Unsupervised Machine Translation Using Monolingual Corpora Only

Lample et. al. 2018
Presenter: Michael Przystupa

Overview

1. Introduction
2. Unsupervised Neural Machine Translation (UNMT)
3. Training
4. Experiments (just results)
5. ~~Related Work~~
6. ~~Conclusion~~ (UNMT approach work better than expected)

Introduction: Background

- Supervised Neural Machine Translation
  - Works great if you have lots of bitext (order of millions of paired sentences)
  - Collecting parallel data can be difficult or costly
- Boosting translation with Monolingual Data
  - Semi-supervised Translation (e.g. back-translation)
  - Auxiliary losses (e.g. reconstruction loss, additional language modeling)
- Zero-resource Translation (i.e. no bitext)
  - Previous work still need labels of SOME kind
  - Deciphering problem (Pourdamghani & Knight 2017)
    - Scalability to longer sequence never tried
    - More specific for closely related languages

Introduction: Research Question

- Can we train a translation systems WITHOUT bitext?
- Motivations:
  1. Applicable for ANY language pair we have monolingual data
  2. Gives lower bound: Performance(Unsupervised) ≤ Performance(Supervised)

Introduction: Approach

- Build a common latent space for both languages
  - Think aligning word embeddings from 2 languages
- Reconstruct outputs confined with 2 objectives
  1. Given sentence X and X_{noisy} :
     - $f(X_{noisy}) = X$
     - Denoising Auto-Encoder objective
  2. Given sentence X and Y, where X & Y are 2 different domains (e.g. English and French):
     - $f(Y_{noisy}) = X$
     - Reconstruct sentence from the target domain
       - achieved with back translation

Introduction: Approach Visualization

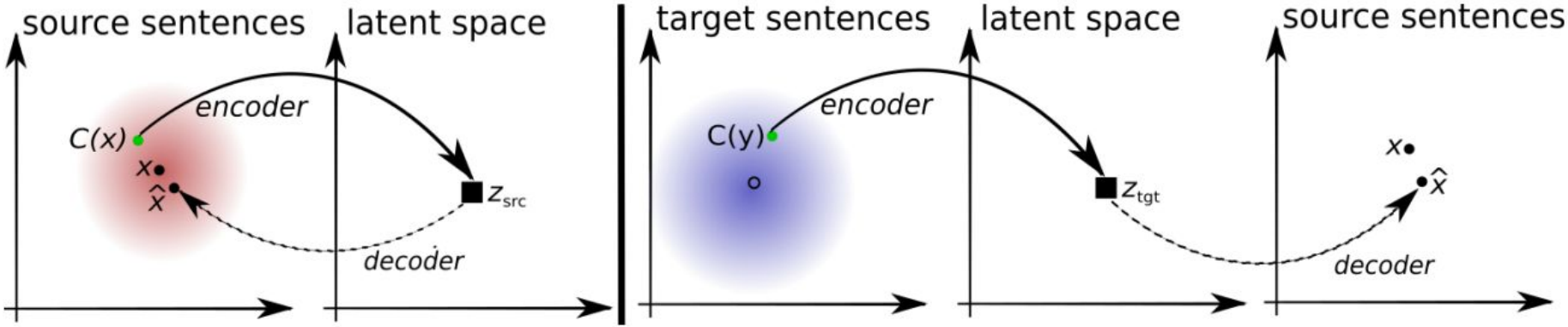


Figure 1: Toy illustration of the principles guiding the design of our objective function. Left (auto-encoding): the model is trained to reconstruct a sentence from a noisy version of it. x is the target, $C(x)$ is the noisy input, $\hat{x}$ is the reconstruction. Right (translation): the model is trained to translate a sentence in the other domain. The input is a noisy translation (in this case, from source-to-target) produced by the model itself, M, at the previous iteration (t), $y = M^{(t)}(x)$. The model is symmetric, and we repeat the same process in the other language. See text for more details.

UNMT: NMT Model

- Decoder - Encoder Model
  - All parameters θ are shared except language embedding matrices
- Encoder: $e(\mathbf{x}, L) = \mathbf{Z} = [Z_1, \ldots, Z_n]$
  - **Z** is the hidden state at each step of encoding
  - L tells encoder which embedding's to use (W_{emb}^{L1} or W_{emb}^{L2})
- Decoder: $d(\mathbf{z}, L') = \mathbf{Y} = [Y_0, Y_1, \ldots, Y_m]$
  - Y_0 is language dependent start token
  - L' is target language we are translating to and is passed as input to LSTM

UNMT: Aside, Bahdanau et al 2015

- We generate matrix **Z** of hidden states **Z** = (z_1 , z_2 …, z_n)
- Hidden states progressively lose info in sequence
- Solution: Generate a weighted sum of **Z** (context)
  - Picture is general idea
  - Equations for it are below
    - a : is a matrix of weights
    - s_i : previous hidden decoder state
    - h_j : encoder hidden state j
    - f,g : RNN's and other parameters
  - FYI: pytorch has a tutorial with this model

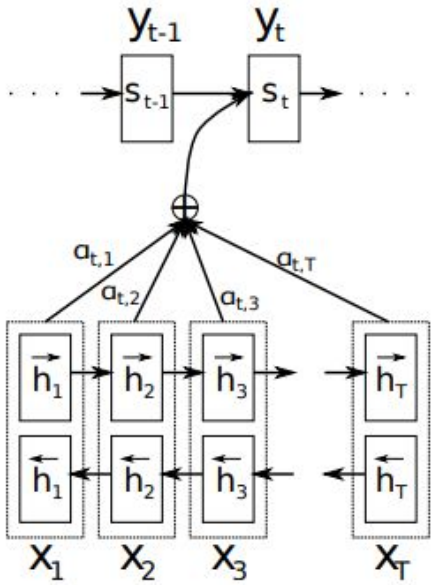
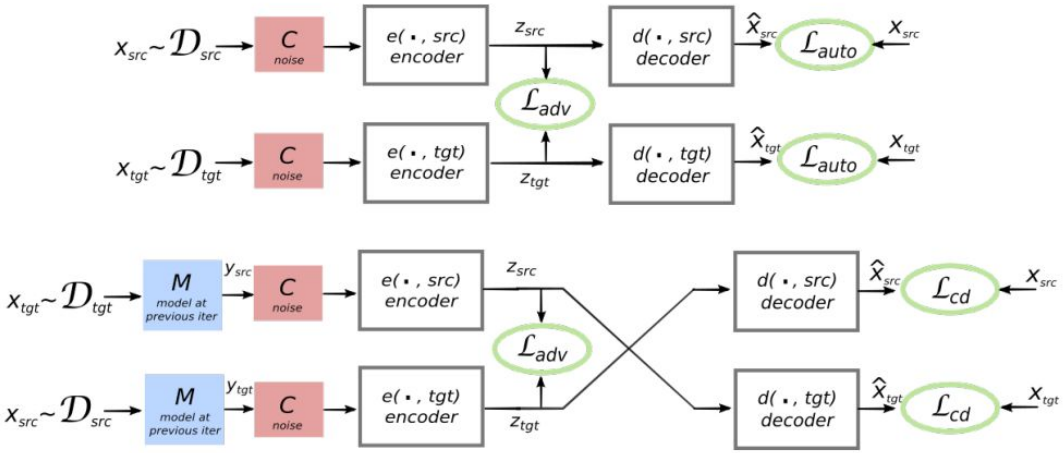


Figure 1: The graphical illustration of the proposed model trying to generate the t-th target word y_t given a source sentence $(x_1, x_2, \ldots, x_T)$.

$$e_{ij} = a(s_{i-1}, h_j) \qquad c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j \qquad s_i = f(s_{i-1}, y_{i-1}, c_i)$$

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})} \qquad p(y_i | y_1, \ldots, y_{i-1}, \mathbf{x}) = g(y_{i-1}, s_i, c_i)$$

UNMT: Overview of Method



Denoising Auto-encoding

Translation Denoising Auto-encoding

UNMT: Denoising Auto-Encoding

- Auto-encoding is too easy
  - Even a random permutation of words can be perfectly reconstructed
- Noise Model C:
  - Drop words with probability p_{wd}
  - Word Shuffled:
    - Condition $\forall i$ in {1, n}, $|\sigma(i) - i| \leq k$
      - i : indexes of sequence of length n
      - k : hyperparameter to of how noisy to make it
    - σ sorts the sequence based on permutation model q
      - $q_i = i + U(0, \alpha)$, α: hyperparameter (k + 1 in paper)
      - σ sorts sequence q and accepts based on condition
      - Is biased sample
  - E.g. with n = 5, k = 3, α = 4:
    - [0, 1, 2, 3, 4] ⇒ [0.885, 3.659, 5.791, 6.933, 5.172] ⇒ sort⇒ [0, 1, 4, 2, 3]

$$\mathcal{L}_{auto}(\theta_{\text{enc}}, \theta_{\text{dec}}, \mathcal{Z}, \ell) = \mathbb{E}_{x \sim \mathcal{D}_\ell, \hat{x} \sim d(e(C(x),\ell),\ell)} \left[\Delta(\hat{x}, x)\right]$$

UNMT: Cross Domain Training

- The thing we actually care about is $x_{src} \Rightarrow y_{tgt}$
- Generate bitext with back translation from model M^t
  - $M^t(x_{src}) = y_{tgt}^{translation}$
- Apply noise model C to y_{tgt} and decode back to x_{src}
  - $d(e(C(y_{tgtt}^{translation}), L_{tgt}) , L_{src}) = x_{src}$ (hopefully)

UNMT: Cross Domain Training

- Here's the loss:
  $\mathcal{L}_{cd}(\theta_{\text{enc}}, \theta_{\text{dec}}, \mathcal{Z}, \ell_1, \ell_2) = \mathbb{E}_{x \sim \mathcal{D}_{\ell_1}, \hat{x} \sim d(e(C(M(x)), \ell_2), \ell_1)} [\Delta(\hat{x}, x)]$
  Δ (a, b) is token level cross entropy

- Example:

  M^t(Michael is the best) = مايكل هو الأفضل

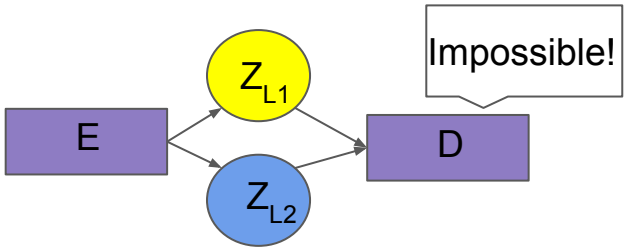
  C(مايكل هو الأفضل) = هوأ الأفلضما

  M^t(هوأ الأفلضما) = Is the flap

  Calculate Δ(Michael is the best, Is the flap)

UNMT: Adversarial Training

- Remember that we are translating bidirectionally
  - L_1 ←Model→ L2
- We need latent space z to be same for both directions
  - Otherwise you need 2 separate models



UNMT: Adversarial Training

- Enforce this with adversarial training:
  - Discriminator must determine which language we encoded from

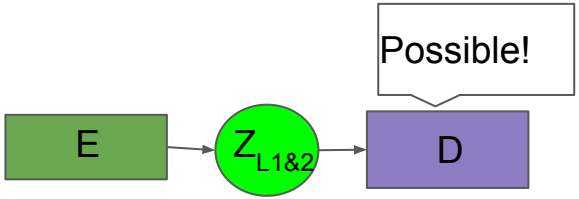
$$\mathcal{L}_D(\theta_D|\theta,\mathcal{Z}) = -\mathbb{E}_{(x_i,\ell_i)}[\log p_D(\ell_i|e(x_i,\ell_i))]$$

Predict correct language

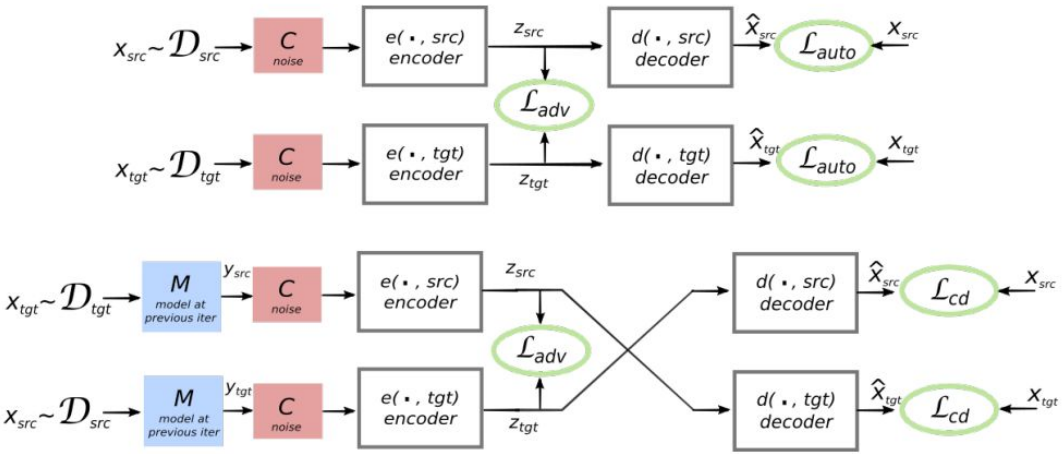
  - Encoder must fool discriminator which forces **z** to look similar in both directions

$$\mathcal{L}_{adv}(\theta_{\text{enc}},\mathcal{Z}|\theta_D) = -\mathbb{E}_{(x_i,\ell_i)}[\log p_D(\ell_j|e(x_i,\ell_i))]$$

Fool D to predict j instead of i

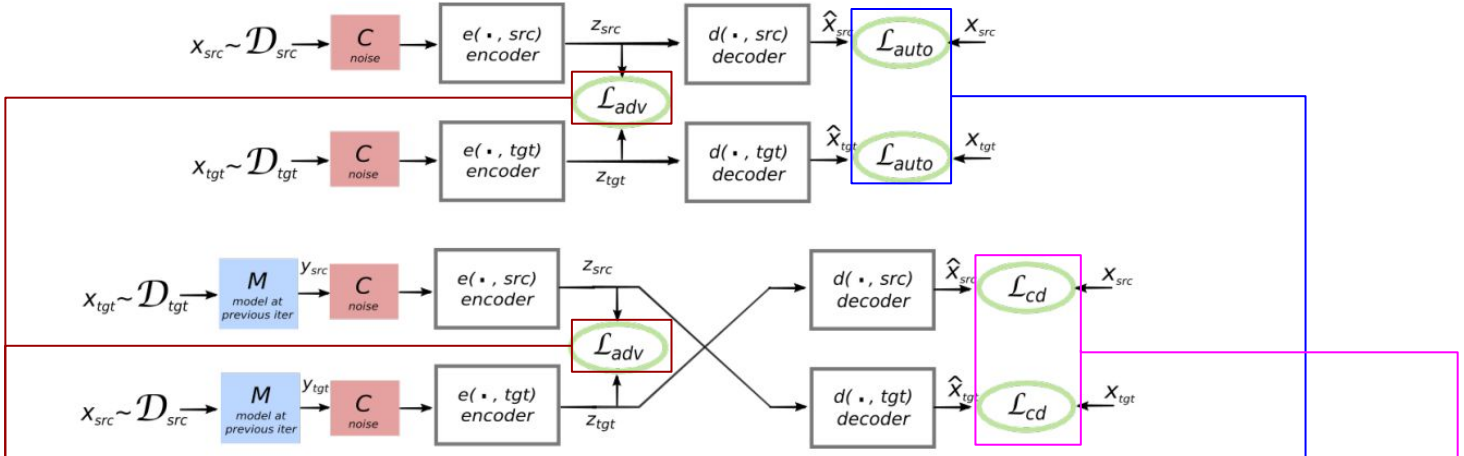


UNMT: Final Objective



$$\begin{aligned}\mathcal{L}(\theta_{\text{enc}}, \theta_{\text{dec}}, \mathcal{Z}) =& \lambda_{auto}[\mathcal{L}_{auto}(\theta_{\text{enc}}, \theta_{\text{dec}}, \mathcal{Z}, src) + \mathcal{L}_{auto}(\theta_{\text{enc}}, \theta_{\text{dec}}, \mathcal{Z}, tgt)] + \\ & \lambda_{cd}[\mathcal{L}_{cd}(\theta_{\text{enc}}, \theta_{\text{dec}}, \mathcal{Z}, src, tgt) + \mathcal{L}_{cd}(\theta_{\text{enc}}, \theta_{\text{dec}}, \mathcal{Z}, tgt, src)] + \\ & \lambda_{adv}\mathcal{L}_{adv}(\theta_{\text{enc}}, \mathcal{Z} | \theta_D)\end{aligned}$$

UNMT: Final Objective



$$\mathcal{L}(\theta_{\text{enc}}, \theta_{\text{dec}}, \mathcal{Z}) = \lambda_{auto}[\mathcal{L}_{auto}(\theta_{\text{enc}}, \theta_{\text{dec}}, \mathcal{Z}, src) + \mathcal{L}_{auto}(\theta_{\text{enc}}, \theta_{\text{dec}}, \mathcal{Z}, tgt)] +$$
$$\lambda_{cd}[\mathcal{L}_{cd}(\theta_{\text{enc}}, \theta_{\text{dec}}, \mathcal{Z}, src, tgt) + \mathcal{L}_{cd}(\theta_{\text{enc}}, \theta_{\text{dec}}, \mathcal{Z}, tgt, src)] +$$
$$\lambda_{adv}\mathcal{L}_{adv}(\theta_{\text{enc}}, \mathcal{Z}|\theta_D)$$

Training: Iterative Training

Algorithm 1 Unsupervised Training for Machine Translation

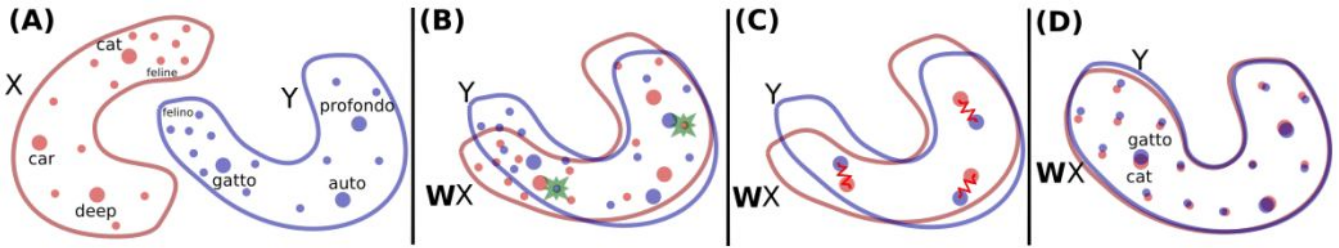
procedure TRAINING($\mathcal{D}_{src}, \mathcal{D}_{tgt}, T$)
    Infer bilingual dictionary using monolingual data (Conneau et al., 2017)
    $M^{(1)} \leftarrow$ unsupervised word-by-word translation model using the inferred dictionary
    for $t = 1, T$ **do**
        using $M^{(t)}$, translate each monolingual dataset
        // discriminator training & model training as in eq. 4
        $\theta_{\text{discr}} \leftarrow \arg\min \mathcal{L}_D, \quad \theta_{\text{enc}}, \theta_{\text{dec}}, \mathcal{Z} \leftarrow \arg\min \mathcal{L}$
        $M^{(t+1)} \leftarrow e^{(t)} \circ d^{(t)}$ // update MT model
    end for
    return $M^{(T+1)}$
end procedure

Algorithm 1 Unsupervised Training for Machine Translation

procedure TRAINING($\mathcal{D}_{src}, \mathcal{D}_{tgt}, T$)
  Infer bilingual dictionary using monolingual data (Conneau et al., 2017)
  $M^{(1)} \leftarrow$ unsupervised word-by-word translation model using the inferred dictionary
  for $t = 1, T$ **do**
    using $M^{(t)}$, translate each monolingual dataset
    // discriminator training & model training as in eq. 4
    $\theta_{\text{discr}} \leftarrow \arg\min \mathcal{L}_D, \quad \theta_{\text{enc}}, \theta_{\text{dec}}, \mathcal{Z} \leftarrow \arg\min \mathcal{L}$
    $M^{(t+1)} \leftarrow e^{(t)} \circ d^{(t)}$ // update MT model
  end for
  return $M^{(T+1)}$
end procedure

Step 1:

We need to align our words to approximate a word to word dictionary (alternatively, have a dictionary of word level translations)



Algorithm 1 Unsupervised Training for Machine Translation

procedure TRAINING($\mathcal{D}_{src}, \mathcal{D}_{tgt}, T$)
  Infer bilingual dictionary using monolingual data (Conneau et al., 2017)
  $M^{(1)} \leftarrow$ unsupervised word-by-word translation model using the inferred dictionary
  for $t = 1, T$ **do**
    using $M^{(t)}$, translate each monolingual dataset
    // discriminator training & model training as in eq. 4
    $\theta_{\text{discr}} \leftarrow \arg\min \mathcal{L}_D, \quad \theta_{\text{enc}}, \theta_{\text{dec}}, \mathcal{Z} \leftarrow \arg\min \mathcal{L}$
    $M^{(t+1)} \leftarrow e^{(t)} \circ d^{(t)}$ // update MT model
  end for
  return $M^{(T+1)}$
end procedure

Step 2:

Initialize our encoder-decoder with word level translation.

Suspected Pseudo-code (probably wrong):

For e in epochs:

For each word pair in our dictionary:

$e(x_{word,}, L_1) \Rightarrow z \Rightarrow d(y_{word}, L_2) \Rightarrow$ *minimize Cross entropy*

Algorithm 1 Unsupervised Training for Machine Translation

procedure TRAINING($\mathcal{D}_{src}, \mathcal{D}_{tgt}, T$)
  Infer bilingual dictionary using monolingual data (Conneau et al., 2017)
  $M^{(1)} \leftarrow$ unsupervised word-by-word translation model using the inferred dictionary
  for $t = 1, T$ **do**
    using $M^{(t)}$, translate each monolingual dataset
    // discriminator training & model training as in eq. 4
    $\theta_{\text{discr}} \leftarrow \arg\min \mathcal{L}_D, \quad \theta_{\text{enc}}, \theta_{\text{dec}}, \mathcal{Z} \leftarrow \arg\min \mathcal{L}$
    $M^{(t+1)} \leftarrow e^{(t)} \circ d^{(t)}$ // update MT model
  end for
  return $M^{(T+1)}$
end procedure

Step 3:

Generate our synthetic bitext (backtranslate) with current model M

Example:

$M^{(t)}$(Michael is a the best) = مايكل هو الأفضل

At training time, we will then do the following as part of updating:

e(مايكل هو الأفضل, Arabic) ⇒ z ⇒ d(z,english) ⇒ Michael are a good guy ⇒ <next slide>

Algorithm 1 Unsupervised Training for Machine Translation

procedure TRAINING($\mathcal{D}_{src}, \mathcal{D}_{tgt}, T$)
  Infer bilingual dictionary using monolingual data (Conneau et al., 2017)
  $M^{(1)} \leftarrow$ unsupervised word-by-word translation model using the inferred dictionary
  for $t = 1, T$ **do**
    using $M^{(t)}$, translate each monolingual dataset
    // discriminator training & model training as in eq. 4
    $\theta_{\text{discr}} \leftarrow \arg\min \mathcal{L}_D, \quad \theta_{\text{enc}}, \theta_{\text{dec}}, \mathcal{Z} \leftarrow \arg\min \mathcal{L}$
    $M^{(t+1)} \leftarrow e^{(t)} \circ d^{(t)}$ // update MT model
  end for
  return $M^{(T+1)}$
end procedure

Step 3:

Minimize our loss functions from before

$$\begin{aligned}\mathcal{L}(\theta_{\text{enc}}, \theta_{\text{dec}}, \mathcal{Z}) = &\lambda_{auto}[\mathcal{L}_{auto}(\theta_{\text{enc}}, \theta_{\text{dec}}, \mathcal{Z}, src) + \mathcal{L}_{auto}(\theta_{\text{enc}}, \theta_{\text{dec}}, \mathcal{Z}, tgt)] + \\ &\lambda_{cd}[\mathcal{L}_{cd}(\theta_{\text{enc}}, \theta_{\text{dec}}, \mathcal{Z}, src, tgt) + \mathcal{L}_{cd}(\theta_{\text{enc}}, \theta_{\text{dec}}, \mathcal{Z}, tgt, src)] + \\ &\lambda_{adv}\mathcal{L}_{adv}(\theta_{\text{enc}}, \mathcal{Z} | \theta_D)\end{aligned}$$

Algorithm 1 Unsupervised Training for Machine Translation

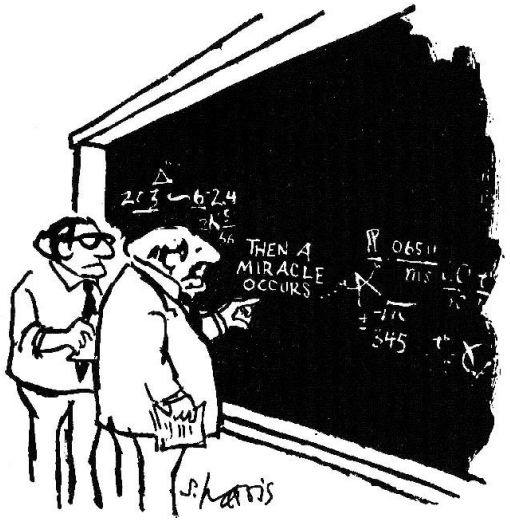
procedure TRAINING($\mathcal{D}_{src}, \mathcal{D}_{tgt}, T$)
  Infer bilingual dictionary using monolingual data (Conneau et al., 2017)
  $M^{(1)} \leftarrow$ unsupervised word-by-word translation model using the inferred dictionary
  for $t = 1, T$ **do**
    using $M^{(t)}$, translate each monolingual dataset
    // discriminator training & model training as in eq. 4
    $\theta_{\text{discr}} \leftarrow \arg\min \mathcal{L}_D, \quad \theta_{\text{enc}}, \theta_{\text{dec}}, \mathcal{Z} \leftarrow \arg\min \mathcal{L}$
    $M^{(t+1)} \leftarrow e^{(t)} \circ d^{(t)}$ // update MT model
  end for
  return $M^{(T+1)}$
end procedure

Step 4:

Update our model and repeat the process a bunch of times

Training: Iterative Training

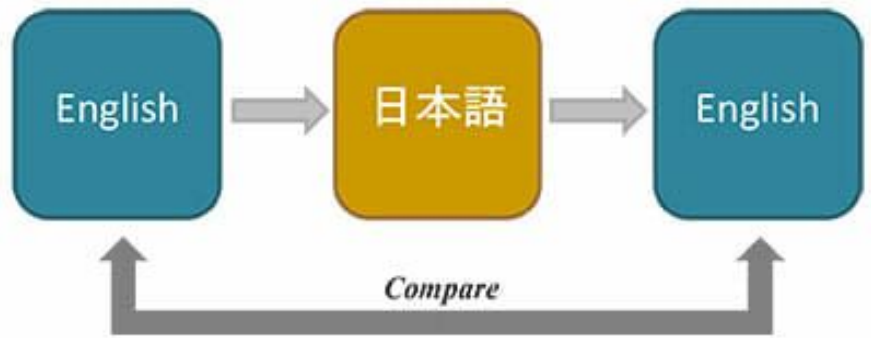
- Why would this work?
  - There isn't actually any rigorous proof that this should work
- Practical explanation is empirically it works
- Intuitive explanation:
  - As long as $M^{(1)}$ contains some semblance of language structure, should be ok
  - Most of our objectives about are cleaning noisy data
    - Our model at least should predict cleaned outputs



"I think you should be more explicit here in step two."

Training: Unsupervised Model Selection Criterion

- For the task, there is 0 parallel data
  - Are learning anything?
- Do actual back translation
  - Translate from $L_1 \Rightarrow L_2 \Rightarrow L_1^{noise}$
  - Calculate Bleu between L_1 & L_1^{noise}
  - Do for L_2 and average them
- Criteria usage:
  - Stopping Training
  - Hyperparameter tuning

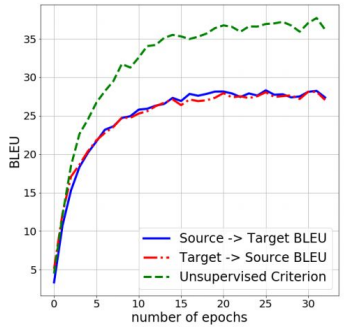


Training: Unsupervised Model Selection Criterion

- The Equation

$$MS(e, d, \mathcal{D}_{src}, \mathcal{D}_{tgt}) = \frac{1}{2}\mathbb{E}_{x\sim\mathcal{D}_{src}}\left[\text{BLEU}(x, M_{src\to tgt} \circ M_{tgt\to src}(x))\right] + \frac{1}{2}\mathbb{E}_{x\sim\mathcal{D}_{tgt}}\left[\text{BLEU}(x, M_{tgt\to src} \circ M_{src\to tgt}(x))\right]$$

- Is criterion actually correlated to actual BLEU?
  - For Stopping: Mostly Yes (Spearman correlation coefficient = .95 between test set BLEU)
  - For Hyperparameter tuning: somewhat yes (SPC = .75 between test set BLEU)



Experiments: Baselines vs UNMT

	Multi30k-Task1				WMT			
	en-fr	fr-en	de-en	en-de	en-fr	fr-en	de-en	en-de
Supervised	56.83	50.77	38.38	35.16	27.97	26.13	25.61	21.33
word-by-word	8.54	16.77	15.72	5.39	6.28	10.09	10.77	7.06
word reordering	-	-	-	-	6.68	11.69	10.84	6.70
oracle word reordering	11.62	24.88	18.27	6.79	10.12	20.64	19.42	11.57
Our model: 1st iteration	27.48	28.07	23.69	19.32	12.10	11.79	11.10	8.86
Our model: 2nd iteration	31.72	30.49	24.73	21.16	14.42	13.49	13.25	9.75
Our model: 3rd iteration	32.76	32.07	26.26	22.74	15.05	14.31	13.33	9.64

Table 2: **BLEU score on the Multi30k-Task1 and WMT datasets** using greedy decoding.

Experiments: Unsupervised vs Supervised NMT

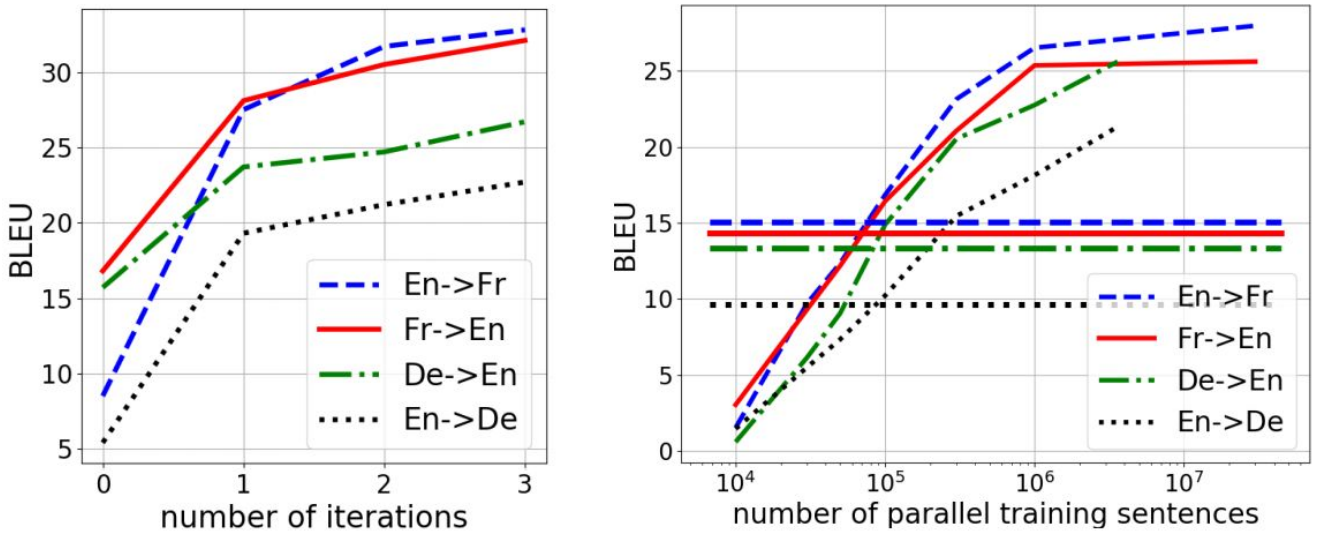


Figure 4: Left: BLEU as a function of the number of iterations of our algorithm on the Multi30k-Task1 datasets. Right: The curves show BLEU as a function of the amount of parallel data on WMT datasets. The unsupervised method which leverages about 15 million monolingual sentences in each language, achieves performance (see horizontal lines) close to what we would obtain by employing 100,000 parallel sentences.

Experiments: Translation Examples

Source	un homme est debout près d' une série de jeux vidéo dans un bar .
Iteration 0	a man is seated near a series of games video in a bar .
Iteration 1	a man is standing near a closeup of other games in a bar .
Iteration 2	a man is standing near a bunch of video video game in a bar .
Iteration 3	a man is standing near a bunch of video games in a bar .
Reference	**a man is standing by a group of video games in a bar .**
Source	une femme aux cheveux roses habillée en noir parle à un homme .
Iteration 0	a woman at hair roses dressed in black speaks to a man .
Iteration 1	a woman at glasses dressed in black talking to a man .
Iteration 2	a woman at pink hair dressed in black speaks to a man .
Iteration 3	a woman with pink hair dressed in black is talking to a man .
Reference	**a woman with pink hair dressed in black talks to a man .**
Source	une photo d' une rue bondée en ville .
Iteration 0	a photo a street crowded in city .
Iteration 1	a picture of a street crowded in a city .
Iteration 2	a picture of a crowded city street .
Iteration 3	a picture of a crowded street in a city .
Reference	**a view of a crowded city street .**

Experiments: Ablation Study

	en-fr	fr-en	de-en	en-de
$\lambda_{cd} = 0$	25.44	27.14	20.56	14.42
Without pretraining	25.29	26.10	21.44	17.23
Without pretraining, $\lambda_{cd} = 0$	8.78	9.15	7.52	6.24
Without noise, $C(x) = x$	16.76	16.85	16.85	14.61
$\lambda_{auto} = 0$	24.32	20.02	19.10	14.74
$\lambda_{adv} = 0$	24.12	22.74	19.87	15.13
Full	**27.48**	**28.07**	**23.69**	**19.32**

Table 4: **Ablation study on the Multi30k-Task1 dataset.**

Conclusion

- The approach works modestly well
  - Can work better compared to low-resource bitext system