FALL 2020: 9/10/2020

TEXAS
The University of Texas at Austin

Unsupervised Generation

CS 395T: Topics in Natural Language Processing

Mao Ye, Akshay Kumar Gupta, The University of Texas at Austin

Similarity:

Obtain a conditional generative model → key structures are similar

Consider unsupervised method

Dissimilarity:

The key idea to unsupervised training is different:

1. Reconstruction error + latent space alignment
2. Assign new task for training (masked key words refill)

Unsupervised Machine Translation Using Monolingual Corpora Only. (Lample et al, 2018)

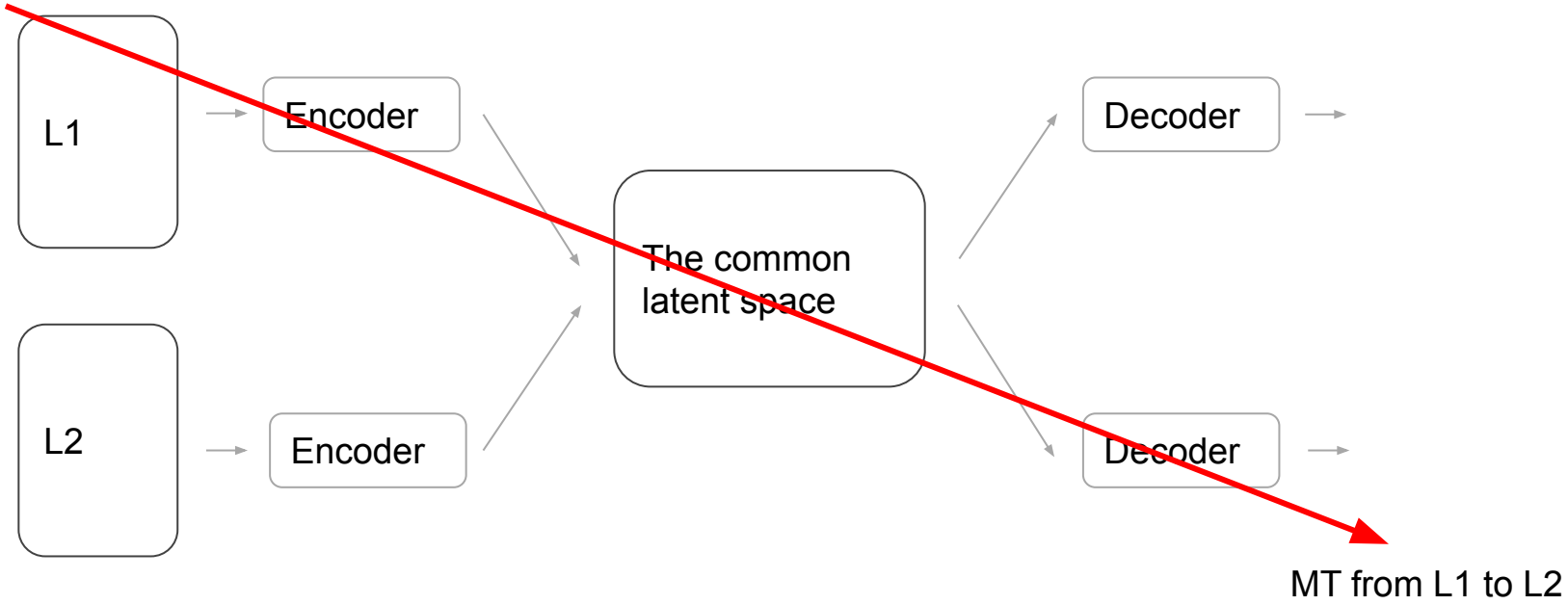
MT is improving thanks to large-scale parallel corpora but it's hard to get thousands of parallel sentences for low-resource language pairs

- Can we learn to translate without parallel sentence data?

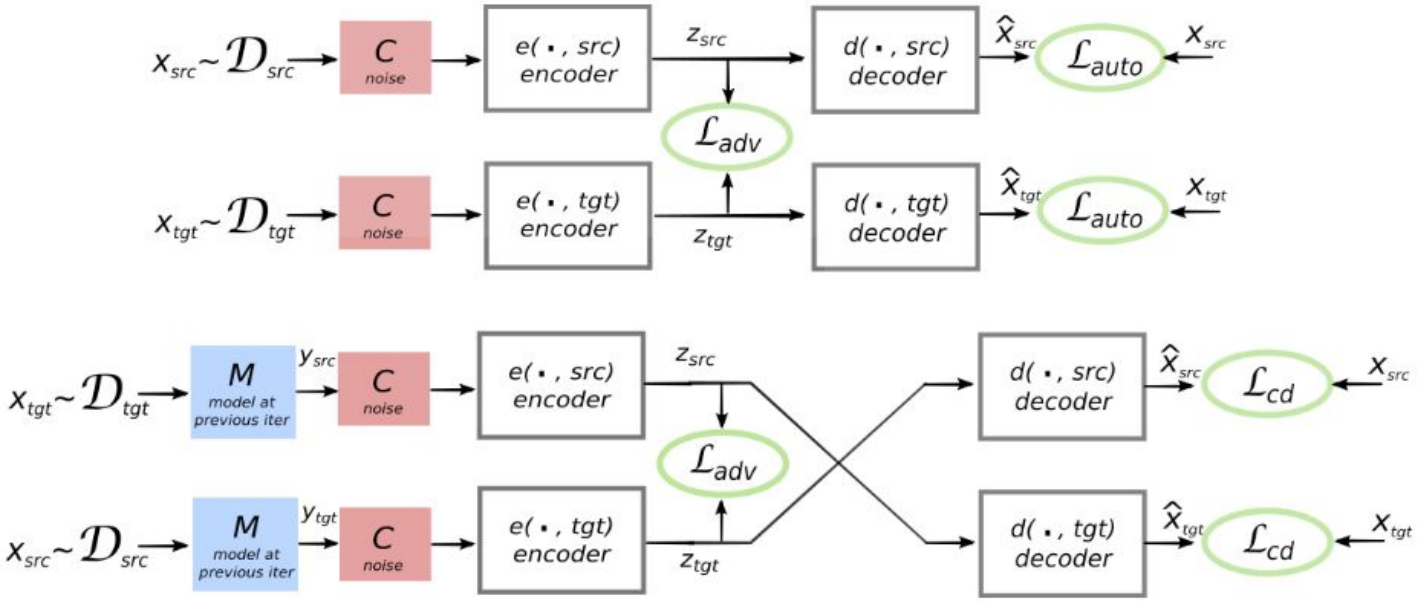
System Overview

- Build encode and decode system such that the latent spaces common for both source and target language.

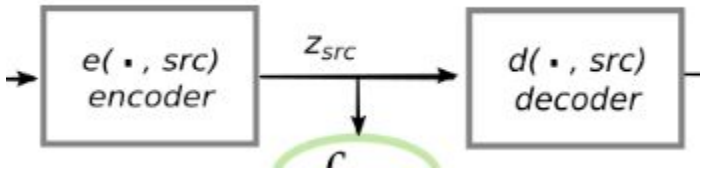
System Overview



System Overview

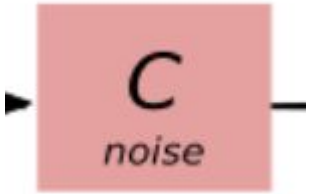


System Overview



Use seq2seq model with attention.

System Overview

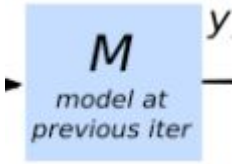


C is a random sampled noise version of sentence.

- Drop a word in a sentence in a certain probability
- Slightly shuffle the input sentence

Why? Better quality supported by ablation study (regularization technique)

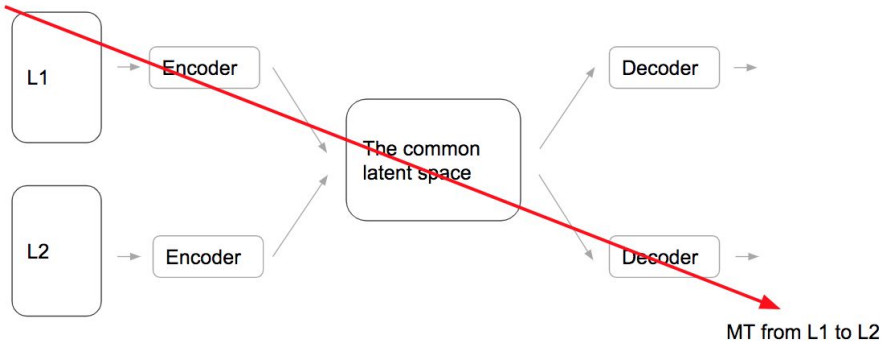
System Overview



Translation model at previous iteration.

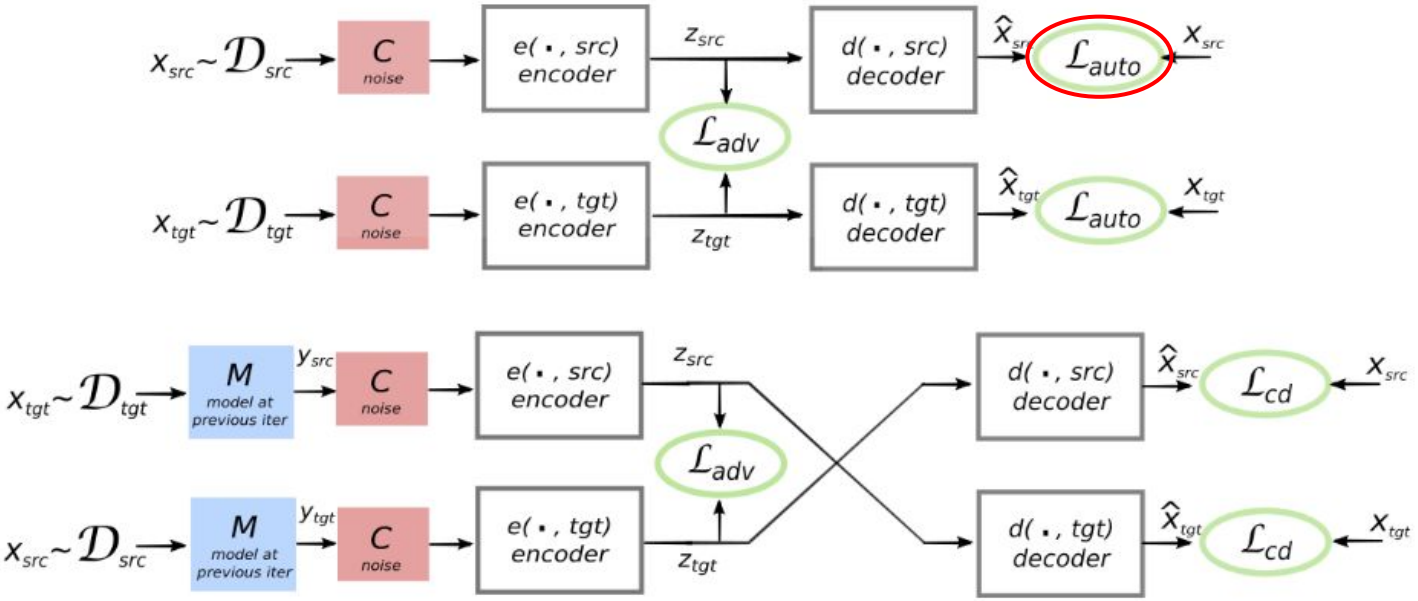
For iteration = 0, use naive word-by-word translation

For iteration > 0:



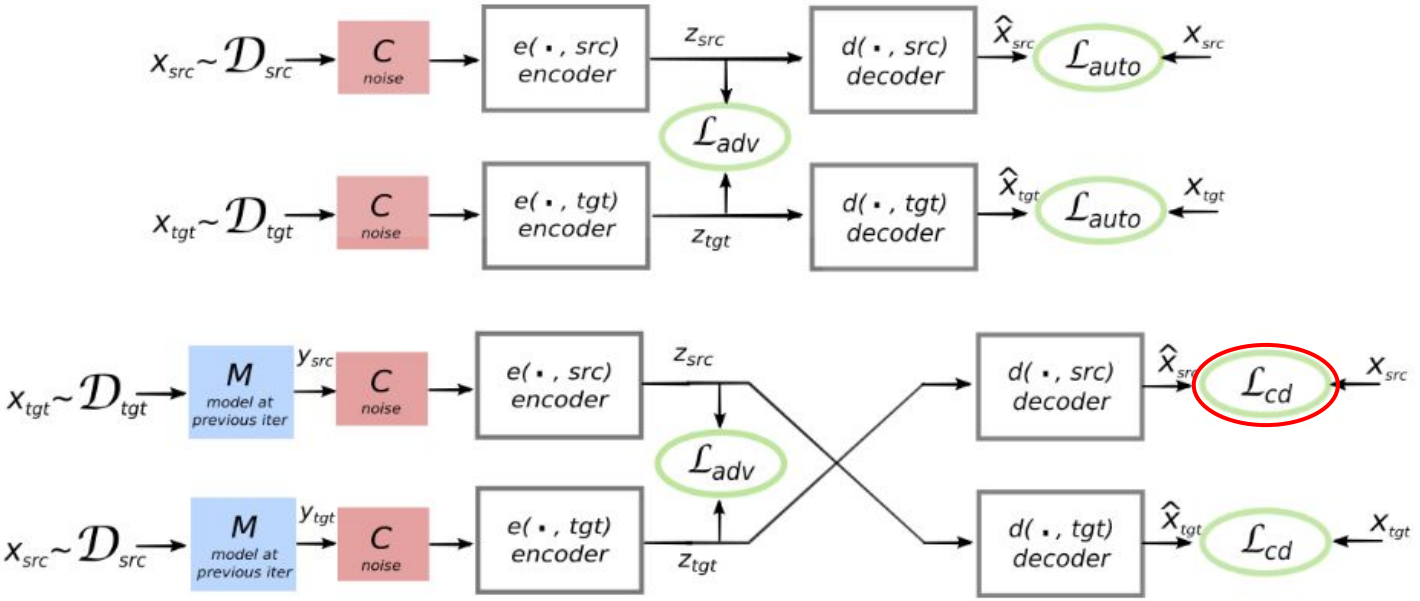
System Overview

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

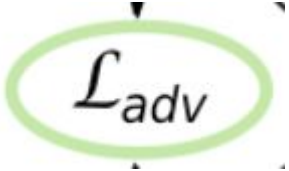


System Overview

$$\mathcal{L}_{cd}(\theta_{\text{enc}}, \theta_{\text{dec}}, \mathcal{Z}, \ell_1, \ell_2) = \mathbb{E}_{x \sim \mathcal{D}_{\ell_1}, \hat{x} \sim d(e(C(M(x)), \ell_2), \ell_1)} [\Delta(\hat{x}, x)]$$



System Overview



Use adversarial training to ensure the latent space are common for both language.

Maintain a network that tries to classify the encoded sentence comes from which language.

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

- **Discriminator Architecture:** Multilayer perceptron with 3 hidden layers of size 1024, Leaky-ReLU and output logistic unit.

System Overview

Final Objective function The final objective function at one iteration of our learning algorithm is thus:

$$\begin{aligned}\mathcal{L}(\theta_{\text{enc}}, \theta_{\text{dec}}, \mathcal{Z}) =& \lambda_{auto}[\mathcal{L}_{auto}(\theta_{\text{enc}}, \theta_{\text{dec}}, \mathcal{Z}, src) + \mathcal{L}_{auto}(\theta_{\text{enc}}, \theta_{\text{dec}}, \mathcal{Z}, tgt)] + \\ & \lambda_{cd}[\mathcal{L}_{cd}(\theta_{\text{enc}}, \theta_{\text{dec}}, \mathcal{Z}, src, tgt) + \mathcal{L}_{cd}(\theta_{\text{enc}}, \theta_{\text{dec}}, \mathcal{Z}, tgt, src)] + \\ & \lambda_{adv}\mathcal{L}_{adv}(\theta_{\text{enc}}, \mathcal{Z}|\theta_D)\end{aligned} \tag{4}$$

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

Parameter/Model Selection

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

Parameter/Model Selection

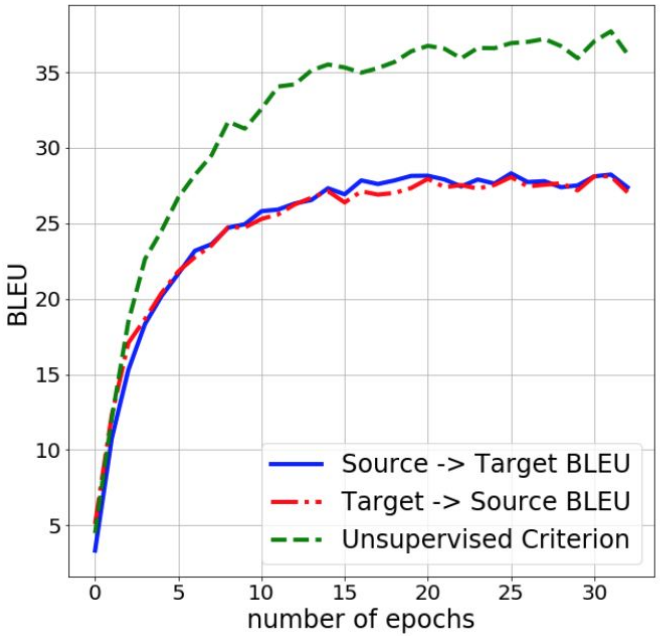


Figure 3: **Unsupervised model selection.** BLEU score of the source to target and target to source models on the Multi30k-Task1 English-French dataset as a function of the number of passes through the dataset at iteration $(t) = 1$ of the algorithm (training $M(2)$ given $M(1)$). BLEU correlates very well with the proposed model selection criterion, see Equation 5.

Datasets

- **WMT'14 English-French:** 36m pairs, reduced to 30m and split for EN/FR sentences. Validation on 3,000 sentences. Test on *newstest2014*.
- **WMT'16 English-German:** Same as above, 1.8m training sentences each. Testing on *newstest2016*.
- **Multi30k-Task1:** 30k image annotations (EN, FR, DE), 29/1/1 split. Split for monolingual corpora (14.5/0.5/0.5k).

	MMT1 en-fr	MMT1 de-en	WMT en-fr	WMT de-en
Monolingual sentences	14.5k	14.5k	15M	1.8M
Vocabulary size	10k / 11k	19k / 10k	67k / 78k	80k / 46k

Table 1: **Multi30k-Task1 and WMT datasets statistics.** To limit the vocabulary size in the WMT en-fr and WMT de-en datasets, we only considered words with more than 100 and 25 occurrences, respectively.

Baselines

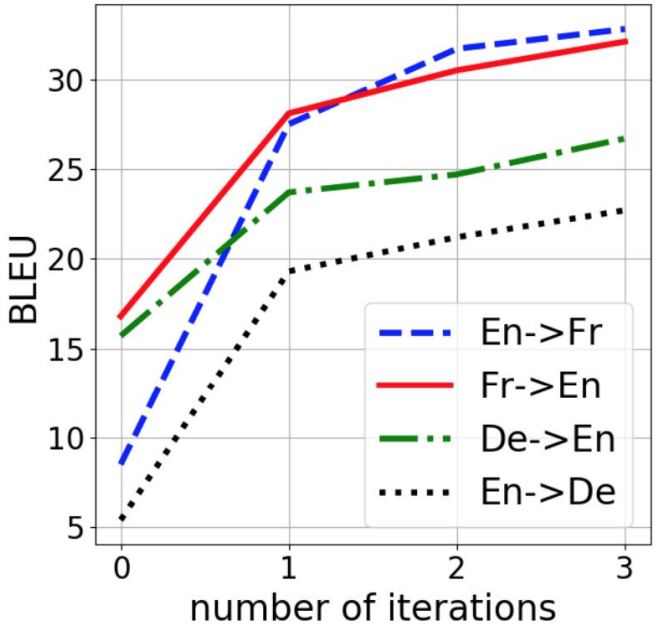
- **Word-by-word translation (WBW):** Using an inferred bilingual dictionary
- **Word reordering (WR):** After WBW, use LSTM-based LM trained on the target side
- **Oracle Word Reordering (OWR):** Using reference, best possible generation using only the words given by WBW
- **Supervised Learning:** Same model trained using the standard cross-entropy loss on the original parallel sentences

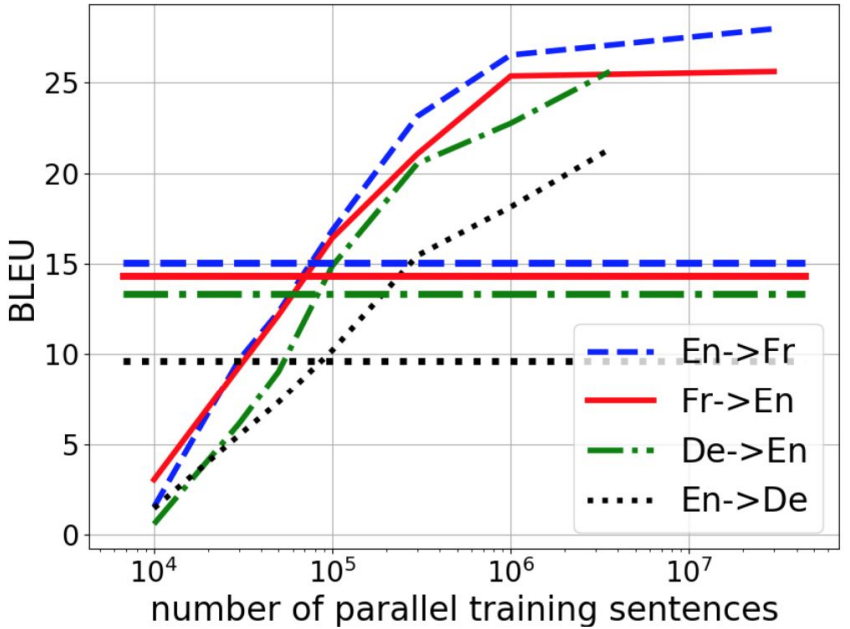
Results

	Multi30k-Task1				WMT			
	en-fr	fr-en	de-en	en-de	en-fr	fr-en	de-en	en-de
upervised	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.

Results


BLEU
30
25
20
15
10
5
0
1
2
3
number of iterations
En->Fr
Fr->En
De->En
En->De


BLEU
25
20
15
10
5
0
10^4
10^5
10^6
10^7
number of parallel training sentences
En->Fr
Fr->En
De->En
En->De

Sample Output

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 .**

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

- **Approach:** Model is learned using monolingual datasets only, with no word/sentence level alignment

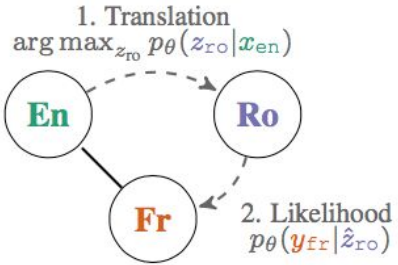
- **Principle:**
  - Start from unsupervised W2W MT model
  - Iteratively improve using reconstruction loss
  - Use discriminator to align latent distributions of src/tgt languages

- **Results:** Can effectively learn translation models without any supervision!

Conclusion

Some future extensions:

1. **Phrase-Based & Neural Unsupervised Machine Translation:** use shared parameter to align laten space; use phrase table to better mapping words in two languages.
2. **A Multilingual View of Unsupervised Machine Translation : Multilingual setting**



(b) Cross-translation

The Summary Loop: Learning to Write Abstractive Summaries Without Examples. (Laban et al, ACL 2020)

Task: Given a source document and length constraint, generate a summary of the source document that satisfies the length constraint. (Unsupervised method)

The Summary Loop: Learning to Write Abstractive Summaries Without Examples. (Laban et al, ACL 2020)

Requirement:

- **Coverage** of the key words
- **Fluency** of the generated language
- **Brevity** of generated summaries

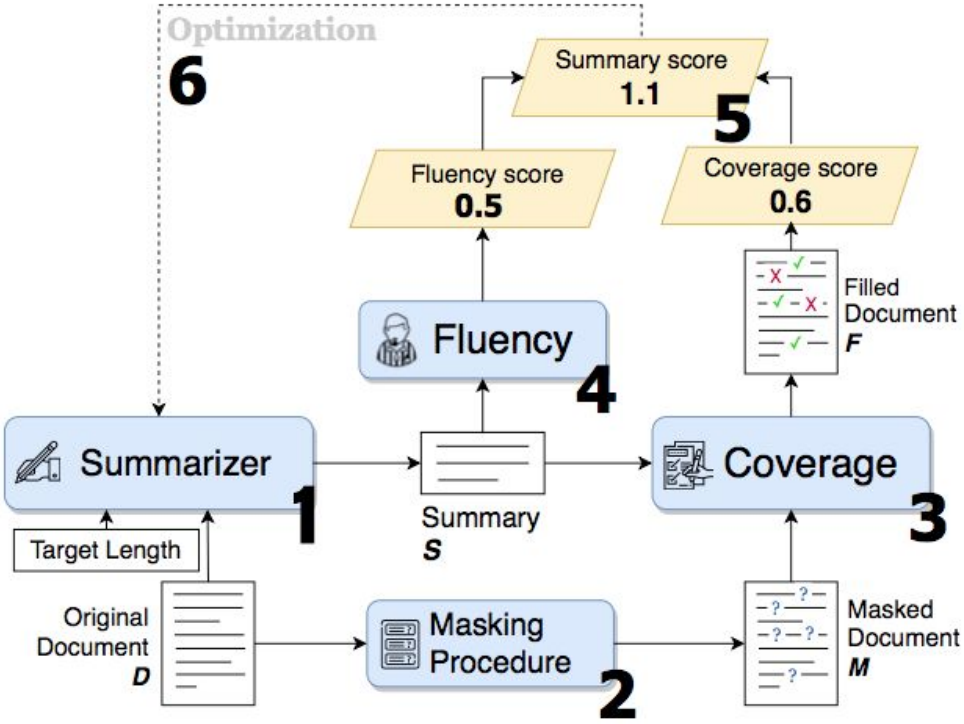
Previous work

1. Extractive method: deleting uninformative words
2. Seq2seq model: usually supervised and thus requires paired examples
3. Miao and Blunsom (2016) train seperate en/decoder for supervised and unsupervised examples (still requires a lot paired examples)

Main Contribution

- Using coverage model to ensure a informative summary
- Unsupervised training procedure
- Specialized training technique for preventing pathological behavior during generating

The proposed Summary Loop



The architectures for each components

Summarizer

Architecture:

Generative Transformer (Radford et al., 2019) as the model architecture of the summarizer.

- Produce one word at a time
- Feed the produced word into the model for the next one
- Repeat until length constraint.

Masking Procedure

Select the top k words with highest tf-idf score and mask them.

t: text

d: document

D: all documents

tf-idf (t,d) = tf(t,d) * idf(t,D)

Masking Procedure

tf(t,d) measures whether text t is ‘special’ to document d

Variants of term frequency (tf) weight

weighting scheme	tf weight
binary	$0, 1$
raw count	$f_{t,d}$
term frequency	$f_{t,d} \Big/ \sum_{t' \in d} f_{t',d}$
log normalization	$\log(1 + f_{t,d})$
double normalization 0.5	$0.5 + 0.5 \cdot \frac{f_{t,d}}{\max_{\{t' \in d\}} f_{t',d}}$
double normalization K	$K + (1 - K)\frac{f_{t,d}}{\max_{\{t' \in d\}} f_{t',d}}$

Coverage Model

idf(t,d) measures whether text t is a ‘general common’ word

$$\operatorname{idf}(t, D) = \log \frac{N}{|\{d \in D : t \in d\}|}$$

Coverage Model

Receives a computationally generated summary and the masked document and attempts to fill in each blank word.

Different from masked language modeling (MLM): summary is given while MLM only use other unmasked info to fill the mask

Use same BERT-like architecture for MLM in (Devlin et al., 2019) .

Coverage Model

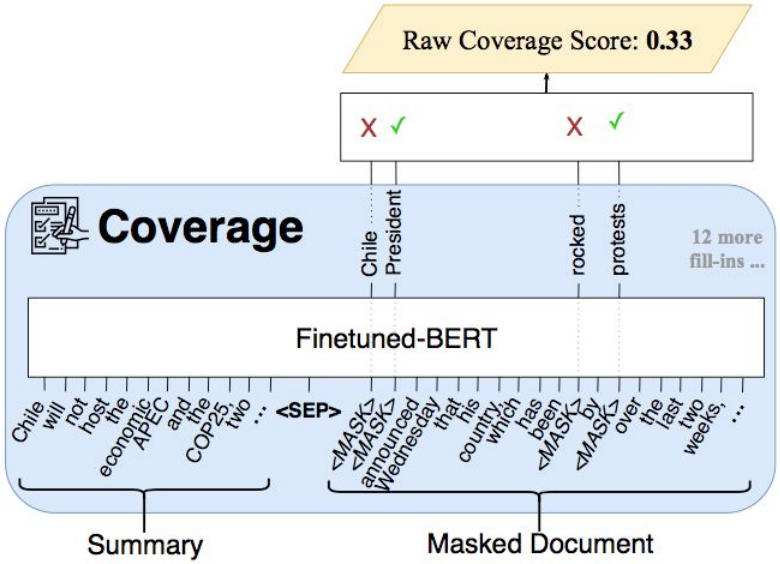


Figure 3: **The Coverage model uses a finetuned BERT model.** The summary is concatenated to the masked document as the input, and the model predicts the identity of each blank from the original document. The accuracy obtained is the *raw coverage score*.

Fluency Model

Fluency Model is to judge the writing quality of the summary, independent of its coverage

Prevent the summary of just being a list of keywords.

Modify language model's probability into a Fluency Score.

Fluency Model

Use generative Transformer (Radford et al., 2019) architecture.

As it can be trained into a powerful language model.

Objective functions

Coverage Score

D: document

M = f(D), the masked document

S: summary

F = g(S,M) the filled document produced by coverage model

$$\text{RawCov}(D, S) = \frac{\|i \in I_M \text{ if } D_i = F_i\|}{\|I_M\|} \quad (1)$$

Coverage Score

Disadvantage:

The model can use information in the unmasked (visible) words of M to predict the masked words.

Calibration:

$$\text{NormCov}(D, S) = \text{RawCov}(D, S) - \text{RawCov}(D, \text{“ ”}) \quad (2)$$

Fluency Score

To produce a uniform Fluency Score, we linearly scale the language model's log-probability of a given summary ($LM(S)$) between an ideal value LP_{low} and a maximum value LP_{high}:

$$\text{Fluency}(S) = 1 - \frac{LM(S) - LP_{low}}{LP_{high} - LP_{low}} \quad (3)$$

Summary Score

The final Summary Score is a weighed sum of the Coverage and Fluency Scores:

$$\text{SummaryScore}(D, S) = \alpha \cdot \text{NormCov}(D, S) + \beta \cdot \text{Fluency}(S) \quad (4)$$

Training procedure

Training Procedure

Train the coverage and fluency model once and them froze their weights during the training of summarizer

Use RL for training summarizer

Training the Coverage Model

Use the first 50 words of unmasked document as a proxy for document summaries (keep the procedure unsupervised)

Because BERT is already trained on the similar MLM task, the Coverage model is able to leverage knowledge accrued by BERT.

Training the Fluency Model

Standard with the proposed fluency score

Training the Summarizer

Use Self-critical sequence training (SCST) method

In SCST, the Summarizer is used to produce two summaries of document D: a greedy summary $\hat{S}$, using a decoding strategy that always picks the most likely next word, and a sampled summary S^s, picking the next word in the summary by sampling from the word distribution.

Summaries are scored using the Summary Loop:

$$\hat{R} = SummaryScore(D, \hat{S})$$
$$R^s = SummaryScore(D, S^s)$$

Then we minimize the following loss:

$$L = (\hat{R} - R^s) \sum_{i=0}^{N} \log p(w_i^s | w_1^s, ..., w_{i-1}^s, D)$$

Training Guard Rails

Prevent summarizer model learns pathological summarization strategies.

If a pathology is detected in a summary, its Summary Score is reduced by a penalty amount

- No-repetition
- Finish-your-sentence
- No-frame-filling

Training Guard Rails

No-repetition:

Raises a penalty on a summary when it contains any repeated 3-gram.

Training Guard Rails

Finish-your-sentence:

Given the length constraint of the generated summary, if the model doesn’t produce the END token, it is likely the last sentence is not completed.

Raise penalty for this case.

Training Guard Rails

No-frame-filling:

During training, the model sometimes learns to overly rely on sentence patterns that achieves high reward as a one size fits all summary.

I.e., X talks with Y about the Z

During training, we keep track of the last 100 summaries produced by the model. We then aggregate the frequency of words for each word position in the 100 summaries. If any word appears more than 50% of the time at a specific word position, we raise the penalty.

Experiment Result

Metric

Summarizer:

ROUGE, or **Recall-Oriented Understudy for Gisting Evaluation**

- ROUGE–N: Overlap of N–grams[2] between the system and reference summaries.
  - ROUGE–1 refers to the overlap of ***unigram*** *(each word)* between the system and reference summaries.
  - ROUGE–2 refers to the overlap of ***bigrams*** between the system and reference summaries.

ROUGE–L: Longest Common Subsequence (LCS)[3] based statistics.

Metric

Coverage:

Use the coverage score defined before

Fluency:

Use the fluency score defined before

Brevity:

Ave # words

Method	R-1	R-2	R-L	Coverage Score	Fluency Score	Brevity (avg words)
Baselines						
Human-written Summaries	100	100	100	0.392	0.612	58.5
𝕏 Lead-3 baseline	40.3	17.7	36.6	**0.421**	**0.656**	**84.0**
Supervised Methods						
Pointer Generator (See et al., 2017)	36.4	15.7	33.4	0.342	0.547	55.6
PG + Coverage (See et al., 2017)	39.5	17.3	36.4	0.377	0.508	61.7
Bottom-Up (Gehrmann et al., 2018)	41.2	18.7	38.3	0.378	0.538	73.9
$PEGASUS_{BASE}$ (Zhang et al., 2019a)	41.8	18.8	38.9	-	-	-
$PEGASUS_{LARGE}$ (Zhang et al., 2019a)	**44.1**	**21.3**	**40.9**	-	-	-
Unsupervised Methods						
𝕏 TextRank (Mihalcea and Tarau, 2004)	35.2	12.9	28.7	0.370	0.612	49.62
GPT2 Zero-Shot (Radford et al., 2019)	29.3	8.3	26.6	-	-	-
Summary Loop 45	**37.7**	**14.8**	**34.7**	0.404	0.627	47.0

Table 2: ROUGE Results (F-1) on the non-anonymized CNN/DM test-set for supervised and unsupervised methods. Extractive methods indicated with 𝕏. Our ROUGE scores have a 95% confidence interval of at most ±0.30. Coverage, Fluency and Brevity (average number of words) included for systems where summaries are available, using Coverage and Fluency models from our work.

Supervision is not the enemy

Initialization Method	R-1	R-2	R-L	Test Loss
28k samples from CNN/DM (10%)				
Random Initialization	7.0	0.9	8.8	6.05
GPT2	37.1	15.9	31.9	2.21
Summary Loop S10	**38.7**	**16.2**	**35.1**	2.07
All of CNN/DN (100%)				
Random Weights	20.4	4.1	19.1	4.22
GPT2	38.4	17.2	35.0	2.02
Summary Loop S100	**41.0**	**18.1**	**37.3**	1.89

Table 5: ROUGE Results on the CNN/DM test-set for supervised generative Transformers. Initializing with the unsupervised Summary Loop outperforms random and GPT2 initializations.


YOU
BELONG
HERE
CNS Diversity and Inclusion • cns.utexas.edu/diversity

Questions?