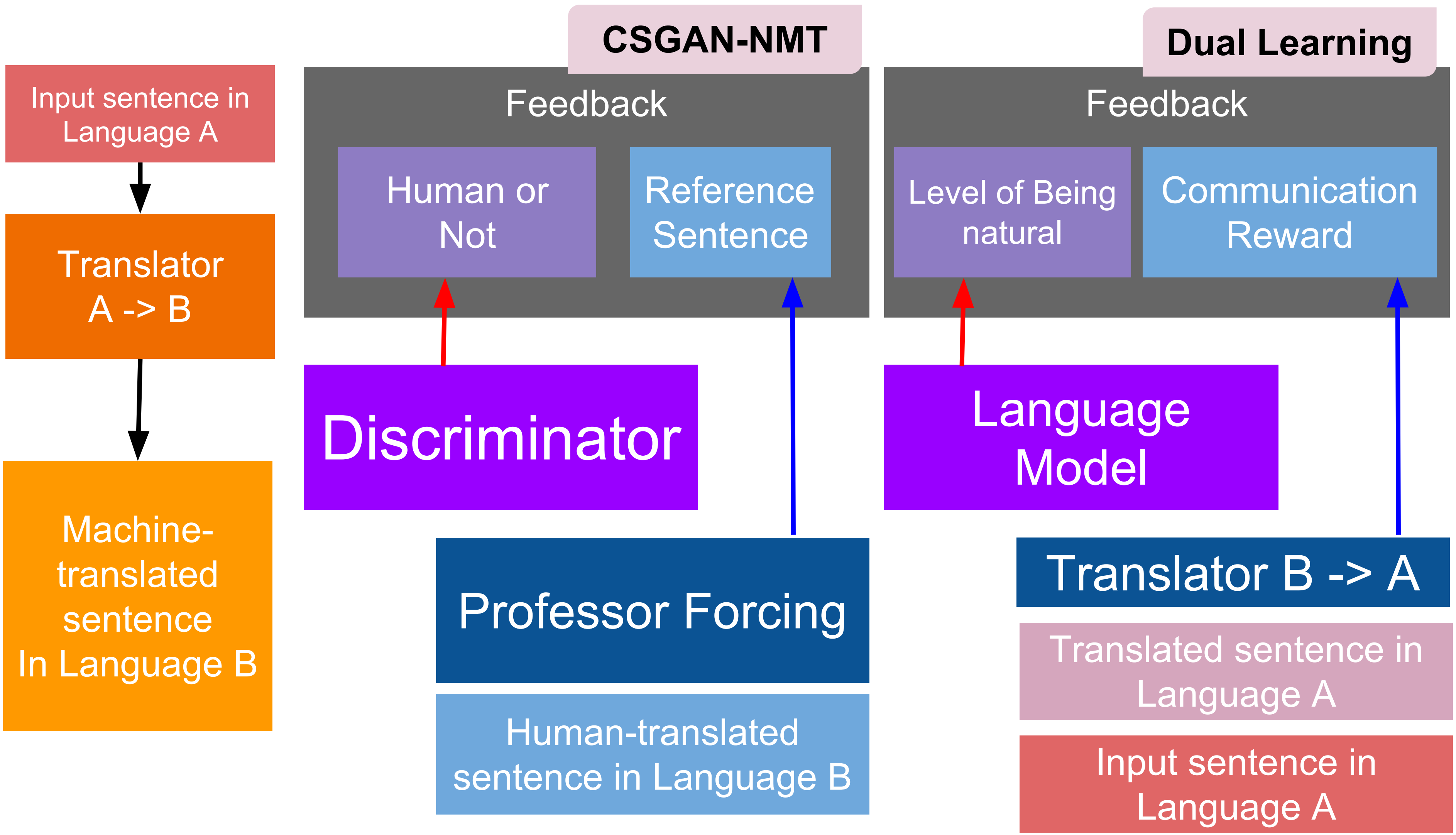


Comparison with Dual Learning



Dual Learning

Comparison with Dual Learning



Conclusion of GAN

- Framework of GAN with Reward of Reinforcement Learning
- Reward for a partially generated sentence
- Problem of strong discriminator - Unstable training
    - CNN-based or RNN-based
    - Professor Forcing
    - Initial accuracy of discriminator

Ongoing work

- Implementation:
  - Generative Adversarial Network
  - Reward for a partially generated sentence: Not MC search.
  - Improving the feedback for Generator: Feedback from Language Model
- Research:
  - Combination between Variational Neural Network and Generative Adversarial Network

Thank you