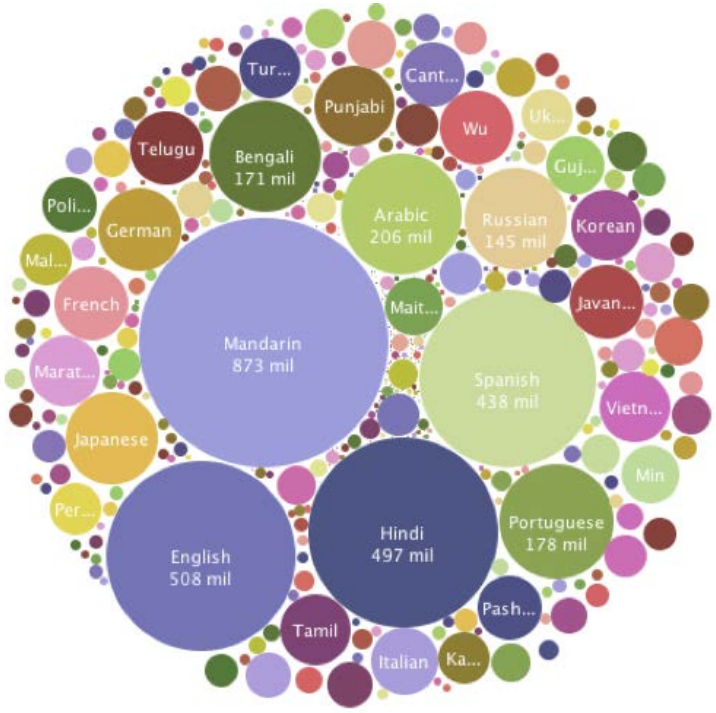
Multilinguality in NLP

Benefiting low resource languages

Sudeshna Sarkar

IIT Kharagpur

Many Languages



- 7000 languages
- India: 122 major languages, 2371 dialects

The Divide

The availability of resources, training data, and benchmarks in English etc leads to a disproportionate focus on a few languages and a neglect of many popular languages

Motivation

Multilingual and Multicultural World

- NLP systems are trained on resources
  - Resources have been created for only the of the world's languages
  - For the majority of languages, no or few resources

Motivation

  - Promote linguistic diversity
  - Enable multilingual access to information
  - Enable multilingual applications
- Need to bootstrap NLP systems for all languages

NLP Tasks

- Language consists of many layers of structures:
  - Sounds, Words, Morphology, POS, Syntax, Semantics, Discourse

Core technologies

- Language modelling
- Part-of-speech tagging
- Syntactic parsing
- Named-entity recognition
- Coreference resolution
- Word sense disambiguation
- Semantic Role Labelling
- …

Applications

- Machine Translation
- Information Retrieval
- Question Answering
- Dialogue Systems
- Information Extraction
- Summarization
- Sentiment Analysis
- …

Paradigm Shifts in NLP

- Logic-based/ Rule-based NLP
- 90s: Statistical NLP:
- Mid 2010s: Neural NLP

- robust in the face of real-world data
- Better performance
- Less engineering of hand-crafted rules/knowledge

- ML models perform poorly in low-resource settings
- Linguistically implausible generalizations

Enabling NLP in resource poor languages

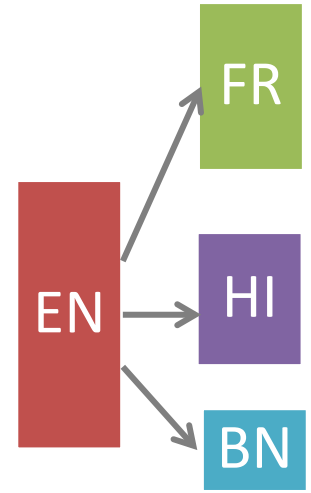
- Standard techniques used in NLP cannot be applied to low-resource languages as large amount of training data are not available.
- But there are various opportunities:
  - You can transfer learned representations from a related task:
    - cross-lingual transfer
    - cross-domain transfer
  - Joint Multilingual, Multi-task learning
  - You can learn useful representations from unlabeled data
    - Unsupervised methods
    - Semi-supervised methods

Transfer learning

- Apply knowledge gained in one context to a different context
  - Different Language
  - Different Task

Transfer Learning

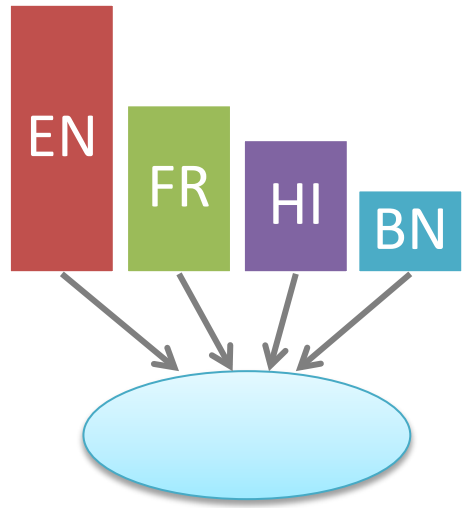
- **Cross-lingual transfer learning** – Transfer of resources and models from resource-rich source to resource-poor target languages
  - Transfer of annotations (e.g., POS tags, syntactic or semantic features) via cross-lingual bridges (e.g., word or phrase alignments)
  - Transfer of models – train a model in a resource-rich language and apply it in a resource-poor language



- **Zero-shot learning** – train a model in one domains and assume it generalizes more or less out-of-the-box in a low-resource domain
- **One-shot learning** – train a model in one domain and use only few examples from a low-resource domain to adapt it

Multilingual Model

- Joint Multilingual Learning: Joint resource-rich and resource-poor learning using a language-universal representation



- An architecture that leverages multilinguality:
  - A single model for a large number of languages
  - Multilingual model outperforms monolingual models

Neural Networks in NLP

Now State of the art in most NLP problems

- Entity and Relation finding
- Parsing
- Machine Translation
- Entity linking
- …

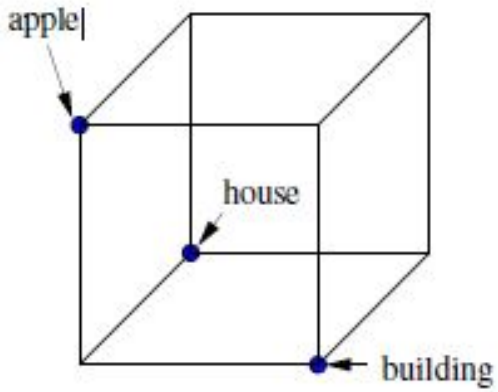
Cross-lingual transfer learning enablers

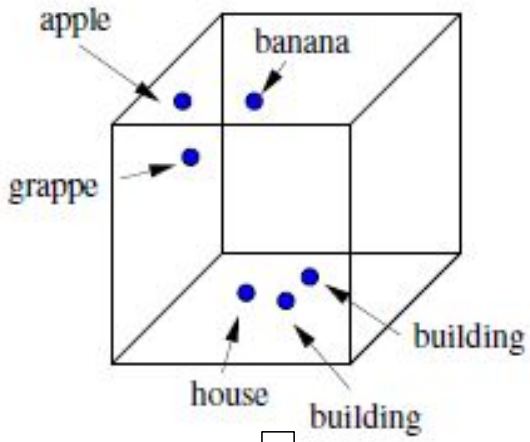
- Shared representation
- Bilingual and multilingual resources

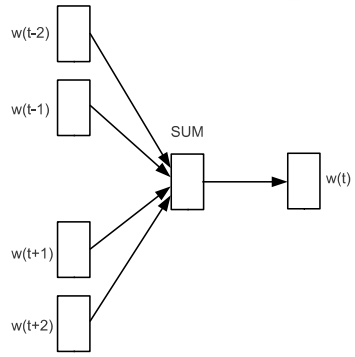
Representation: Text Embedding

A key secret sauce for the success of many NLP systems across tasks

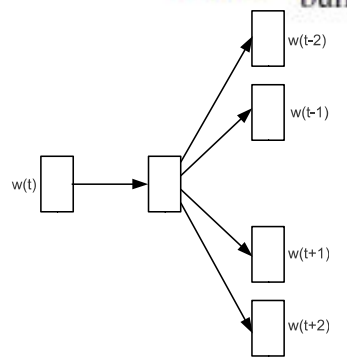
- Continuous vector space representation of word, sentence, etc as dense vector $x_i \in R^d$
- The quest for Universal Embeddings: embeddings that are pre-trained on a large corpus and can be plugged in a variety of downstream task models







CBOW



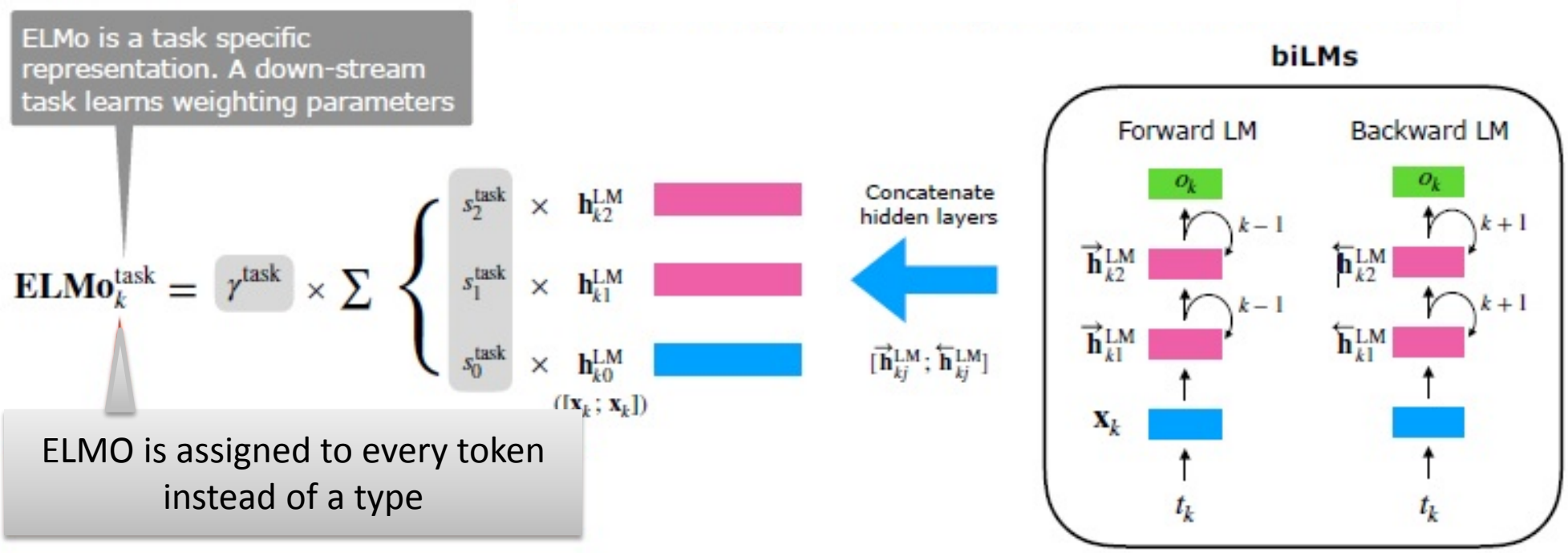
Skip-gram

1. Word2vec: Learn these embeddings in a way that similar words are nearby in that space.
   - Skipgram, CBOW
2. Fasttext (Bojanowski 2016) uses character ngrams for subword encoding

Distributed Representations of Words and Phrases and their Compositionality; Mikolov etal, NIPS 2013

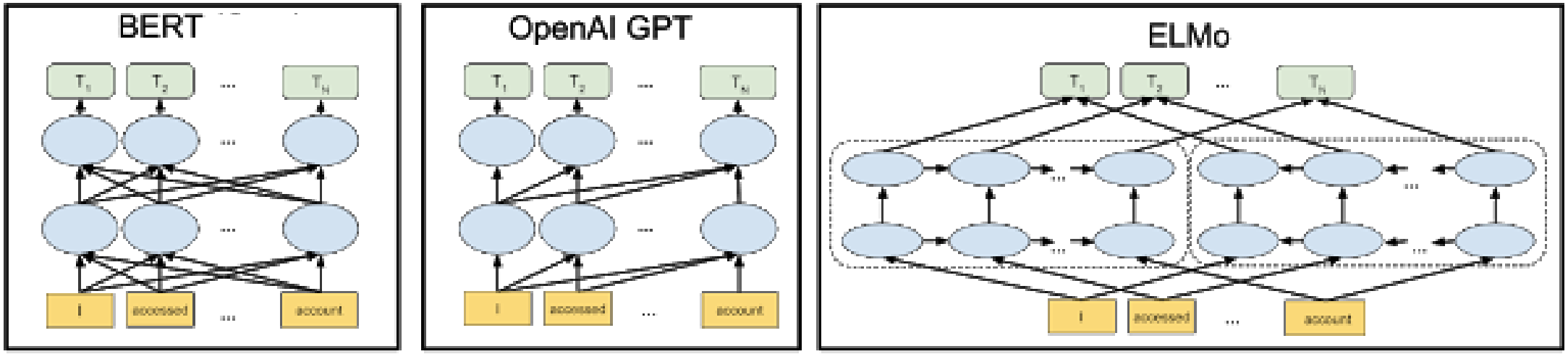
Contextualized representations: ELMO

- ELMO represents a word as a linear combination of corresponding hidden layers (inc. its embedding)



- ELMo can be integrated to almost all neural NLP tasks with simple concatenation to the embedding layer

Contextual Representations: Elmo, BERT, …



A. ELMo (Peters et al., 2018)

B. OpenAI GPT (Radford et al., 2018)

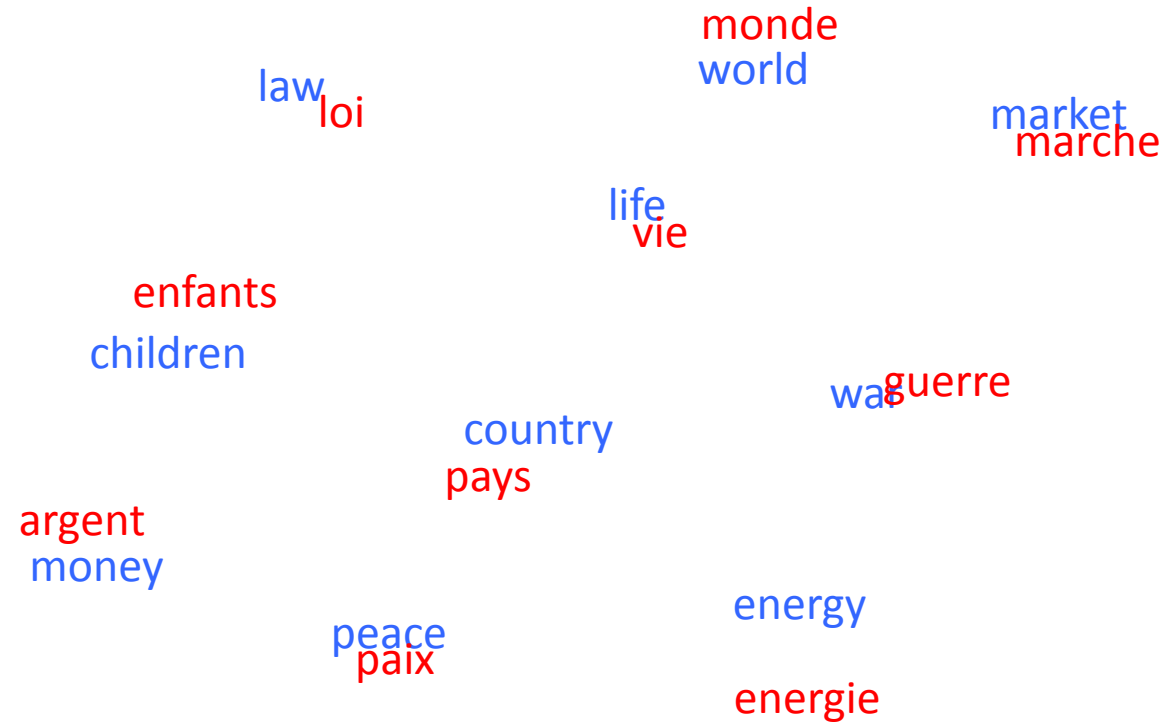
C. BERT: (Devlin 2018 from Google) Bidirectional Encoder Representations from Transformers

1. include pre-trained representations as additional features
2. introduce task-specific parameters and fine-tune the pre-trained parameters

Multilingual Embeddings

- Learn a shared embedding space between words in all languages.
- Many benefits:
  - Transfer learning
  - Cross lingual information retrieval
  - MT

Mikolov etal, 2013; Faruqui & Dyer, 2014; etc

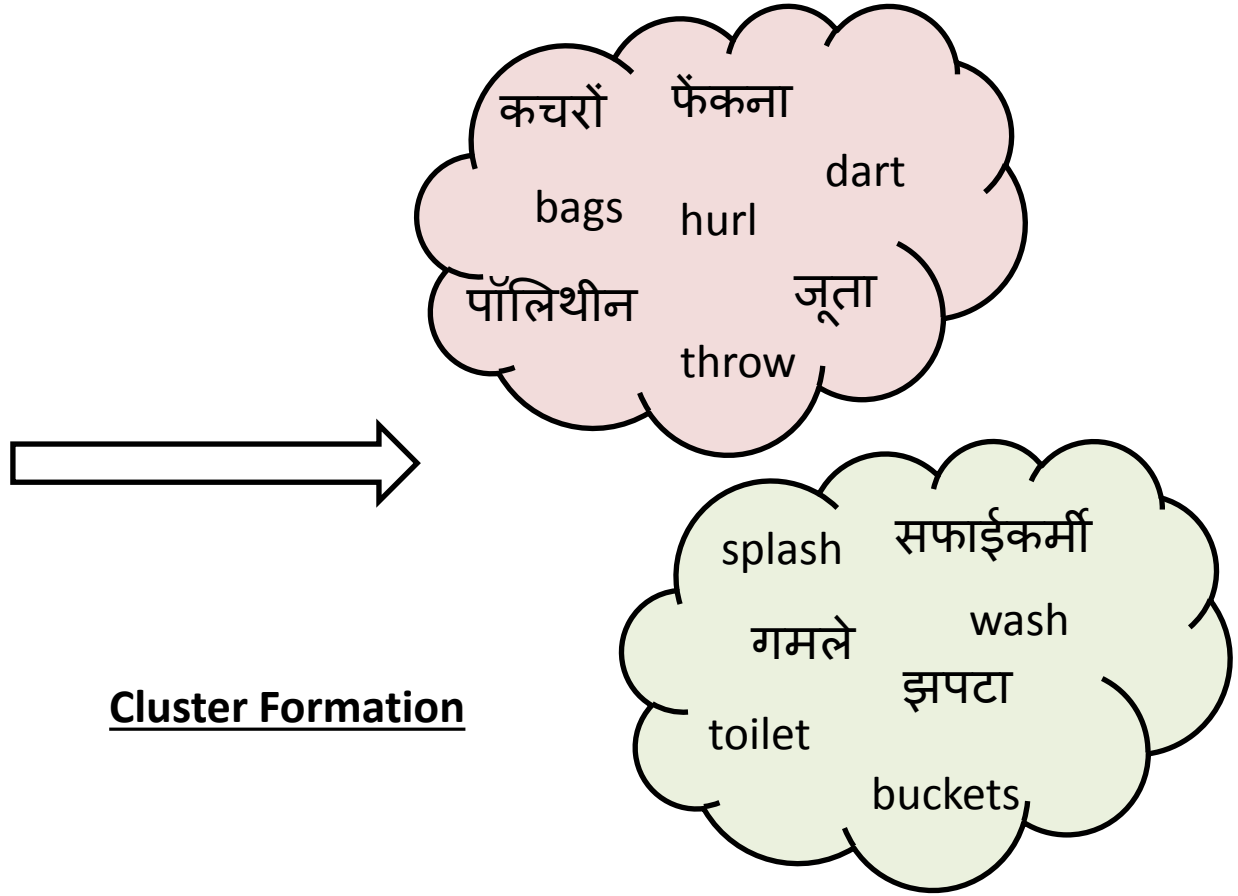


Multilingual word embedding

- Resources:
  - Word aligned data
  - Sentence aligned data
  - Dictionary
  - Document Aligned Corpora
  - Unsupervised

- Methods
- Monolingual mapping
  - Train monolingual word embeddings
  - Learn linear mapping or CCA
- Pseudo-cross-lingual
  - Train on pseudo-cross-lingual corpus by mixing contexts of different languages
- Cross lingual training
  - Train on parallel corpus
- Joint Optimization
  - Jointly optimize a combination of monolingual and cross-lingual losses.

Multilingual Clustering using Word Embeddings



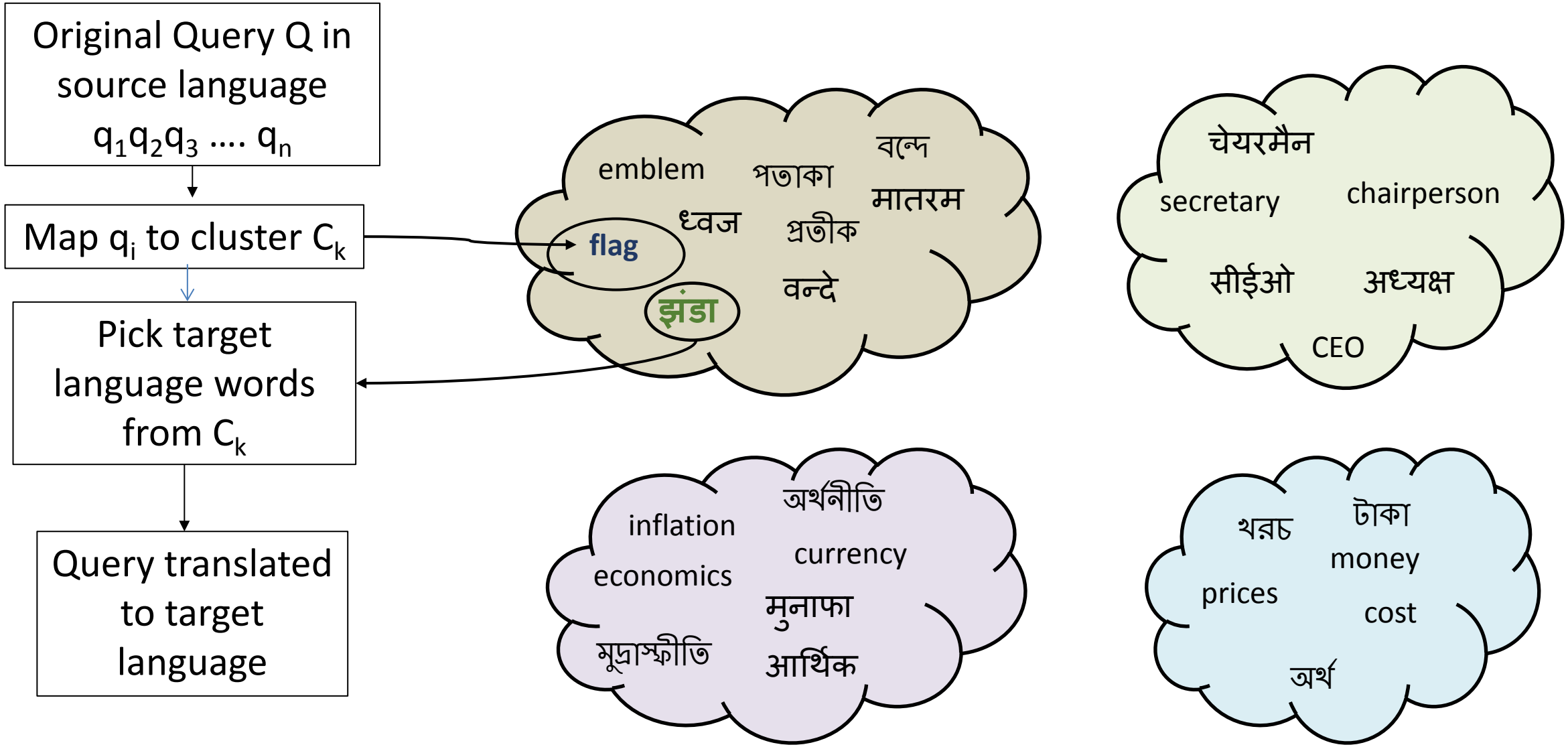
Cluster Formation

- Construct a graph G = (V, E)
- V = vertices, words from the languages
- E = edges, exist if the similarity between the vertices is above a particular threshold
- Edges are weighted as cosine similarity value

- Use Louvain [Blondel et al., 2008] for cluster detection
- Performs hard clustering

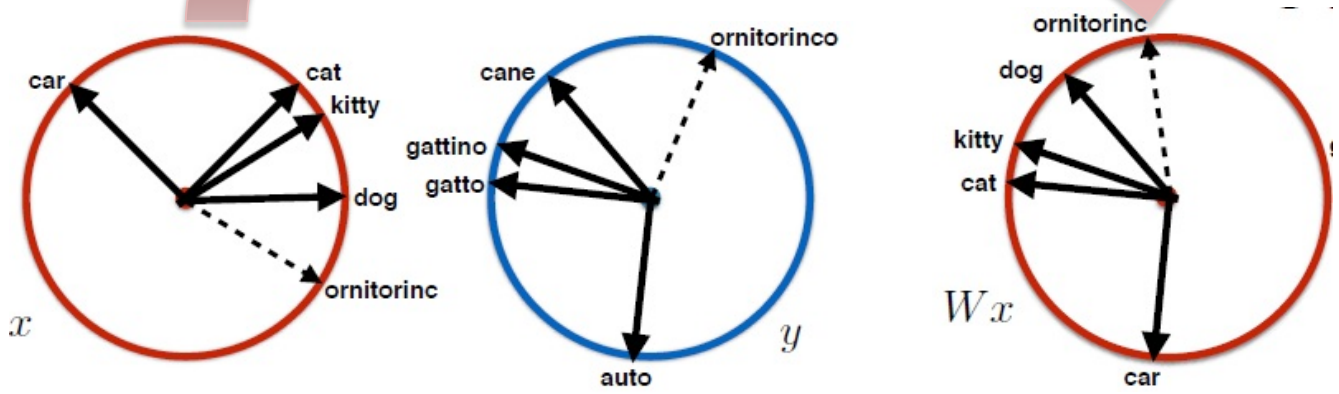
Using Communities of Words Derived from Multilingual Word Vectors for Cross-Language Information Retrieval in Indian Languages Bhattacharya , Goyal , Sarkar , Dec 2018

Application to Cross-Language IR



Unsupervised Word Translation

- The context of a word, is often similar across languages (the same underlying physical world)
- Use only monolingual corpora for two languages.
- Separate embeddings are trained for each language using monolingual data



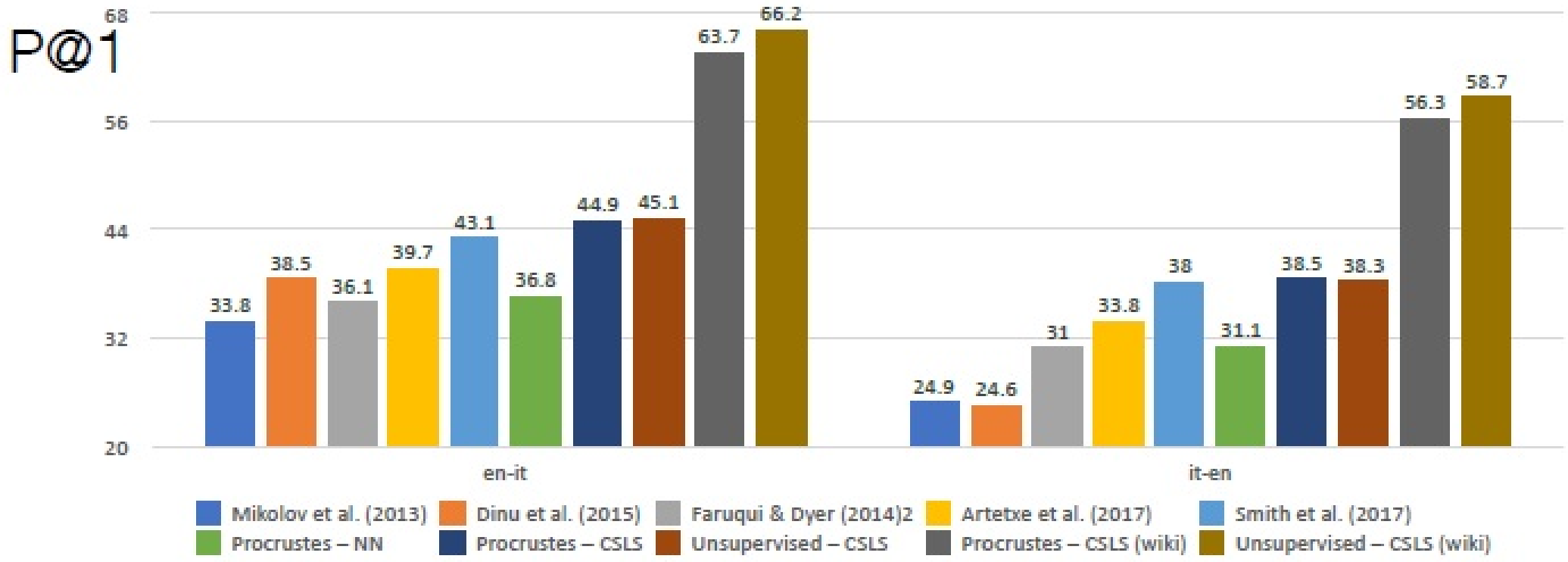
1. Learn a rotation matrix to roughly align the two domains via adversarial training:
- pick a word at random from each language, embed them, project one of the two, and make sure distributions match.

2. Iterative refinement via orthogonal Procrustes, using the most frequent words.
- Pick most frequent words, translate them via nearest neighbor, solve least square, and iterate

$$W_t = argmin\|W_{t-1}X - Y\|^2 \quad \text{s.t. } W_t W_t^T = 1$$

Word Translation Without Parallel Data; Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, Hervé Jégou. Facebook, Le Mans, Sorbonne. ArXiv 2017.

Results on Word Translation



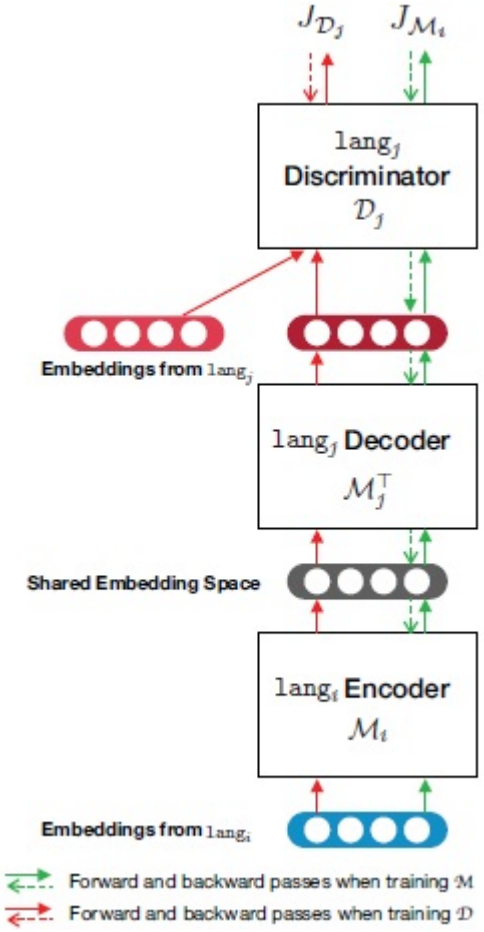
By using more anchor points and lots of unlabeled data, they even outperform supervised approaches!

Key Idea: 1. Learn representations of each domain.
2. Force representations to match in order to translate.

Unsupervised Multilingual Word Embedding

- Learn a single multilingual embedding space (UMWE)
- Uses only monolingual word embeddings.
- The method exploits the interdependencies between two languages and maps all monolingual embeddings into a shared multilingual embedding space via a two-stage algorithm
  i. Multilingual Adversarial Training (MAT)
  ii. Multilingual Pseudo-Supervised Refinement (MPSR)

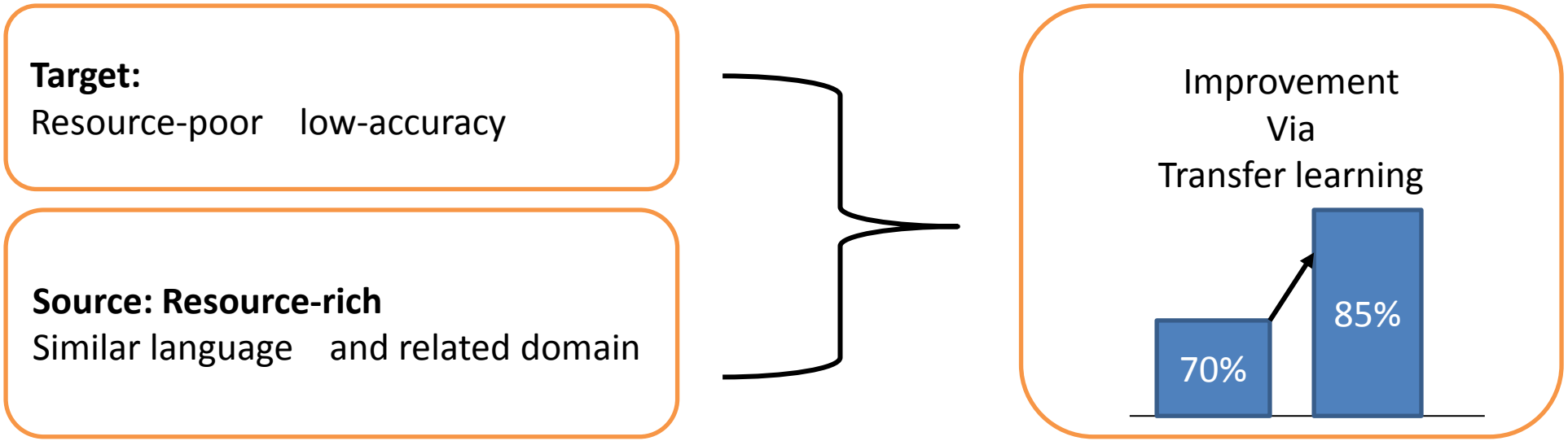
  Induce a dictionary of highly confident word pairs
  for every language pair, used as pseudo supervision
  to improve the embeddings learned by MAT



Chen, Xilun, and Claire Cardie. "Unsupervised Multilingual Word Embeddings. EMNLP 2018.

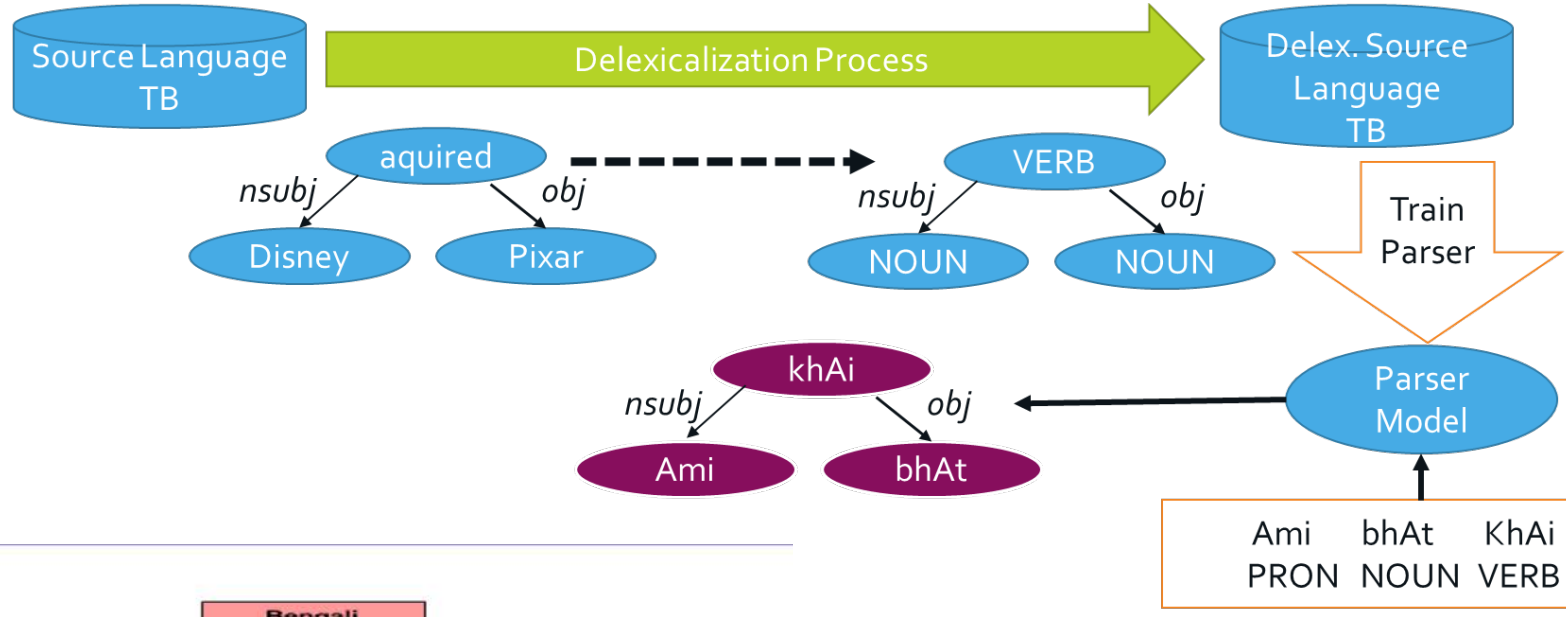
NLP for Low-resource Languages by Transfer

- Good quality NLP tools require large amount of data.
- **Transfer learning:** Use resources in related source language tasks to improve performance of the task in target language.

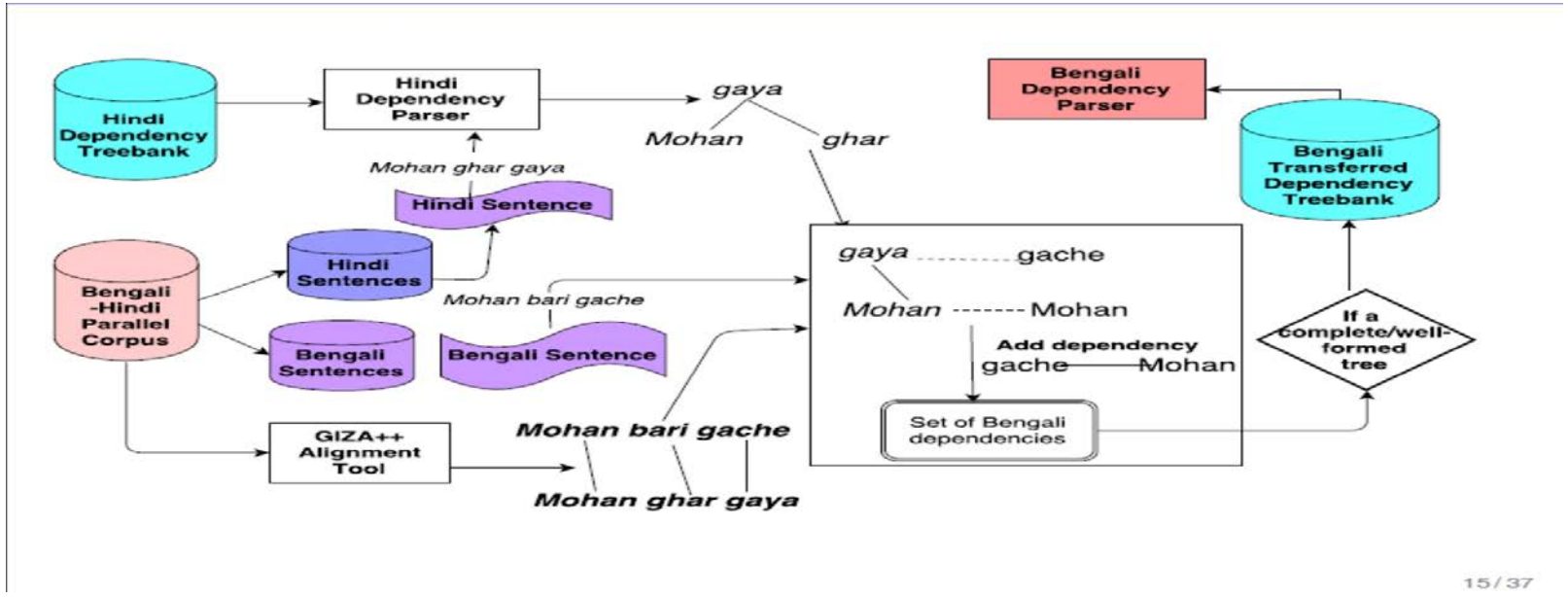


Transfer methods for Dependency Parsing

1. Delexicalized Transfer

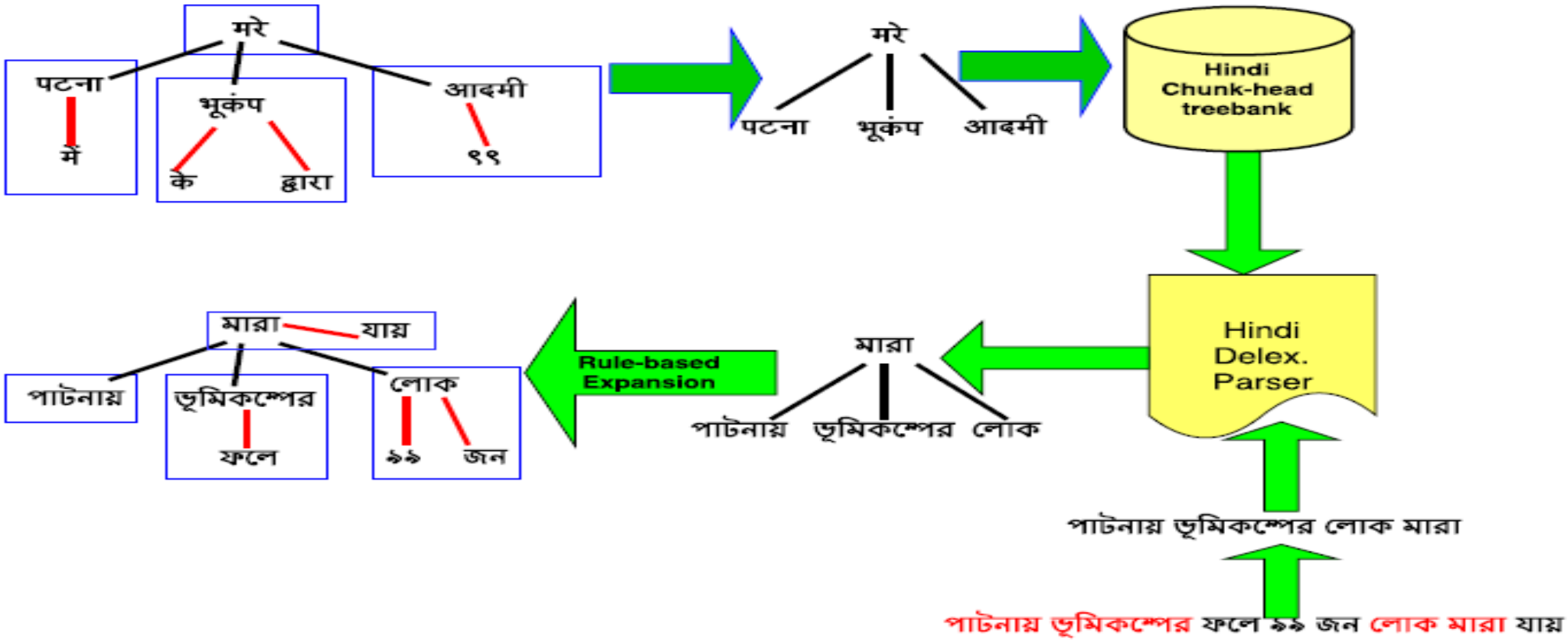


1. Annotation Projection



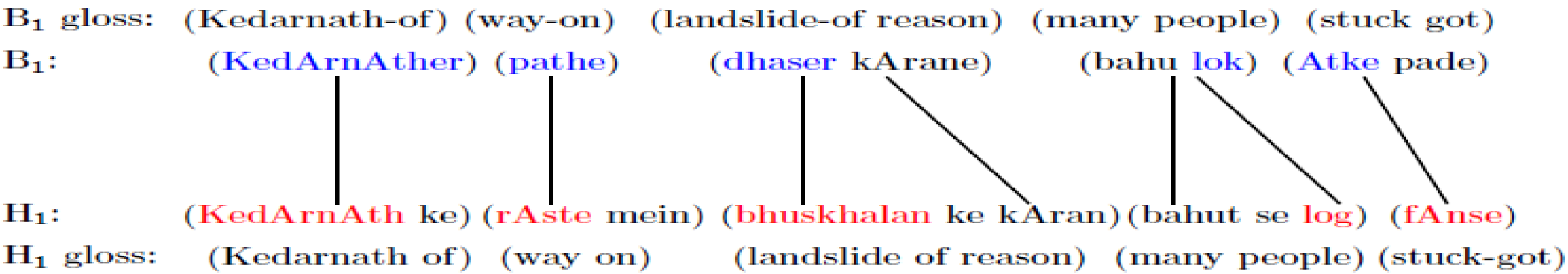
Delex. Transfer from Hindi to Bengali by chunking

English : 99 people died due earthquake in Patna
Hindi : पटना में भूकम्प के द्वारा ९९ लोग मरे
Bengali : পাটনায় ভূমিকম্পের ফলে ৯৯ জন লোক মারা যায়

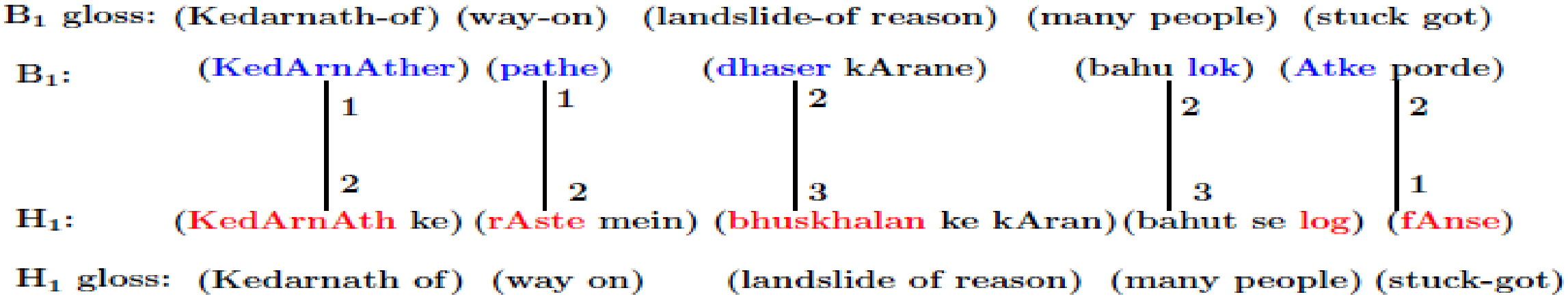


Annotation Projection using Chunking (Example)

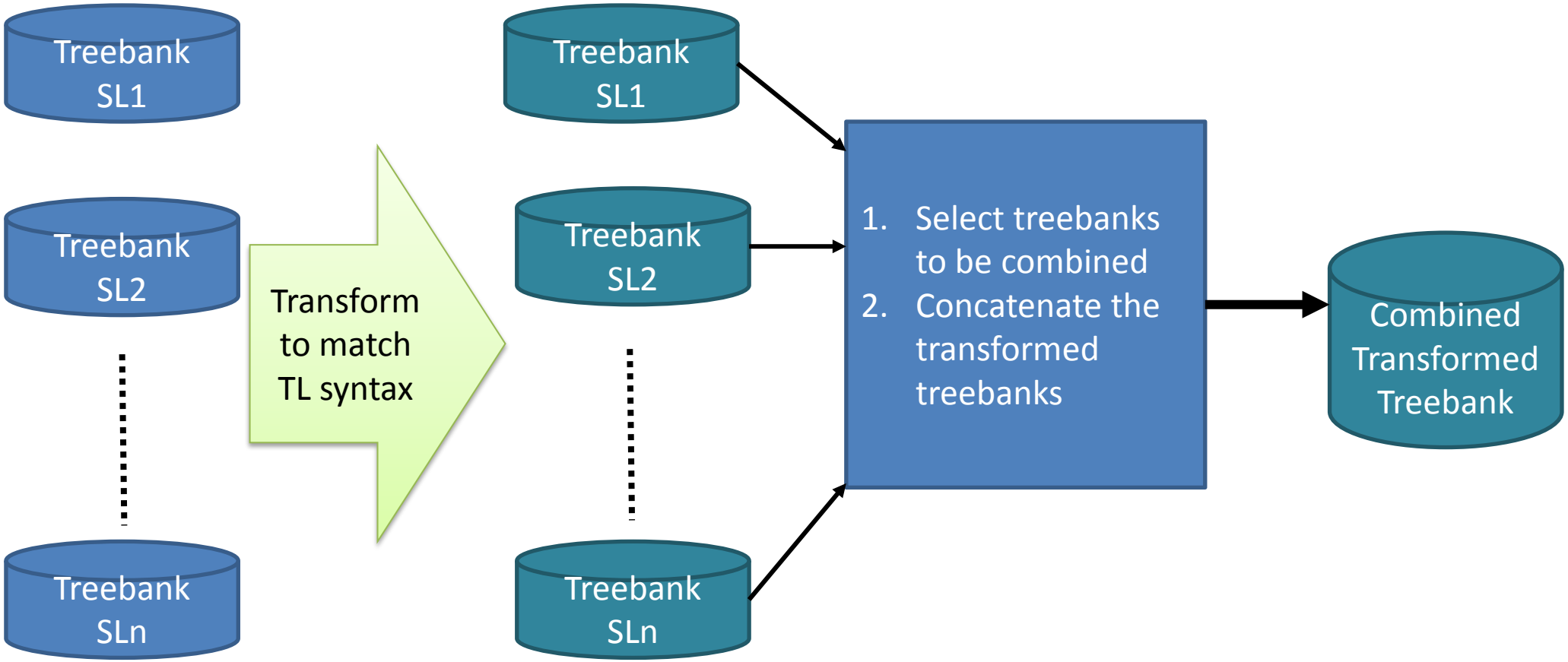
Word Alignment



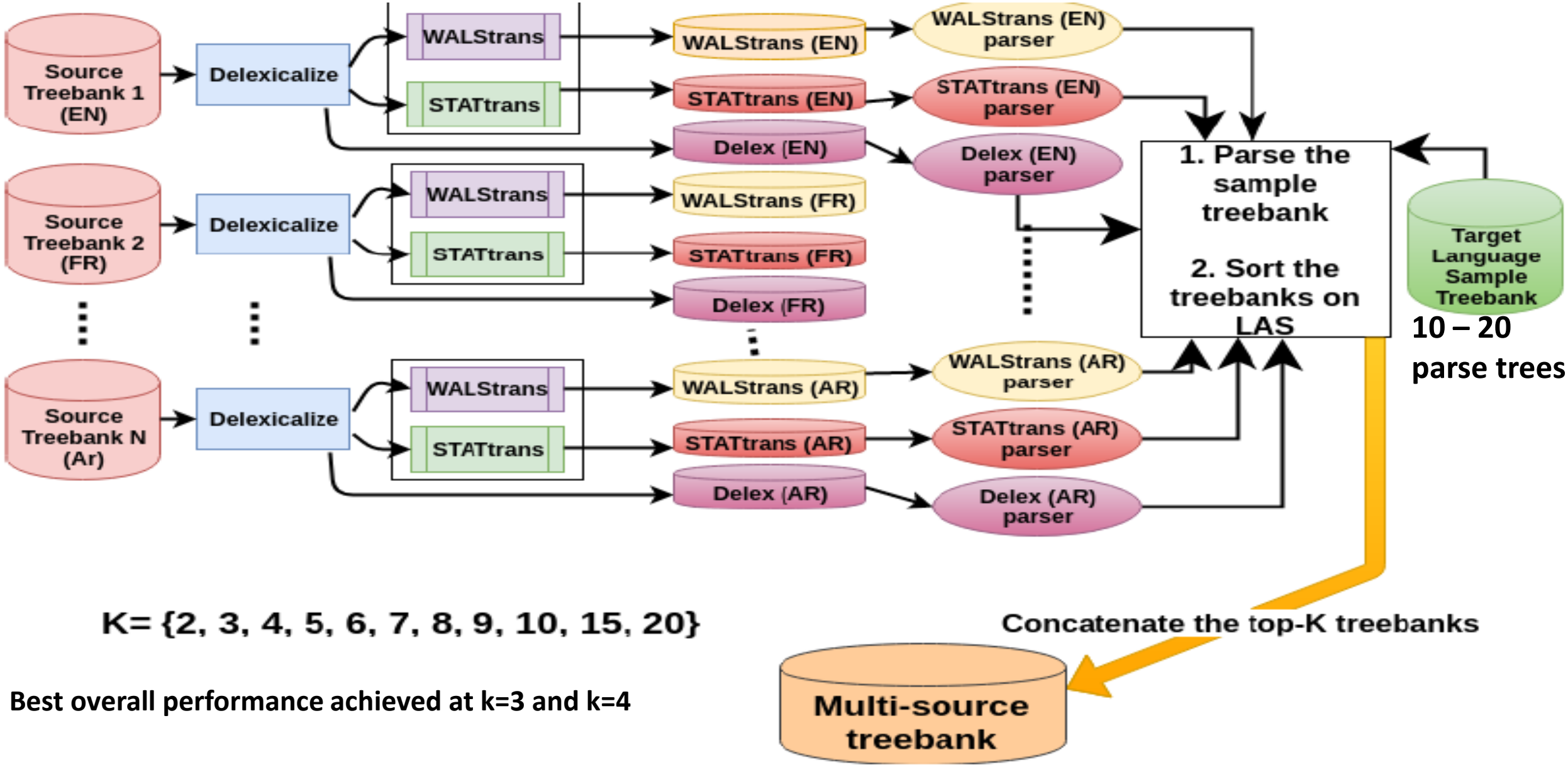
Chunk Alignment



Transform, Combine and Transfer



Multi-lingual transfer using transformation

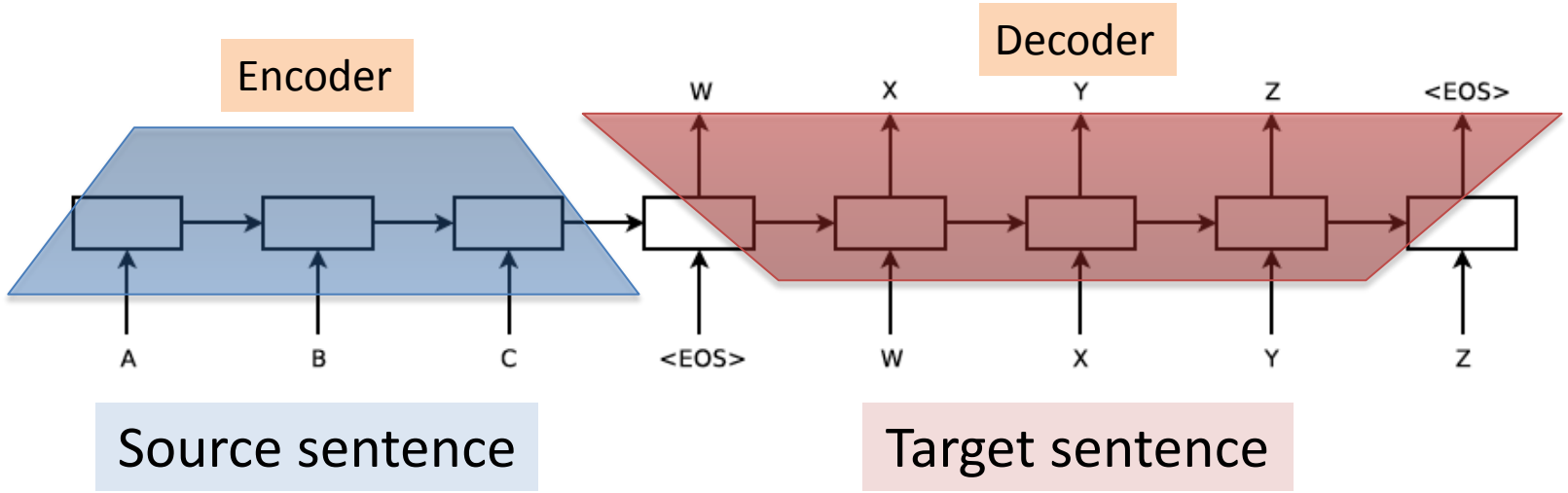


Best overall performance achieved at k=3 and k=4

NEURAL MACHINE TRANSLATION AND LOW RESOURCE LANGUAGES

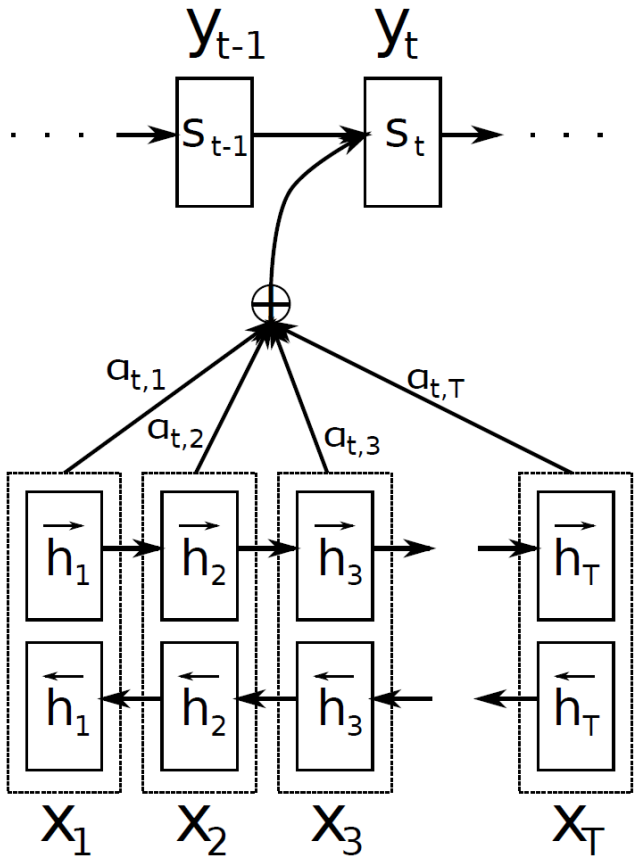
Neural Machine Translation
Encoder-Decoder with Attention

- Encoder Decoder Model



  - End-to-end training: All parameters are simultaneously optimized
  - Distributed representations share strength
  - Better exploitation of context

Encoder-Decoder-Attention for MT



Context vector (input to decoder):

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j$$

Mixture weights:

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})}$$

Alignment score (how well do input words near j match output words at position i):

$$e_{ij} = a(s_{i-1} h_j)$$

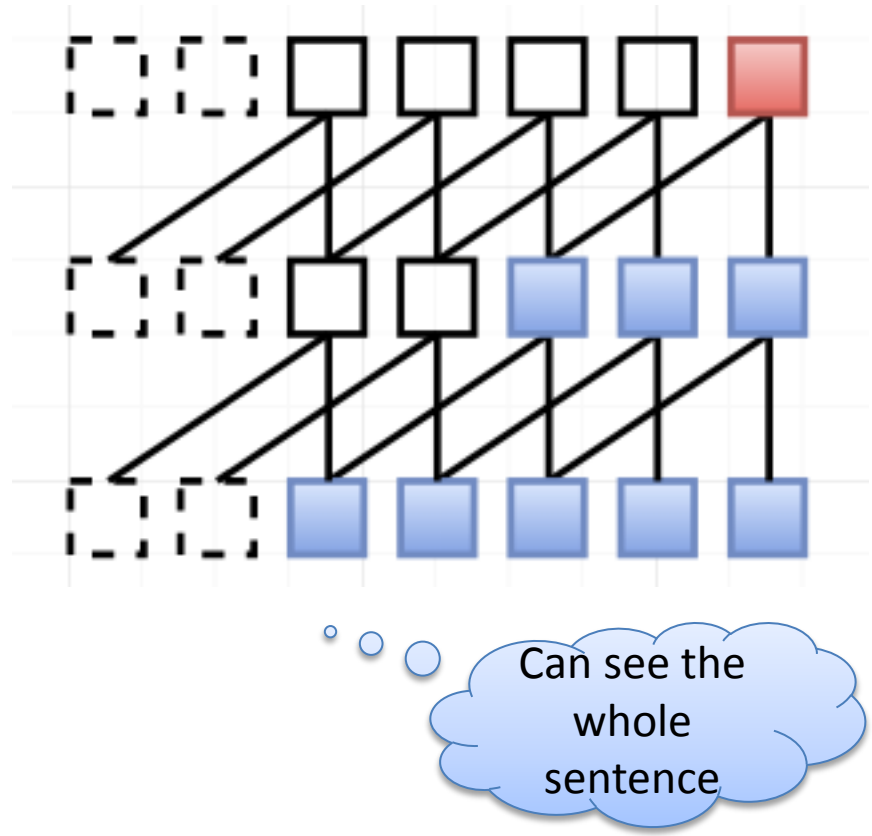
Bahdanau et al, “Neural Machine Translation by Jointly Learning to Align and Translate”, ICLR 2015

Alternative Neural MT Models

- LSTM seq2seq with attention: Bahdanau, 2014
- ConvS2S: (Facebook) May 2017
  - CNN: Convolutional Sequence to Sequence Learning
- Transformer: (Google) June 2017
  - Attention Is All You Need
- Universal Transformer: (Google) Aug 2018

ConvS2S

- Faster translation with multi-hop attention and gating
  - Do not depend on previous steps => parallelization
  - Hierarchical structure provides a shorter path to capture long-range dependencies
  - O(n) -> O(n/k)

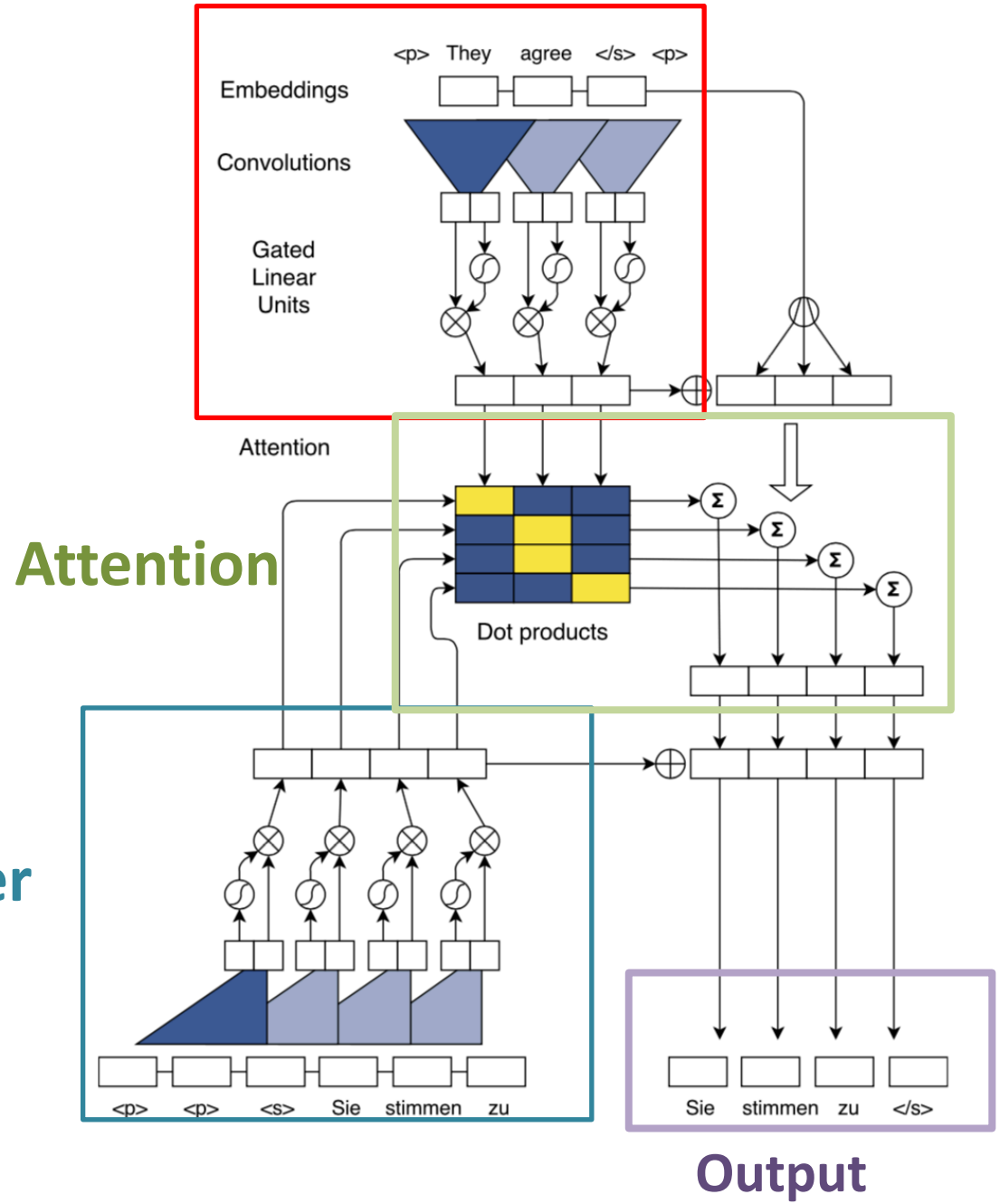


Convolutional Sequence to Sequence Learning. Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, Yann N. Dauphin. arXiv, 2017

Conv S2S Architecture

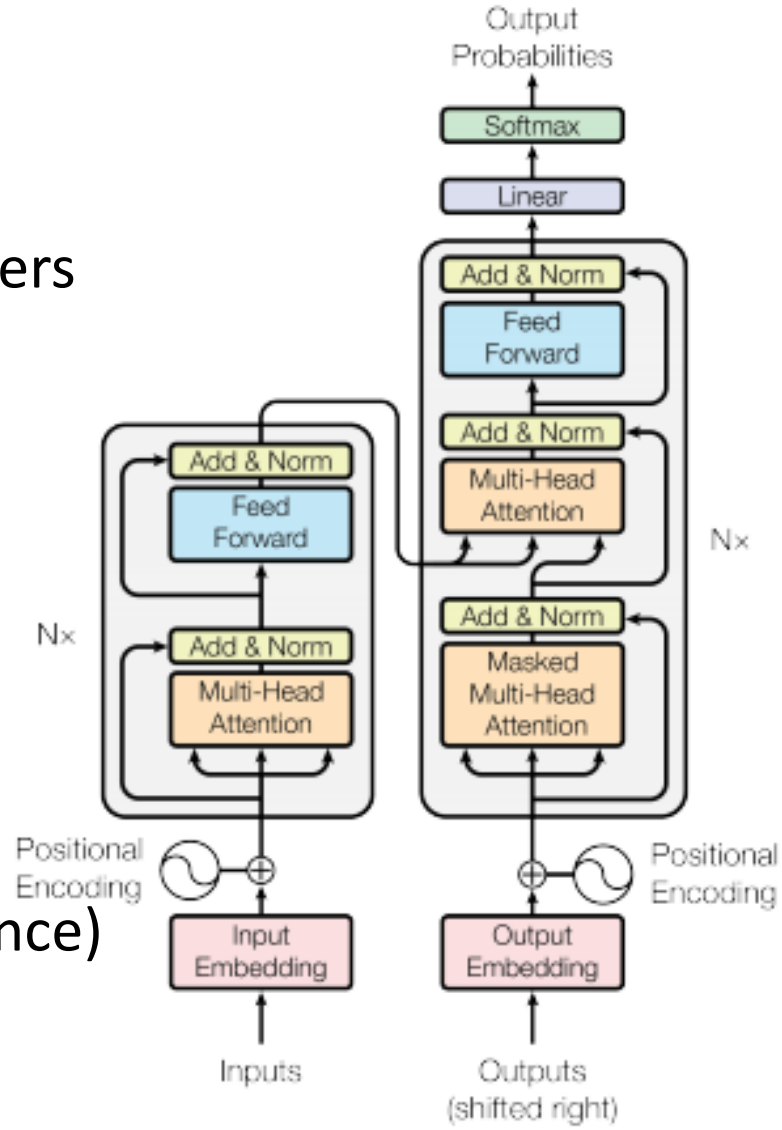
Encoder

- Input: word + position
- GLU (gated linear units)
  - Non-linearity
  - input field focus on fewer elements; gated
  - Sigmoid(b) control which inputs A are relevant



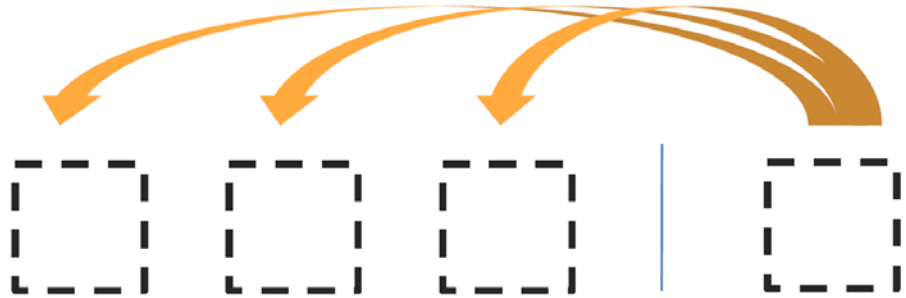
The Transformer

- Encoder: A stack of N=6 layers. Each layer has 2 sublayers
  1. A multi-head self-attention mechanism,
  2. A position-wise FC FFNN
- Decoder: N = 6 identical layers. 3 sublayers:
  3. Multi-head attention over the output of the encoder stack (Mask subsequent positions)
- Attention
- Output words fed back as input shifted right
- Positional Encoding (for introducing notion of sequence)
- Masked attention



Vaswani et al. “Attention is all you need”, arXiv 2017

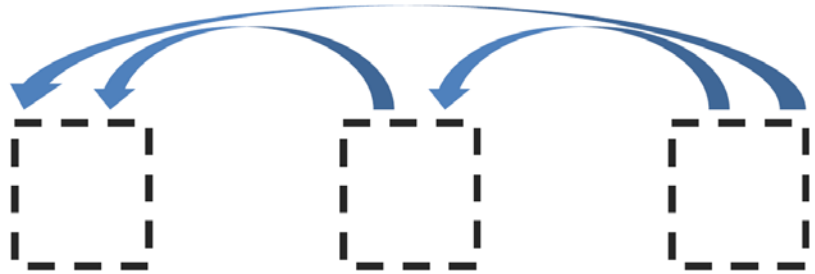
Attention in Transformer Networks



Encoder-Decoder Attention

Replaces word recurrence in encoder and decoder

Encoder Self-Attention



MaskedDecoder Self-Attention

Masking limits attention to earlier units:
y_i depends only on y_j for $j < i$.

image from Lukas Kaiser, Stanford NLP seminar

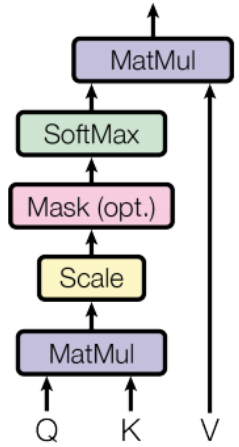
The Transformer

- Attention: Map query and a key-value pair to an output
- Encoder-decoder layer: the queries come from the previous decoder layer, and the memory keys and values come from the output of the encoder. (Similar to Bahdanau).
- Self-attention layer: all of the keys, values and queries come from the previous layer in the encoder.

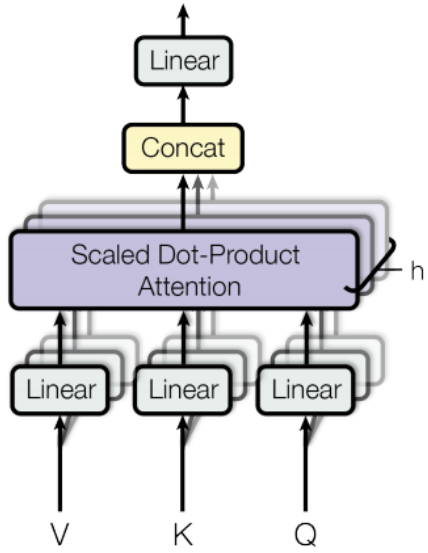
Q = query vector
K = key
V = value

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Scaled Dot-Product Attention



Multi-Head Attention



Multi-head attention:

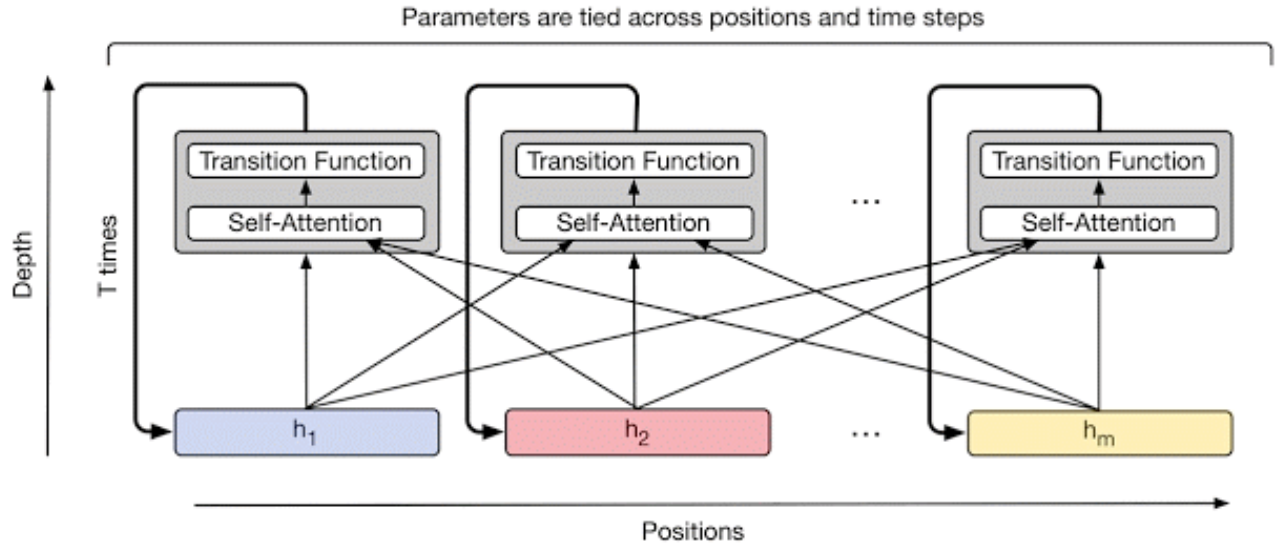
- Linearly project the Q, K and V h times with different, learned linear projections.
- These are concatenated and once again projected, resulting in the final values.

Multi-head attention allows the model to jointly attend to information from different representation subspaces at different positions

Universal Transformer

Extend the standard Transformer to be computationally universal (Turing complete) using parallel-in-time recurrence.

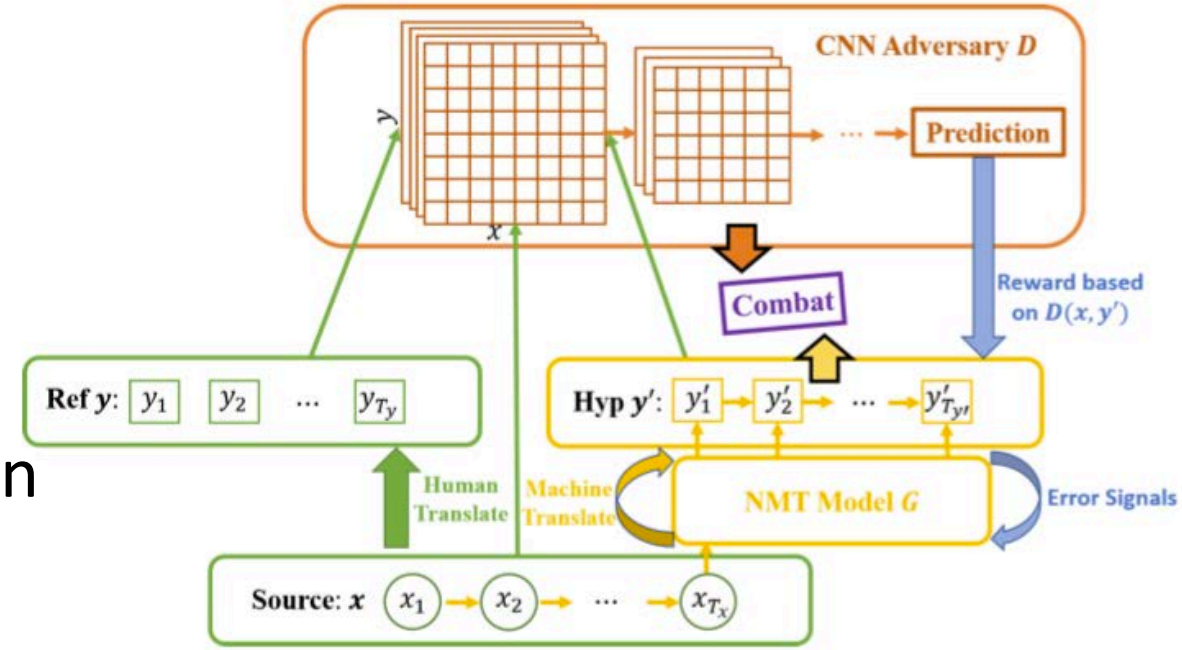
- Processes all symbols at the same time
- Refines its interpretation of every symbol in parallel over a variable number of recurrent processing steps using self-attention.



This parallel-in-time recurrence mechanism is both faster than the serial recurrence used in RNNs, and also makes the Universal Transformer more powerful than the standard feedforward Transformer.

Adversarial Machine Translation

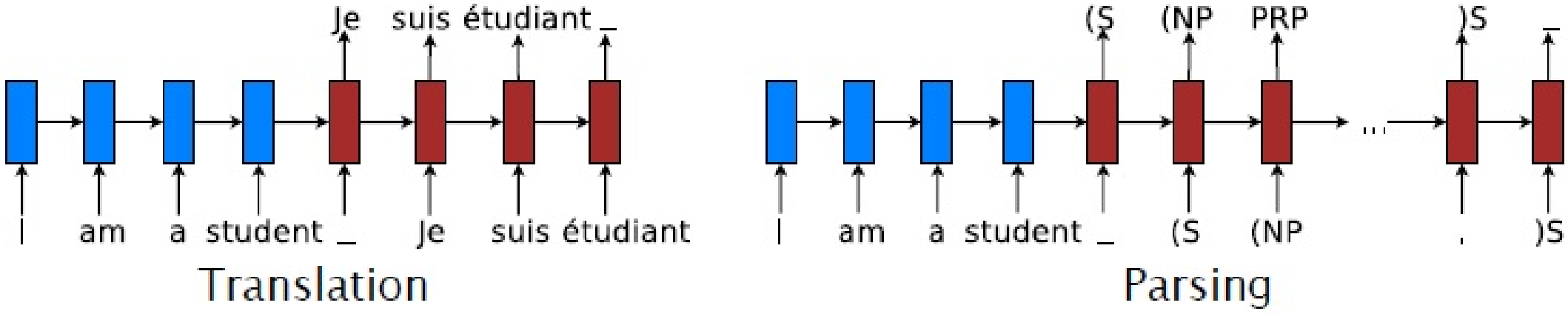
- Generator: any NMT model
- Discriminator: CNN
- The generator learns to generate correct and natural translation results so that the discriminator can not tell which is generated by the generator. And the discriminator learns to not be fooled by the generator.



Wu et al. Adversarial Neural Machine Translation. arXiv 2017

Utilize more data sources for MT

- Multi-lingual: learn from many language pairs
- Utilize monolingual data
- Multi-task learning: combine seq2seq tasks

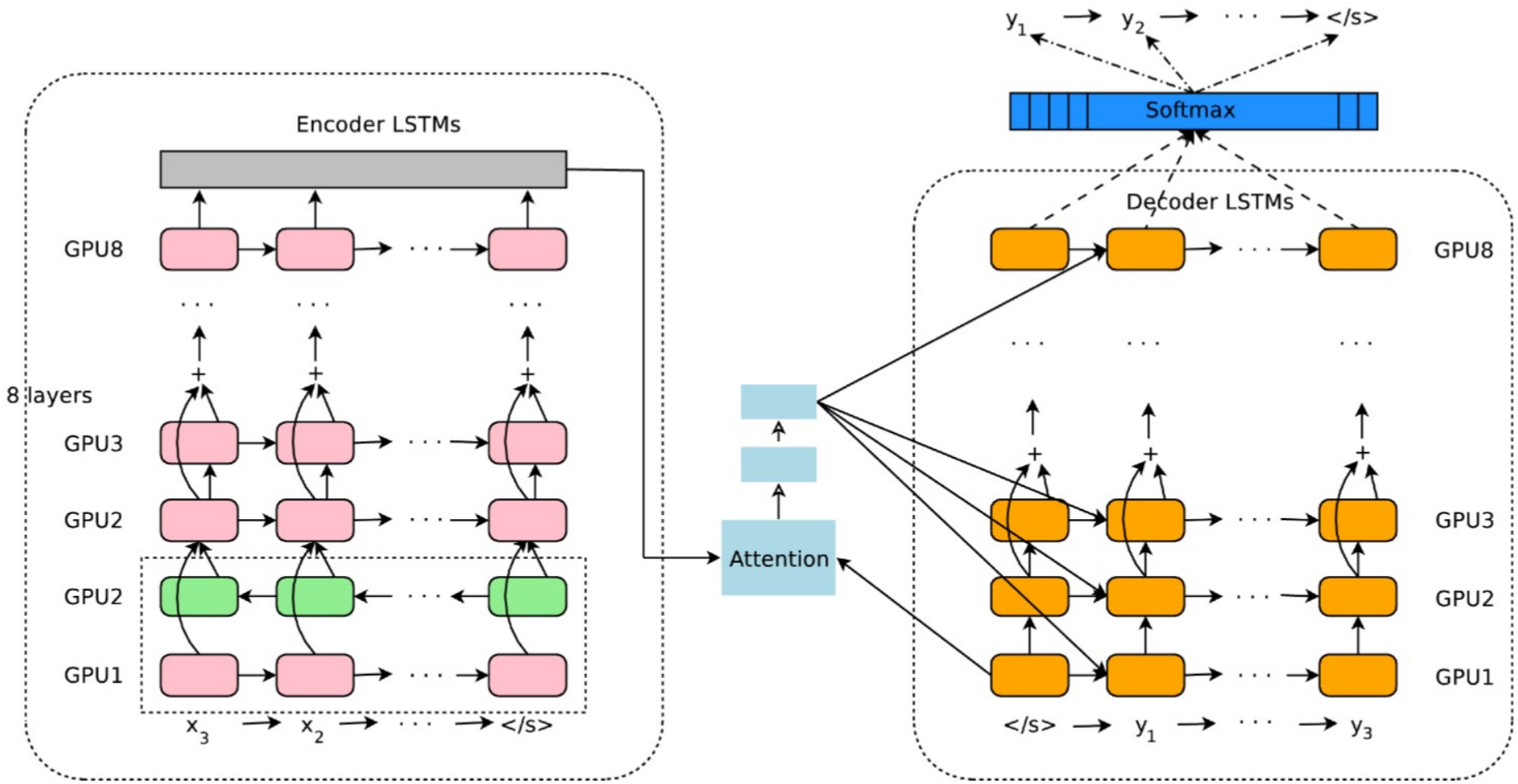


Multilingual Machine Translation

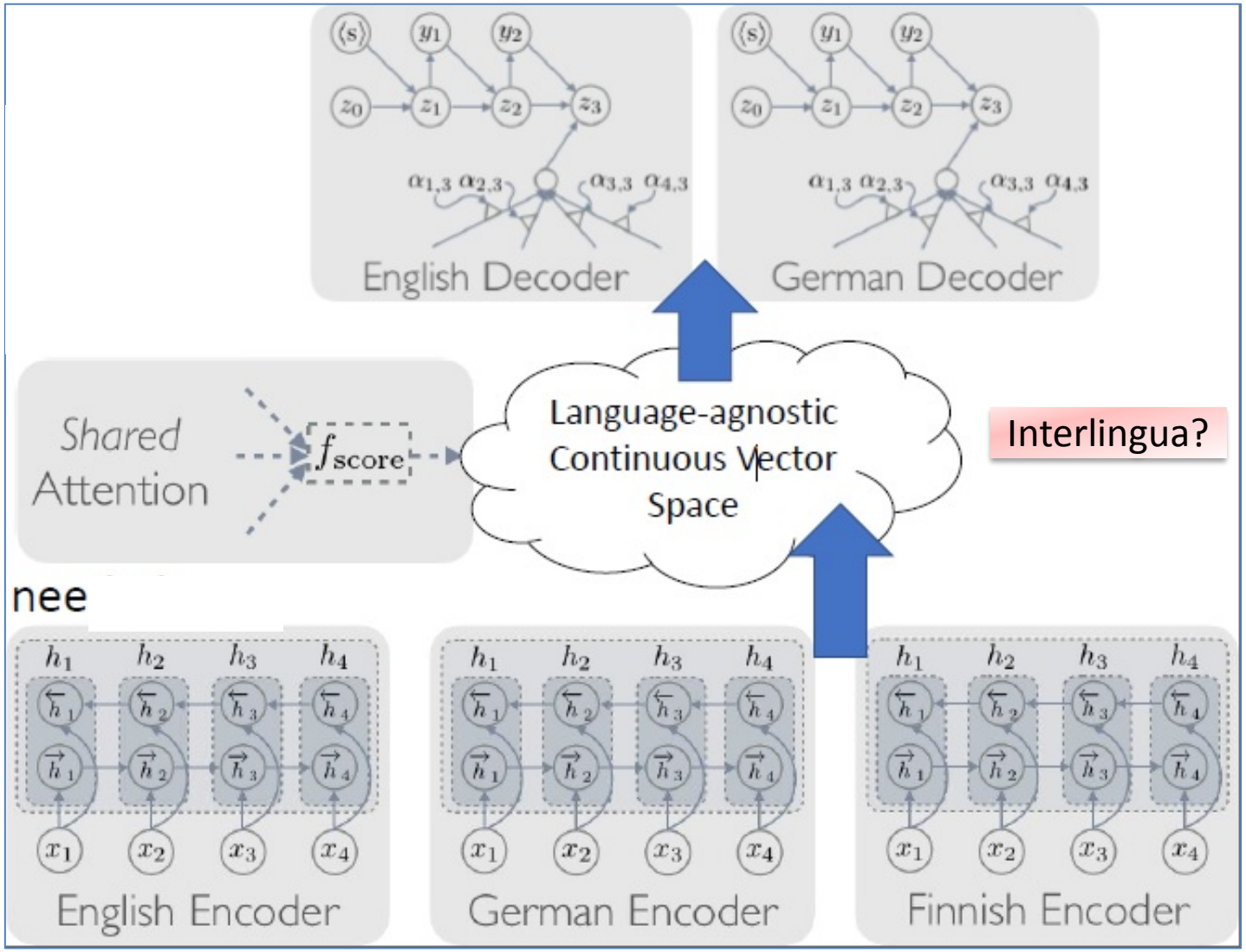
- Enabling Zero-Shot Translation; by Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, Jeffrey Dean; Transactions of the Association of Computational Linguistics – Volume 5, Issue 1, Oct 2017 (Google)

- Multi-Way, Multilingual Neural Machine Translation;
  by Firat, Kyunghyun Cho, B sankaran, FTY Vural, Yoshua Bengio, Journal; Computer Speech and Language, Volume 45 Issue C, September 2017

GNMT


Encoder LSTMs
GPU8
8 layers
GPU3
GPU2
GPU2
GPU1
x3
x2
</s>
Attention
y1
y2
</s>
Softmax
Decoder LSTMs
GPU8
GPU3
GPU2
GPU1
</s>
y1
y3

Multiway Multilingual Machine Translation



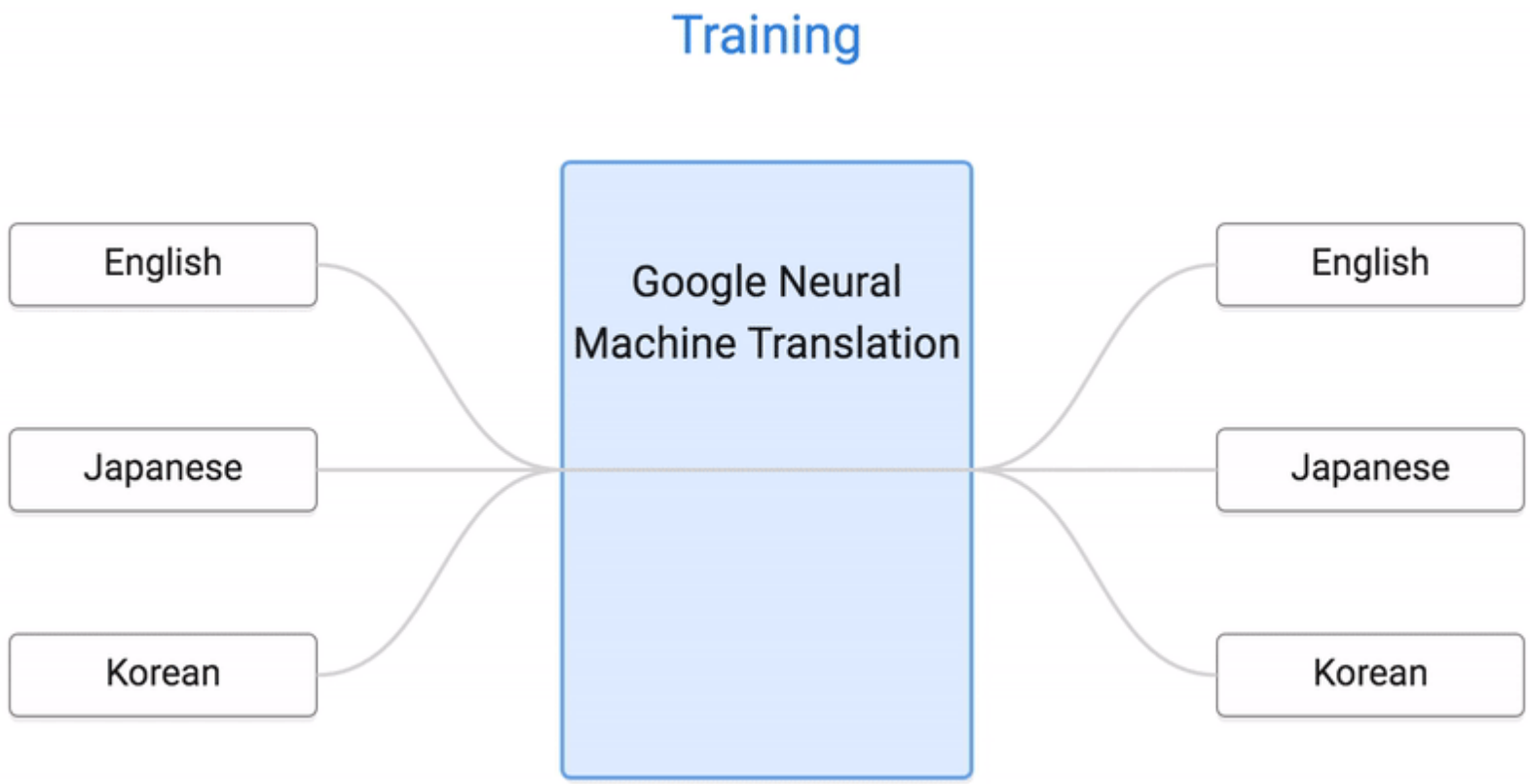
- One Encoder per source language
- One decoder per target language
- Shared Attention Mechanism

  Target hidden state, source context vector

  → Attention weight
- Bilingual sentence pairs only
- Each sentence pair activates/updates one encoder, decoder and shared attention

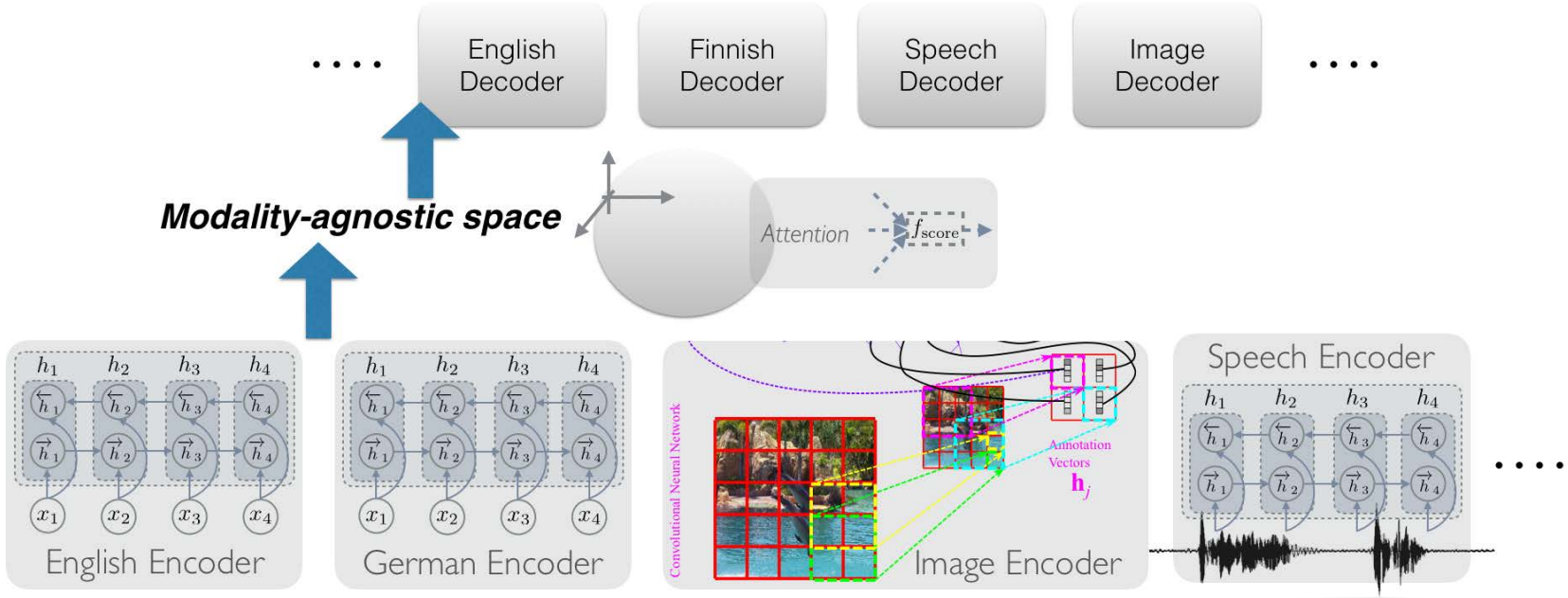
- **Low-resource translation**: Positive language transfer from high-resource to low-resource language pair-directions
- **Zero-resource translation**: Translation without any direct parallel resource

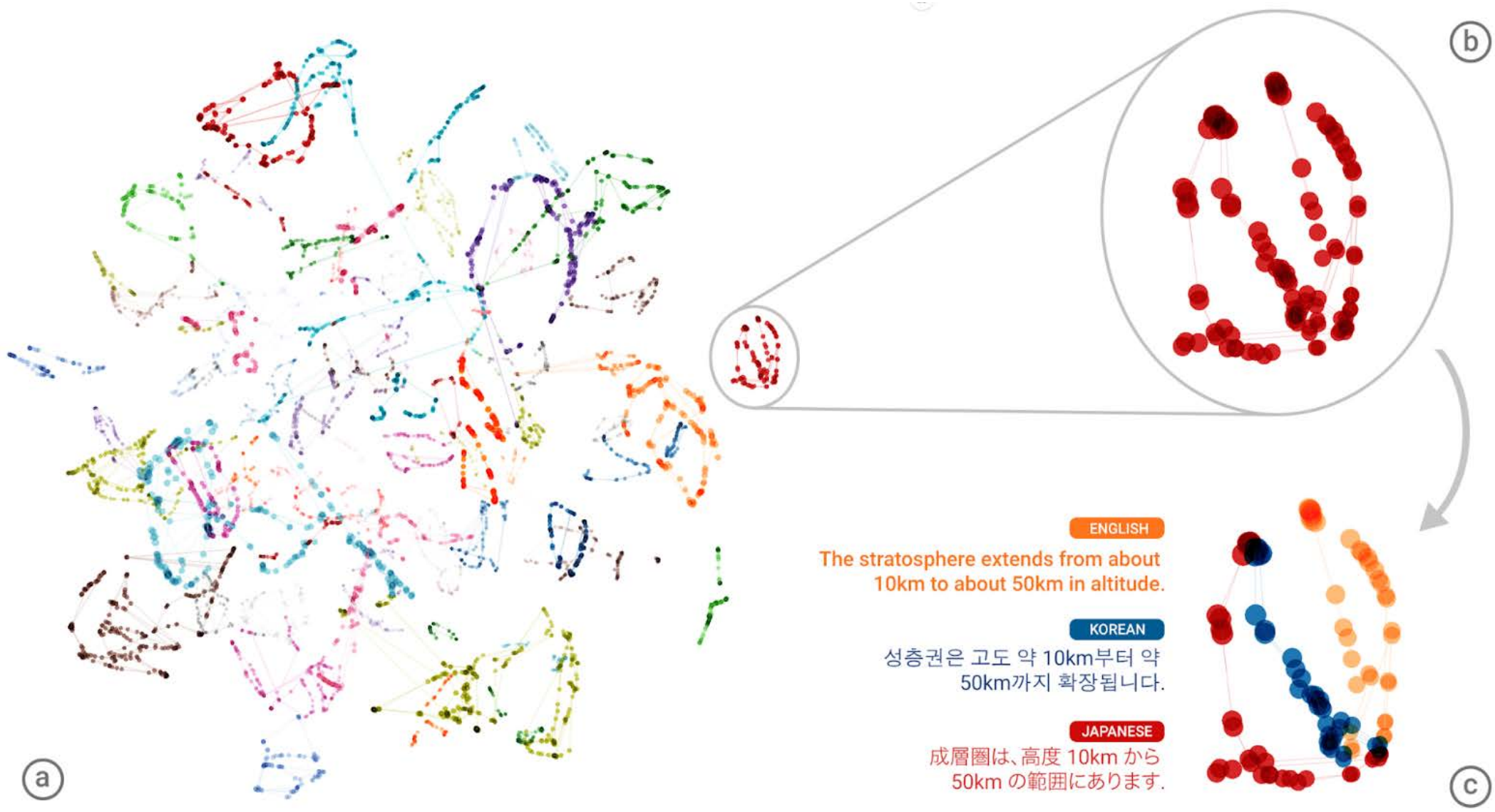
Zero-shot


Training
English
Japanese
Korean
Google Neural Machine Translation
English
Japanese
Korean

One model to learn them all

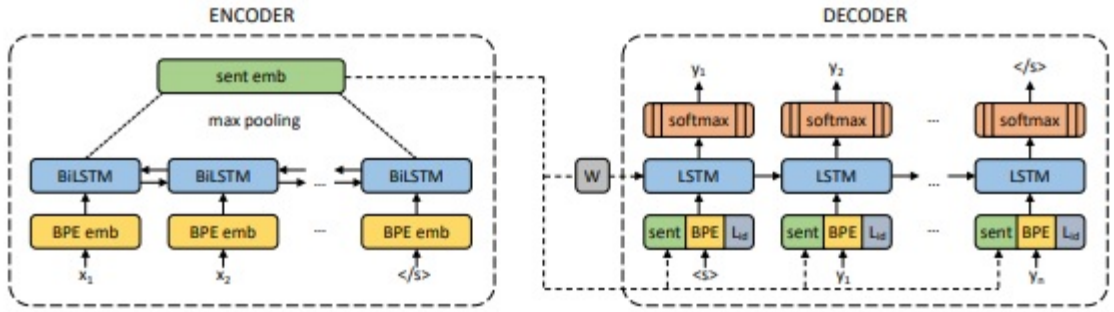
- Multi-modal, multi-task: Text, speech, image… all converging to a common paradigm.
- Or you may train them together to achieve zero-shot AI.
  - Translation without any direct parallel resource




a
b
c
ENGLISH
The stratosphere extends from about 10km to about 50km in altitude.
KOREAN
성층권은 고도 약 10km부터 약 50km까지 확장됩니다.
JAPANESE
成層圏は、高度 10km から 50km の範囲にあります.

Multilingual Sentence Embedding

- Learn multilingual sentence embedding
  - N-way aligned corpora
  - Use a shared encoder (3-layer BiLSTM) (Discard the decoders)



1. Use the cosine distance between sentences in different languages
2. Consider the margin between the cosine of a given sentence pair and that of its k nearest neighbors

$$\text{score}(x,y) = \text{margin}\left(\cos(x,y), \sum_{z \in NN_k(x)} \frac{\cos(x,z)}{2k} + \sum_{z \in NN_k(y)} \frac{\cos(y,z)}{2k}\right)$$

$$\text{margin1}(a,b) = 1 - b; \;\; \text{margin2}(a,b) = {}^{a}/_{b}$$

- Application: Filter noisy parallel data and mine for parallel data

1. Filtering and Mining Parallel Data in a Joint Multilingual Space; H Schwenk - arXiv:1805.09822, 2018
2. Margin-based Parallel Corpus Mining with Multilingual Sentence Embeddings; M Artetxe, H Schwenk - arXiv preprint arXiv:1811.01136, Nov 2018

Unsupervised Neural Machine Translation

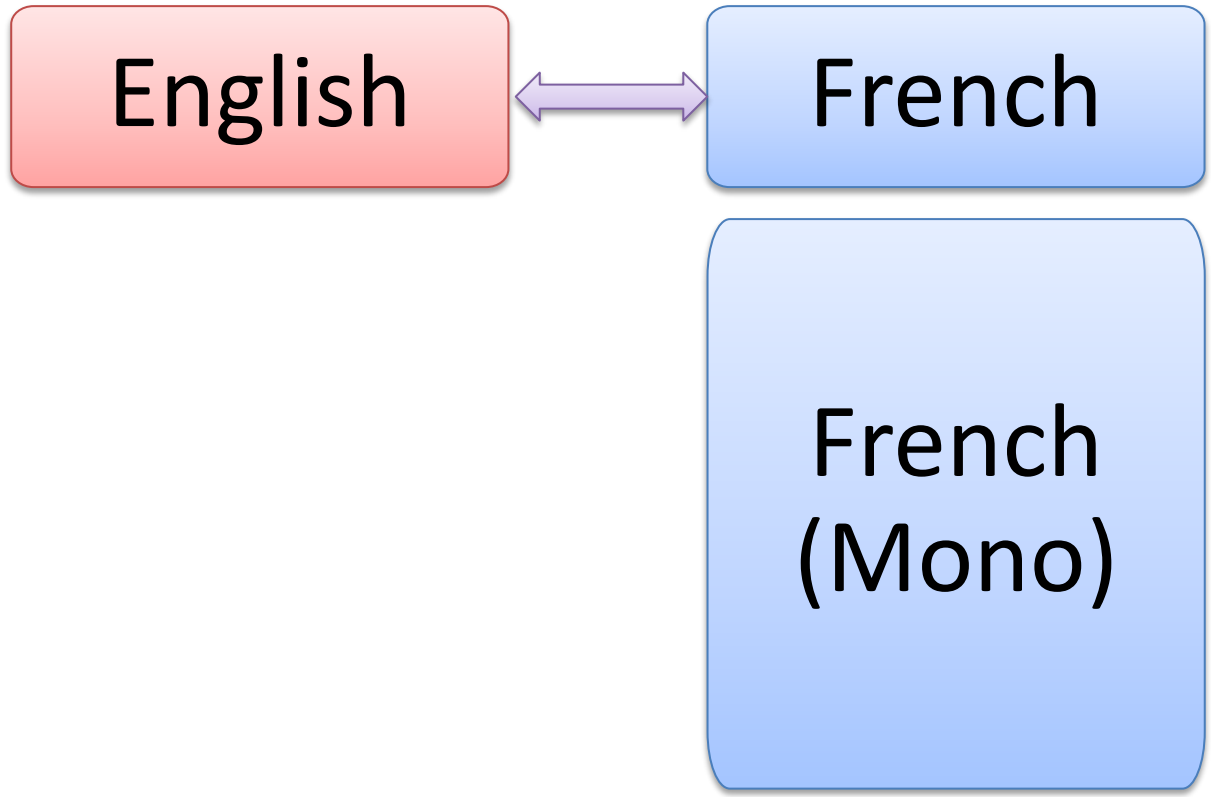
UNSUPERVISED MACHINE TRANSLATION

Unsupervised Machine Translation: Motivation

- Neural machine translation works well for language pairs with a lot of parallel data
- Performance drops when parallel data is scarce
- Monolingual data easier to get
- Can we learn without any parallel sentence?

Back-translation: Sennrich etal 2015

- Small Parallel Dataset or None
- Huge monolingual corpus in the target language

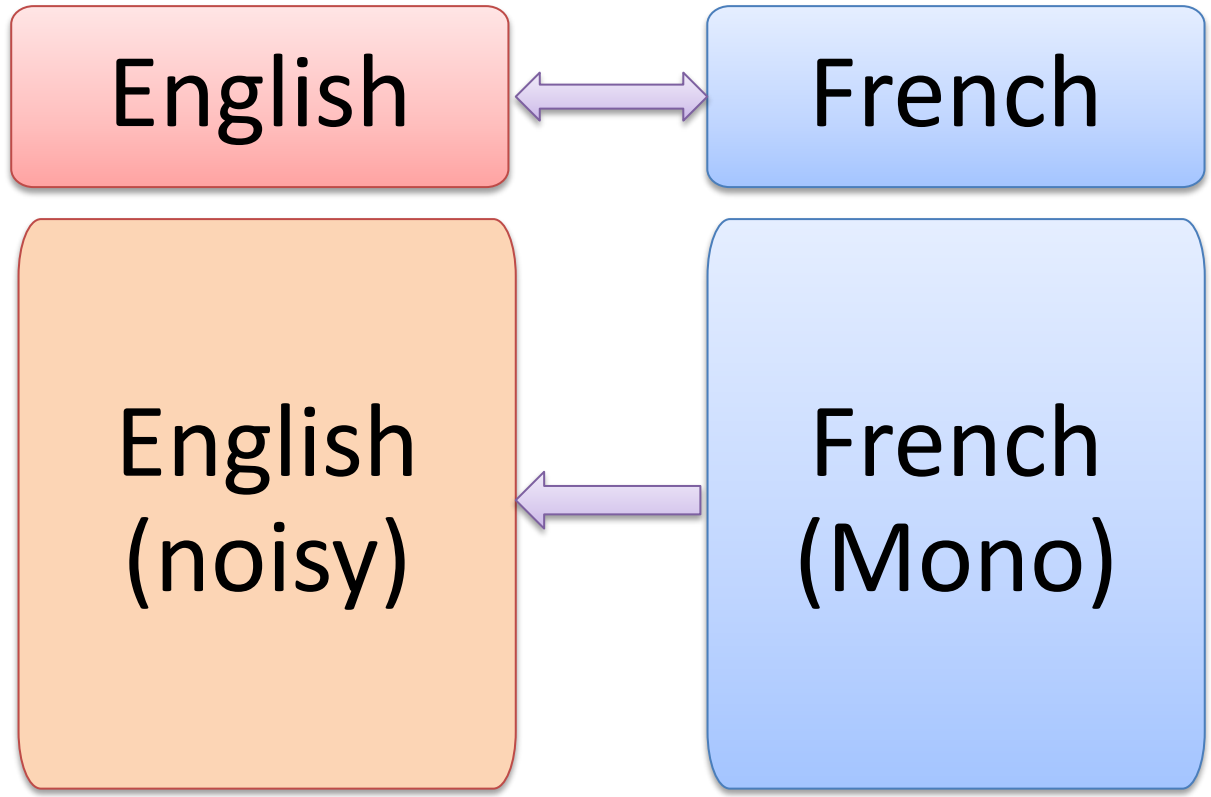


Back-translation: Sennrich etal 2015

- Small Parallel Dataset or None
- Huge monolingual corpus in the target language

- Train a (target → source) model Mt2s
- Use Mt2s to translate the target monolingual corpus

- Use the two parallel datasets to train Ms2t



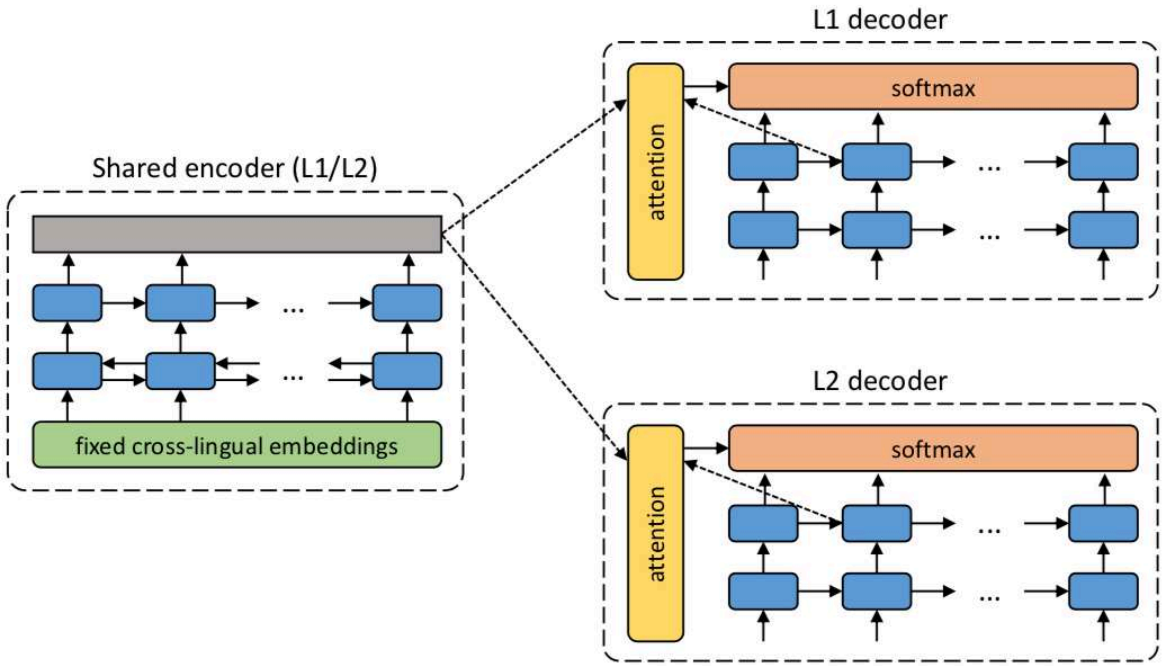
Unsupervised Neural MachineTranslation

- *Word Translation Without Parallel Data;* Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, Herve Jegou, ICLR 2018

1. *Unsupervised Machine Translation Using Monolingual Corpora Only;* Guillaume Lample, Alexis Conneau, Ludovic Denoyer, Marc'Aurelio Ranzato; ICLR 2018

2. *Unsupervised Neural Machine Translation*; Mikel Artetxe, Gorka Labaka & Eneko Agirre, Kyunghyun Cho; ICLR 2018

Unsupervised NMT



For each sentence in L1, trained in alternating steps:

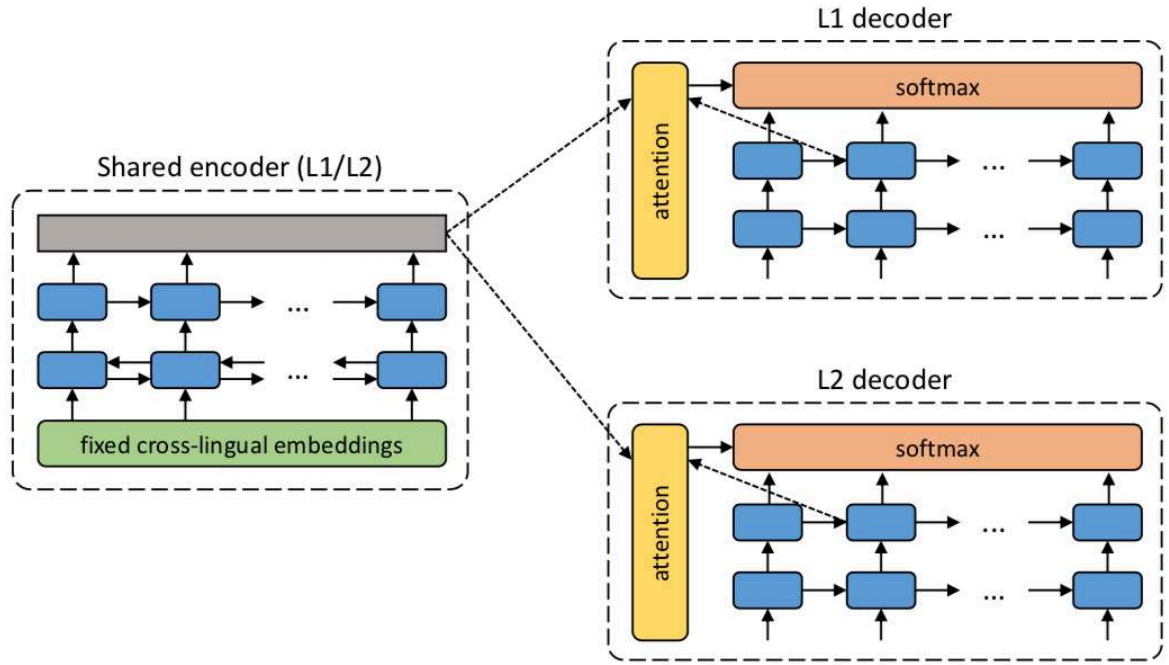
1. Denoising: encoding a noised version of the sentence (N/2 random swaps) with the shared encoder and reconstructing it with the L1 decoder
2. On-the-fly backtranslation: translates the sentence in inference mode (encoding it with the shared encoder and decoding it with the L2 decoder) and then optimizes the probability of encoding this translated sentence with the shared encoder and recovering the original sentence with the L1 decoder.

Alternate training between L1 and L2

Unsupervised Neural Machine Translation; Mikel Artetxe, Gorka Labaka & Eneko Agirre, Kyunghyun Cho; ICLR 2018

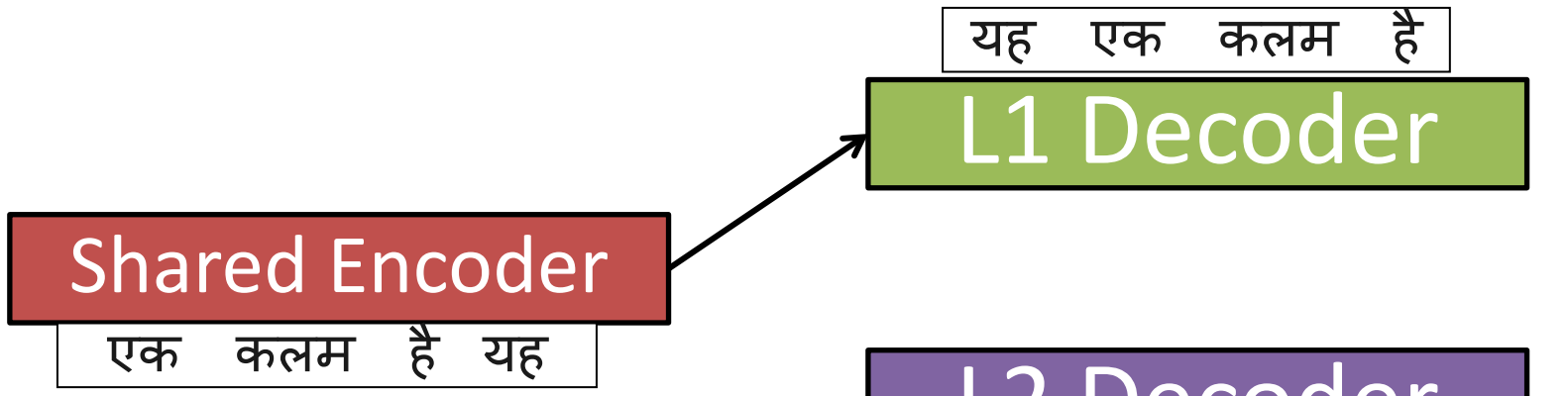
Unsupervised NMT

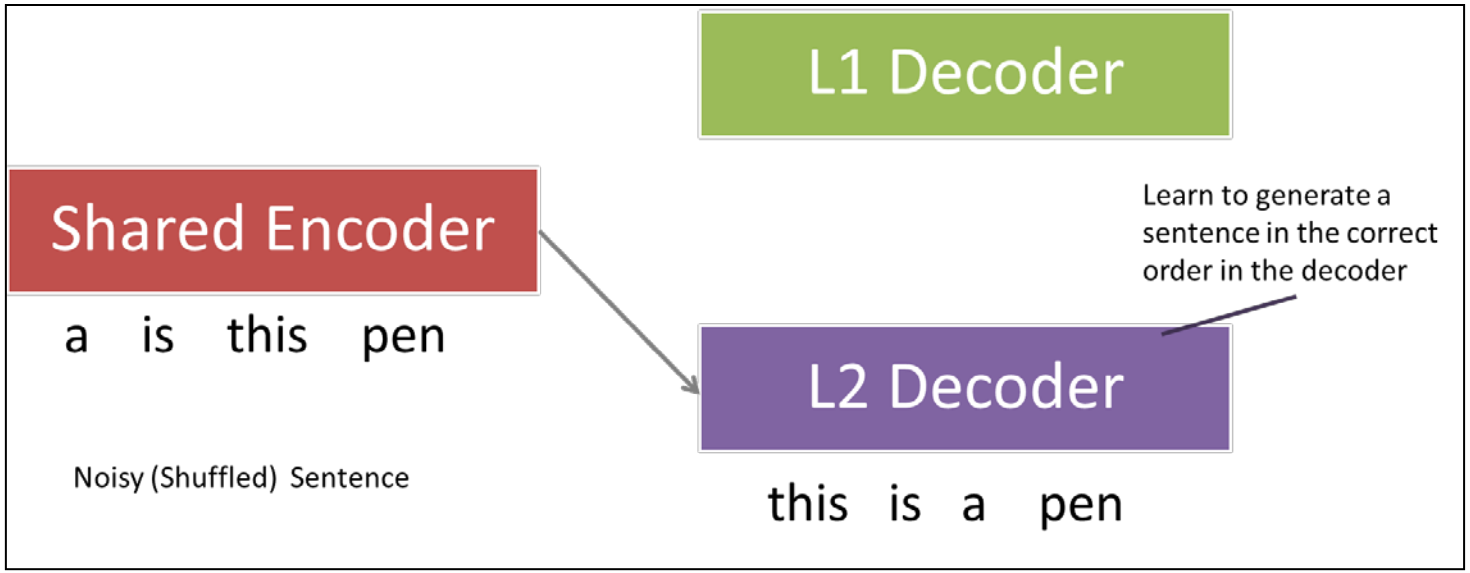
- System Architecture: Encoder-decoder with attention
  - Dual structure: Learn bi-directional transformation
  - Shared encoder: Encoder should learn to compose the embeddings of both languages in a language-independent fashion
  - Fixed embeddings in the encoder: use pre-trained crosslingual embedding in the encoder that are kept fixed during training.



Decoder should learn to decompose this representation into their corresponding language.

Denoising


यह एक कलम है
L1 Decoder
Shared Encoder
एक कलम है यह
L2 Decoder


L1 Decoder
Shared Encoder
Learn to generate a sentence in the correct order in the decoder
a is this pen
L2 Decoder
Noisy (Shuffled) Sentence
this is a pen

Procedure

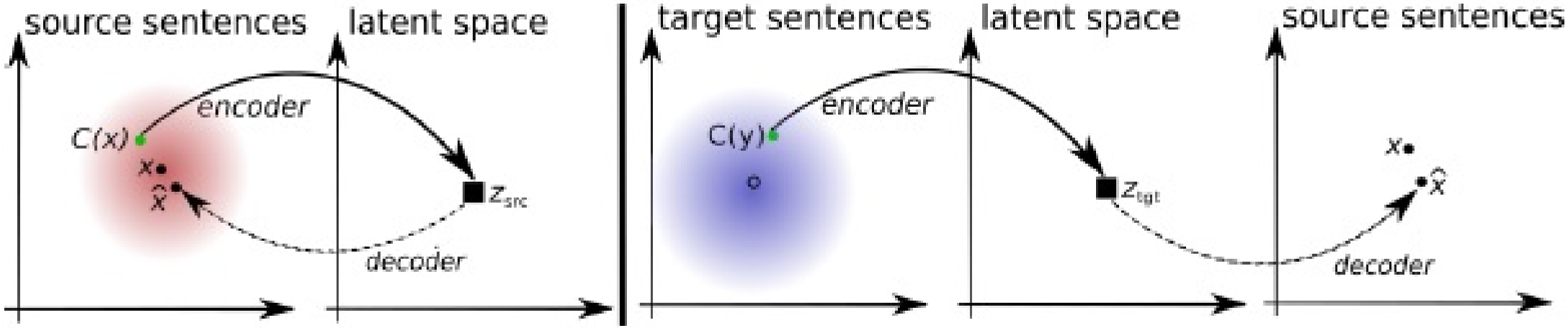
- Learn NMT for Language L1 and Language L2
  1. Denoising (L1)
  2. Denoising (L2)
  3. On-the-fly backtranslation (L1 -> L2)
     1. Translate from L1 to L2 (do not learn)
     2. Translation from L2 to original L1 statement
  4. On-the-fly backtranslation (L2 -> L1)
     1. Translate from L2 to L1 (do not learn)
     2. Translation from L1 to original L2 sentence

Results

		FR-EN	EN-FR	DE-EN	EN-DE
Unsupervised	1. Baseline (emb. nearest neighbor)	9.98	6.25	7.07	4.39
	2. Proposed (denoising)	7.28	5.33	3.64	2.40
	3. Proposed (+ backtranslation)	15.56	15.13	10.21	6.55
	4. Proposed (+ BPE)	15.56	14.36	10.16	6.89
Semi-supervised	5. Proposed (full) + 10k parallel	18.57	17.34	11.47	7.86
	6. Proposed (full) + 100k parallel	21.81	21.74	15.24	10.95
Supervised	7. Comparable NMT (10k parallel)	1.88	1.66	1.33	0.82
	8. Comparable NMT (100k parallel)	10.40	9.19	8.11	5.29
	9. Comparable NMT (full parallel)	20.48	19.89	15.04	11.05
	10. GNMT (Wu et al., 2016)	-	38.95	-	24.61

Unsupervised MT Using Monolingual Corpora only

- The model takes sentences from monolingual corpora in two different languages and maps them into the same latent space.
  i. Reconstruct a sentence from its noisy version (DAE)
  ii. Reconstruct a source sentence given its noisy translation in TL and vice versa



- Left: Model trained to reconstruct a sentence x as $\hat{x}$ from its noisy version $C(x)$
- Right: Trained to translate a sentence in TL. Input is $M^t(x)$ in TL as (C(y))

Denoising Auto-Encoding: Objective function

$$L_{auto}(\theta_{enc}, \theta_{dec}, Z, l) = E_{x \sim D_l, \hat{x} \sim d(e(C(x),l),l)}[\Delta(\hat{x}, x)]$$

- Δ: sum of token level cross-entropy losses
- Noise model:
  - Drop every word with probability p_{wd}
  - Shuffle words: Swap word pairs

Cross Domain Training:

$$L_{cd}(\theta_{enc}, \theta_{dec}, Z, l1, l2) = E_{x \sim D_{l1}, \hat{x} \sim d(e(C(M(x)),l2),l1)}[\Delta(\hat{x}, x)]$$

Adversarial Training :

Adversarial Training

- Jointly train a discriminator to classify the language given the encoding of source sentences and the encoding of target sentences (Ganin et al. 2016). In detail, the discriminator operates on a sequence of encoded hidden state vectors, and produces a binary prediction (0: source; 1: target). The discriminator is trained by minimizing the cross-entropy loss:

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

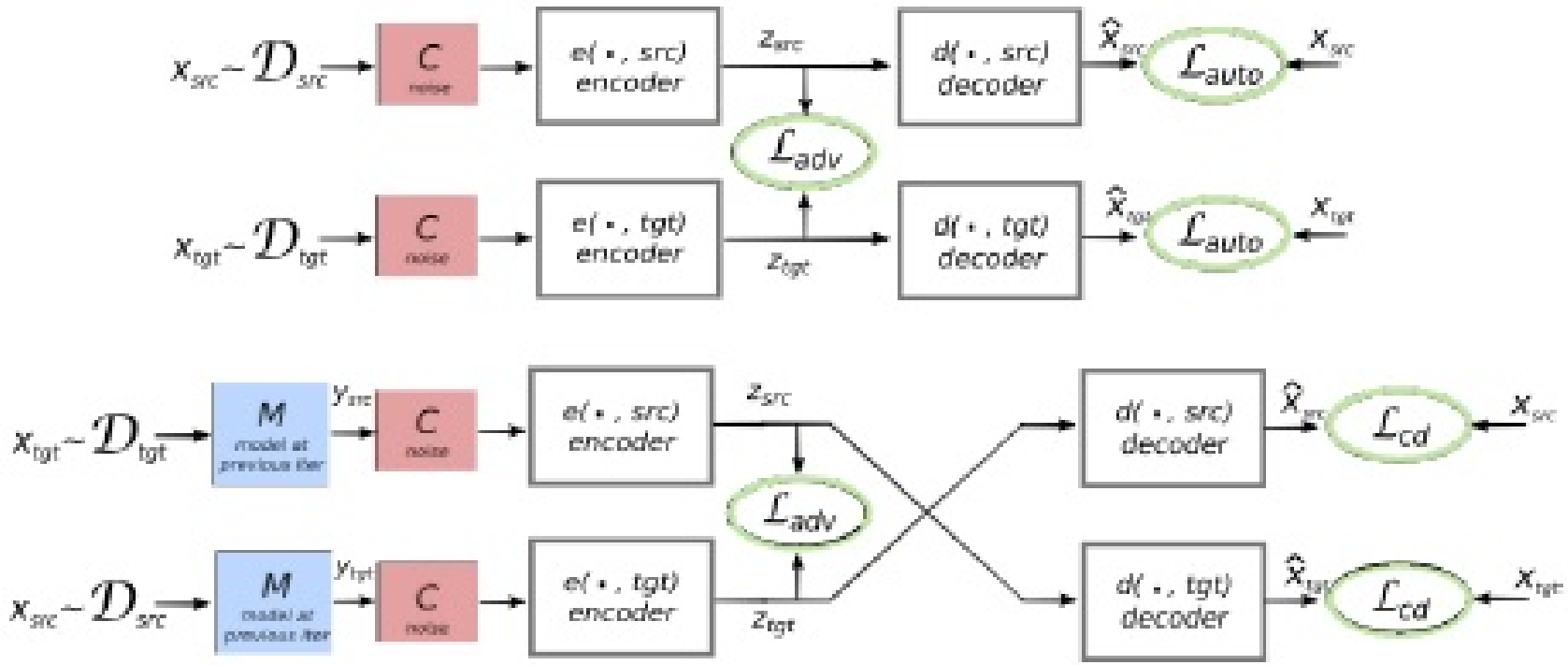
- The encoder is trained instead to fool the discriminator:

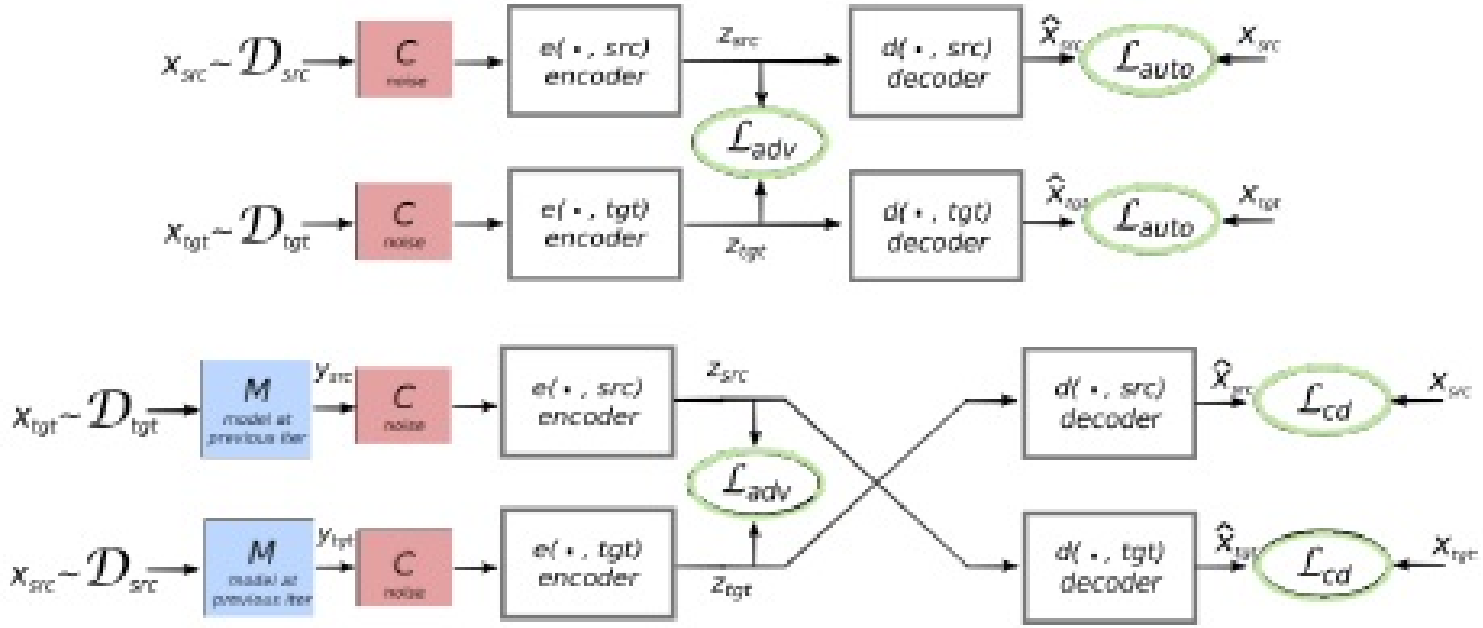
$$\mathcal{L}_{adv}(\theta_{enc},\mathcal{Z}|\theta_D) = -\mathbb{E}_{(x_i,\ell_i)}[\log p_D(\ell_j|e(x_i,\ell_i))] \quad (3)$$

with $\ell_j = \ell_1$ if $\ell_i = \ell_2$, and vice versa.

- Final Objective

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


$x_{src} \sim \mathcal{D}_{src}$
C
noise
e(• , src)
encoder
z_{src}
d(• , src)
decoder
$\hat{x}_{src}$
$\mathcal{L}_{auto}$
x_{src}
$\mathcal{L}_{adv}$
$x_{tgt} \sim \mathcal{D}_{tgt}$
C
noise
e(• , tgt)
encoder
z_{tgt}
d(• , tgt)
decoder
$\hat{x}_{tgt}$
$\mathcal{L}_{auto}$
x_{tgt}
$x_{tgt} \sim \mathcal{D}_{tgt}$
M
model at
previous iter
y_{src}
C
noise
e(• , src)
encoder
z_{src}
d(• , src)
decoder
$\hat{x}_{src}$
$\mathcal{L}_{cd}$
x_{src}
$\mathcal{L}_{adv}$
$x_{src} \sim \mathcal{D}_{src}$
M
model at
previous iter
y_{tgt}
C
noise
e(• , tgt)
encoder
z_{tgt}
d(• , tgt)
decoder
$\hat{x}_{tgt}$
$\mathcal{L}_{cd}$
x_{tgt}



- Start with an unsupervised naive translation model obtained by making word-by-word translation.
- At each iteration, the model are trained by minimizing an objective function that measures their ability to both reconstruct and translate from a noisy input sentence.

Results

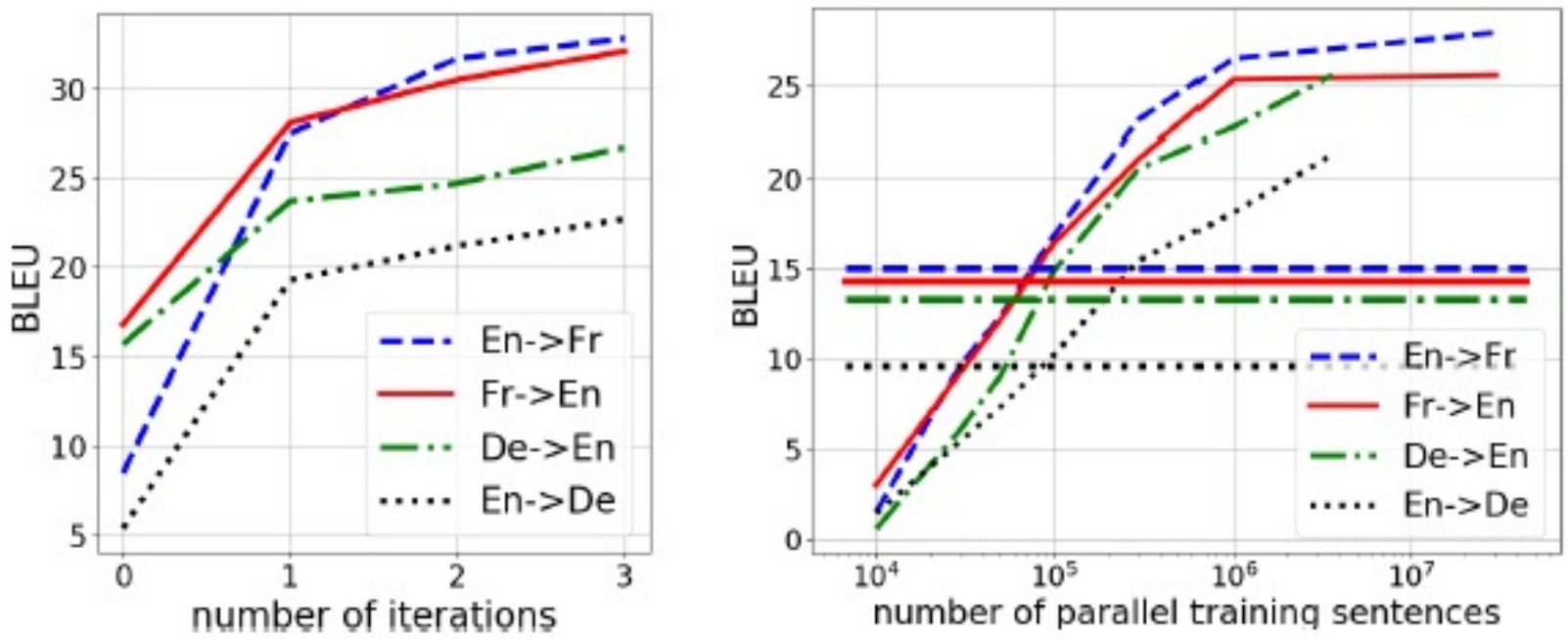


Figure 4: Left: BLEU as a function of the number of iterations of our algorithm on the Multi30k-Task1 datasets. Right: The curves show BLEU as a function of the amount of parallel data on WMT datasets. The unsupervised method which leverages about 15 million monolingual sentences in each language, achieves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

Phrase-Based & Neural Unsupervised Machine Translation, Lample 2018, EMNLP 2018

1. Language models: Learn language models Ps and Pt
2. Initial translation models: Leveraging Ps and Pt, learn two initial translation models, $P_{s\to t}{}^{(0)}$ and $P_{t\to s}{}^{(0)}$
3. for k=1 to N do
   4. Back-translation: Generate source and target sentences using the current translation models, $P_{s\to t}{}^{(k-1)}$ and $P_{t\to s}{}^{(k-1)}$
   5. Train new translation models $P_{s\to t}{}^{(k)}$ and $P_{t\to s}{}^{(k)}$ using the generated sentences and leveraging Ps and Pt
4. end

Phrase-Based & Neural Unsupervised Machine Translation, Lample 2018, EMNLP 2018

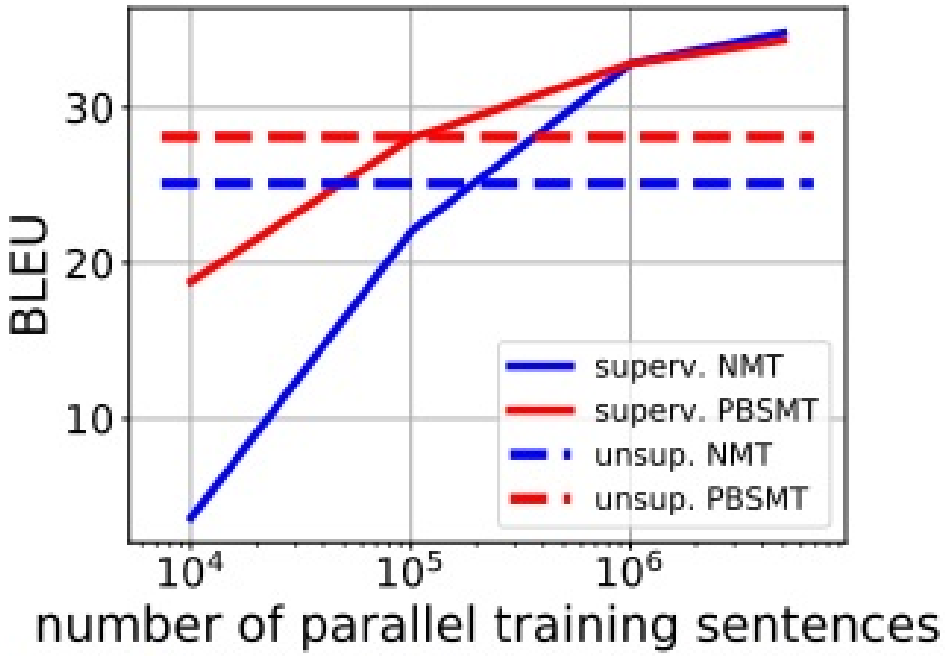


Figure 2: Comparison between supervised and unsupervised approaches on WMT'14 En-Fr, as we vary the number of parallel sentences for the supervised methods.

Model	en-fr	fr-en	en-de	de-en
(Artetxe et al., 2018)	15.1	15.6	-	-
(Lample et al., 2018)	15.0	14.3	9.6	13.3
(Yang et al., 2018)	17.0	15.6	10.9	14.6
NMT (LSTM)	24.5	23.7	14.7	19.6
NMT (Transformer)	25.1	24.2	17.2	21.0
PBSMT (Iter. 0)	16.2	17.5	11.0	15.6
PBSMT (Iter. n)	**28.1**	27.2	17.9	22.9
NMT + PBSMT	27.1	26.3	17.5	22.1
PBSMT + NMT	27.6	**27.7**	**20.2**	**25.2**

Table 2: **Comparison with previous approaches.** BLEU score for different models on the $en - fr$ and $en - de$ language pairs. Just using the unsupervised phrase table, and without back-translation (PBSMT (Iter. 0)), the PBSMT outperforms previous approaches. Combining PBSMT with NMT gives the best results.

Conclusion

- Systems may be greatly improved with limited supervised data.
- Great promise in working on various tasks involving low resource languages.
- Multilingual systems that work for multiple languages.
- Underlying principles may go well beyond MT.

Thank You

Question?