CPSC 532P / LING 530A: Deep Learning for Natural Language Processing (DL-NLP)

Muhammad Abdul-Mageed

muhammad.mageed@ubc.ca

Natural Language Processing Lab

The University of British Columbia

Table of Contents

Autoencoders

Autoencoders: Networks That Copy Their Input to Their Output

- A neural network is trained to copy its input to its output.

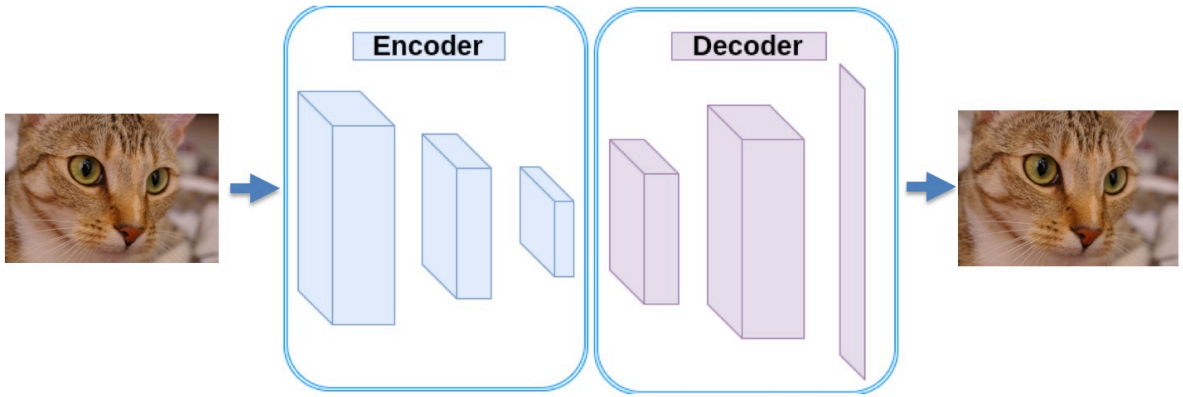


Figure: An Autoencoder.

Autoencoders: Machines With Bottleneck

Autoencoders: Bottleneck in Latent Space Representation

- Has a hidden layer h that describes a **code** representing the input.
- Has an **encoder** function $h = f^{(x)}$ and a **decoder** that produces a reconstruction $r = g^{(h)}$.

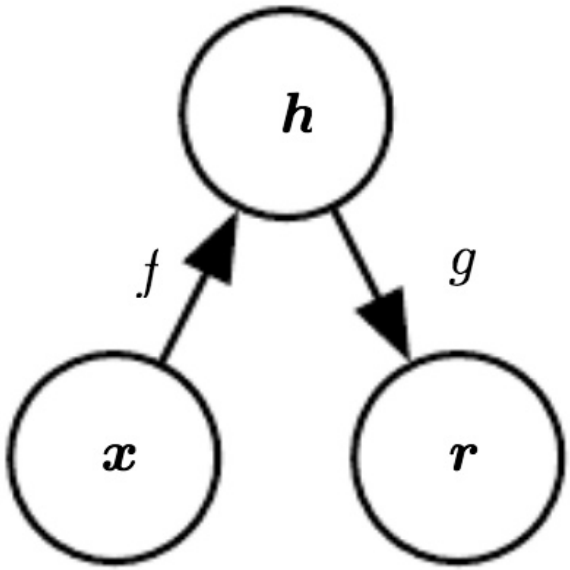


Figure: [Goodfellow et al., 2016]

Autoencoders as Stochastic Mappings

Autoencoders: Stochastic Mappings

- We are not interested in trivial copying.
- We can force the model to learn **only useful properties** of the data.
- We can do this e.g., by constraining **h** to have smaller dimension than **x** (**undercomplete autoencoder**).
- Autoencoders are **stochastic** mappings between:

$$p_{encoder}(h|x)$$

and

$$p_{decoder}(x|h).$$

Rgularizing Autoencoders

There are at least three methods for preventing AEs from copying:

And Three For The Road...

1. **Limiting the model capacity** by keeping the encoder and decoder shallow
2. **Keeping the code size small** (i.e., using an h with smaller dimension than x)
3. **Regularization**

GANs

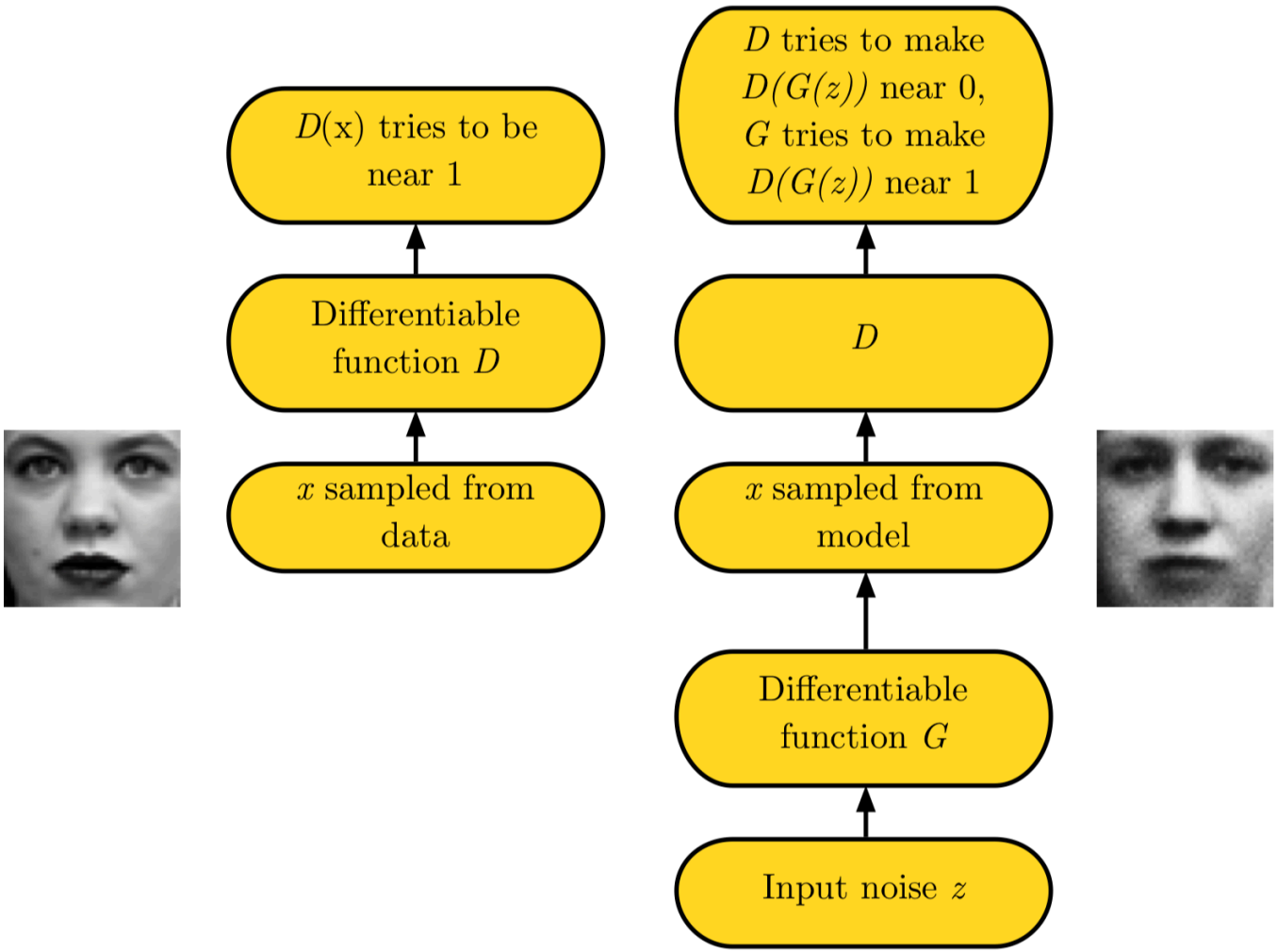


Figure: Adversarial nets framework (Goodfellow 2016; see tutorial)

Training GANs

- **simultaneous SGD**
- On each step, **two minibatches are sampled**:
  - a **minibatch of x values** from the dataset
  - a **minibatch of z values** drawn from the model's prior over latent variables
- Then **two gradient steps are made simultaneously**:
  - one **updating $\theta(D)$ to reduce J(D)**
  - one **updating $\theta(G)$ to reduce J(G)**

Sample GAN-Generated Images

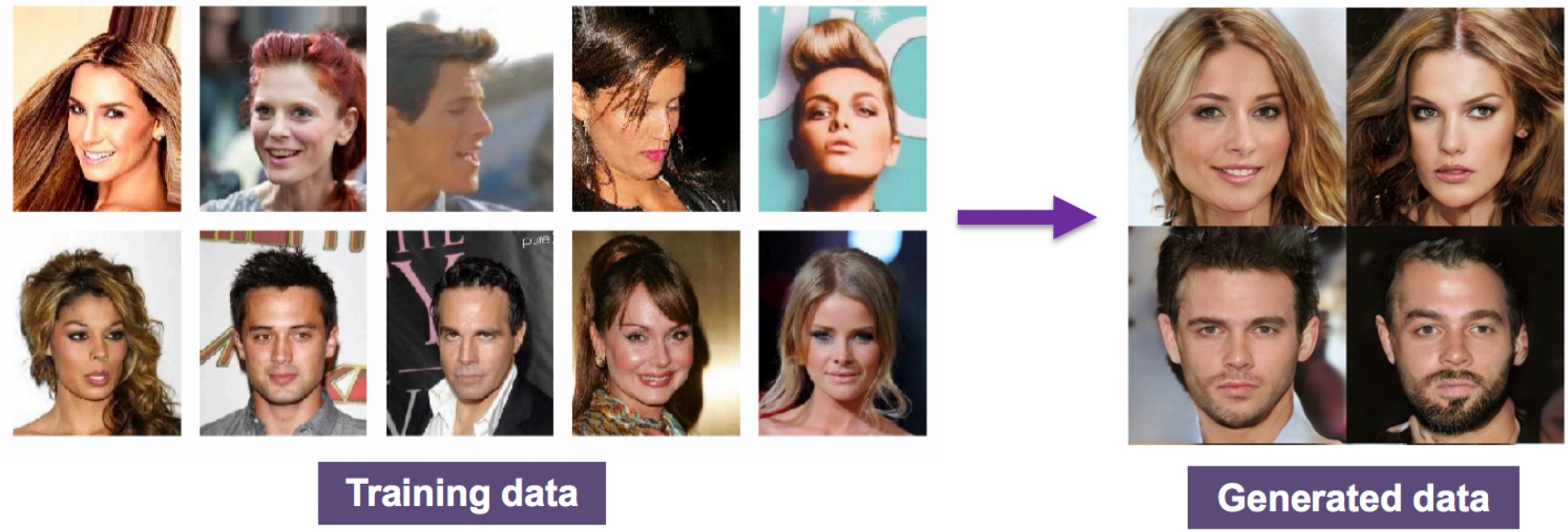


Figure: [From Karras et al, 2017; see paper]

Sample GAN-Generated Images *Contd.*

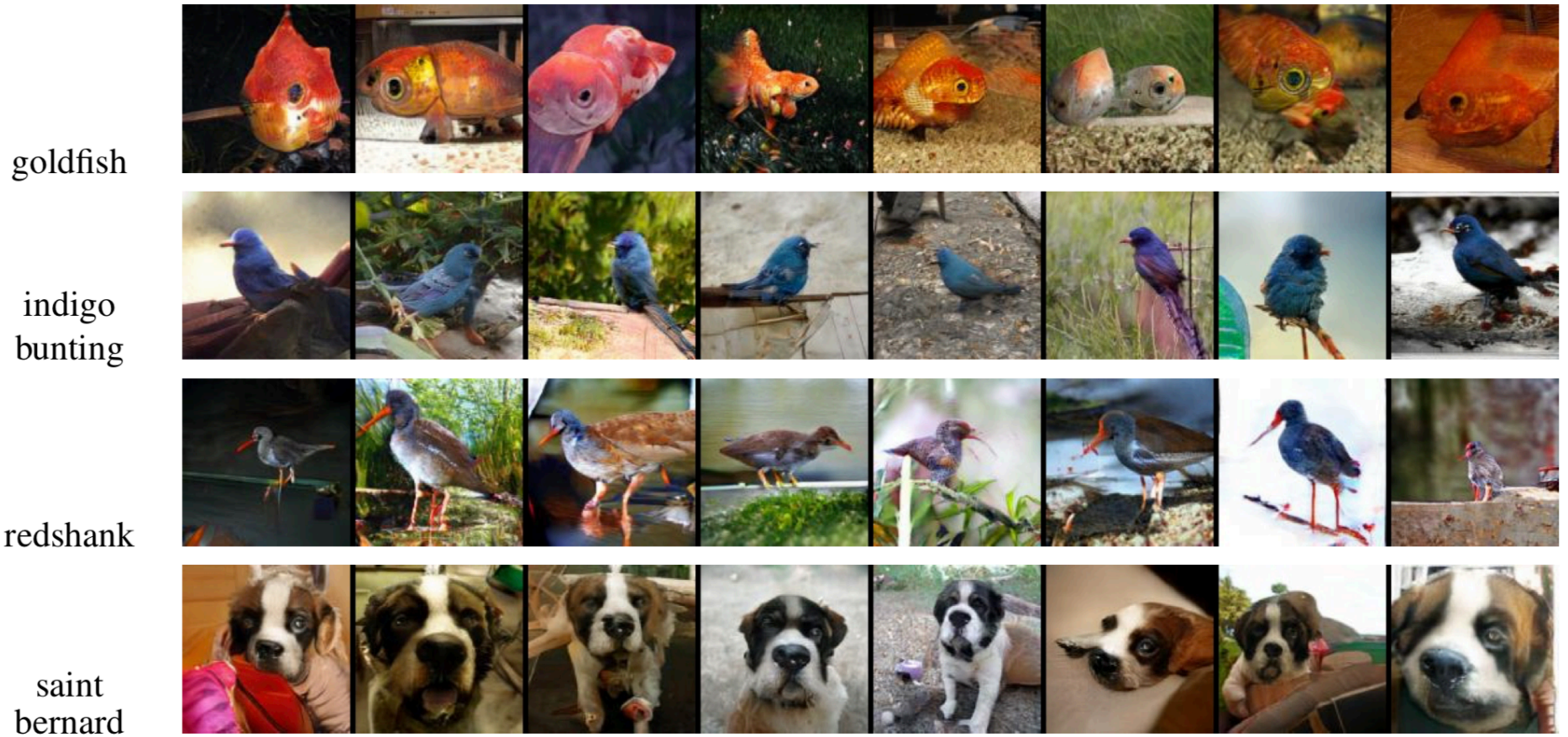


Figure: [From Zhang et al, 2018; see paper]

Multi-Task Learning (**DiaNet**)

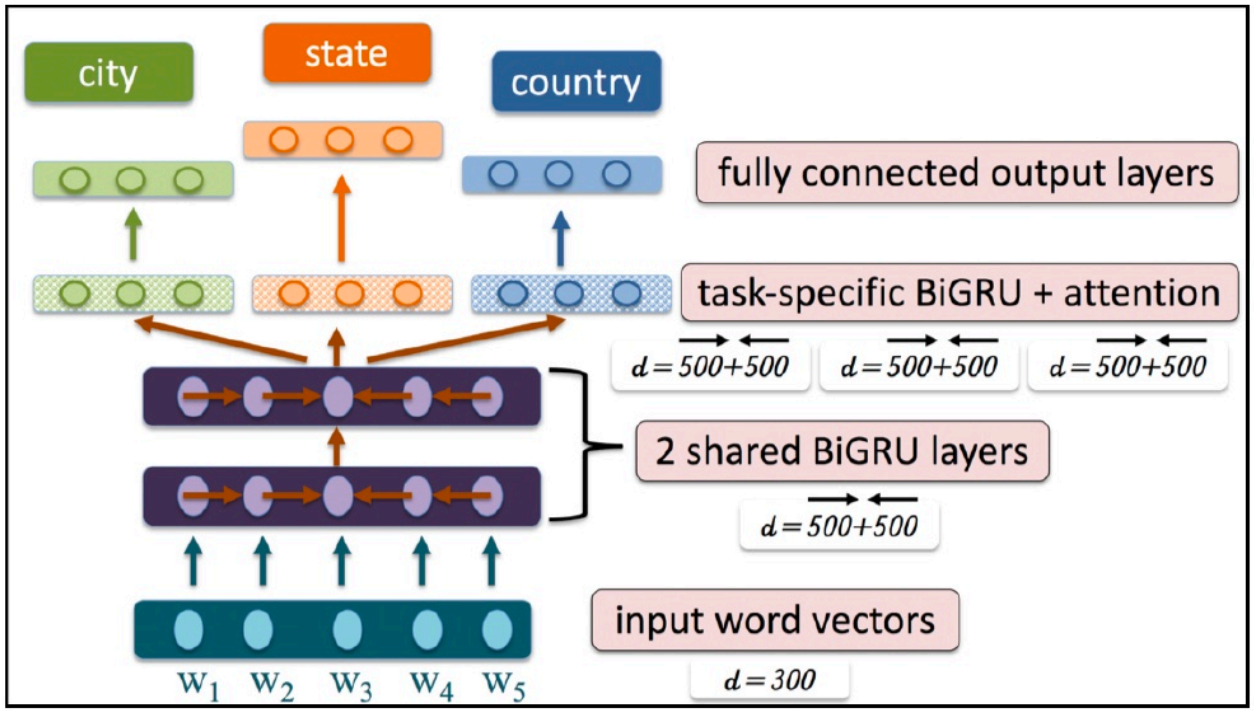


Figure 3: Our MTL network for city, state, and country. The three tasks share 2 hidden layers, with each task having its independent attention layer.

DiaNet Coverage

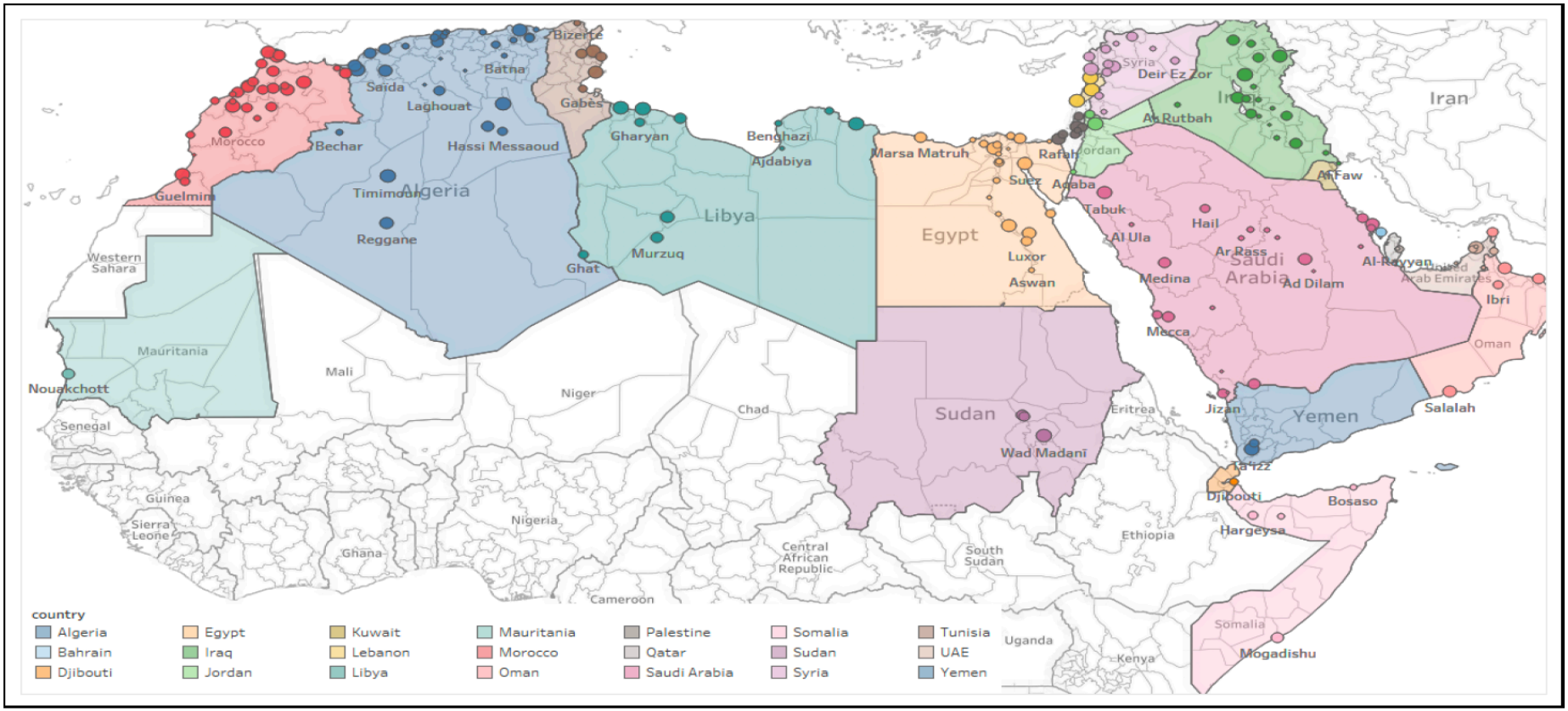


Figure: A total of 319 cities covered in DiaNet, coming from 21 Arab countries.

Continuous Word Representations **Across Languages**

Background

- **Distributional hypothesis (Harris, 1954):** Words occurring in similar contexts tend to have similar meanings
- Exploited in **Word2vec** (Mikolov et al., 2013c;a) and **GloVe** (Pennington et al., 2014); and **FastText** (Bojanowski et al., 2017)
- **Exciting discovery!**: Continuous word embedding spaces **exhibit similar structures across languages**, even when considering distant language pairs like English and Vietnamese (Mikolov et al., 2013b)
- Mikolov et al. (2013b) use a **linear mapping from a source to a target embedding space** with a parallel vocabulary of 5K words as anchor points to learn this mapping
- Mikolov et al. (2013b) **evaluate on a word translation task**

Supervised Learning of XL Word Embeddings

Studies Relying on Bilingual Word Lexica

- Faruqui & Dyer (2014); Xing et al. (2015); Lazaridou et al. (2015); Ammar et al. (2016); Artetxe et al. (2016); Smith et al. (2017)

Reducing Reliance on Bilingual Lexica

- **Using identical character strings** to form a parallel vocabulary (Smith et al., 2017)
- **Using aligned digits** to gradually align embedding spaces (Artetxe et al.,2017)
- Mostly limited to **similar languages sharing a common alphabet**, such as European languages.

Unsupervised, But Less Successful!

Unsupervised

- **Using a distribution-based approach** (Cao et al., 2016)
- **Using adversarial training** (Zhang et al., 2017b)
- Both are **less successful than supervised methods**
- **Conneau et al., 2018**: **(On par with supervised methods!)**
  1. **adversarial training**
  2. **synthetic parallel vocabulary**
  3. **cross-domain similarity local scaling (CSLS)**
- With **two sets of embeddings trained independently on monolingual data**
- **Learn a mapping between the two sets such that translations are close in the shared space**

Learning a Mapping W Between S & T

Finding in Mikolov et al. (2013b)

- Let $\mathcal{X} = \{x_1, ..., x_n\}$ and $\mathcal{Y} = \{y_1, ..., y_m\}$ be **two sets of n and m word embeddings** coming from a source and a target language
- We can **exploit similarities of monolingual embedding spaces to learn a mapping W between source and target space**.
- P.S. They **use a dict of $n = 5000$ pairs of words** $\{x_i, y_i\}_{i \in \{1,n\}}$ to learn a linear mapping such that: (see next slide)

Loss in Mikolov et al. (2013b)

1: Main Loss

$$W^{\star} = \underset{W \in M_d(\mathbb{R})}{\operatorname{argmin}} \|WX - Y\|_{\mathrm{F}}$$

Where:

- d: dimension of the embeddings
- $M_d(\mathbb{R})$: space of $d \times d$ matrices of real numbers
- X and Y: aligned matrices of $d \times n$ with embeddings of the words in parallel vocab
- **Translation t of any source word s** defined as: $t = \operatorname{argmax}_t \cos(Wx_s, y_t)$.

Conneau et al. (2018)

Published as a conference paper at ICLR 2018

WORD TRANSLATION WITHOUT PARALLEL DATA

Alexis Conneau[*†‡], **Guillaume Lample**[*†§],
Marc'Aurelio Ranzato[†], **Ludovic Denoyer**[§], **Hervé Jégou**[†]
{aconneau,glample,ranzato,rvj}@fb.com
ludovic.denoyer@upmc.fr

ABSTRACT

State-of-the-art methods for learning cross-lingual word embeddings have relied on bilingual dictionaries or parallel corpora. Recent studies showed that the need for parallel data supervision can be alleviated with character-level information. While these methods showed encouraging results, they are not on par with their supervised counterparts and are limited to pairs of languages sharing a common alphabet. In this work, we show that we can build a bilingual dictionary between two languages without using any parallel corpora, by aligning monolingual word embedding spaces in an unsupervised way. Without using any character information, our model even outperforms existing supervised methods on cross-lingual tasks for some language pairs. Our experiments demonstrate that our method works very well also for distant language pairs, like English-Russian or English-Chinese. We finally describe experiments on the English-Esperanto low-resource language pair, on which there only exists a limited amount of parallel data, to show the potential impact of our method in fully unsupervised machine translation. Our code, embeddings and dictionaries are publicly available[1].

Conneau et al. (2018) *Contd.*

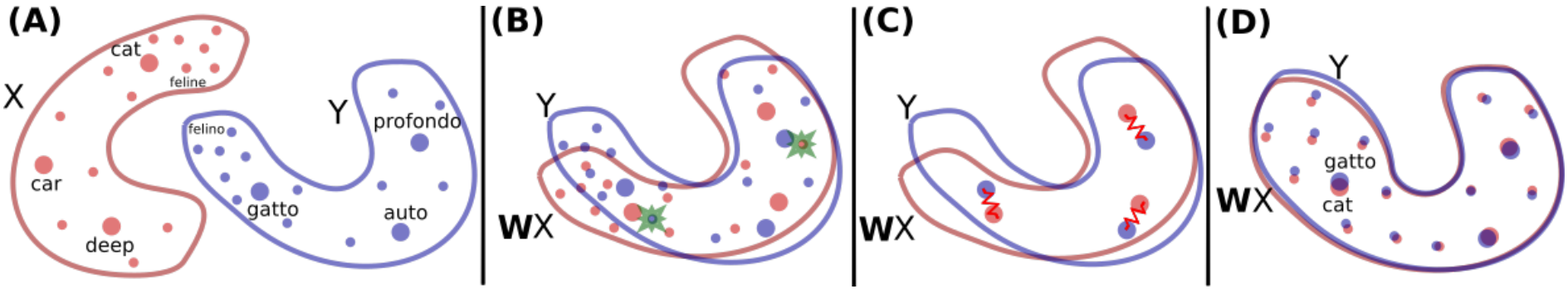


Figure: **(A)** English words in red denoted by X and Italian words in blue denoted by Y. Size of each word dot represents freq in train. **(B)** Use adversarial learning to learn a rotation matrix W which roughly aligns the two distributions. The green stars are randomly selected words fed to the discriminator to determine whether their embeddings come from the same distribution. **(C)** The mapping W is further refined via Procrustes method. **(D)** Finally, translate by using the mapping W and a distance metric that expands the space where there is high density of points (like the area around the word "cat"), so that "hubs" (like the word "cat") become less close to other word vectors than they would otherwise (compare to the same region in panel A).

Domain-adversarial Approach

Learning a Mapping W Between S & T Space

- They use **Deep Adversarial Networks**
- **Discriminator:** Trained to discriminate between elements randomly sampled from $W\mathcal{X} = \{Wx_1, ..., Wx_n\}$ and $\mathcal{Y}$.
- **Mapping W:** W trained to prevent the discriminator from making accurate predictions (Recall Generator)
- **A two-player game:** Discriminator aims at maximizing its ability to identify the origin of an embedding, and W aims at preventing the discriminator from doing so by making $W\mathcal{X}$ and $\mathcal{Y}$ as *similar* as possible

Discriminator

Discriminator Objective

They consider discriminator parameters to be θ_D, and the probability $P_{\theta_D}\left(\text{source} = 1 \middle| z\right)$ that a vector z is the mapping of a source embedding (as opposed to a target embedding) according to the discriminator.

2: Discriminator Loss

$$\mathcal{L}_D(\theta_D|W) = -\frac{1}{n}\sum_{i=1}^{n} \log P_{\theta_D}\left(\text{source} = 1 \middle| Wx_i\right) -$$

$$\frac{1}{m}\sum_{i=1}^{m} \log P_{\theta_D}\left(\text{source} = 0 \middle| y_i\right).$$

Mapping

Mapping Objective

In the unsupervised setting, W is now trained so that the discriminator is unable to accurately predict the embedding origins:

3: Mapping Loss

$$\mathcal{L}_W(W|\theta_D) = -\frac{1}{n}\sum_{i=1}^{n} \log P_{\theta_D}\left(\text{source} = 0 \middle| Wx_i\right) -$$

$$\frac{1}{m}\sum_{i=1}^{m} \log P_{\theta_D}\left(\text{source} = 1 \middle| y_i\right).$$

Refinement

Refining Mapping W

- Adversarial approach tries to align all words **irrespective of their frequencies**
- **Rare words** are updated less frequently, and occur in different contexts in each corpus. (harder to align)
- **Solution:** Use most freq words to acquire **synthetic parallel vocab** using W just learned with adversarial training
- They **retain only mutual nearest neighbors**, to ensure a high-quality dictionary
- Apply the **Procrustes algorithm** on dict and possibly repeat
- Procrustes offers a closed form solution obtained from the singular value decomposition (SVD) of YX^T (see paper)

Hubness Problem

Hubness Problem: Points tending to be nearest neighbors of many points in high-dimensional spaces

- Need to **improve comparison metric** such that the nearest neighbor of a source word, in the target language, is more likely to have as a nearest neighbor this particular source word
- **Problem: Nearest neighbors are asymmetric**: y being a K-NN of x does not imply that x is a K-NN of y.
- Some vectors, dubbed *hubs*, are with high probability nearest neighbors of many other points, while others (*anti-hubs*) are not nearest neighbors of any point.

Cross-Domain Similarity Local Scaling (CSLS)

Bi-partite Neighborhood Graph

- They consider a **bi-partite neighborhood graph** where each word of a given dictionary is connected to its K nearest neighbors in the other language.
- $\mathcal{N}_T(Wx_s)$: The neighborhood, on the bi-partite graph, associated with a mapped source word embedding Wx_s.
- All K elements of $\mathcal{N}_T(Wx_s)$ are words from the target language.
- Similarly, $\mathcal{N}_S(y_t)$ is the neighborhood associated with a word t of the target language.

Mean Similarity of Source Embedding

4: Mean similarity of source embedding x_s to its target neighborhood

$$r_T(Wx_s) = \frac{1}{K} \sum_{y_t \in \mathcal{N}_T(Wx_s)} \cos(Wx_s, y_t)$$

Likewise $r_S(y_t)$ is the mean similarity of a target word y_t to its neighborhood.

Compute Mean Similarities

- **Compute MS quantities for all source and target word vectors with their neighbors**, and use them to define a similarity measure $CSLS(.,.)$ between mapped source words and target words as: $CSLS(Wx_s, y_t) = 2\cos(Wx_s, y_t) - r_T(Wx_s) - r_Sx(y_t)$

CSLS

Compute Mean Similarities

- The CSLS update **increases the similarity associated with isolated word vectors**
- Conversely, **it decreases the ones of vectors lying in dense areas**
- CSLS **significantly increases the accuracy for word translation retrieval**, while not requiring any parameter tuning

Back-Translation

Improving Neural Machine Translation Models with Monolingual Data

Rico Sennrich and **Barry Haddow** and **Alexandra Birch**
School of Informatics, University of Edinburgh
{rico.sennrich,a.birch}@ed.ac.uk, bhaddow@inf.ed.ac.uk

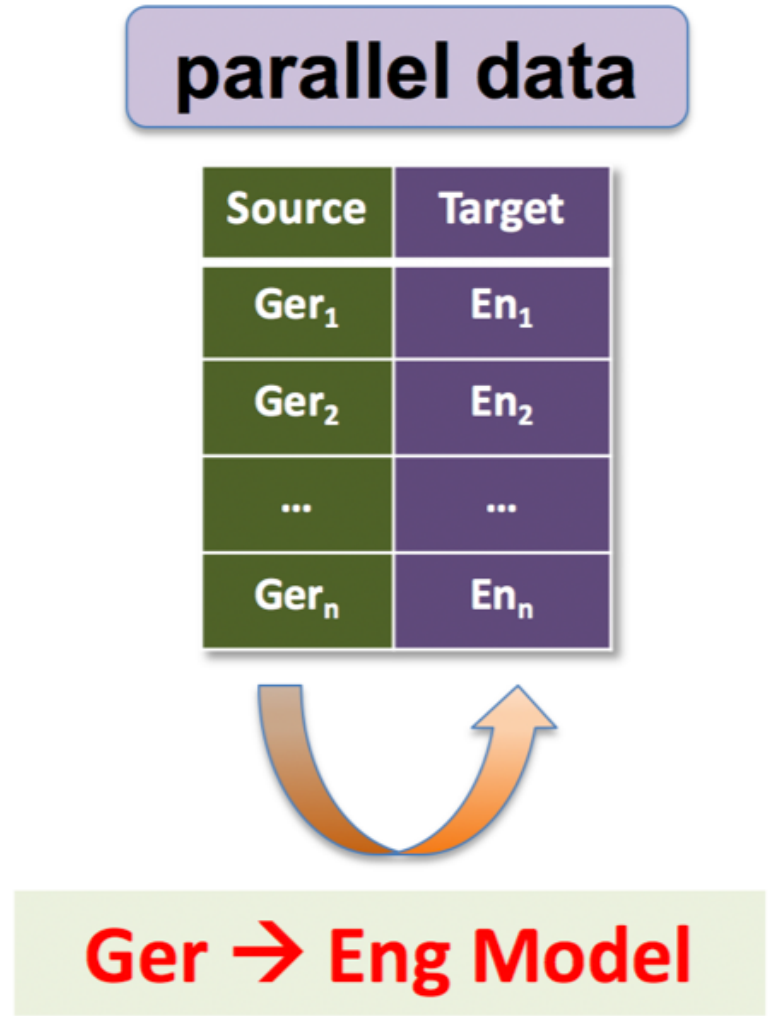
Abstract

Neural Machine Translation (NMT) has obtained state-of-the art performance for several language pairs, while only using parallel data for training. Target-side monolingual data plays an important role in boosting fluency for phrase-based statistical machine translation, and we investigate the use of monolingual data for NMT. In contrast to previous work, which combines NMT models with separately trained language models, we note that encoder-decoder NMT architectures already have the capacity to learn the same information as a language model, and we explore strategies to train with monolingual data without changing the neural network

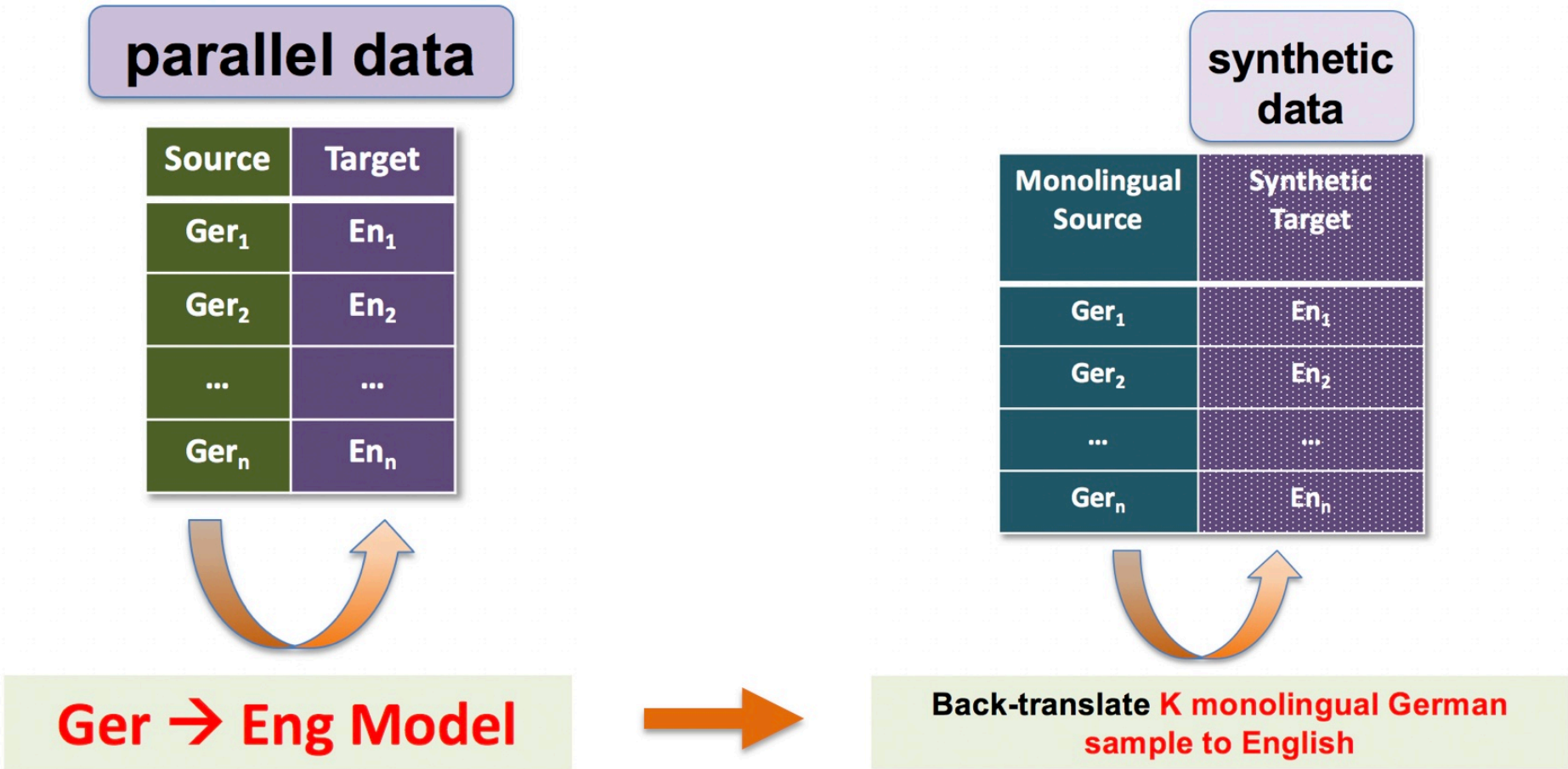
cal machine translation, and we investigate the use of monolingual data for NMT.

Language models trained on monolingual data have played a central role in statistical machine translation since the first IBM models (Brown et al., 1990). There are two major reasons for their importance. Firstly, word-based and phrase-based translation models make strong independence assumptions, with the probability of translation units estimated independently from context, and language models, by making different independence assumptions, can model how well these translation units fit together. Secondly, the amount of available monolingual data in the target language typically far exceeds the amount of parallel data, and models typically improve when trained on more data, or data more similar to the translation task.

Step 1: Translate With Parallel Data



Step 2: Back-Translate L1 Monolingual Data



Step 3: Train New Model With Mixed Data || **Step 4:** Back-Translate L2 Monolingual Data