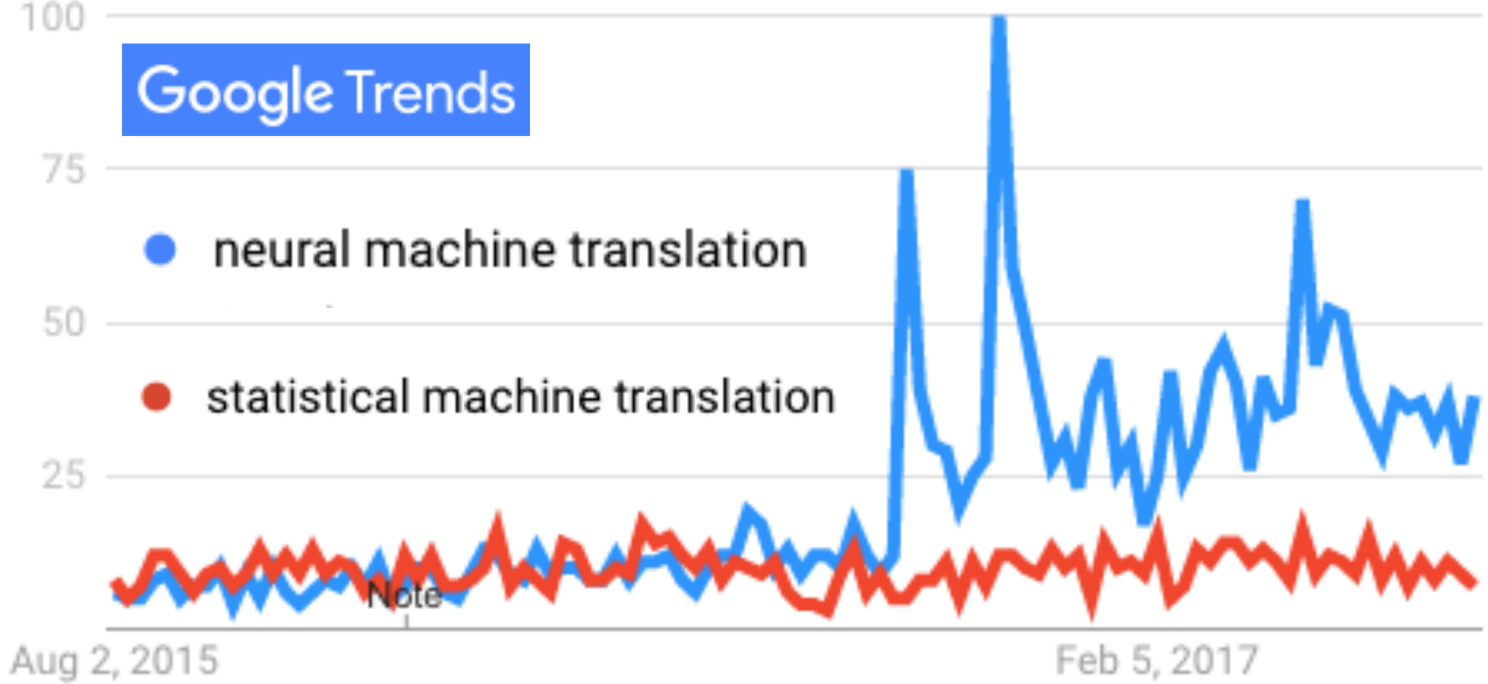
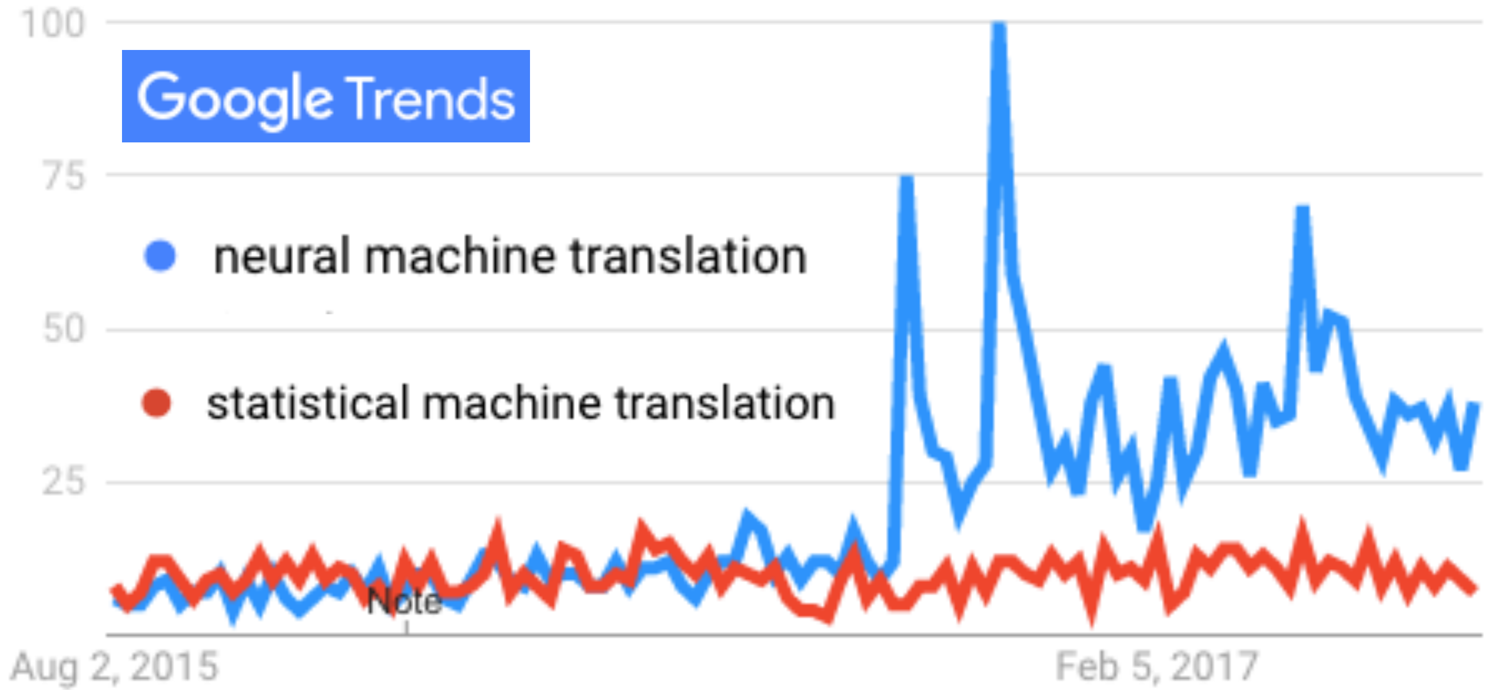
Unsupervised Neural Machine Translation: Are We There Yet?

Roee Aharoni
Natural Language Processing Lab,
Bar Ilan University

Neural Machine Translation

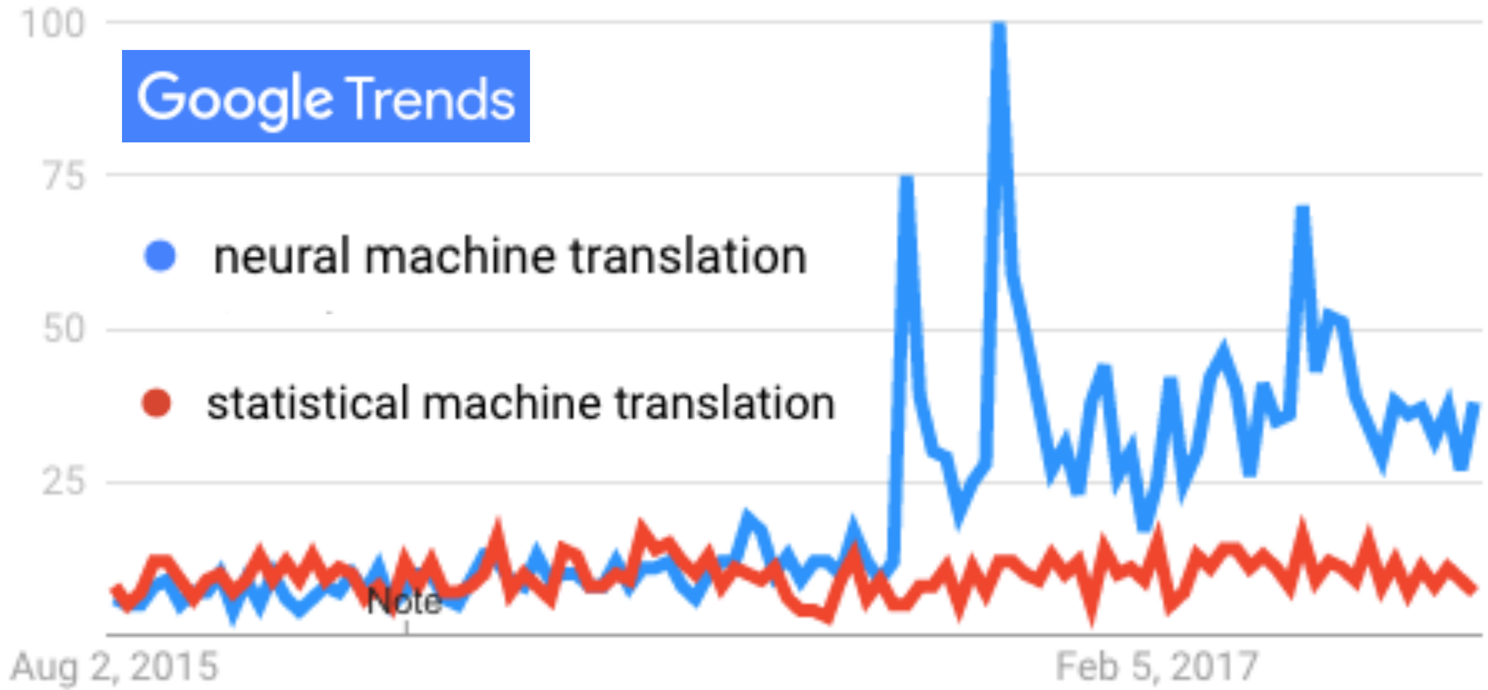


Neural Machine Translation



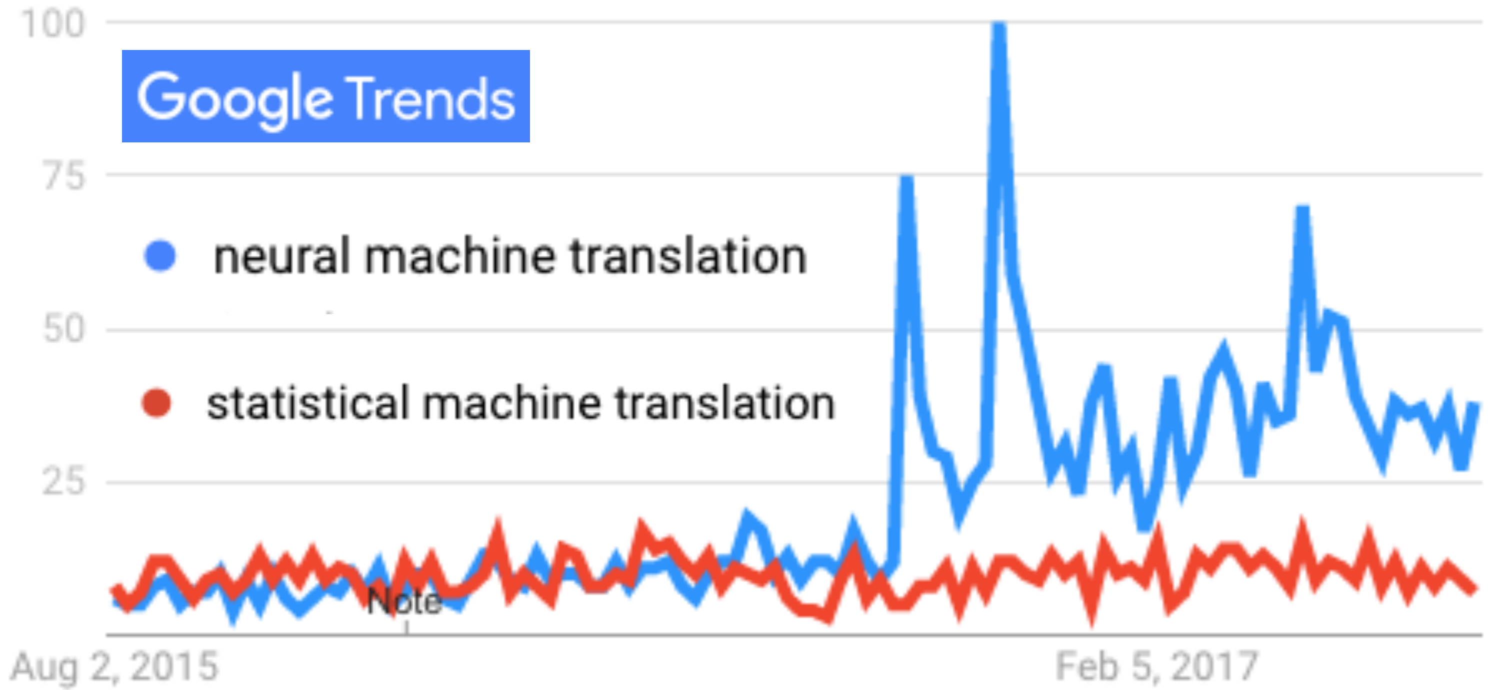
- Introduced in 2014

Neural Machine Translation



- Introduced in 2014
- Driving the current state of the art

Neural Machine Translation



- Introduced in 2014
- Driving the current state of the art
- Widely adopted in industry (Google Translate, Facebook…)

Sequence 2 Sequence Learning

Sequence 2 Sequence Learning

- Inspired by RNN language modeling

Sequence 2 Sequence Learning

- Inspired by RNN language modeling
- First (modern) models for NMT presented by Kalchbrenner et. al. 2013, Sutskever et al., 2014, Cho et al., 2014

Sequence 2 Sequence Learning

- Inspired by RNN language modeling
- First (modern) models for NMT presented by Kalchbrenner et. al. 2013, Sutskever et al., 2014, Cho et al., 2014
- 2 RNN’s, one for “reading” the input and one for “writing” the output (a.k.a the encoder-decoder architecture)

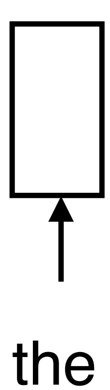
Sequence 2 Sequence Learning

- Inspired by RNN language modeling
- First (modern) models for NMT presented by Kalchbrenner et. al. 2013, Sutskever et al., 2014, Cho et al., 2014
- 2 RNN's, one for "reading" the input and one for "writing" the output (a.k.a the encoder-decoder architecture)

Encoder

Sequence 2 Sequence Learning

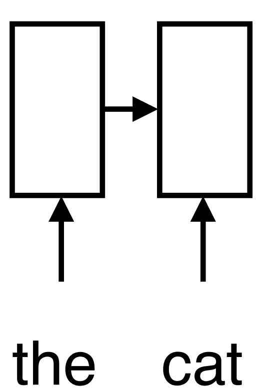
- Inspired by RNN language modeling
- First (modern) models for NMT presented by Kalchbrenner et. al. 2013, Sutskever et al., 2014, Cho et al., 2014
- 2 RNN’s, one for “reading” the input and one for “writing” the output (a.k.a the encoder-decoder architecture)



Encoder

Sequence 2 Sequence Learning

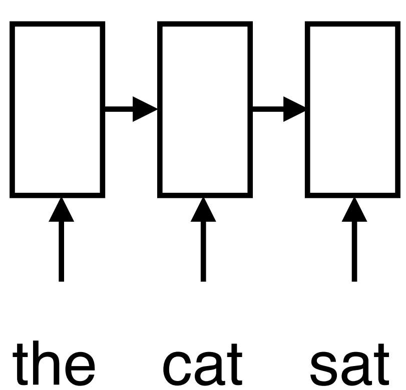
- Inspired by RNN language modeling
- First (modern) models for NMT presented by Kalchbrenner et. al. 2013, Sutskever et al., 2014, Cho et al., 2014
- 2 RNN's, one for "reading" the input and one for "writing" the output (a.k.a the encoder-decoder architecture)



Encoder

Sequence 2 Sequence Learning

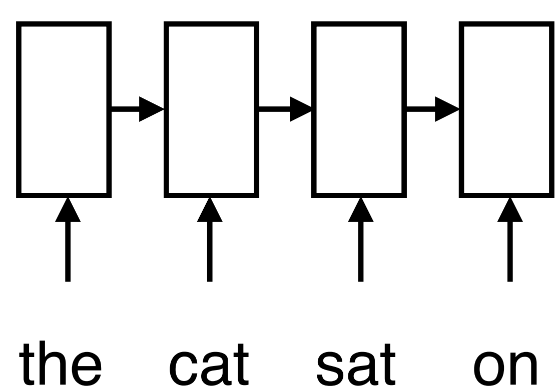
- Inspired by RNN language modeling
- First (modern) models for NMT presented by Kalchbrenner et. al. 2013, Sutskever et al., 2014, Cho et al., 2014
- 2 RNN’s, one for “reading” the input and one for “writing” the output (a.k.a the encoder-decoder architecture)



Encoder

Sequence 2 Sequence Learning

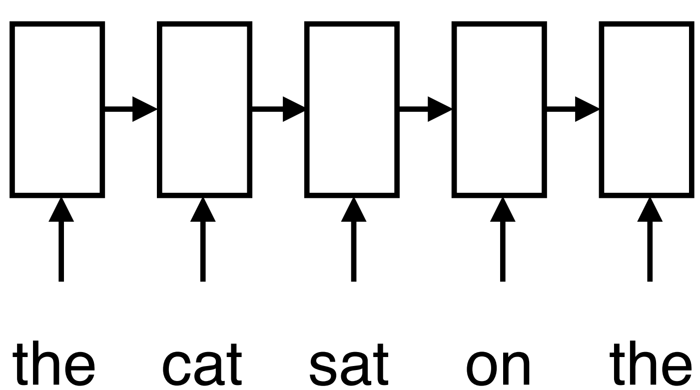
- Inspired by RNN language modeling
- First (modern) models for NMT presented by Kalchbrenner et. al. 2013, Sutskever et al., 2014, Cho et al., 2014
- 2 RNN’s, one for “reading” the input and one for “writing” the output (a.k.a the encoder-decoder architecture)



Encoder

Sequence 2 Sequence Learning

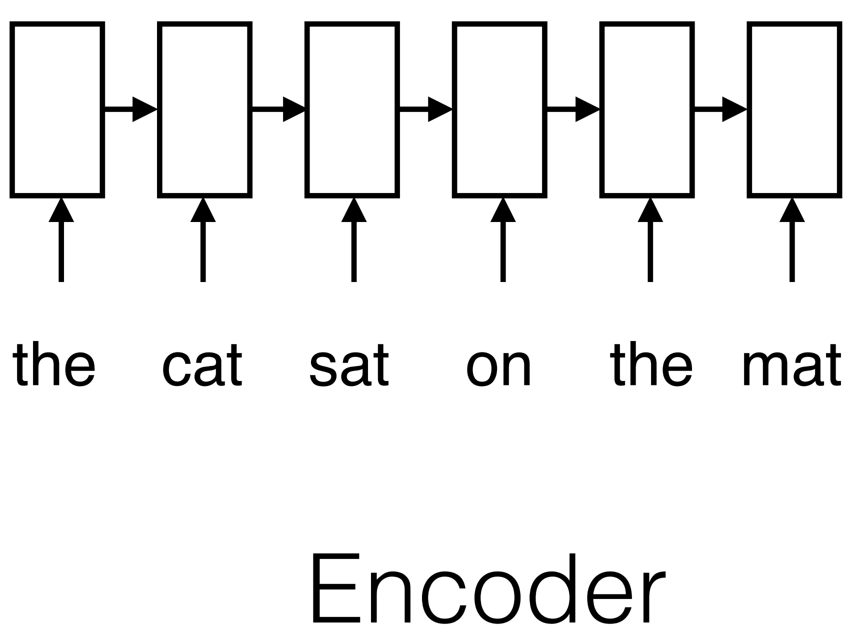
- Inspired by RNN language modeling
- First (modern) models for NMT presented by Kalchbrenner et. al. 2013, Sutskever et al., 2014, Cho et al., 2014
- 2 RNN's, one for "reading" the input and one for "writing" the output (a.k.a the encoder-decoder architecture)



Encoder

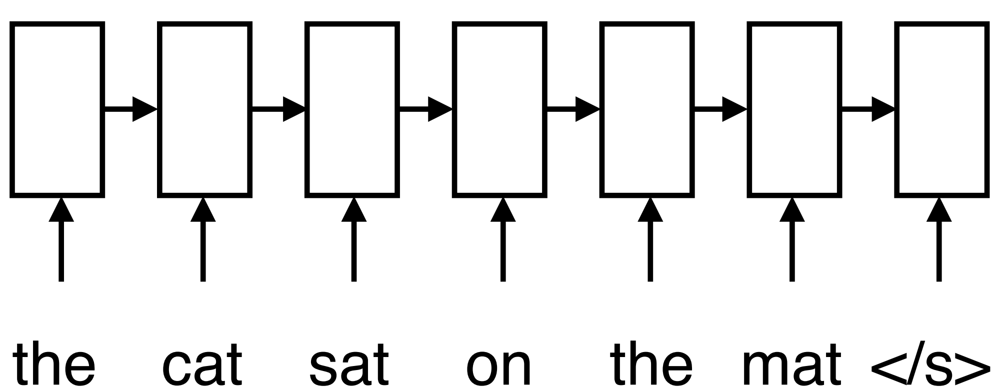
Sequence 2 Sequence Learning

- Inspired by RNN language modeling
- First (modern) models for NMT presented by Kalchbrenner et. al. 2013, Sutskever et al., 2014, Cho et al., 2014
- 2 RNN’s, one for “reading” the input and one for “writing” the output (a.k.a the encoder-decoder architecture)



Sequence 2 Sequence Learning

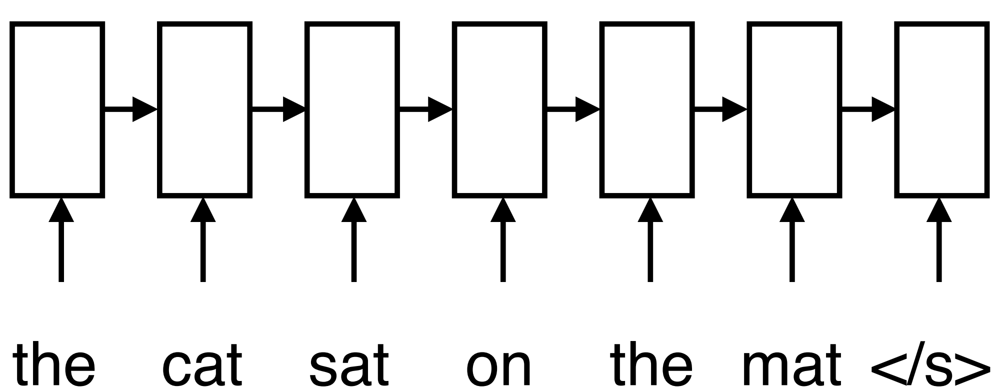
- Inspired by RNN language modeling
- First (modern) models for NMT presented by Kalchbrenner et. al. 2013, Sutskever et al., 2014, Cho et al., 2014
- 2 RNN’s, one for “reading” the input and one for “writing” the output (a.k.a the encoder-decoder architecture)



Encoder

Sequence 2 Sequence Learning

- Inspired by RNN language modeling
- First (modern) models for NMT presented by Kalchbrenner et. al. 2013, Sutskever et al., 2014, Cho et al., 2014
- 2 RNN's, one for "reading" the input and one for "writing" the output (a.k.a the encoder-decoder architecture)

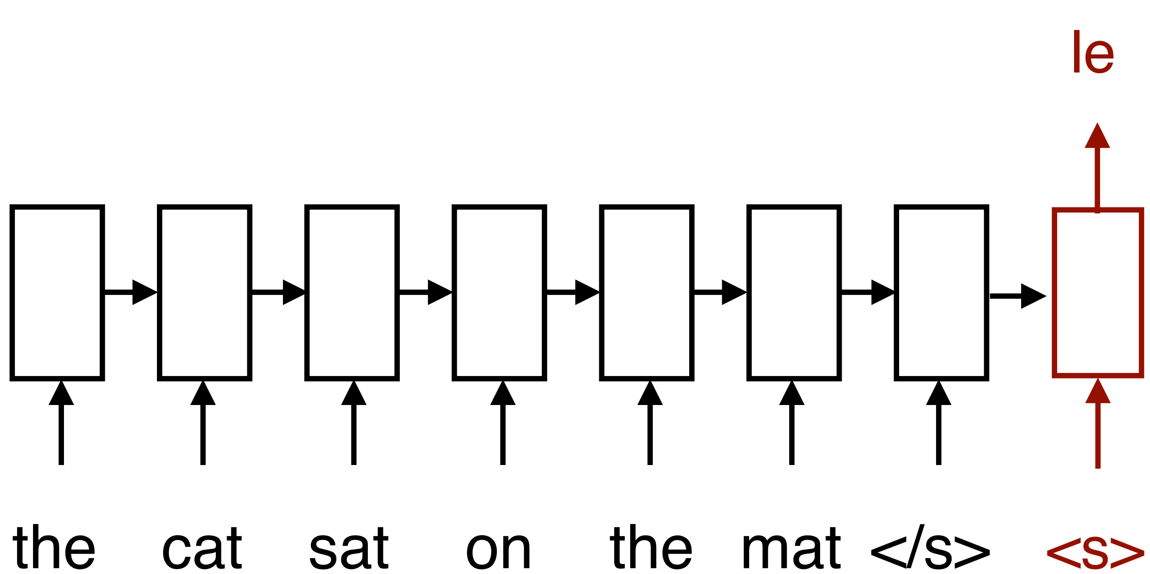


Encoder

Decoder

Sequence 2 Sequence Learning

- Inspired by RNN language modeling
- First (modern) models for NMT presented by Kalchbrenner et. al. 2013, Sutskever et al., 2014, Cho et al., 2014
- 2 RNN’s, one for “reading” the input and one for “writing” the output (a.k.a the encoder-decoder architecture)

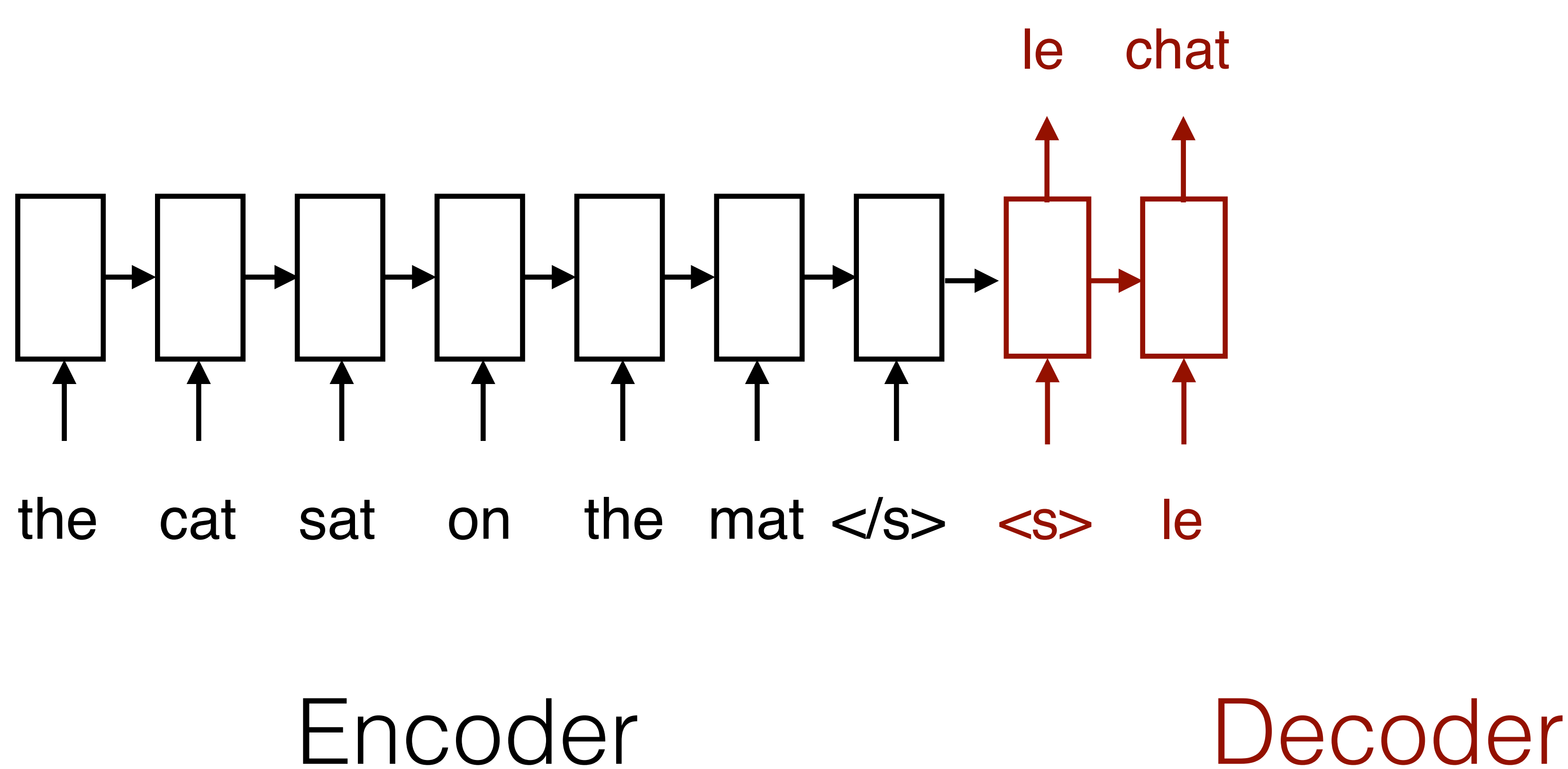


Encoder

Decoder

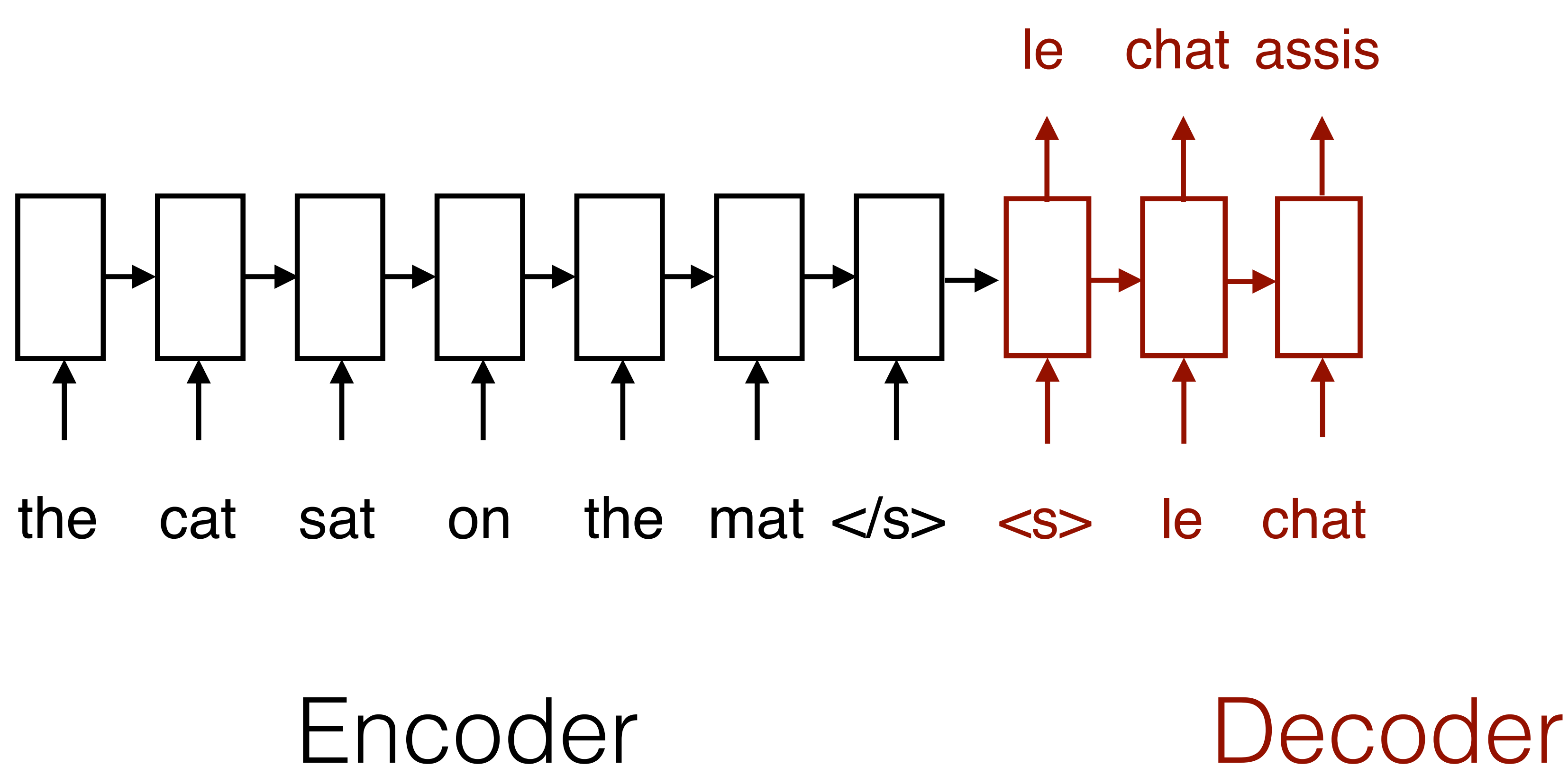
Sequence 2 Sequence Learning

- Inspired by RNN language modeling
- First (modern) models for NMT presented by Kalchbrenner et. al. 2013, Sutskever et al., 2014, Cho et al., 2014
- 2 RNN's, one for "reading" the input and one for "writing" the output (a.k.a the encoder-decoder architecture)



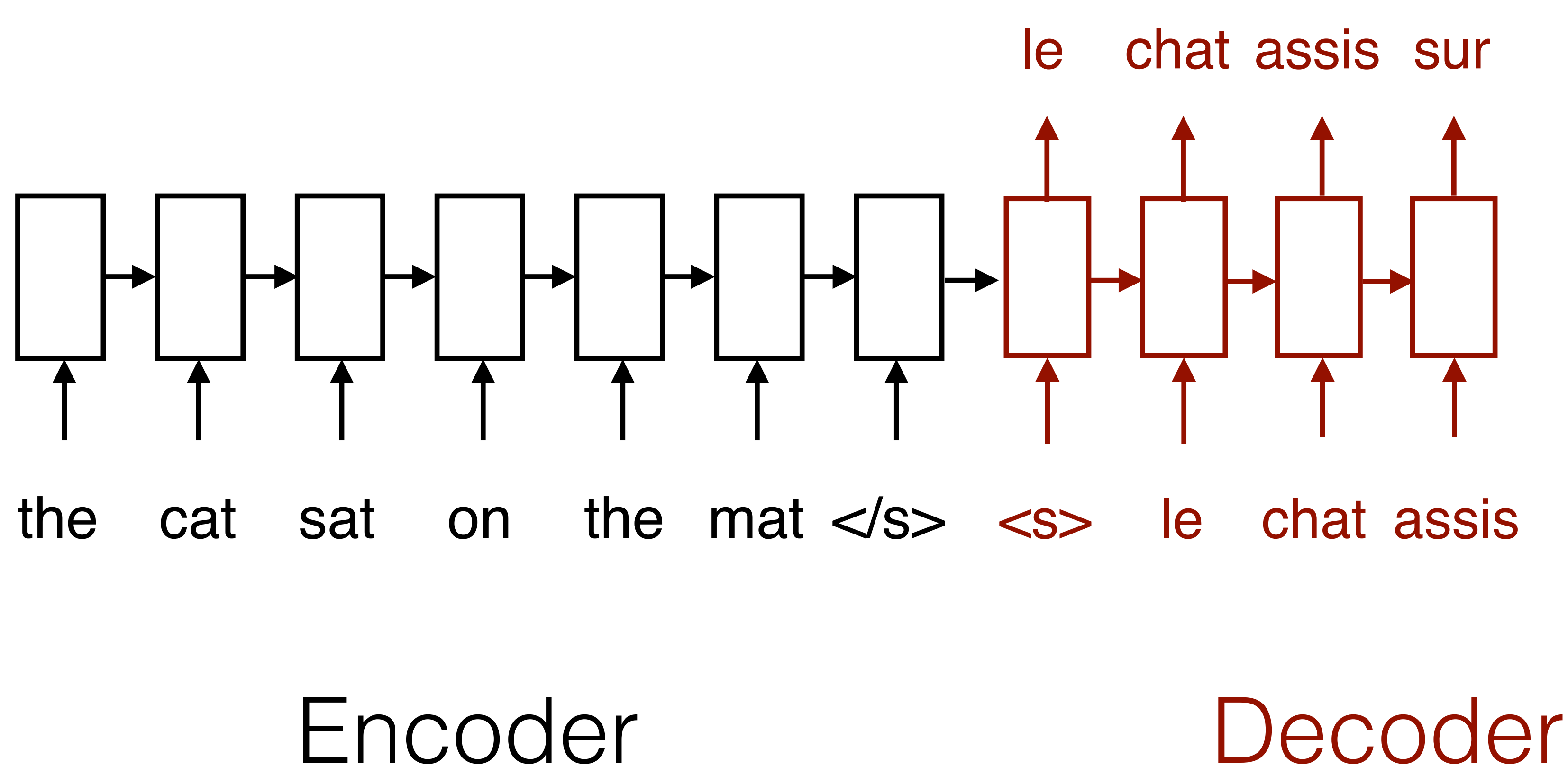
Sequence 2 Sequence Learning

- Inspired by RNN language modeling
- First (modern) models for NMT presented by Kalchbrenner et. al. 2013, Sutskever et al., 2014, Cho et al., 2014
- 2 RNN’s, one for “reading” the input and one for “writing” the output (a.k.a the encoder-decoder architecture)



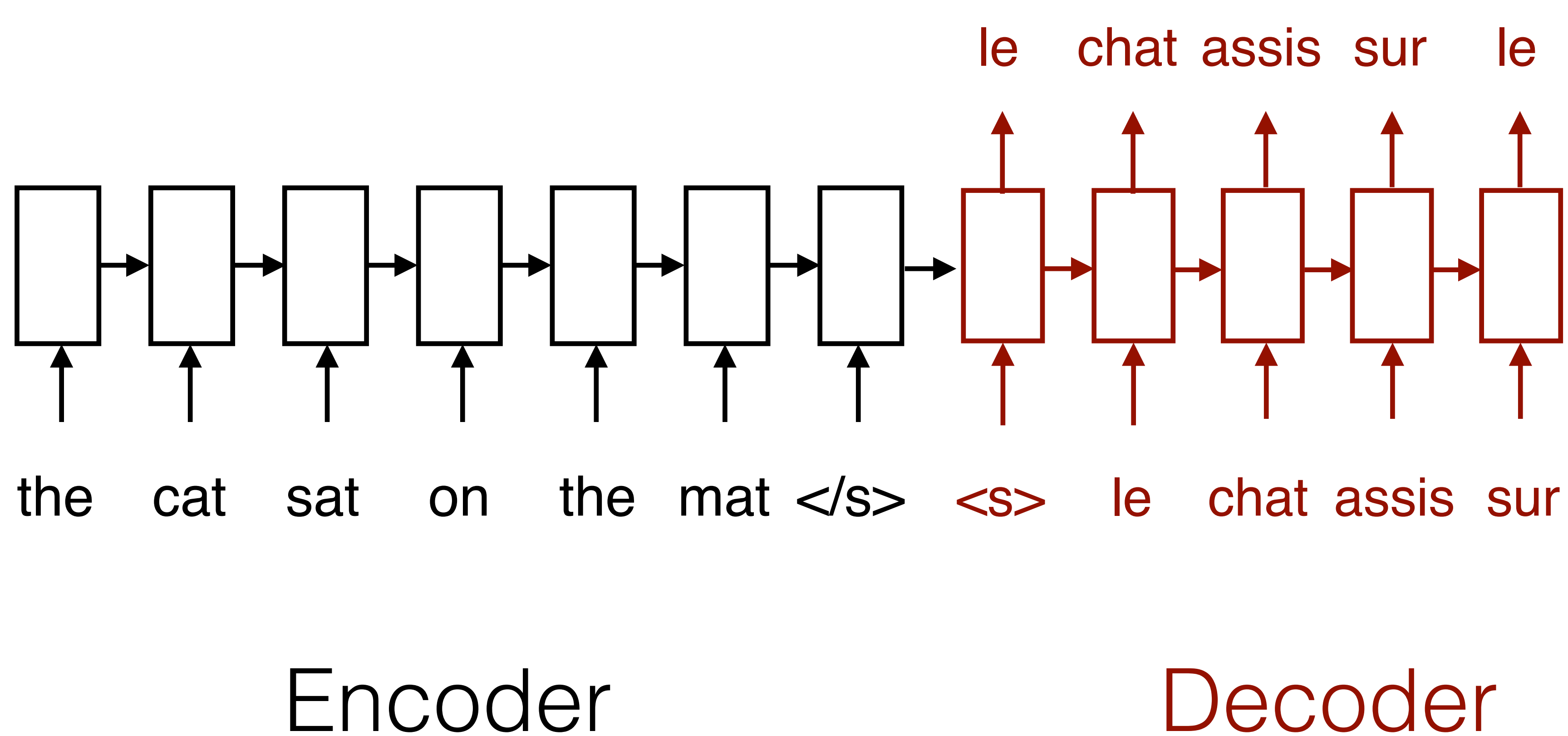
Sequence 2 Sequence Learning

- Inspired by RNN language modeling
- First (modern) models for NMT presented by Kalchbrenner et. al. 2013, Sutskever et al., 2014, Cho et al., 2014
- 2 RNN's, one for "reading" the input and one for "writing" the output (a.k.a the encoder-decoder architecture)



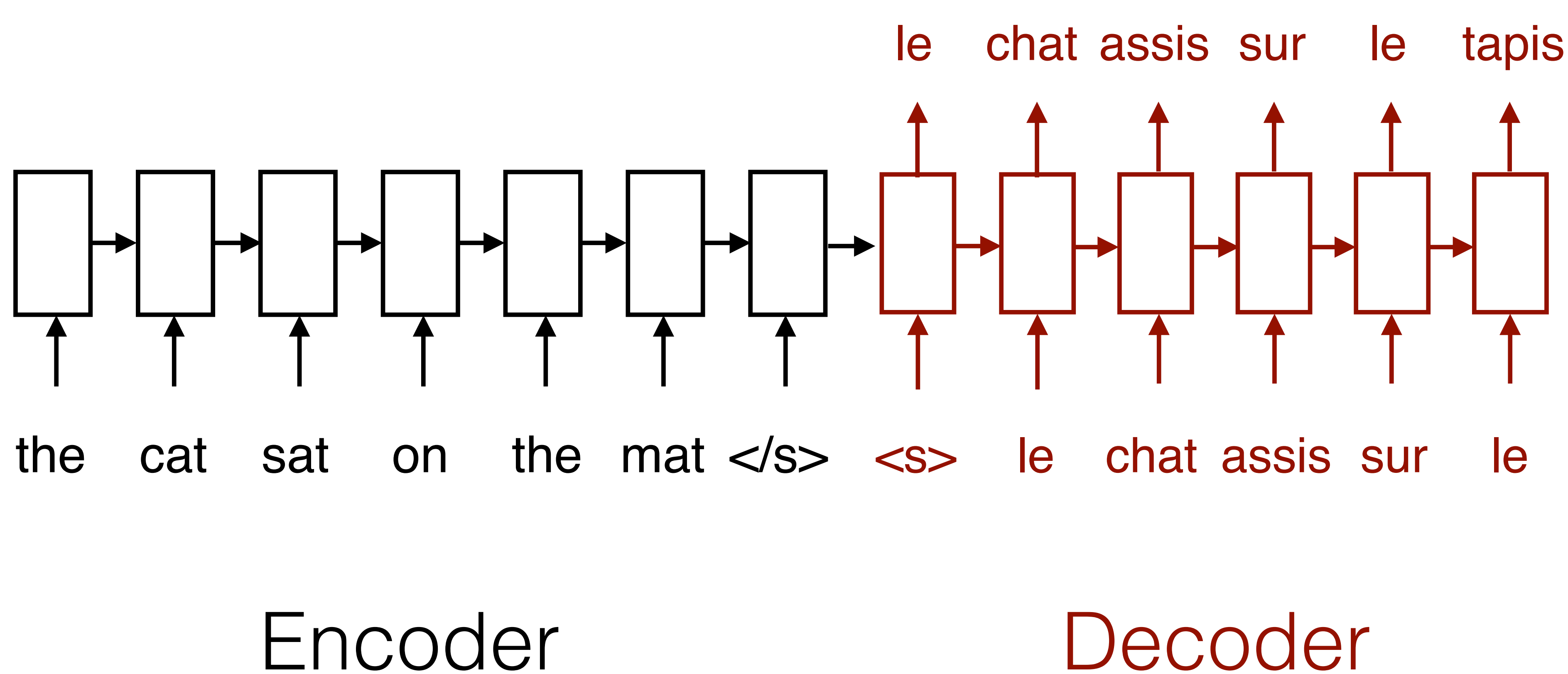
Sequence 2 Sequence Learning

- Inspired by RNN language modeling
- First (modern) models for NMT presented by Kalchbrenner et. al. 2013, Sutskever et al., 2014, Cho et al., 2014
- 2 RNN’s, one for “reading” the input and one for “writing” the output (a.k.a the encoder-decoder architecture)



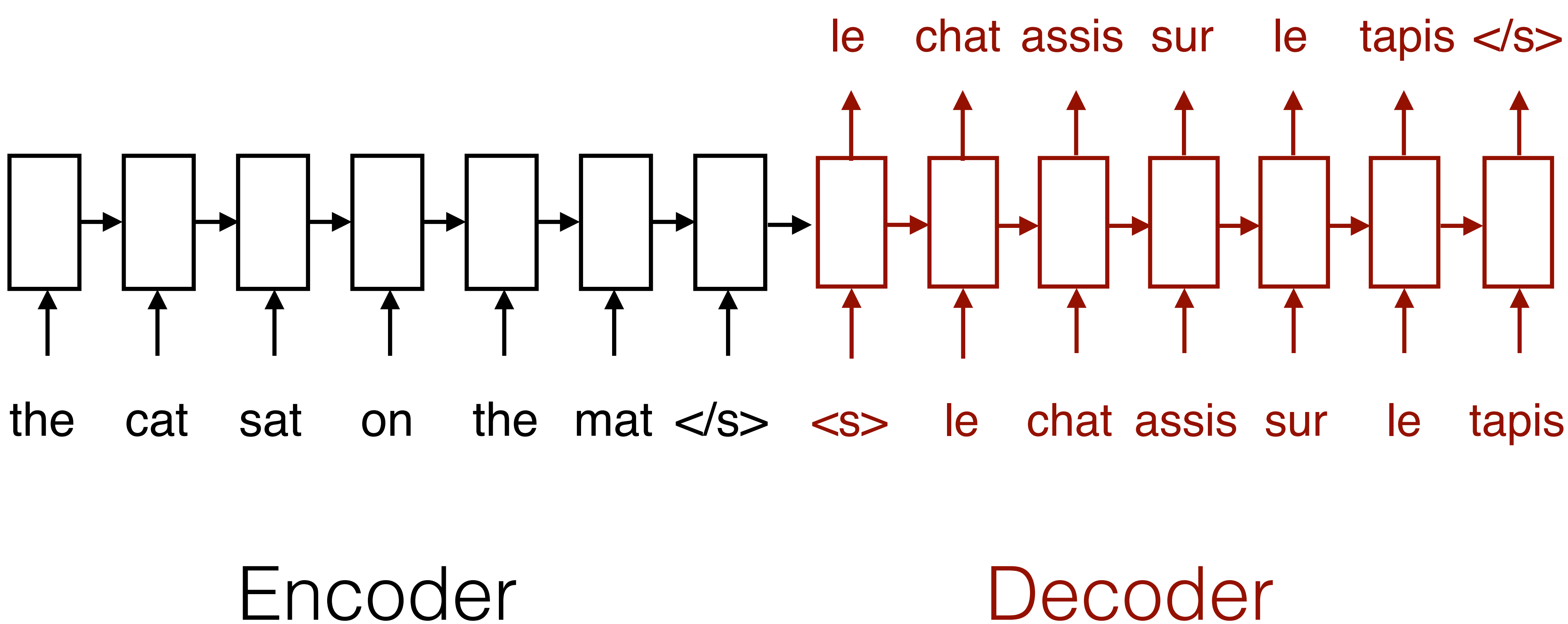
Sequence 2 Sequence Learning

- Inspired by RNN language modeling
- First (modern) models for NMT presented by Kalchbrenner et. al. 2013, Sutskever et al., 2014, Cho et al., 2014
- 2 RNN's, one for "reading" the input and one for "writing" the output (a.k.a the encoder-decoder architecture)



Sequence 2 Sequence Learning

- Inspired by RNN language modeling
- First (modern) models for NMT presented by Kalchbrenner et. al. 2013, Sutskever et al., 2014, Cho et al., 2014
- 2 RNN’s, one for “reading” the input and one for “writing” the output (a.k.a the encoder-decoder architecture)



The problem with “vanilla” seq2seq

“You can’t cram the meaning of a whole %&!$# sentence into a single $&!# vector!”* Ray Mooney

The Attention Mechanism

The Attention Mechanism

- Instead of using a single vector as a fixed representation of the input sequence, **“attend”** at each step to the relevant parts of the input

The Attention Mechanism

- Instead of using a single vector as a fixed representation of the input sequence, **“attend”** at each step to the relevant parts of the input

- The **“relevance”** of each input element to the current prediction is computed via a feed-forward network that gets the input element and the current decoder state

The Attention Mechanism

- Instead of using a single vector as a fixed representation of the input sequence, **“attend”** at each step to the relevant parts of the input

- The **“relevance”** of each input element to the current prediction is computed via a feed-forward network that gets the input element and the current decoder state

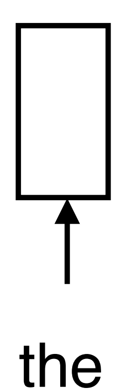
- Coined as **“Resolution Preserving”** - longer sequences get longer representations

The Attention Mechanism

- Instead of using a single vector as a fixed representation of the input sequence, **“attend”** at each step to the relevant parts of the input

- The **“relevance”** of each input element to the current prediction is computed via a feed-forward network that gets the input element and the current decoder state

- Coined as **“Resolution Preserving”** - longer sequences get longer representations

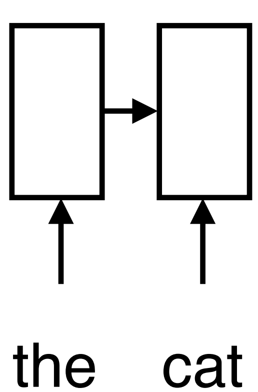


The Attention Mechanism

- Instead of using a single vector as a fixed representation of the input sequence, **"attend"** at each step to the relevant parts of the input

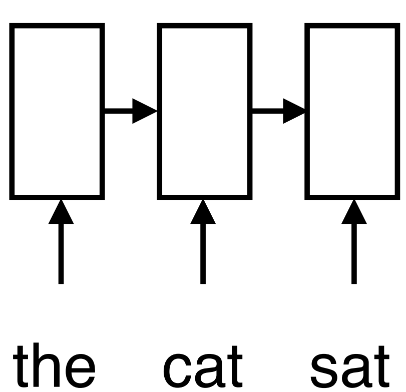
- The **"relevance"** of each input element to the current prediction is computed via a feed-forward network that gets the input element and the current decoder state

- Coined as **"Resolution Preserving"** - longer sequences get longer representations



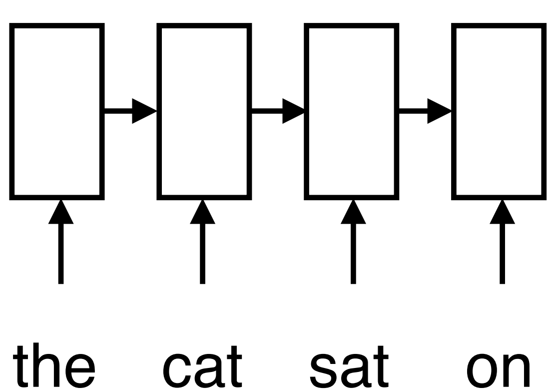
The Attention Mechanism

- Instead of using a single vector as a fixed representation of the input sequence, **“attend”** at each step to the relevant parts of the input
- The **“relevance”** of each input element to the current prediction is computed via a feed-forward network that gets the input element and the current decoder state
- Coined as **“Resolution Preserving”** - longer sequences get longer representations



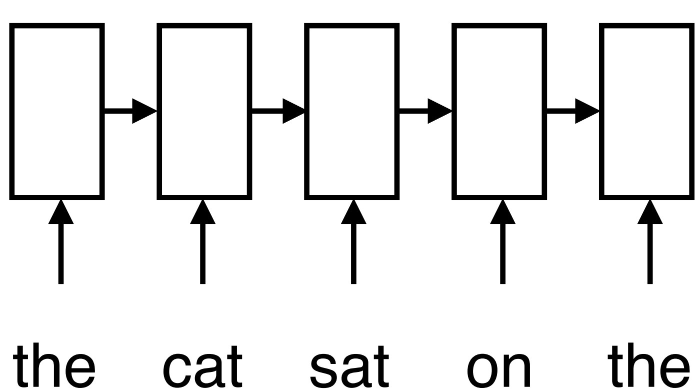
The Attention Mechanism

- Instead of using a single vector as a fixed representation of the input sequence, **"attend"** at each step to the relevant parts of the input
- The **"relevance"** of each input element to the current prediction is computed via a feed-forward network that gets the input element and the current decoder state
- Coined as **"Resolution Preserving"** - longer sequences get longer representations



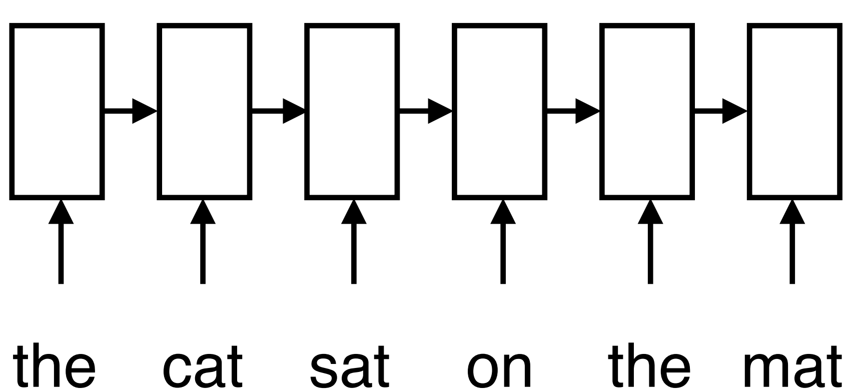
The Attention Mechanism

- Instead of using a single vector as a fixed representation of the input sequence, **“attend”** at each step to the relevant parts of the input
- The **“relevance”** of each input element to the current prediction is computed via a feed-forward network that gets the input element and the current decoder state
- Coined as **“Resolution Preserving”** - longer sequences get longer representations



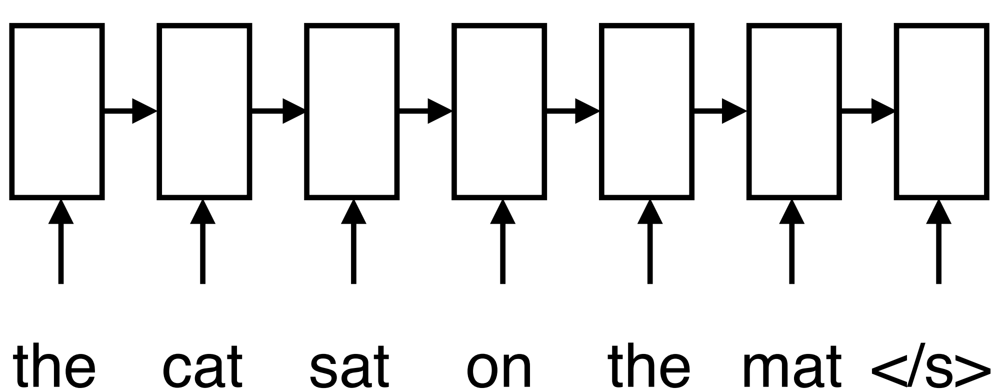
The Attention Mechanism

- Instead of using a single vector as a fixed representation of the input sequence, **"attend"** at each step to the relevant parts of the input
- The **"relevance"** of each input element to the current prediction is computed via a feed-forward network that gets the input element and the current decoder state
- Coined as **"Resolution Preserving"** - longer sequences get longer representations



The Attention Mechanism

- Instead of using a single vector as a fixed representation of the input sequence, **“attend”** at each step to the relevant parts of the input
- The **“relevance”** of each input element to the current prediction is computed via a feed-forward network that gets the input element and the current decoder state
- Coined as **“Resolution Preserving”** - longer sequences get longer representations

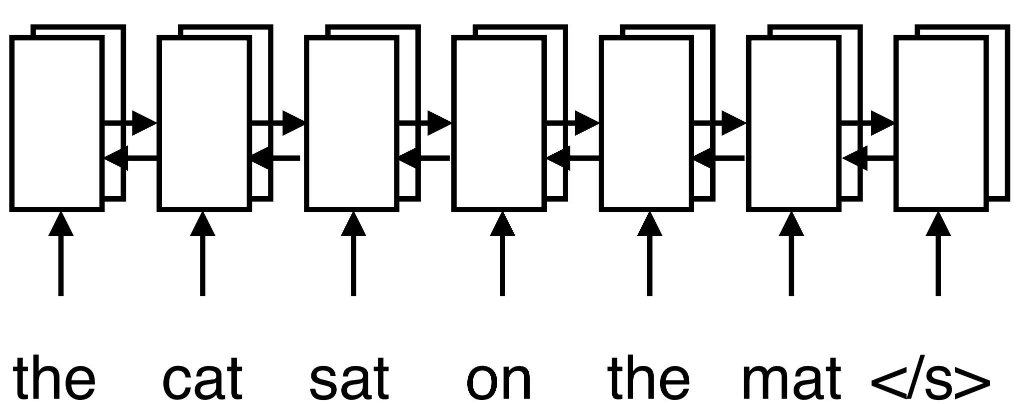


The Attention Mechanism

- Instead of using a single vector as a fixed representation of the input sequence, **“attend”** at each step to the relevant parts of the input

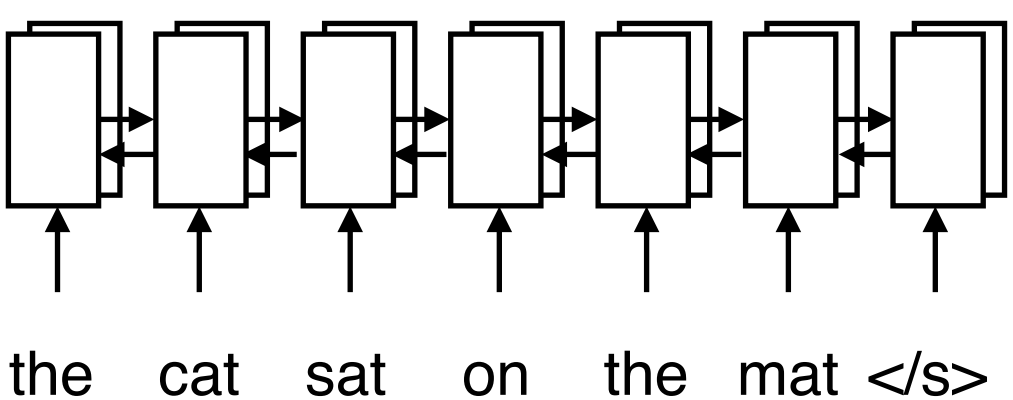
- The **“relevance”** of each input element to the current prediction is computed via a feed-forward network that gets the input element and the current decoder state

- Coined as **“Resolution Preserving”** - longer sequences get longer representations



The Attention Mechanism

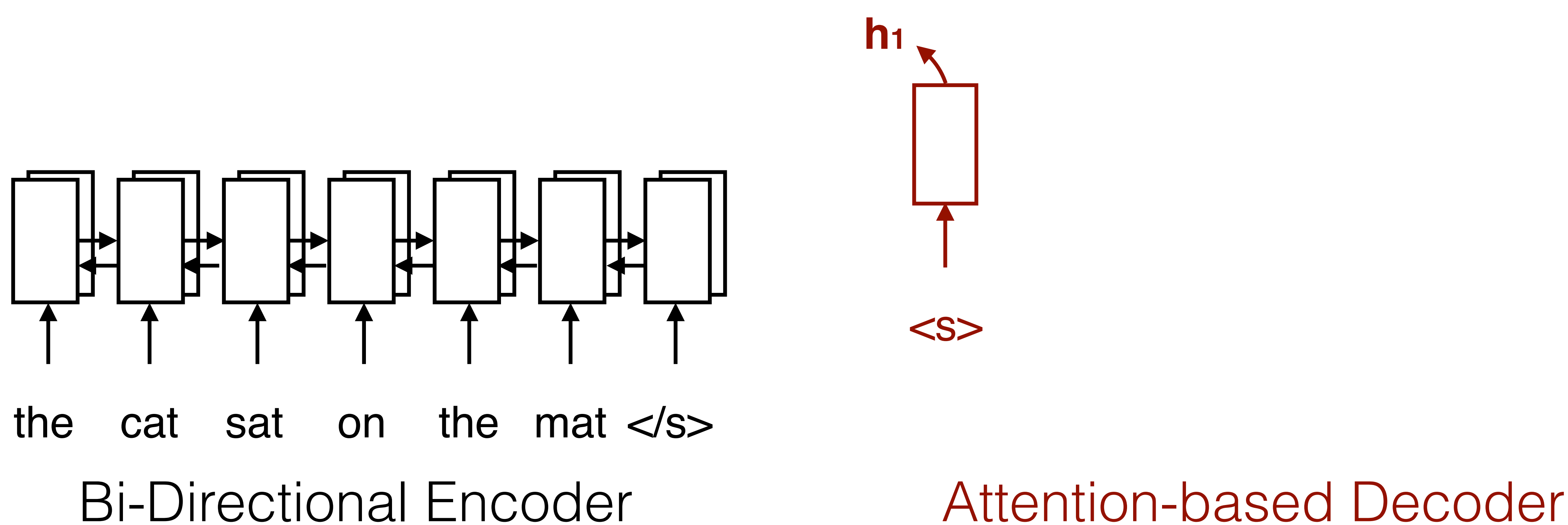
- Instead of using a single vector as a fixed representation of the input sequence, "**attend**" at each step to the relevant parts of the input
- The "**relevance**" of each input element to the current prediction is computed via a feed-forward network that gets the input element and the current decoder state
- Coined as "**Resolution Preserving**" - longer sequences get longer representations



Bi-Directional Encoder

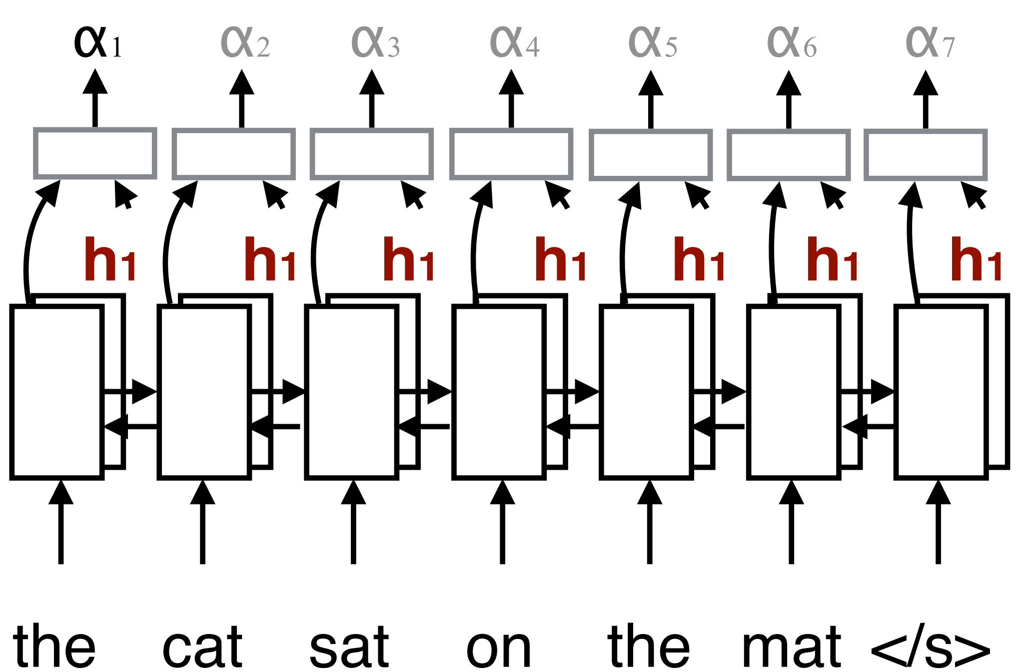
The Attention Mechanism

- Instead of using a single vector as a fixed representation of the input sequence, “**attend**” at each step to the relevant parts of the input
- The “**relevance**” of each input element to the current prediction is computed via a feed-forward network that gets the input element and the current decoder state
- Coined as “**Resolution Preserving**” - longer sequences get longer representations

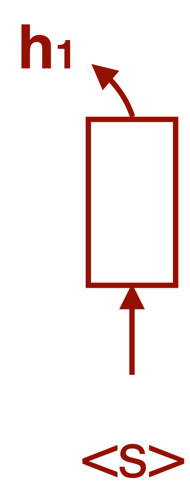


The Attention Mechanism

- Instead of using a single vector as a fixed representation of the input sequence, “**attend**” at each step to the relevant parts of the input
- The “**relevance**” of each input element to the current prediction is computed via a feed-forward network that gets the input element and the current decoder state
- Coined as “**Resolution Preserving**” - longer sequences get longer representations



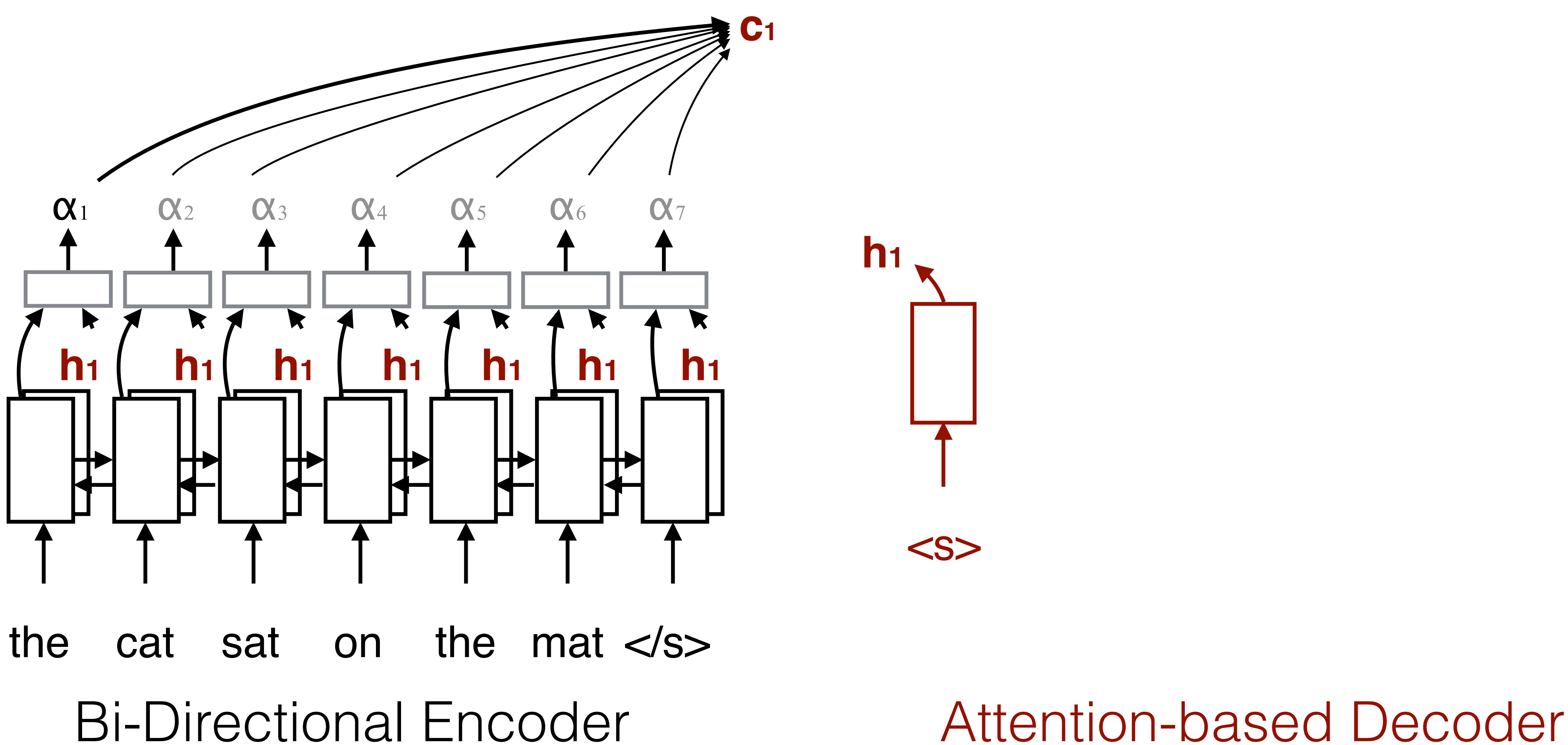
Bi-Directional Encoder



Attention-based Decoder

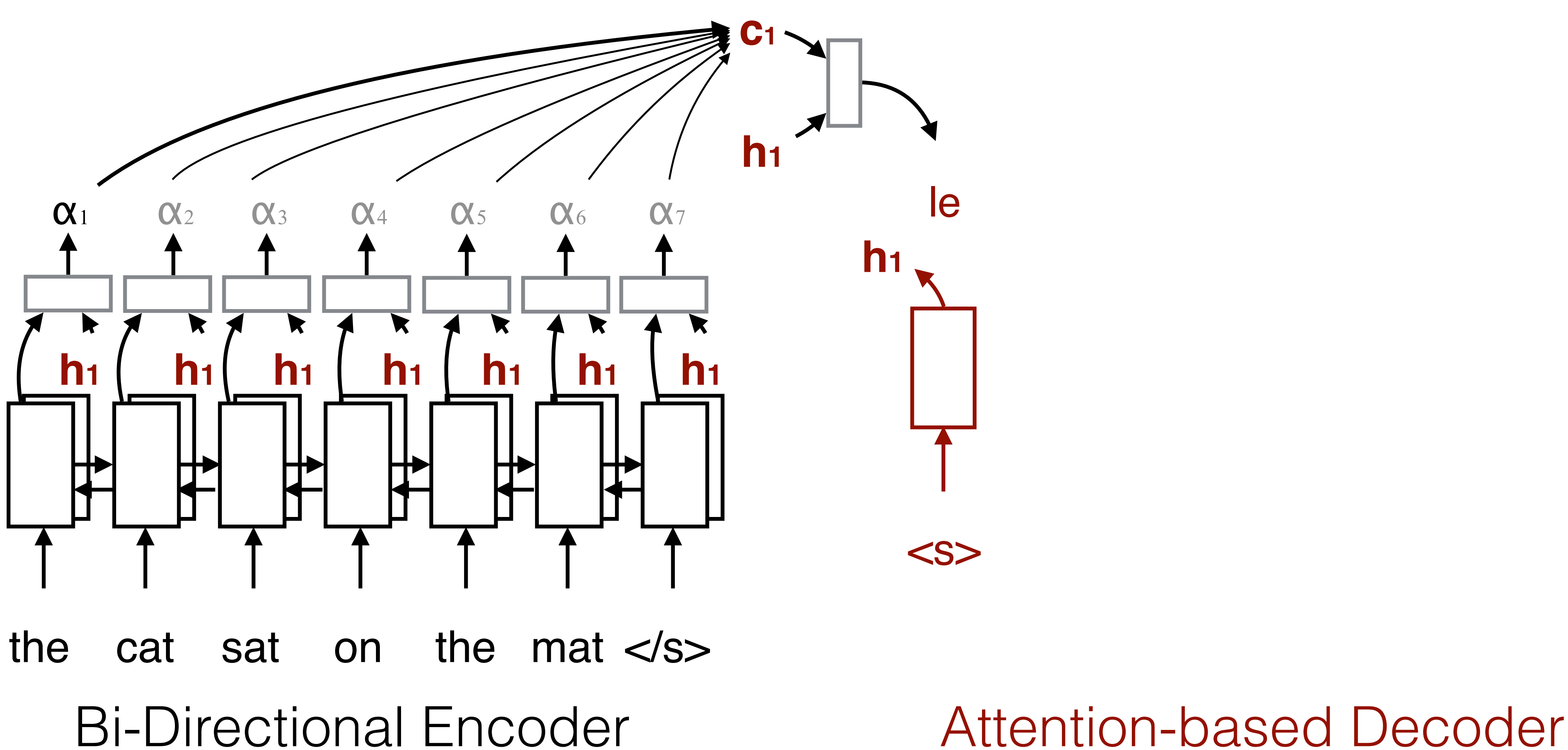
The Attention Mechanism

- Instead of using a single vector as a fixed representation of the input sequence, “**attend**” at each step to the relevant parts of the input
- The “**relevance**” of each input element to the current prediction is computed via a feed-forward network that gets the input element and the current decoder state
- Coined as “**Resolution Preserving**” - longer sequences get longer representations



The Attention Mechanism

- Instead of using a single vector as a fixed representation of the input sequence, "**attend**" at each step to the relevant parts of the input
- The "**relevance**" of each input element to the current prediction is computed via a feed-forward network that gets the input element and the current decoder state
- Coined as "**Resolution Preserving**" - longer sequences get longer representations

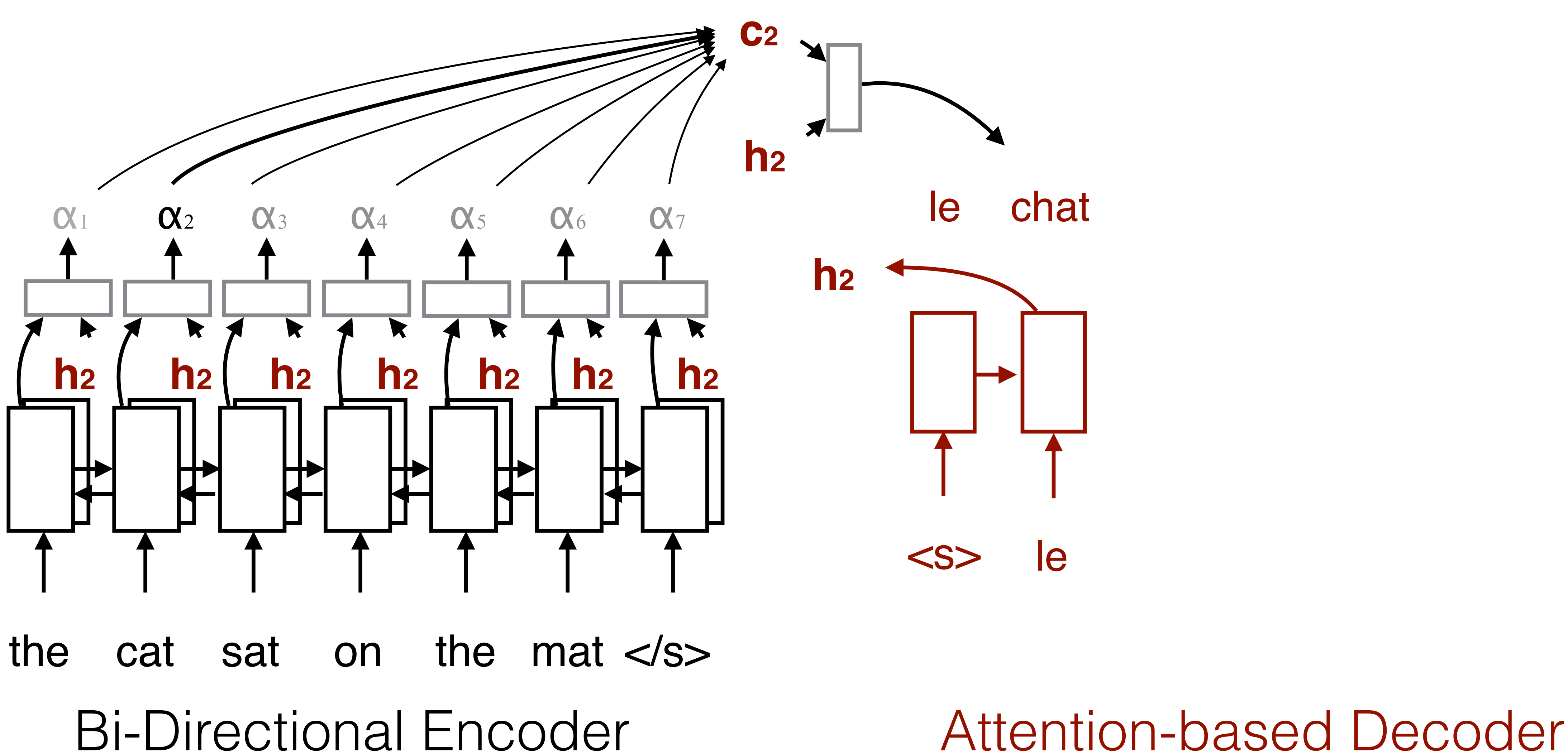


The Attention Mechanism

- Instead of using a single vector as a fixed representation of the input sequence, "**attend**" at each step to the **relevant parts** of the input

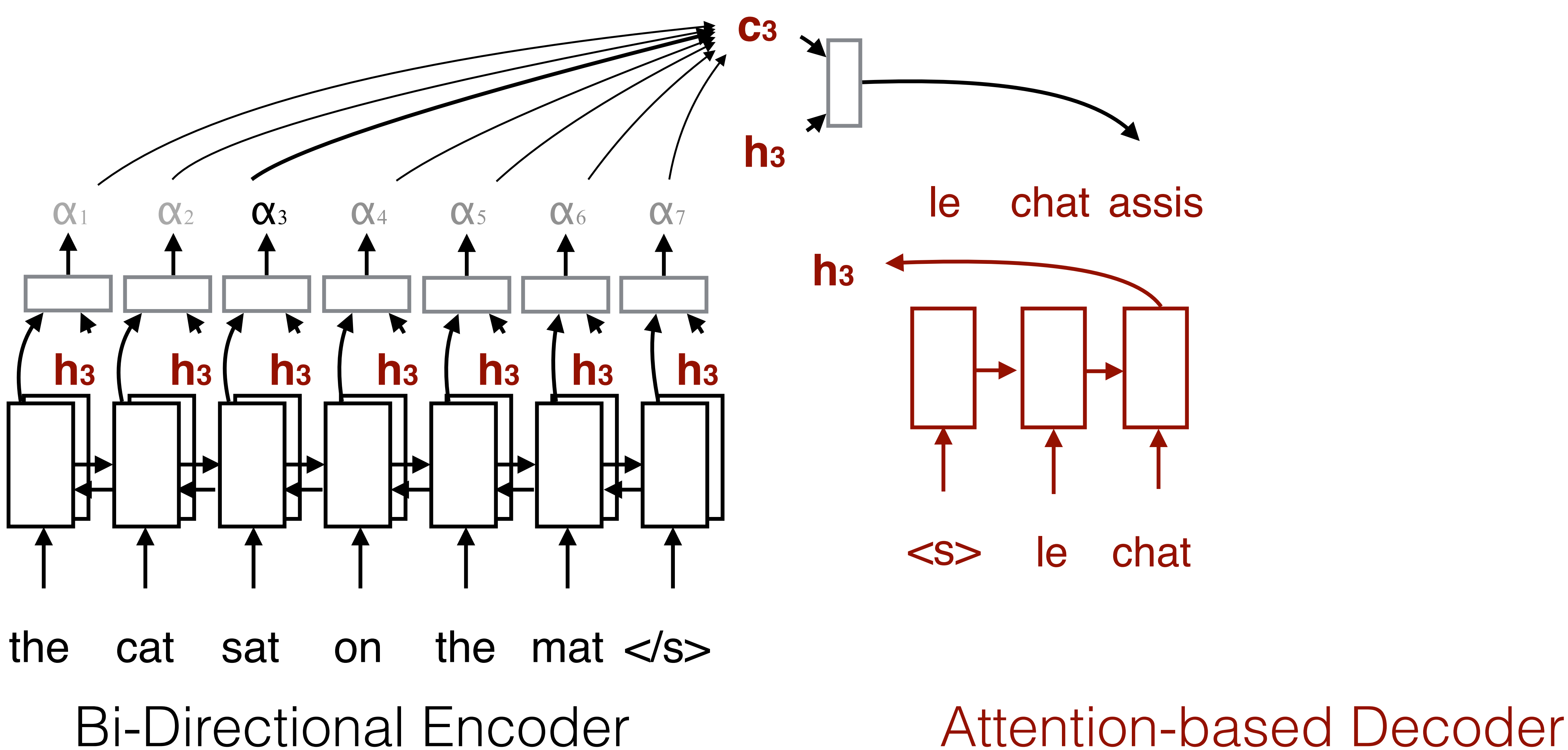
- The "**relevance**" of each input element to the current prediction is **computed** via a **feed-forward network** that gets the input element and the current decoder state

- Coined as "**Resolution Preserving**" - longer sequences get longer representations



The Attention Mechanism

- Instead of using a single vector as a fixed representation of the input sequence, “**attend**” at each step to the **relevant parts** of the input
- The “**relevance**” of each input element to the current prediction is **computed** via a **feed-forward network** that gets the input element and the current decoder state
- Coined as “**Resolution Preserving**” - longer sequences get longer representations

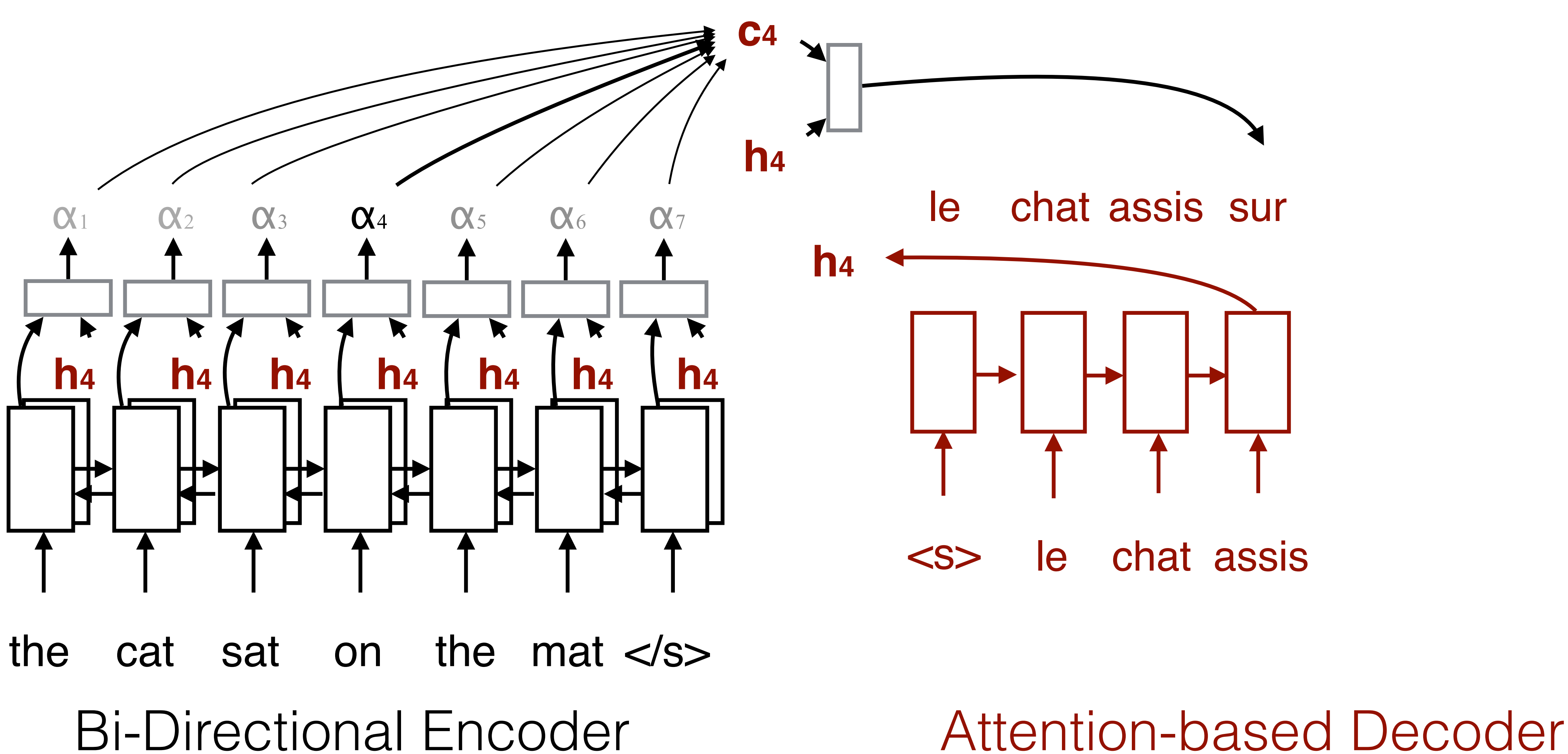


The Attention Mechanism

- Instead of using a single vector as a fixed representation of the input sequence, "**attend**" at each step to the **relevant parts** of the input

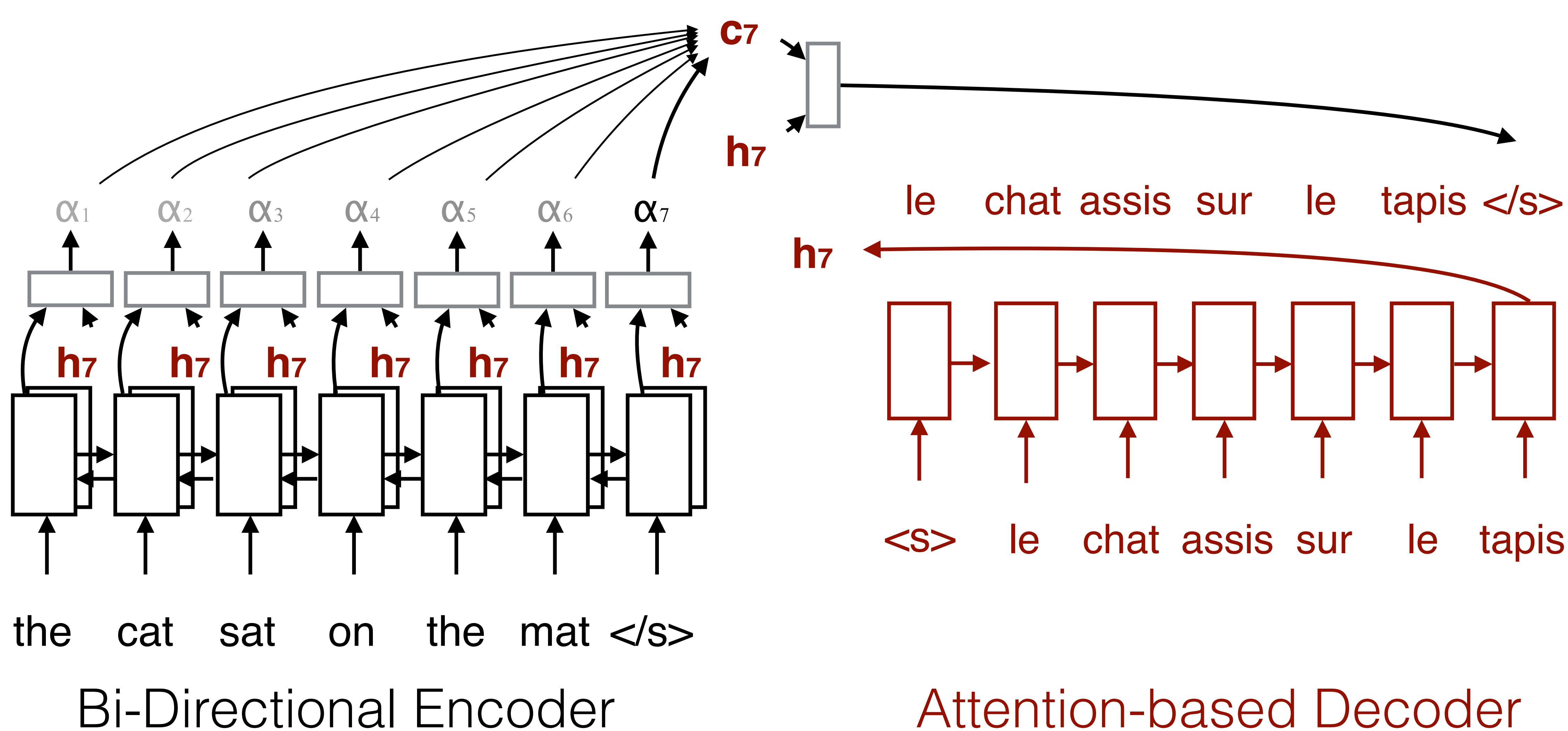
- The "**relevance**" of each input element to the current prediction is **computed** via a **feed-forward network** that gets the input element and the current decoder state

- Coined as "**Resolution Preserving**" - longer sequences get longer representations



The Attention Mechanism

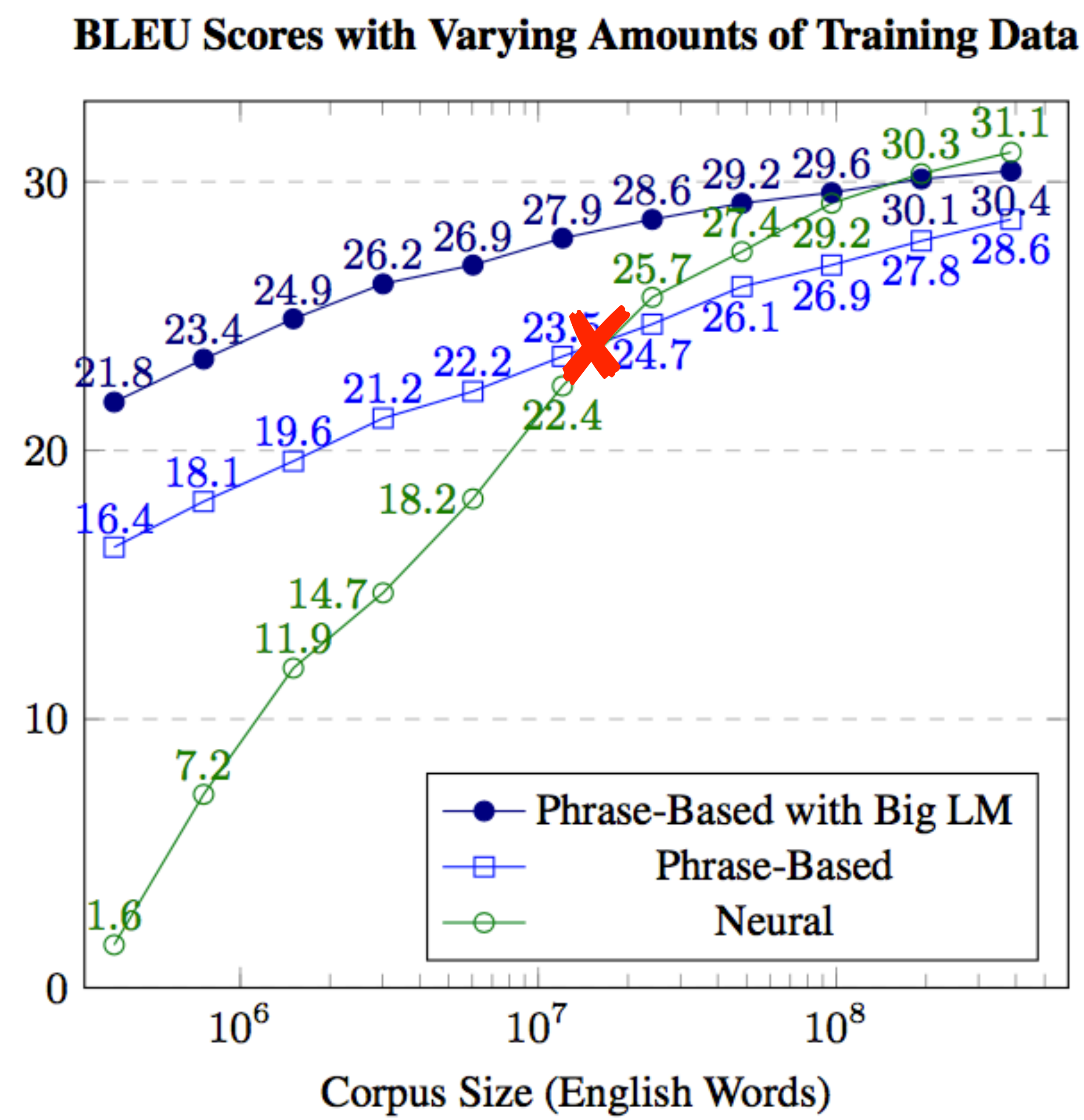
- Instead of using a single vector as a fixed representation of the input sequence, "**attend**" at each step to the **relevant parts** of the input
- The "**relevance**" of each input element to the current prediction is **computed** via a **feed-forward network** that gets the input element and the current decoder state
- Coined as "**Resolution Preserving**" - longer sequences get longer representations



A Price to Pay: (Parallel) Data

A Price to Pay: (Parallel) Data

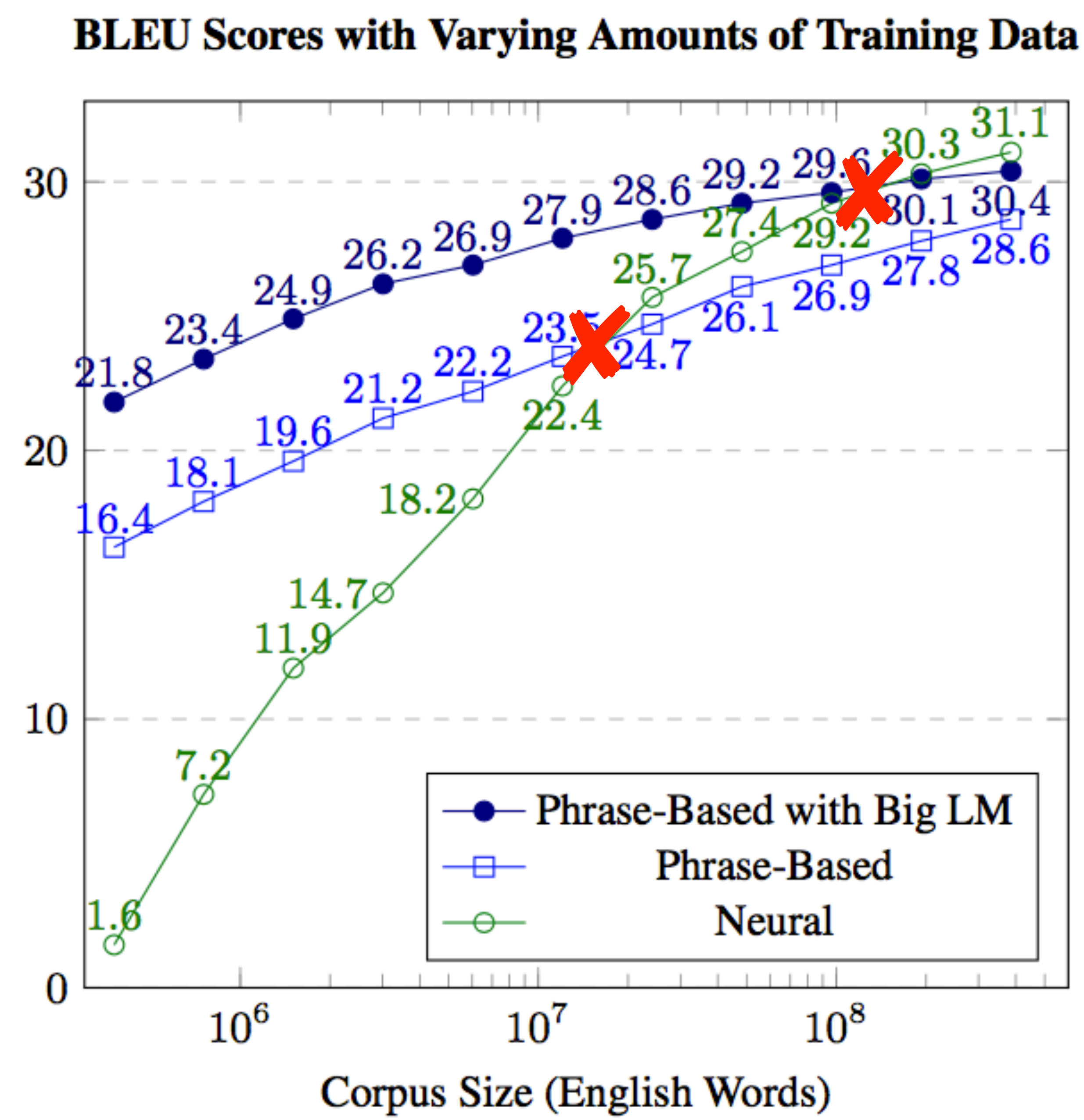
- NMT is better than SMT only when given >10m parallel words



Koehn & Knowles, 2017

A Price to Pay: (Parallel) Data

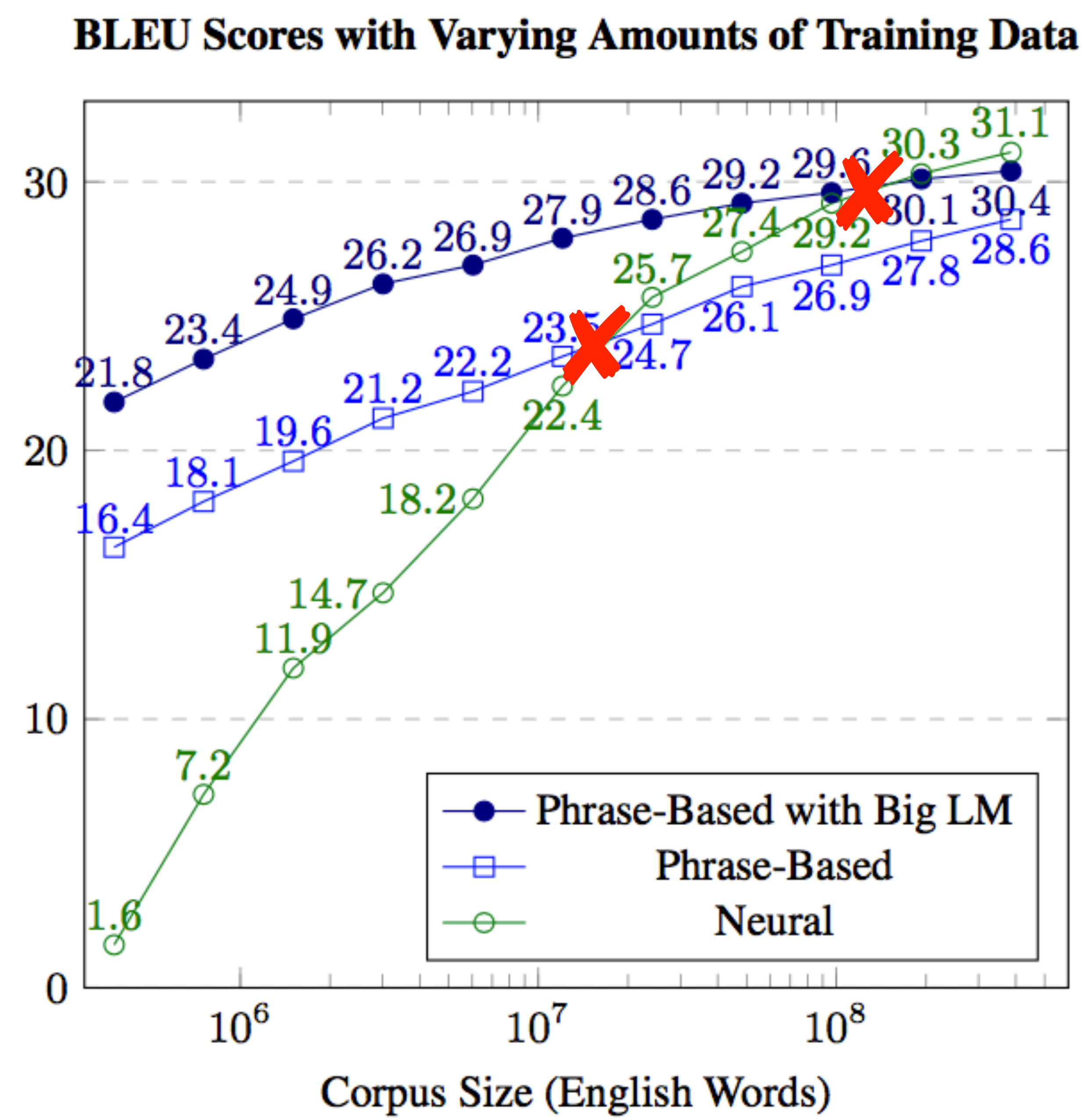
- NMT is better than SMT only when given >10m parallel words
- NMT is better than “Semi Supervised” SMT (SMT + a large language model) only when given >100m parallel words



Koehn & Knowles, 2017

A Price to Pay: (Parallel) Data

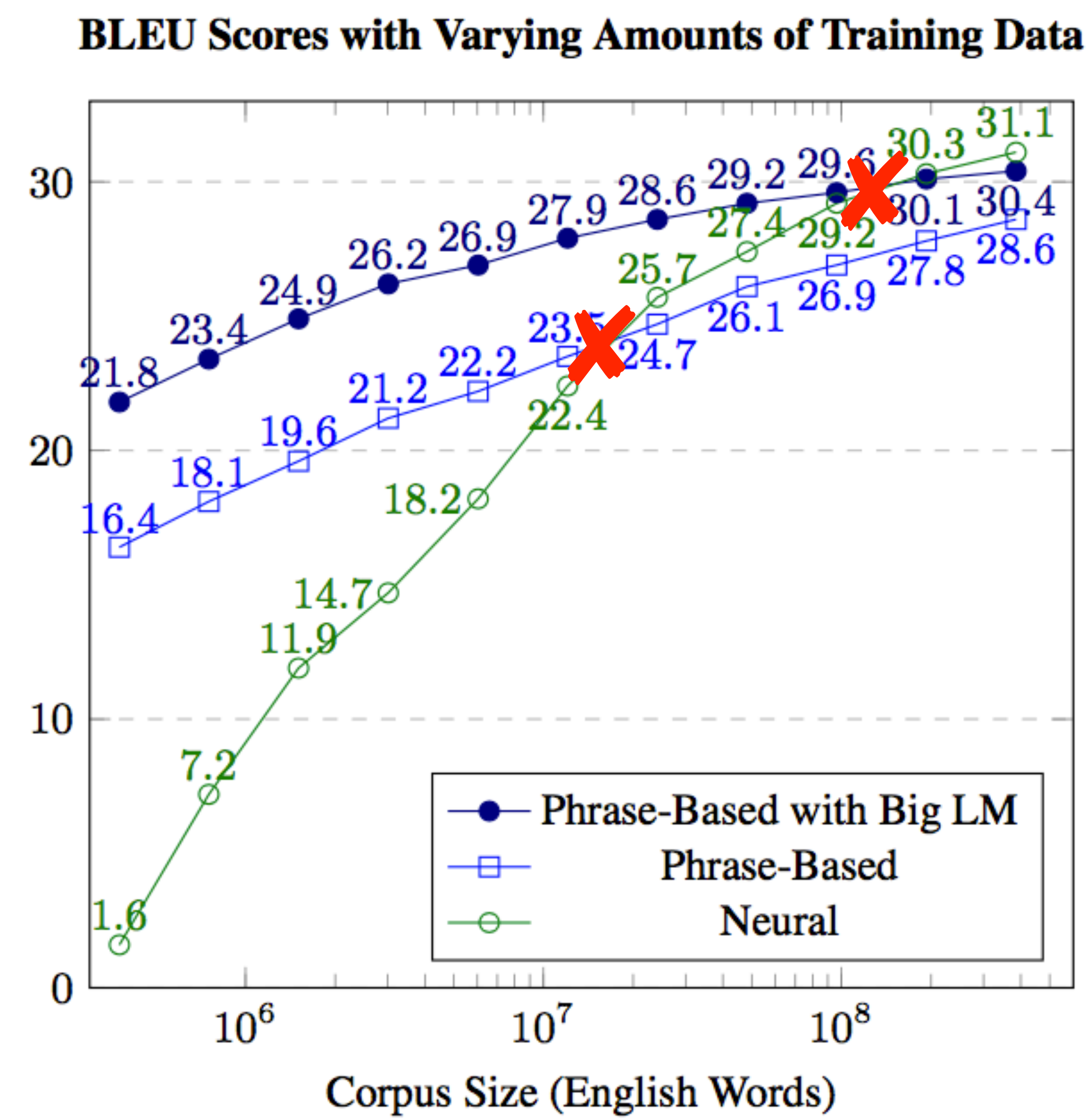
- NMT is better than SMT only when given >10m parallel words
- NMT is better than “Semi Supervised” SMT (SMT + a large language model) only when given >100m parallel words
- But getting parallel data is expensive!



Koehn & Knowles, 2017

A Price to Pay: (Parallel) Data

- NMT is better than SMT only when given >10m parallel words
- NMT is better than “Semi Supervised” SMT (SMT + a large language model) only when given >100m parallel words
- But getting parallel data is expensive!
- Can we do well using only **monolingual data?**



Koehn & Knowles, 2017

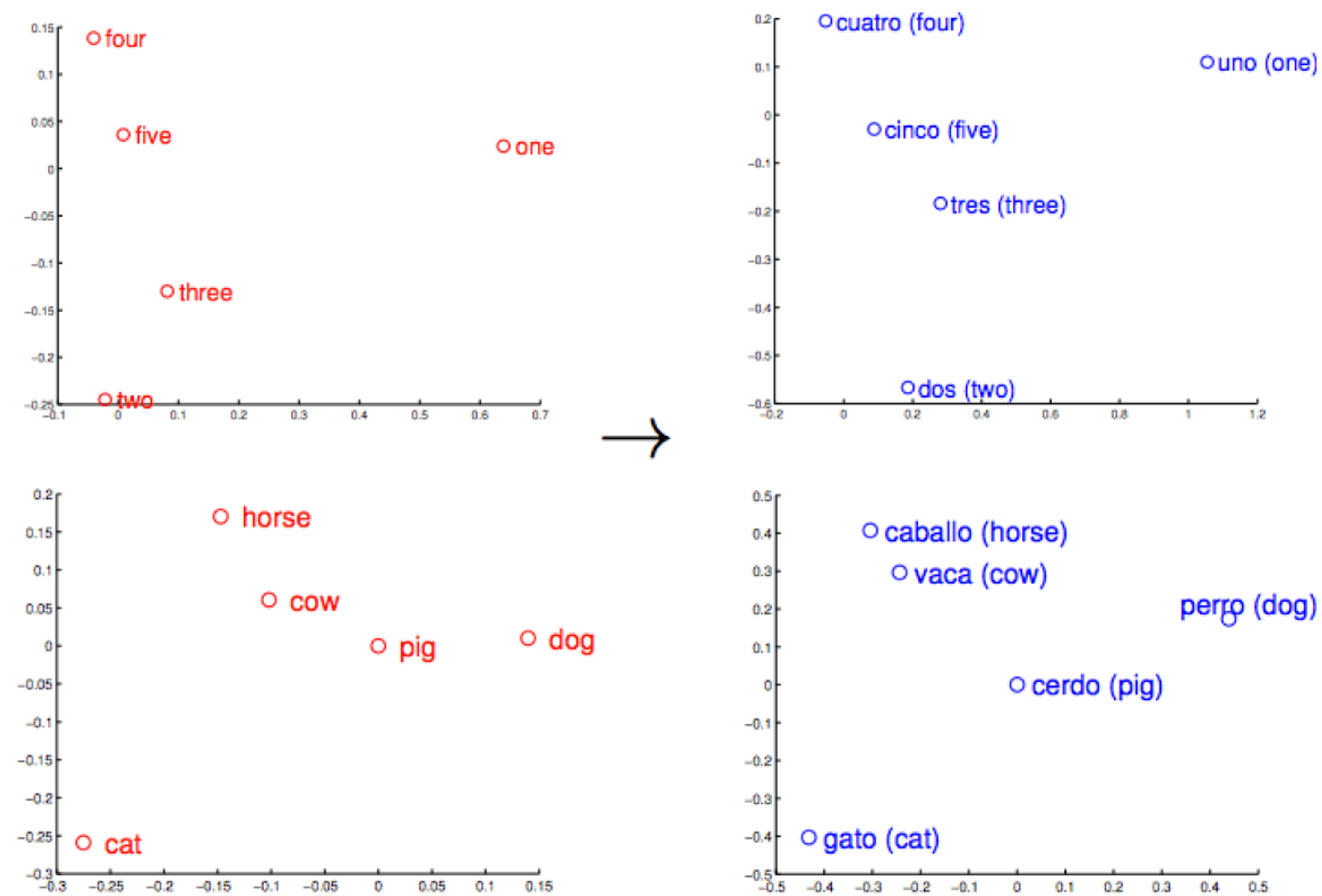
Motivation - Mikolov et al. 2013

Motivation - Mikolov et al. 2013

- “Exploiting Similarities among Languages for Machine Translation” - Mikolov, Le & Sutskever, 2013

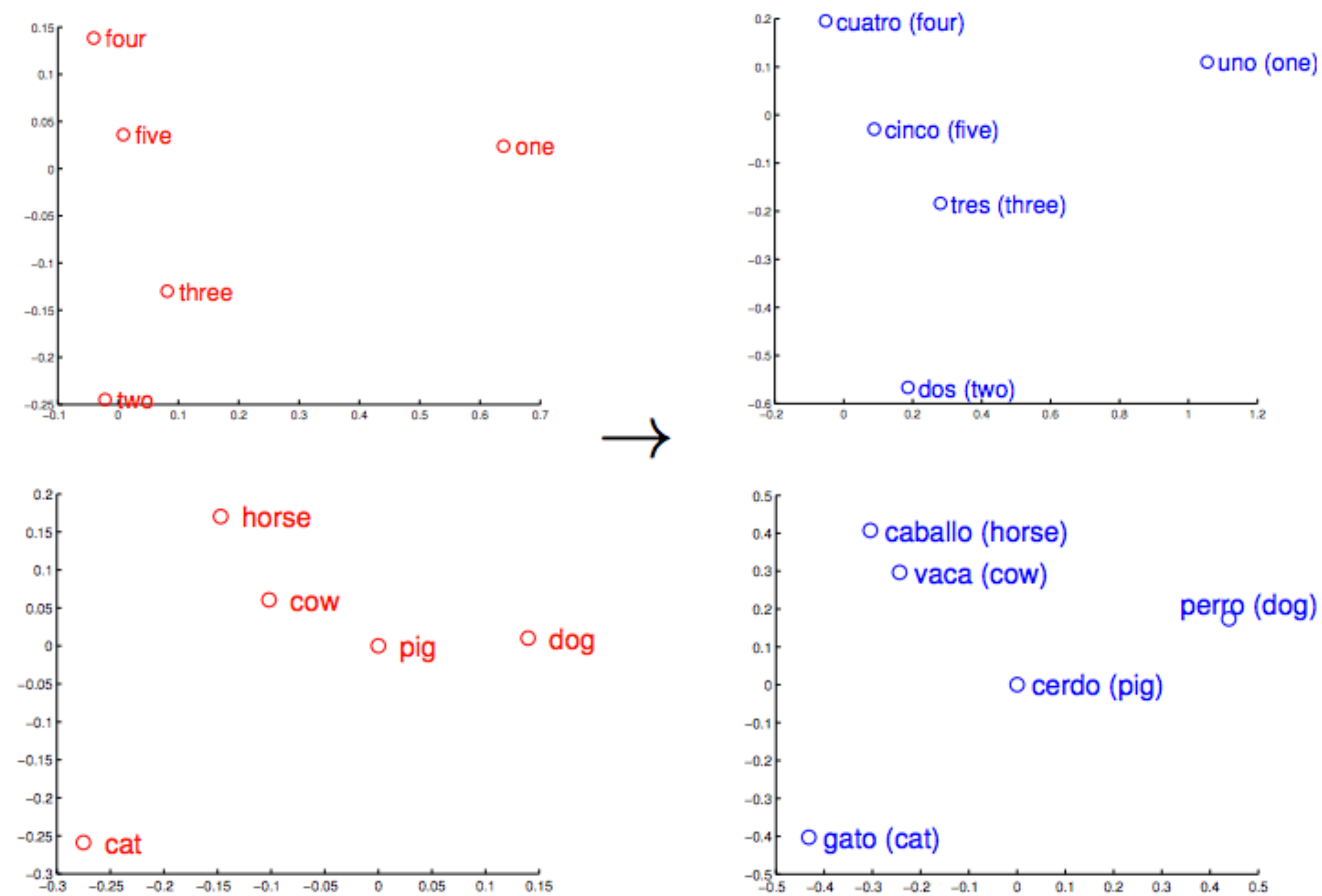
Motivation - Mikolov et al. 2013

- “Exploiting Similarities among Languages for Machine Translation” - Mikolov, Le & Sutskever, 2013
- Observed a **similar structure in unsupervised embedding spaces of different languages,** after rotation



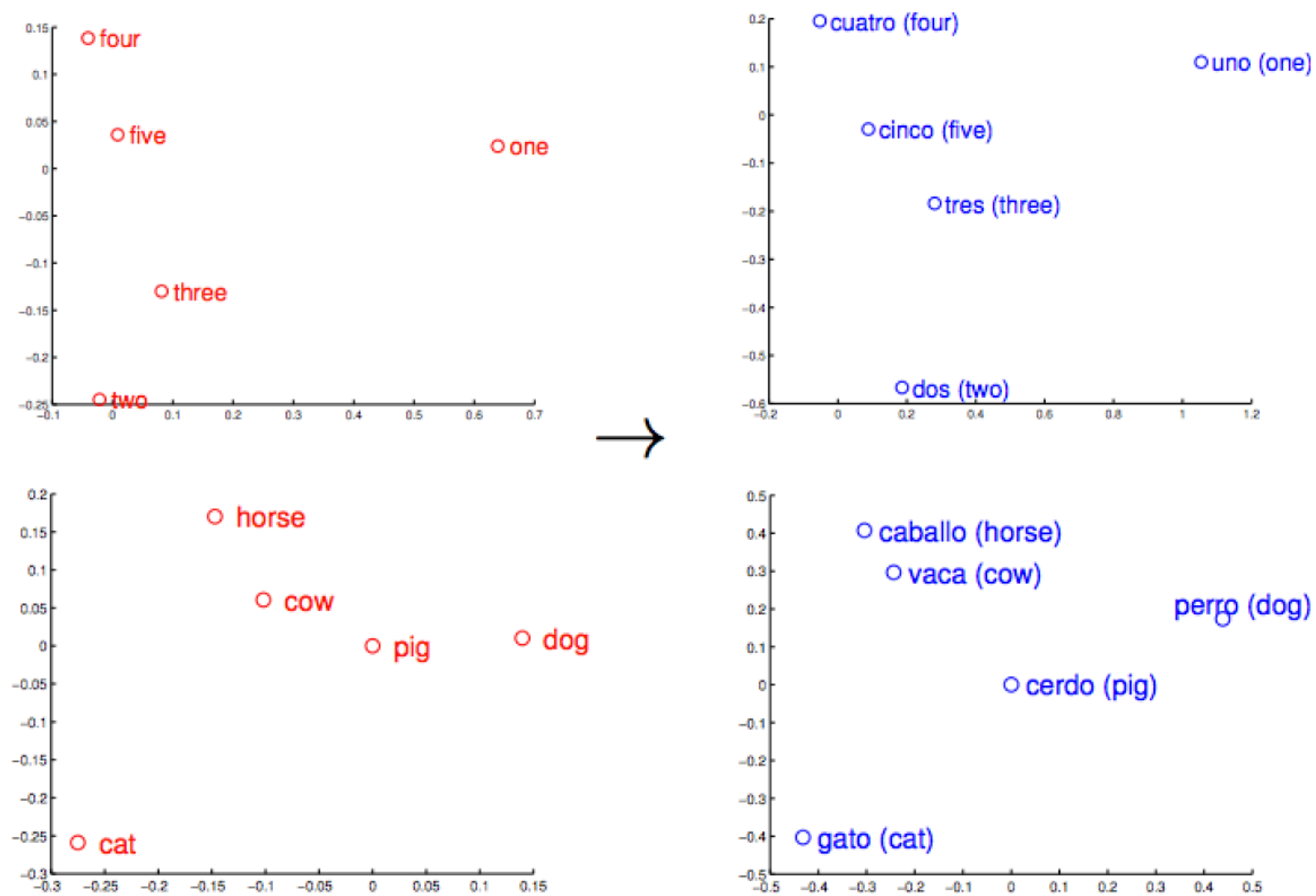
Motivation - Mikolov et al. 2013

- “Exploiting Similarities among Languages for Machine Translation” - Mikolov, Le & Sutskever, 2013
- Observed a **similar structure in unsupervised embedding spaces of different languages**, after rotation
- Learned a rotation matrix to translate words from one embedding space to another with some success



Motivation - Mikolov et al. 2013

- “Exploiting Similarities among Languages for Machine Translation” - Mikolov, Le & Sutskever, 2013
- Observed a **similar structure in unsupervised embedding spaces of different languages**, after rotation
- Learned a rotation matrix to translate words from one embedding space to another with some success
- Weakly supervised - requires a small dictionary (5000 entries)



Unsupervised NMT: A Tale of Two Papers

Unsupervised NMT: A Tale of Two Papers

- Both recently submitted to ICLR 2018 with critical acclaim (October 2017)

Unsupervised NMT: A Tale of Two Papers

- Both recently submitted to ICLR 2018 with critical acclaim (October 2017)
- Similar motivations - both try to tackle:

Unsupervised NMT: A Tale of Two Papers

- Both recently submitted to ICLR 2018 with critical acclaim (October 2017)

- Similar motivations - both try to tackle:

  - **Structure/Fluency** - how to determine the correct word order in the output?

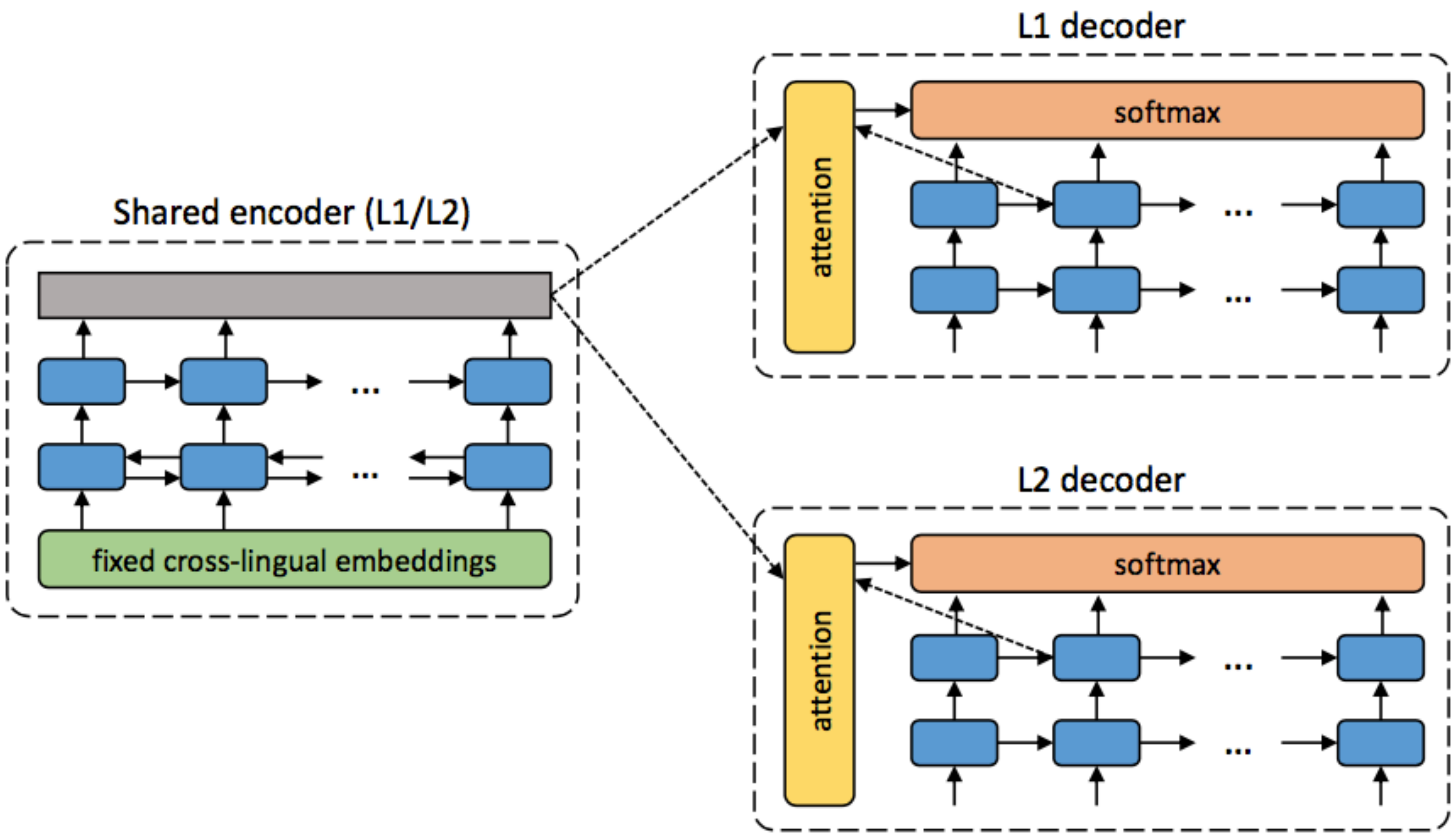
Unsupervised NMT: A Tale of Two Papers

- Both recently submitted to ICLR 2018 with critical acclaim (October 2017)
- Similar motivations - both try to tackle:
  - **Structure/Fluency** - how to determine the correct word order in the output?
  - **Semantics/Adequacy** - how to pick the correct translations given the source?

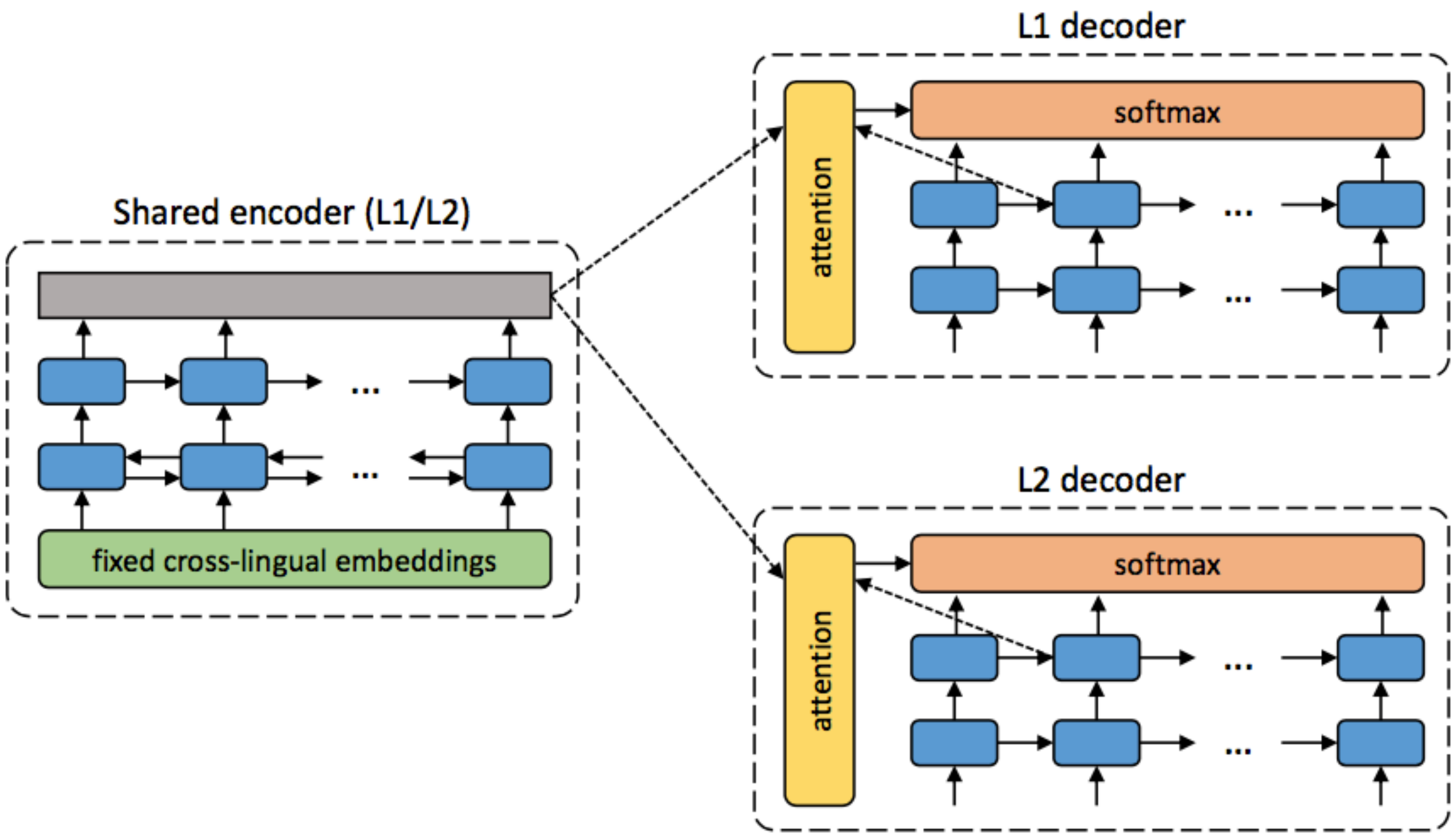
Unsupervised NMT: A Tale of Two Papers

- Both recently submitted to ICLR 2018 with critical acclaim (October 2017)
- Similar motivations - both try to tackle:
  - **Structure/Fluency** - how to determine the correct word order in the output?
  - **Semantics/Adequacy** - how to pick the correct translations given the source?
- Very similar modeling tricks (with slight differences)

Paper I: Artetxe, Labaka, Agirre & Cho

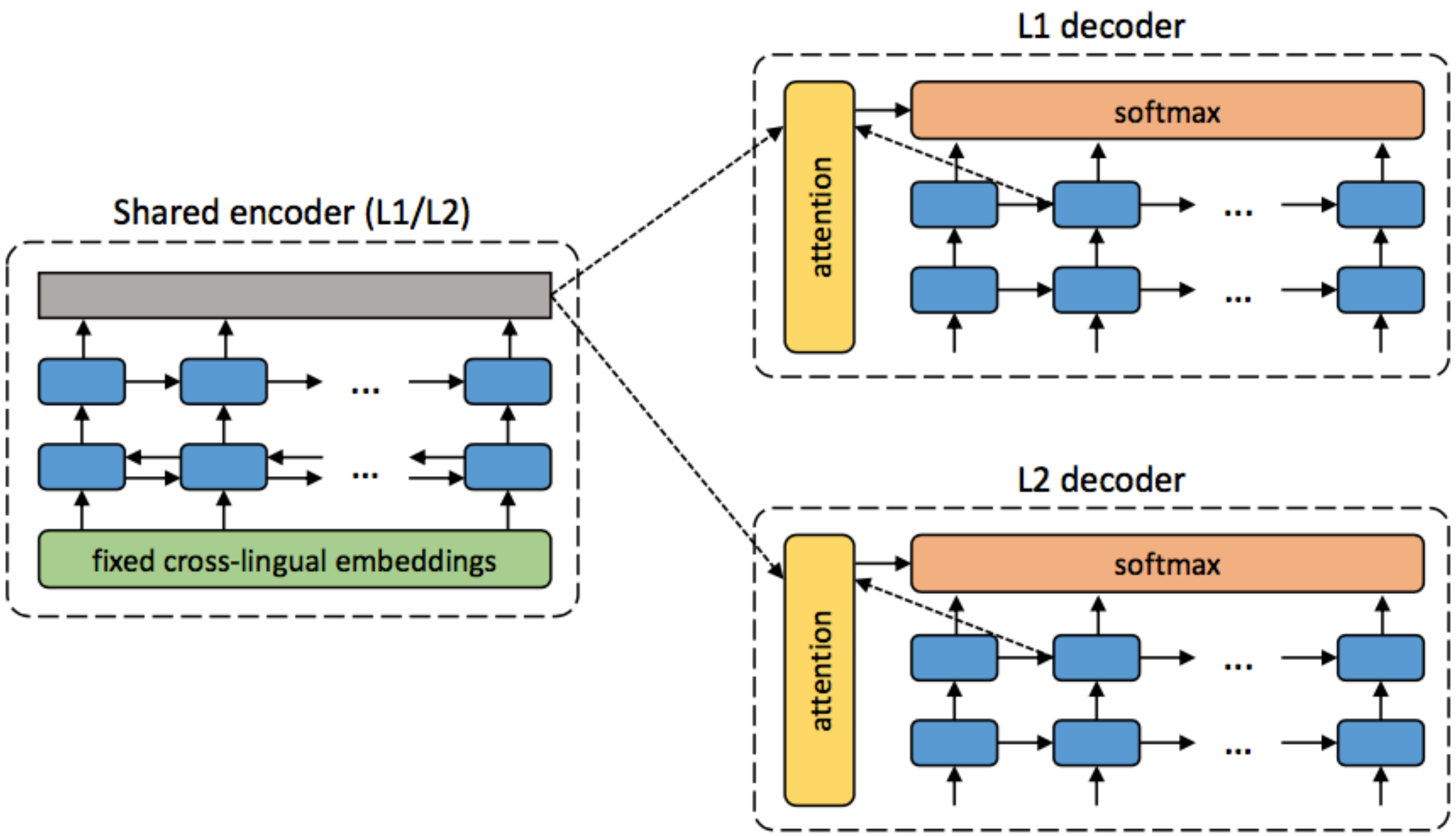


Paper I: Artetxe, Labaka, Agirre & Cho



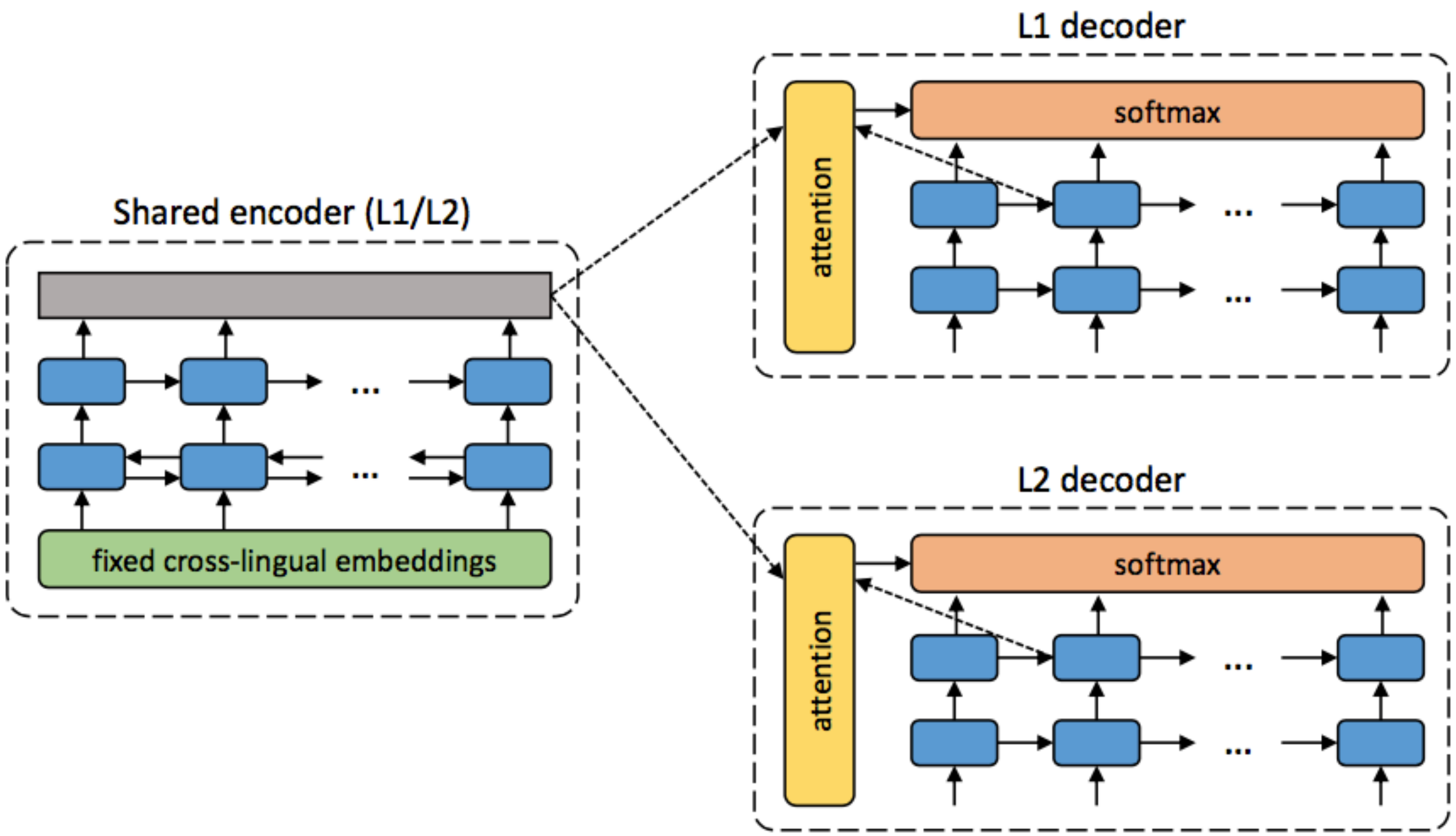
- **Model Architecture:**

Paper I: Artetxe, Labaka, Agirre & Cho



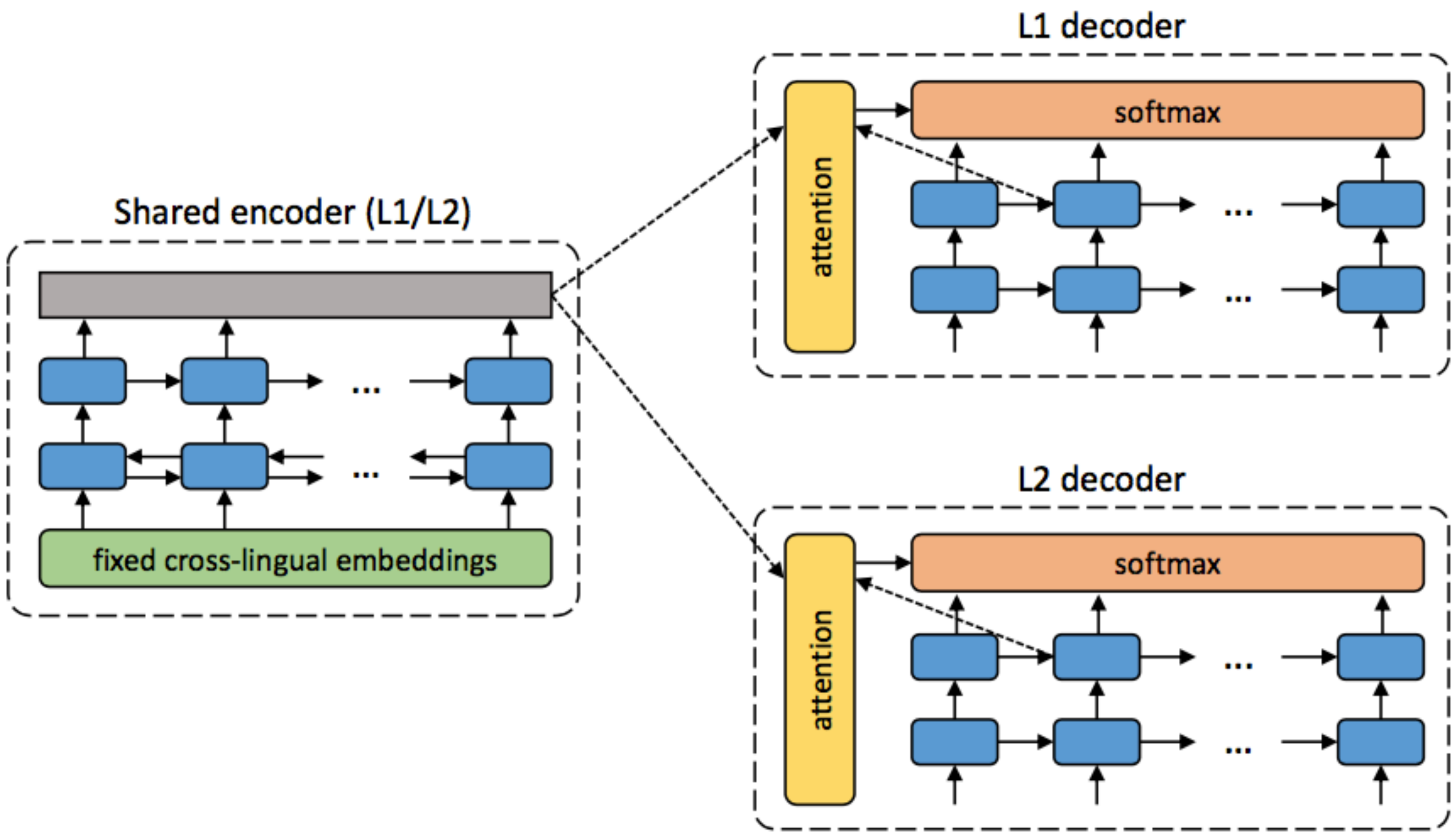
- **Model Architecture:**
  - **Shared** GRU encoder, **Separate** GRU decoders

Paper I: Artetxe, Labaka, Agirre & Cho



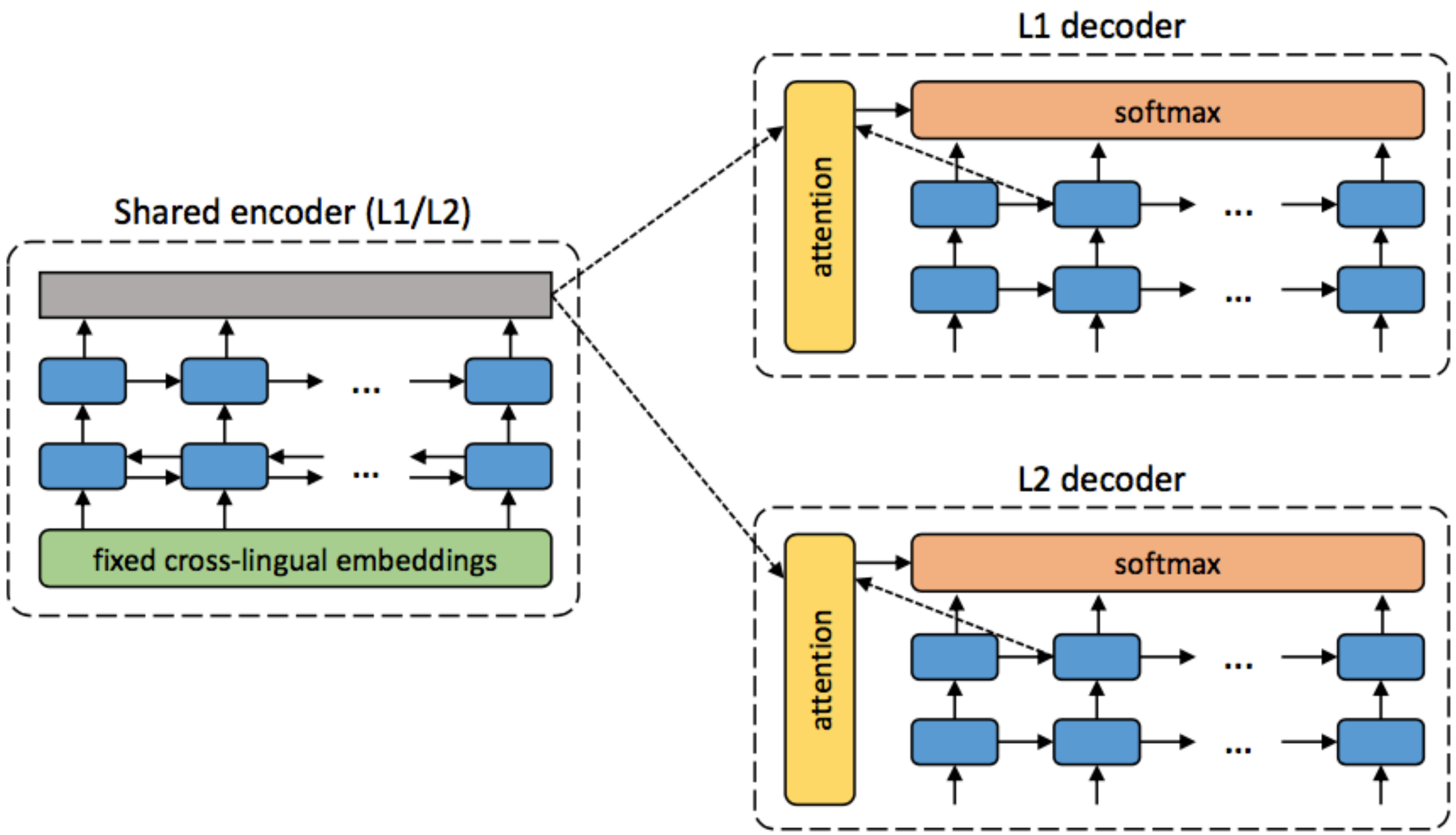
- **Model Architecture:**
  - **Shared** GRU encoder, **Separate** GRU decoders
  - Attention

Paper I: Artetxe, Labaka, Agirre & Cho



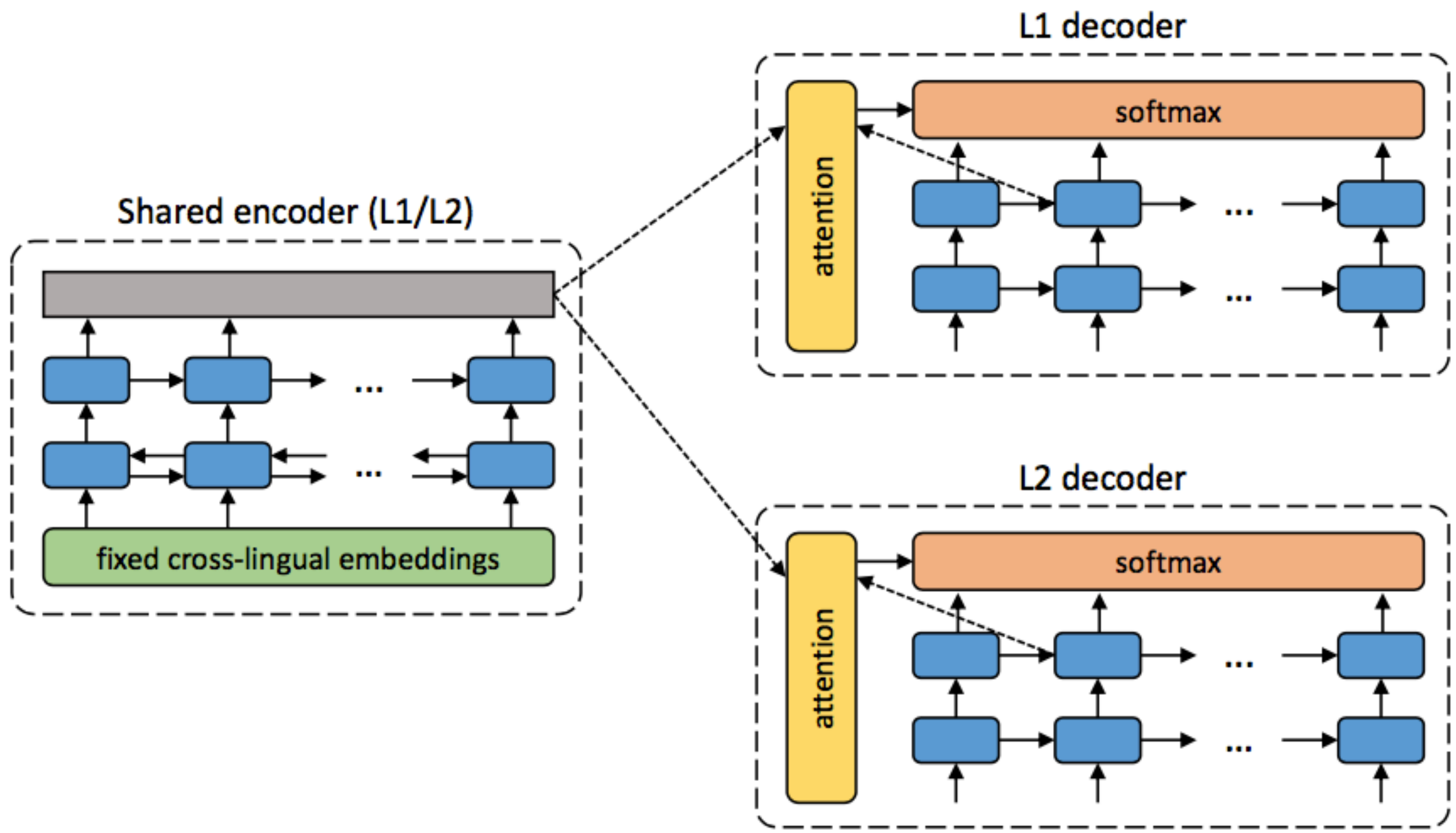
- **Model Architecture:**
  - **Shared** GRU encoder, **Separate** GRU decoders
  - Attention
- **Main "Tricks":**

Paper I: Artetxe, Labaka, Agirre & Cho



- **Model Architecture:**
    - **Shared** GRU encoder, **Separate** GRU decoders
    - Attention
- **Main "Tricks":**
    - Fixed, unsupervised cross-lingual embeddings (**Adequacy**)

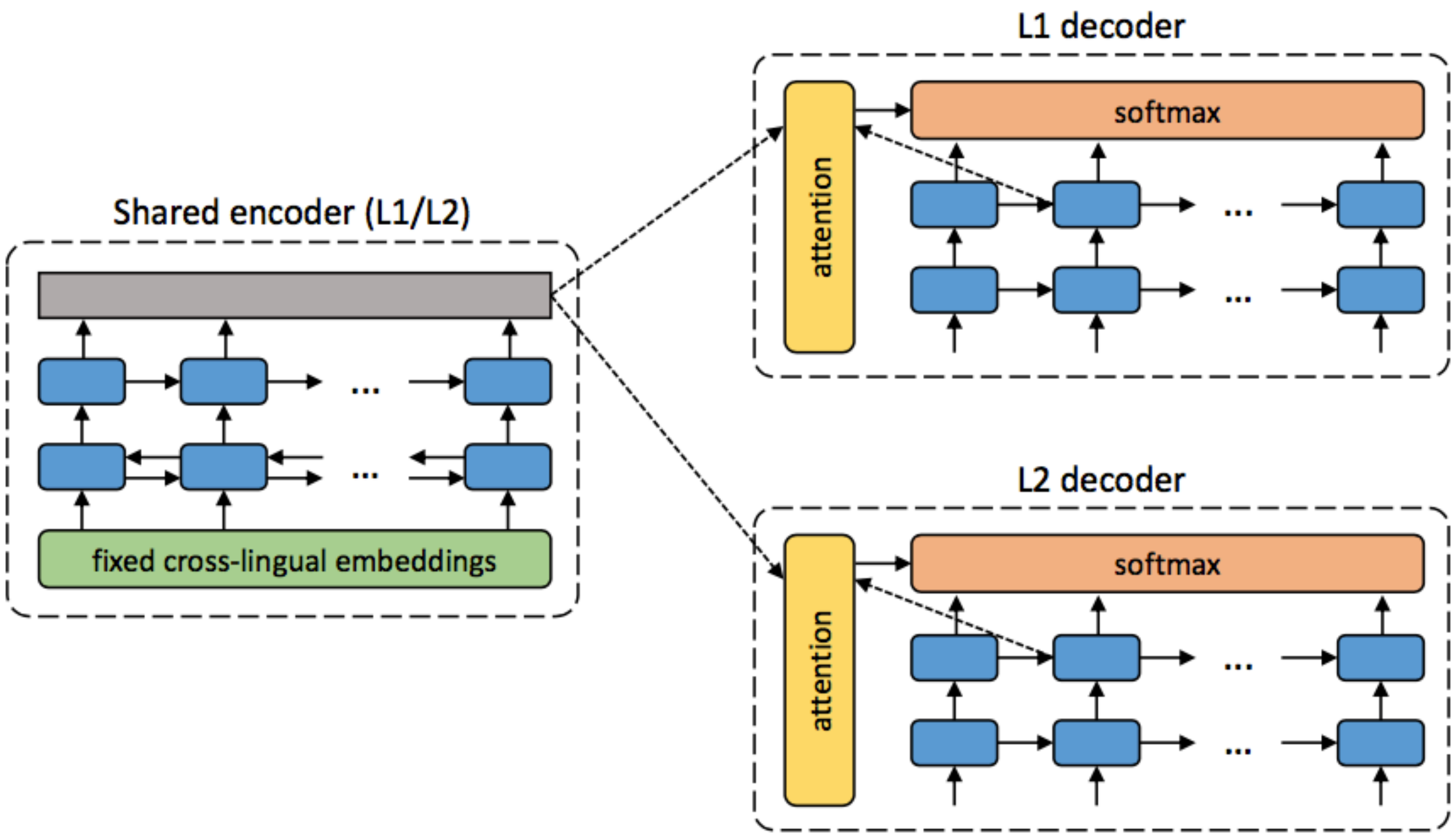
Paper I: Artetxe, Labaka, Agirre & Cho



- **Model Architecture:**
  - **Shared** GRU encoder, **Separate** GRU decoders
  - Attention

- **Main "Tricks":**
  - Fixed, unsupervised cross-lingual embeddings (**Adequacy**)
  - Backtranslation loss (**Adequacy**)

Paper I: Artetxe, Labaka, Agirre & Cho



- **Model Architecture:**
  - **Shared** GRU encoder, **Separate** GRU decoders
  - Attention

- **Main "Tricks":**
  - Fixed, unsupervised cross-lingual embeddings (**Adequacy**)
  - Backtranslation loss (**Adequacy**)
  - Denoising auto-encoder loss (**Fluency**)

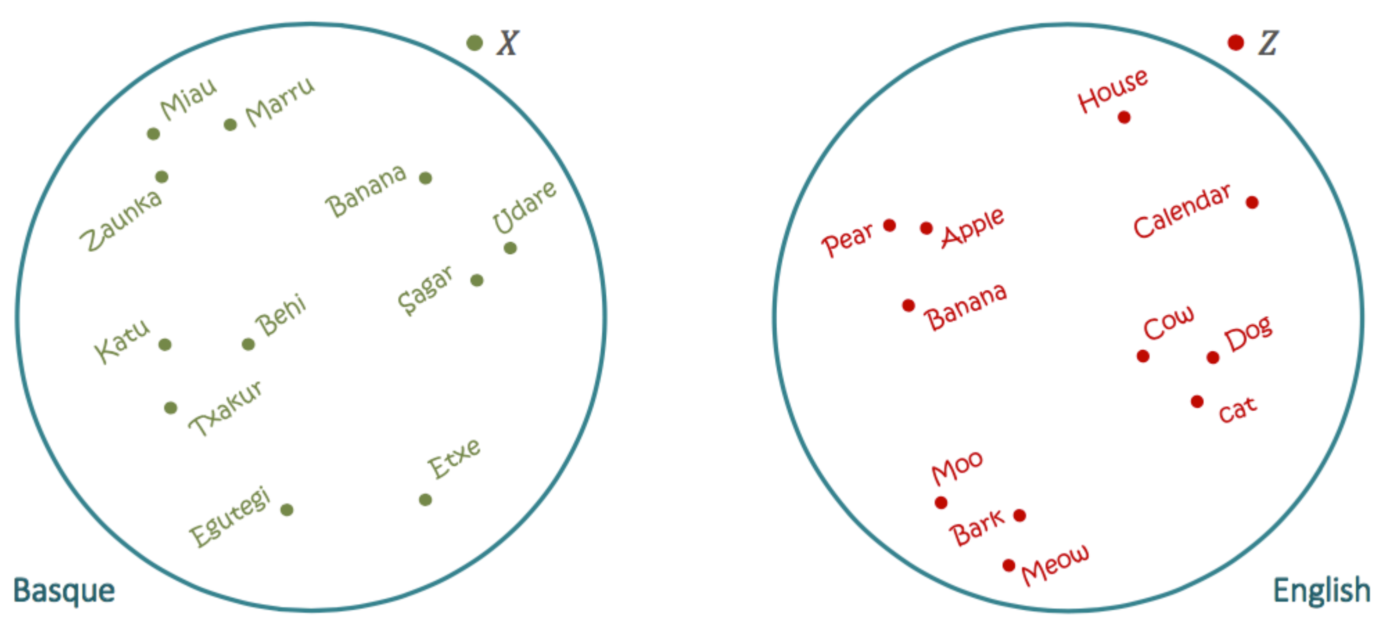
Unsupervised Cross-Lingual Word Embeddings

Unsupervised Cross-Lingual Word Embeddings

- Artetxe, Labake & Agirre, ACL 2017

Unsupervised Cross-Lingual Word Embeddings

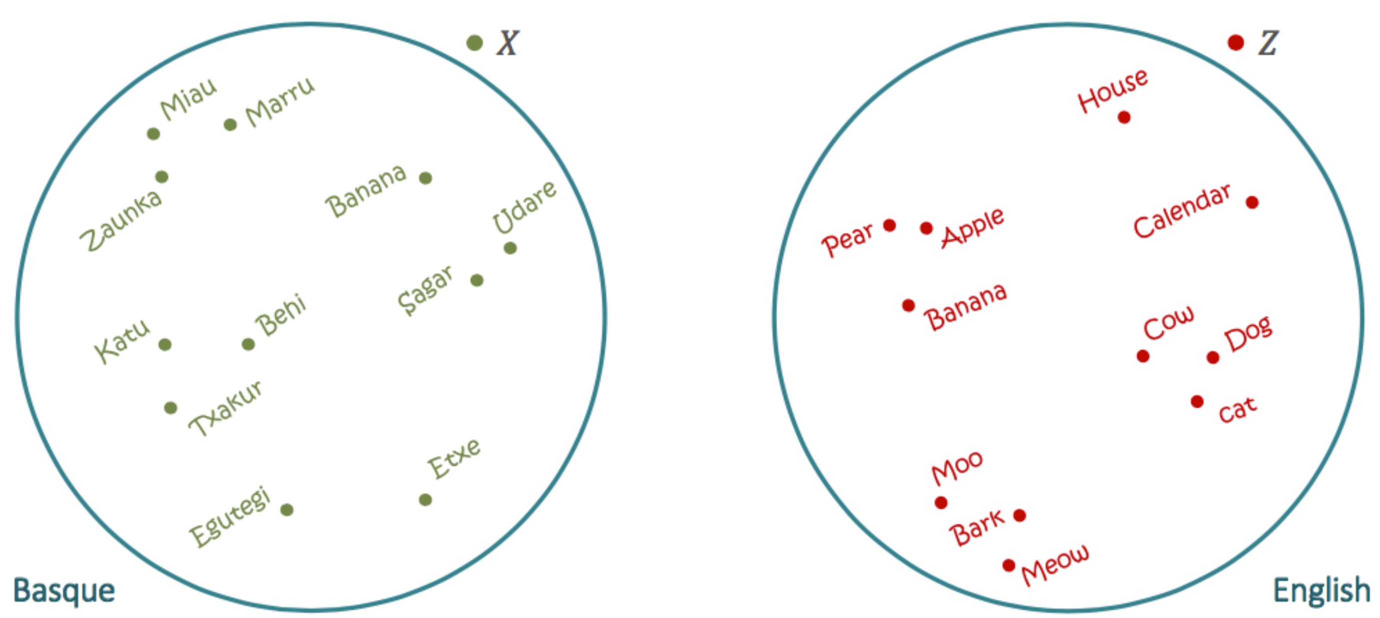
- Artetxe, Labake & Agirre, ACL 2017
- Start with monolingual embedding spaces in two languages (trained using word2vec)



From Artetxe, ACL 2017

Unsupervised Cross-Lingual Word Embeddings

- Artetxe, Labake & Agirre, ACL 2017
- Start with monolingual embedding spaces in two languages (trained using word2vec)
- Learn a linear mapping from one language to the other:



From Artetxe, ACL 2017

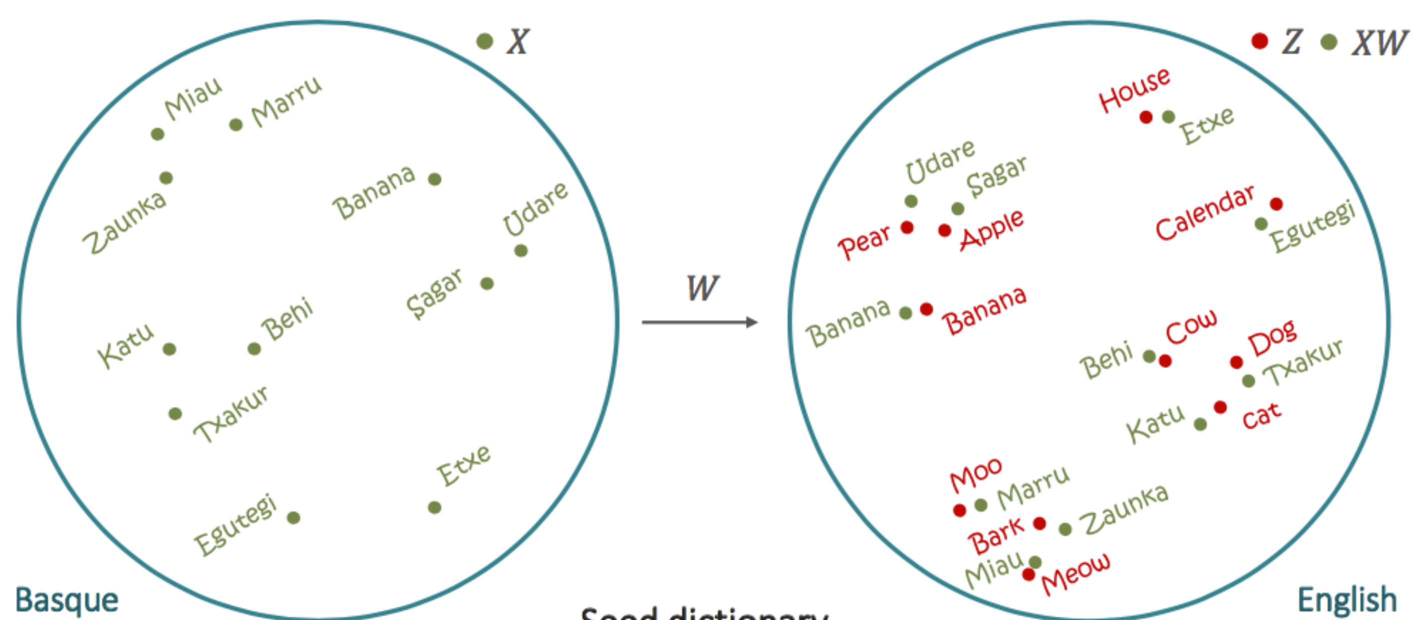
Unsupervised Cross-Lingual Word Embeddings

- Artetxe, Labake & Agirre, ACL 2017

- Start with monolingual embedding spaces in two languages (trained using word2vec)

- Learn a linear mapping from one language to the other:

  - Start with a **seed dictionary**. Clever idea: use numerals (5-5, 1989-1989…) as seed dictionary - fully unsupervised



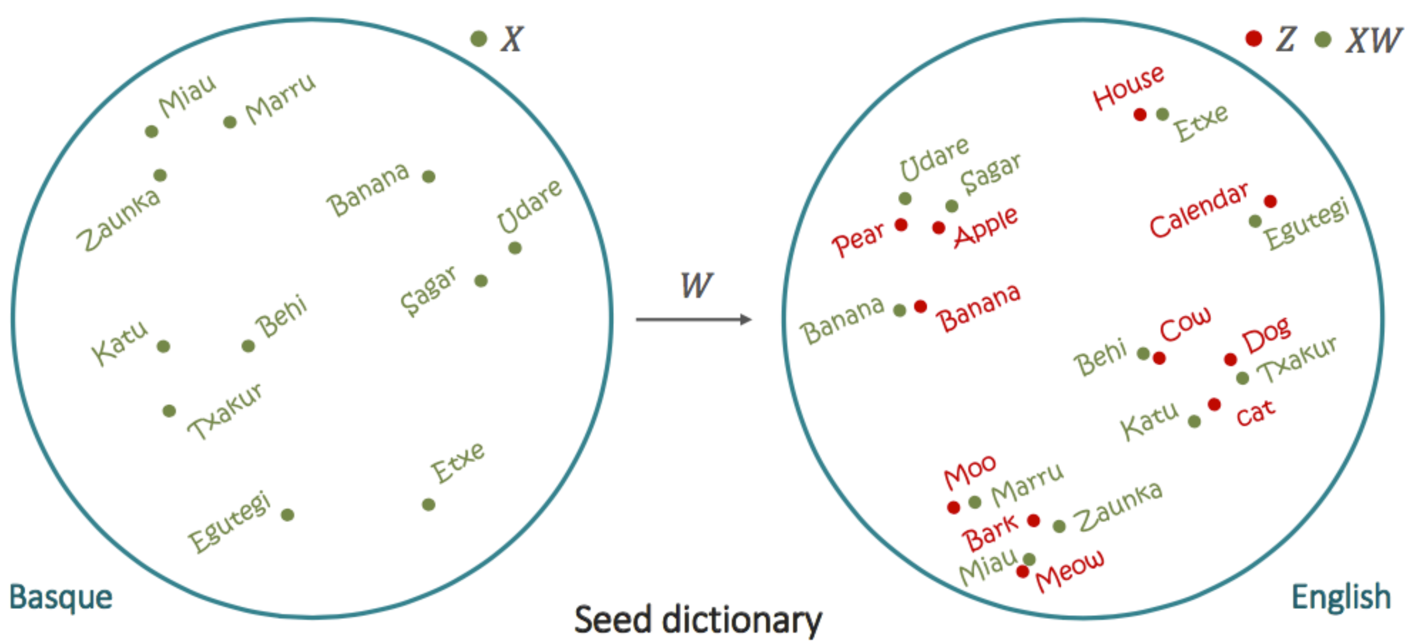
Seed dictionary

$$\begin{array}{r} \text{Txakur} \\ \text{Sagar} \\ \vdots \\ \text{Egutegi} \end{array} \begin{bmatrix} X_{1,*} \\ X_{2,*} \\ \vdots \\ X_{n,*} \end{bmatrix} \quad \Bigg| \quad \begin{bmatrix} Z_{1,*} \\ Z_{2,*} \\ \vdots \\ Z_{n,*} \end{bmatrix} \begin{array}{l} \text{Dog} \\ \text{Apple} \\ \vdots \\ \text{Calendar} \end{array}$$

From Artetxe, ACL 2017

Unsupervised Cross-Lingual Word Embeddings

- Artetxe, Labake & Agirre, ACL 2017
- Start with monolingual embedding spaces in two languages (trained using word2vec)
- Learn a linear mapping from one language to the other:
  - Start with a **seed dictionary**. Clever idea: use numerals (5-5, 1989-1989…) as seed dictionary - fully unsupervised
  - Optimize the mapping W w.r.t the dictionary: $$\underset{W\in O(n)}{\arg\min} \sum_i \left\|X_{i*}W - Z_{j*}\right\|^2$$



Seed dictionary

$$\begin{array}{r} \text{Txakur} \\ \text{Sagar} \\ \vdots \\ \text{Egutegi} \end{array} \begin{bmatrix} X_{1,*} \\ X_{2,*} \\ \vdots \\ X_{n,*} \end{bmatrix} \quad \Bigg| \quad \begin{bmatrix} Z_{1,*} \\ Z_{2,*} \\ \vdots \\ Z_{n,*} \end{bmatrix} \begin{array}{l} \text{Dog} \\ \text{Apple} \\ \vdots \\ \text{Calendar} \end{array}$$

From Artetxe, ACL 2017

Unsupervised Cross-Lingual Word Embeddings

- Artetxe, Labake & Agirre, ACL 2017

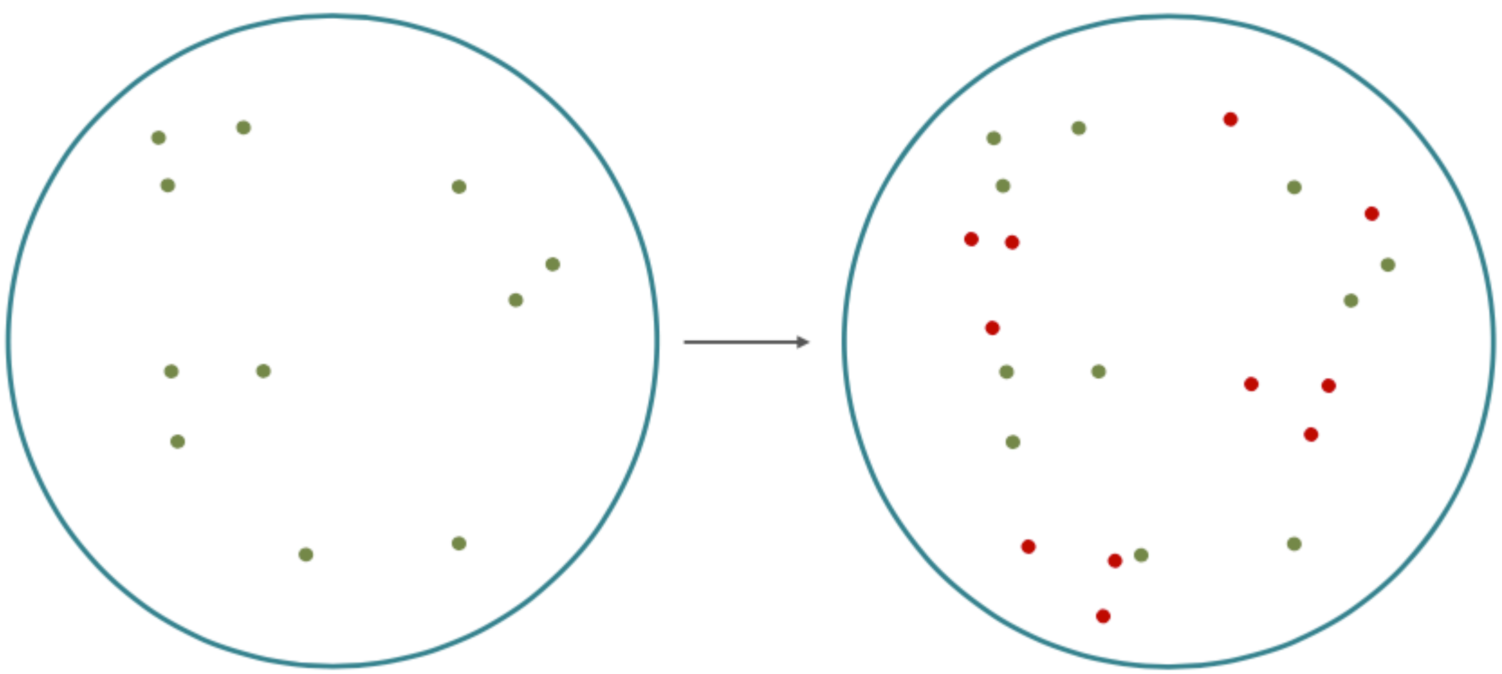
- Start with monolingual embedding spaces in two languages (trained using word2vec)

- Learn a linear mapping from one language to the other:

  - Start with a **seed dictionary**. Clever idea: use numerals (5-5, 1989-1989…) as seed dictionary - fully unsupervised

  - Optimize the mapping W w.r.t the dictionary: $\underset{W \in O(n)}{\arg\min} \sum_i \left\| X_{i*}W - Z_{j*} \right\|^2$

- Extract a new dictionary and **repeat iteratively** until a convergence threshold is met



From Artetxe, ACL 2017

Unsupervised Cross-Lingual Word Embeddings

- Artetxe, Labake & Agirre, ACL 2017

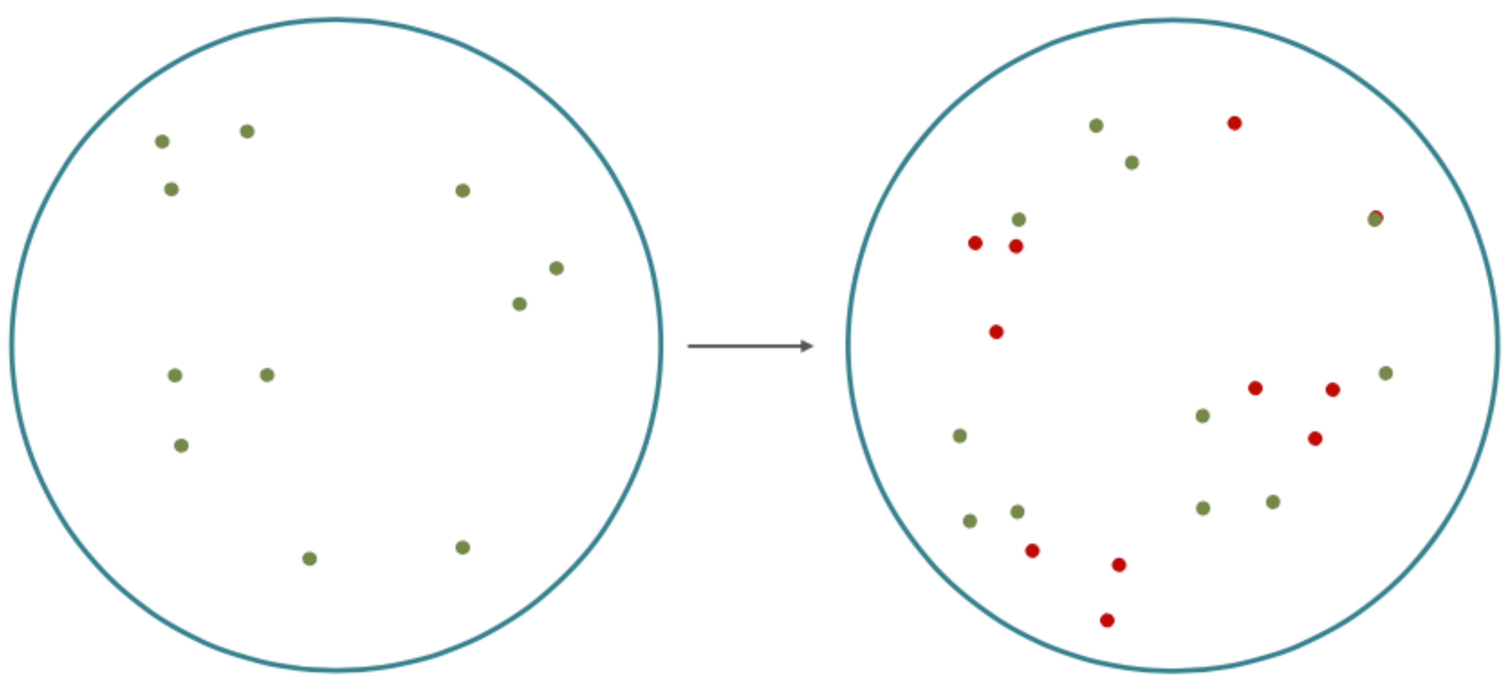
- Start with monolingual embedding spaces in two languages (trained using word2vec)

- Learn a linear mapping from one language to the other:

    - Start with a **seed dictionary**. Clever idea: use numerals (5-5, 1989-1989…) as seed dictionary - fully unsupervised

    - Optimize the mapping W w.r.t the dictionary: $\arg\min_{W \in O(n)} \sum_i \left\| X_{i*}W - Z_{j*} \right\|^2$

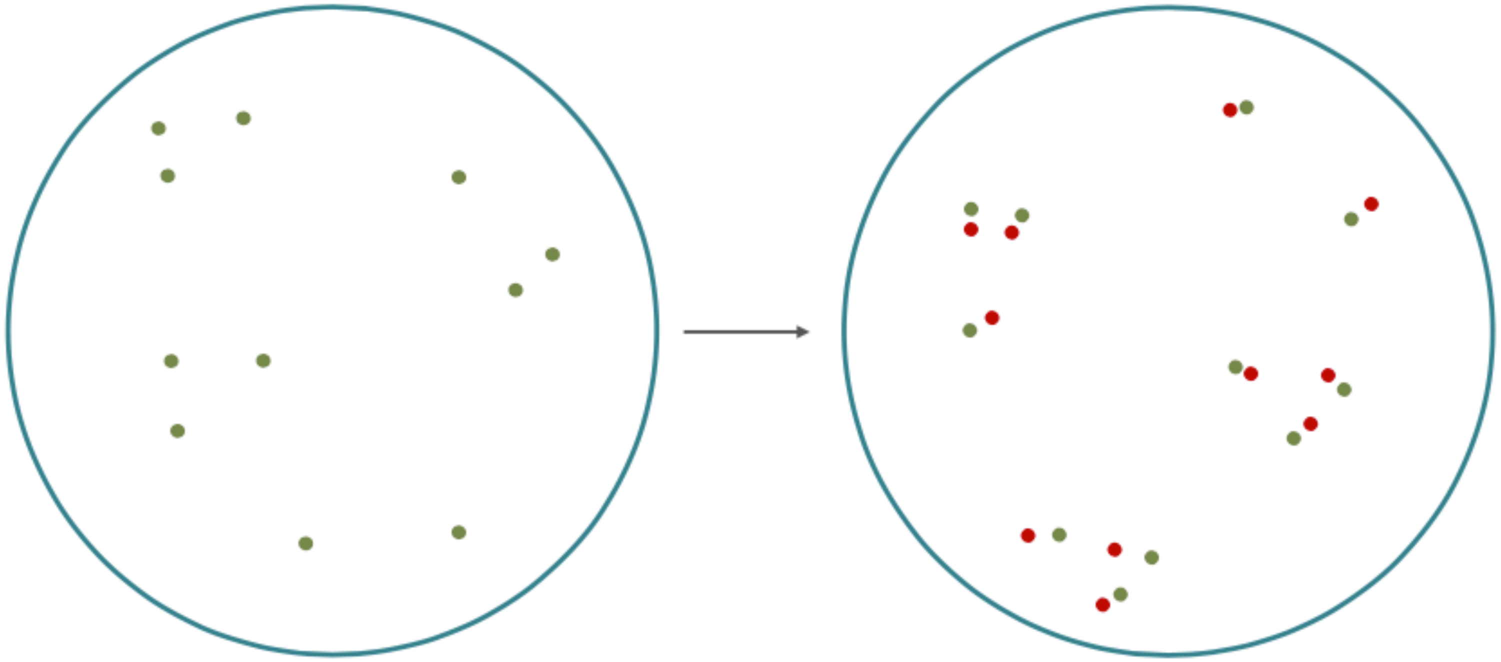
- Extract a new dictionary and **repeat iteratively** until a convergence threshold is met



From Artetxe, ACL 2017

Unsupervised Cross-Lingual Word Embeddings

- Artetxe, Labake & Agirre, ACL 2017
- Start with monolingual embedding spaces in two languages (trained using word2vec)
- Learn a linear mapping from one language to the other:
  - Start with a **seed dictionary**. Clever idea: use numerals (5-5, 1989-1989…) as seed dictionary - fully unsupervised
  - Optimize the mapping W w.r.t the dictionary: $\underset{W \in O(n)}{\arg\min} \sum_i \left\| X_{i*} W - Z_{j*} \right\|^2$
- Extract a new dictionary and **repeat iteratively** until a convergence threshold is met



From Artetxe, ACL 2017

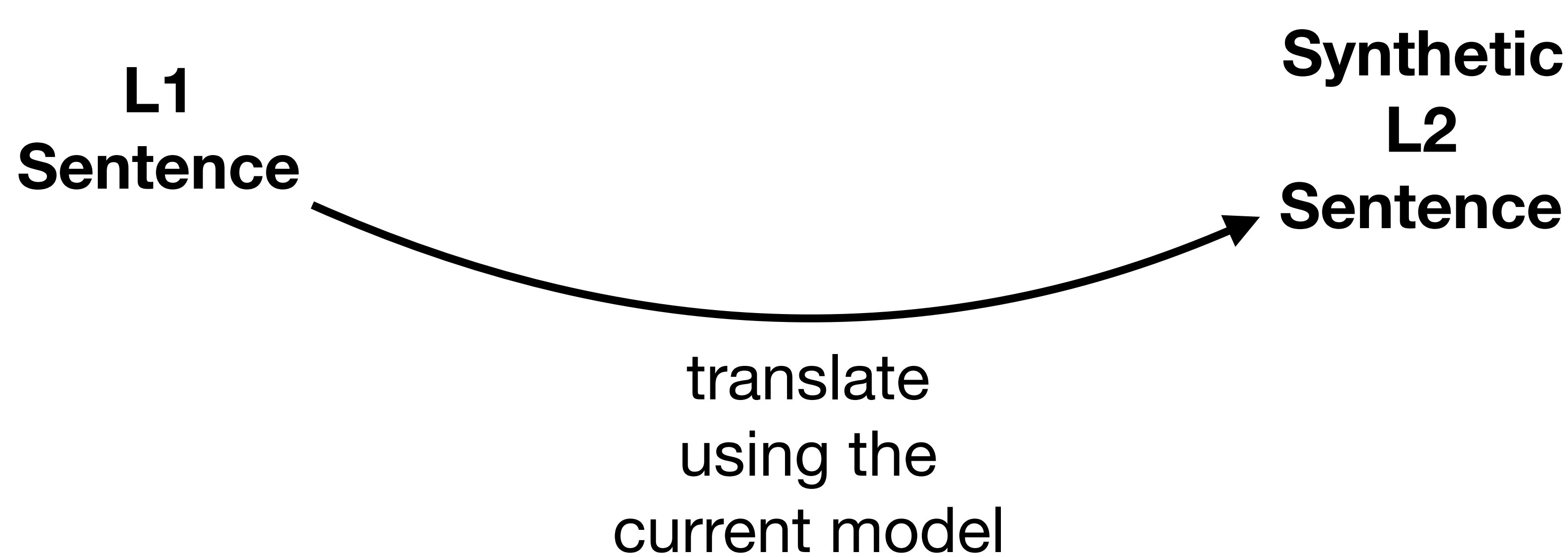
Learning Semantics - Back-translation

Learning Semantics - Back-translation

- We need to learn a mapping from one language (L1) into another (L2), but we don’t have parallel data

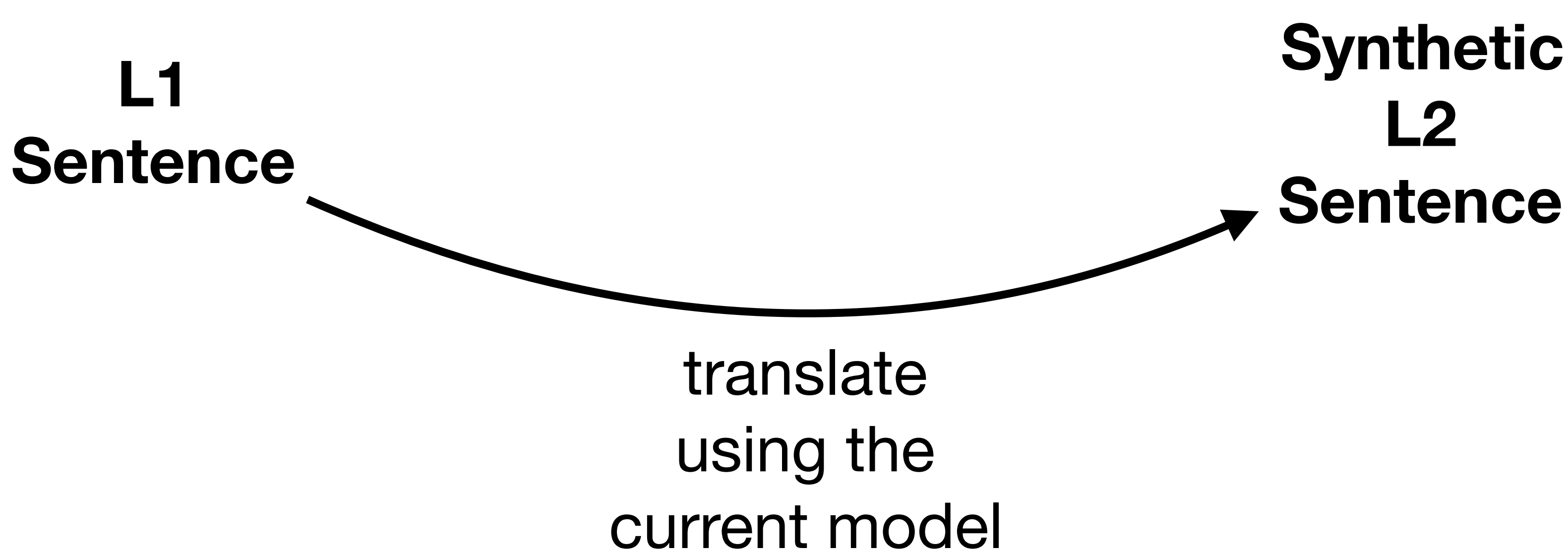
Learning Semantics - Back-translation

- We need to learn a mapping from one language (L1) into another (L2), but we don't have parallel data
- Solution: create **synthetic** parallel data by translating with the current model (possible since the model is bidirectional)



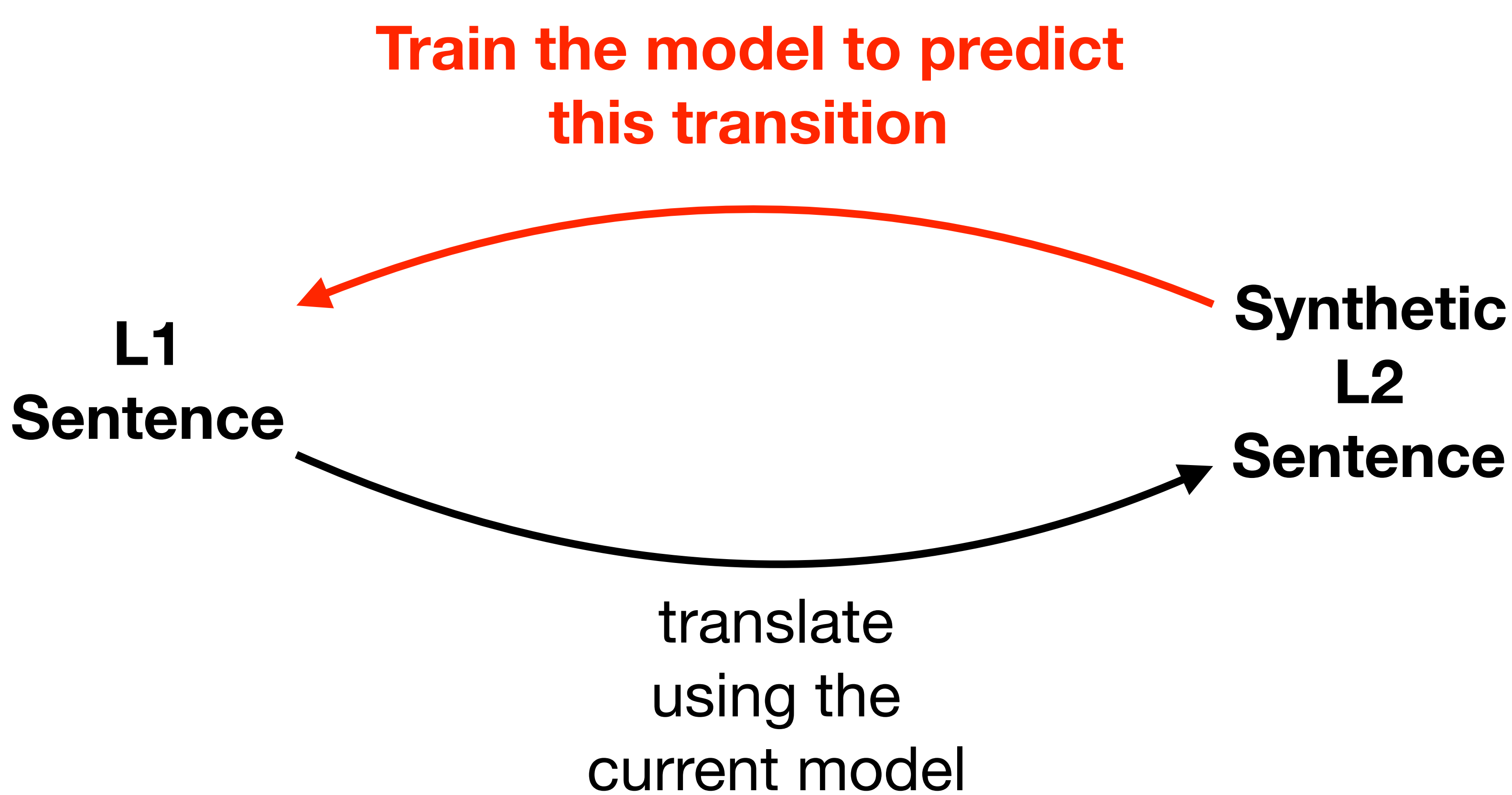
Learning Semantics - Back-translation

- We need to learn a mapping from one language (L1) into another (L2), but we don't have parallel data
- Solution: create **synthetic** parallel data by translating with the current model (possible since the model is bidirectional)
- Use the synthetic data for training using cross entropy loss



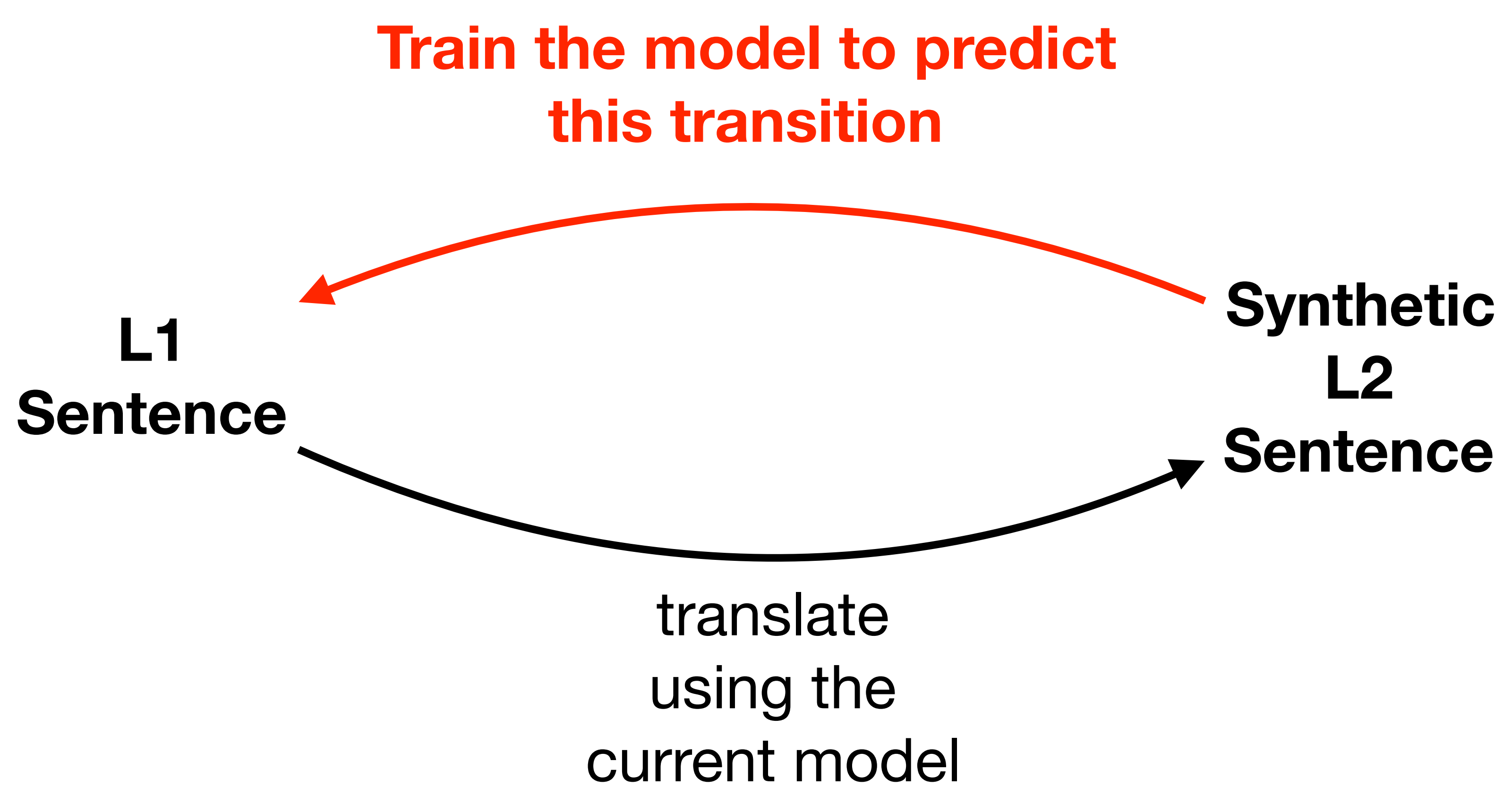
Learning Semantics - Back-translation

- We need to learn a mapping from one language (L1) into another (L2), but we don't have parallel data
- Solution: create **synthetic** parallel data by translating with the current model (possible since the model is bidirectional)
- Use the synthetic data for training using cross entropy loss



Learning Semantics - Back-translation

- We need to learn a mapping from one language (L1) into another (L2), but we don't have parallel data
- Solution: create **synthetic** parallel data by translating with the current model (possible since the model is bidirectional)
- Use the synthetic data for training using cross entropy loss
- **This is not entirely useless** since the cross-lingual embeddings do carry some alignment signal



Learning Structure - “Denoising” Auto-encoders

Learning Structure - “Denoising” Auto-encoders

- The decoder needs to learn how to organize the words on the target side

Learning Structure - “Denoising” Auto-encoders

- The decoder needs to learn how to organize the words on the target side
- We could train it to predict a sentence given itself - auto-encoding

Learning Structure - “Denoising” Auto-encoders

- The decoder needs to learn how to organize the words on the target side
- We could train it to predict a sentence given itself - auto-encoding
  - But this would lead it to learn trivial copying!

Learning Structure - “Denoising” Auto-encoders

- The decoder needs to learn how to organize the words on the target side
- We could train it to predict a sentence given itself - auto-encoding
  - But this would lead it to learn trivial copying!
- **Introduce “noise”** (by randomly swapping adjacent words, N/2 times) in the input, to force the decoder to learn word ordering

Learning Structure - “Denoising” Auto-encoders

- The decoder needs to learn how to organize the words on the target side
- We could train it to predict a sentence given itself - auto-encoding
  - But this would lead it to learn trivial copying!
- **Introduce “noise”** (by randomly swapping adjacent words, N/2 times) in the input, to force the decoder to learn word ordering

cat The on sat mat the → **The cat sat on the mat**

manger J'aime croissants des → **J'aime manger des croissants**

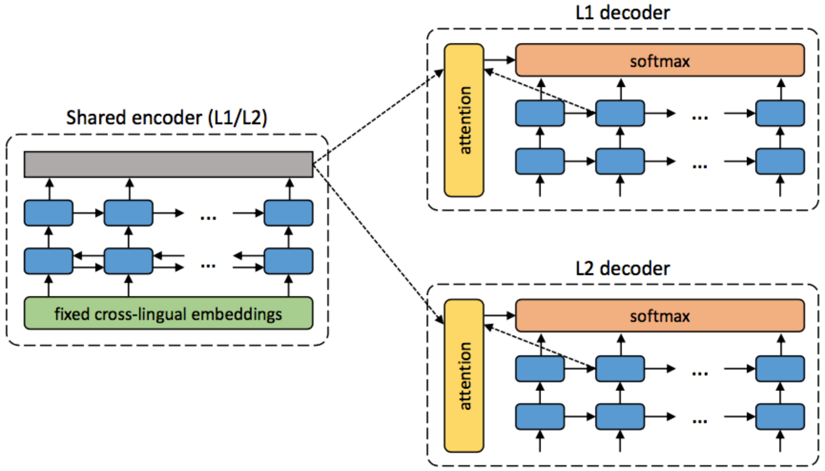
Learning Structure - “Denoising” Auto-encoders

- The decoder needs to learn how to organize the words on the target side
- We could train it to predict a sentence given itself - auto-encoding
  - But this would lead it to learn trivial copying!
- **Introduce “noise”** (by randomly swapping adjacent words, N/2 times) in the input, to force the decoder to learn word ordering
- Using conventional cross entropy loss

cat The on sat mat the → **The cat sat on the mat**

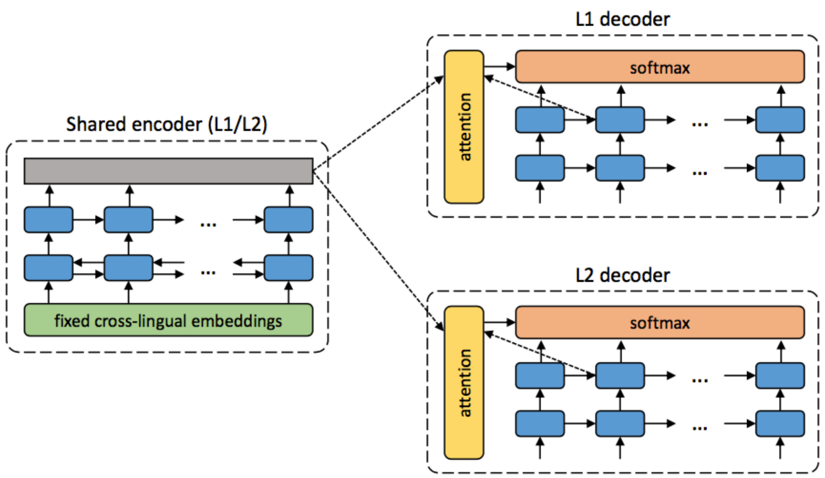
manger J'aime croissants des → **J'aime manger des croissants**

Putting it all together - Iterative Training



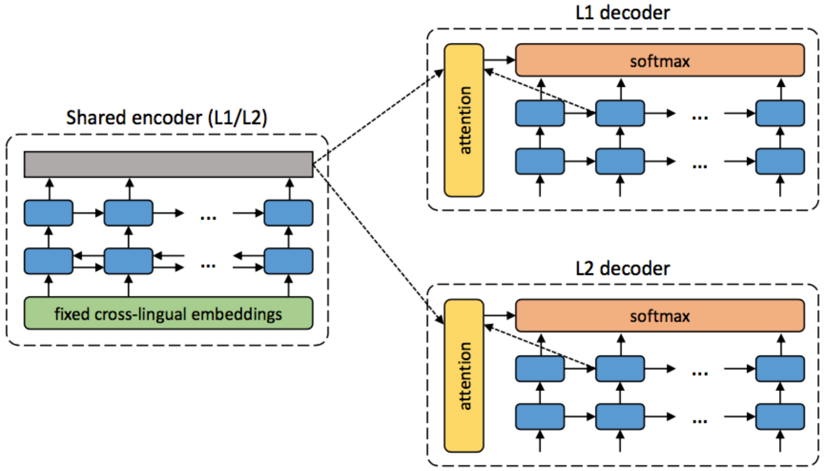
Putting it all together - Iterative Training

- Training iterations alternate between denoising and back-translation



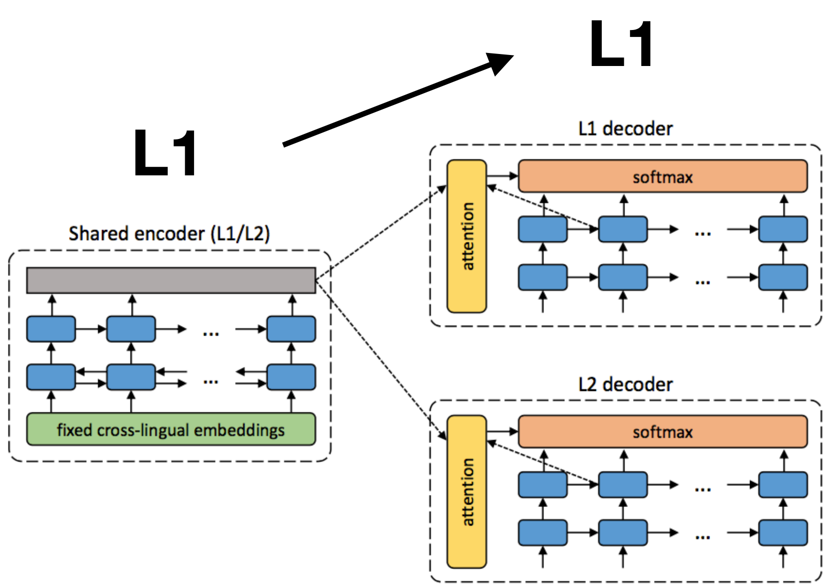
Putting it all together - Iterative Training

- Training iterations alternate between denoising and back-translation
- Each training iteration is composed of:



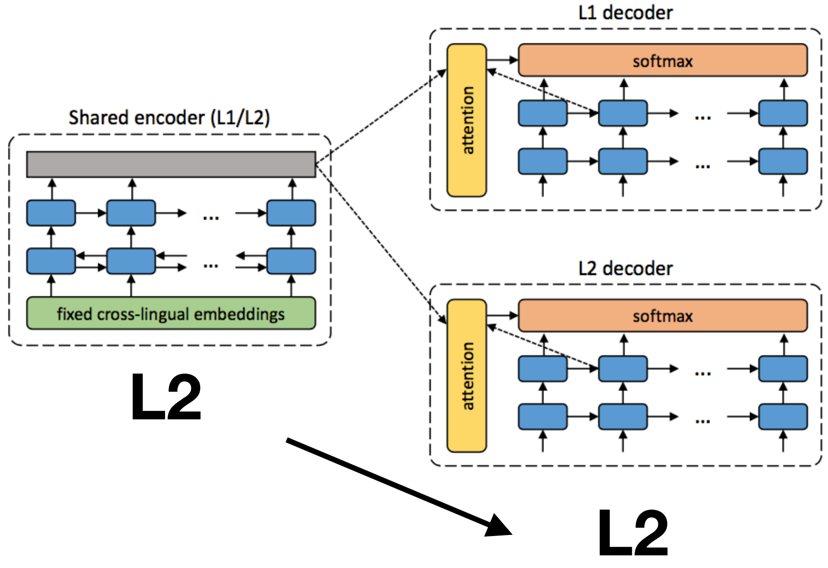
Putting it all together - Iterative Training

- Training iterations alternate between denoising and back-translation
- Each training iteration is composed of:
  - Denoising batch: L1 to L1



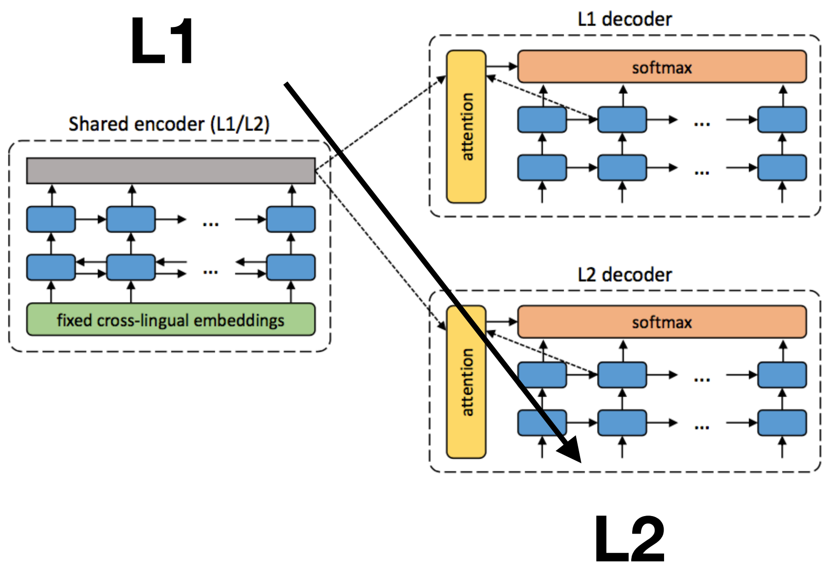
Putting it all together - Iterative Training

- Training iterations alternate between denoising and back-translation
- Each training iteration is composed of:
    - Denoising batch: L1 to L1
    - Denoising batch: L2 to L2



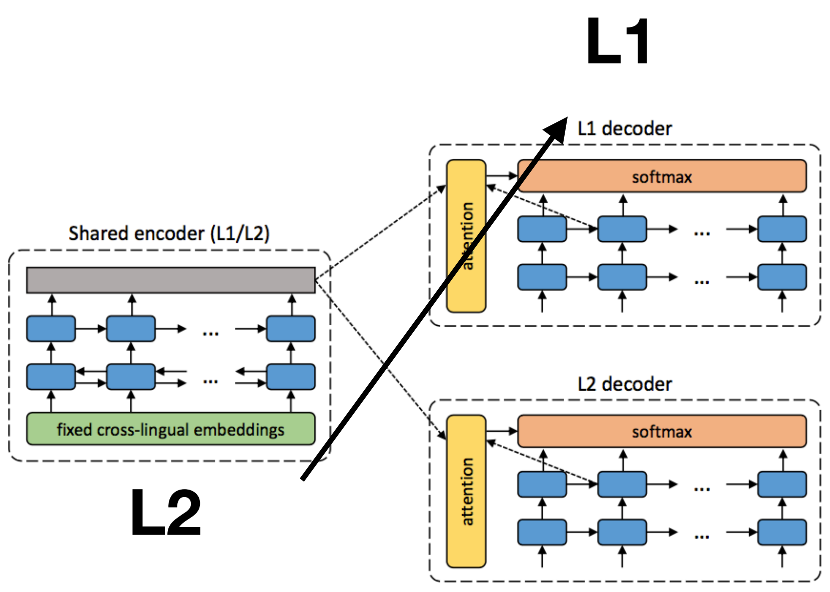
Putting it all together - Iterative Training

- Training iterations alternate between denoising and back-translation
- Each training iteration is composed of:
  - Denoising batch: L1 to L1
  - Denoising batch: L2 to L2
  - Back-translation batch: L1 to L2



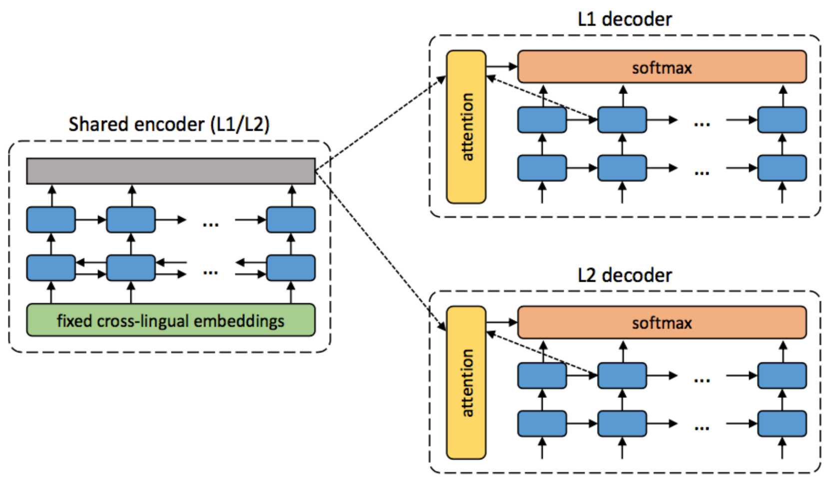
Putting it all together - Iterative Training

- Training iterations alternate between denoising and back-translation
- Each training iteration is composed of:
    - Denoising batch: L1 to L1
    - Denoising batch: L2 to L2
    - Back-translation batch: L1 to L2
    - Back-translation batch: L2 to L1



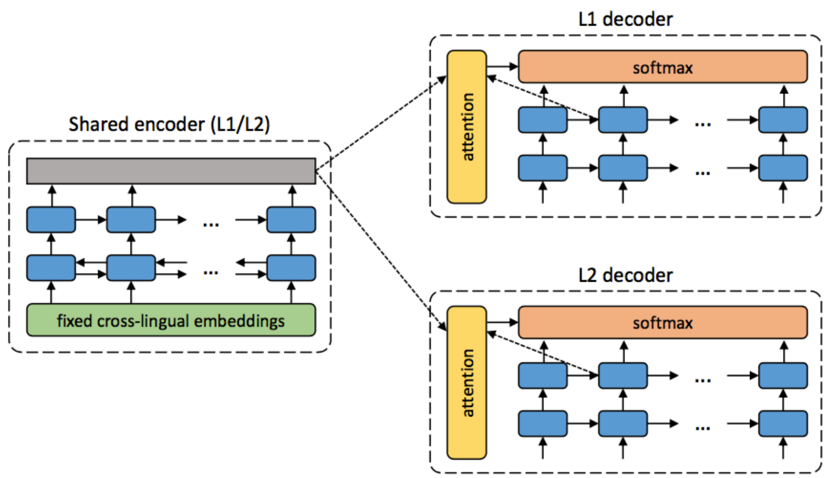
Putting it all together - Iterative Training

- Training iterations alternate between denoising and back-translation
- Each training iteration is composed of:
  - Denoising batch: L1 to L1
  - Denoising batch: L2 to L2
  - Back-translation batch: L1 to L2
  - Back-translation batch: L2 to L1
- When do we stop? Can’t use parallel validation set!



Putting it all together - Iterative Training

- Training iterations alternate between denoising and back-translation
- Each training iteration is composed of:
  - Denoising batch: L1 to L1
  - Denoising batch: L2 to L2
  - Back-translation batch: L1 to L2
  - Back-translation batch: L2 to L1



- When do we stop? Can’t use parallel validation set!
  - Train for a fixed amount of iterations (300k)

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) + 100k parallel	21.81	21.74	15.24	10.95
Supervised	6. Comparable NMT	20.48	19.89	15.04	11.05
	7. GNMT (Wu et al., 2016)	-	38.95	-	24.61

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) + 100k parallel	21.81	21.74	15.24	10.95
Supervised	6. Comparable NMT	20.48	19.89	15.04	11.05
	7. GNMT (Wu et al., 2016)	-	38.95	-	24.61

- Denoising alone degrades performance of embeddings nearest-neighbor

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) + 100k parallel	21.81	21.74	15.24	10.95
Supervised	6. Comparable NMT	20.48	19.89	15.04	11.05
	7. GNMT (Wu et al., 2016)	-	38.95	-	24.61

- Denoising alone degrades performance of embeddings nearest-neighbor
- Denoising+Back-translation improves results significantly

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) + 100k parallel	21.81	21.74	15.24	10.95
Supervised	6. Comparable NMT	20.48	19.89	15.04	11.05
	7. GNMT (Wu et al., 2016)	-	38.95	-	24.61

- Denoising alone degrades performance of embeddings nearest-neighbor
- Denoising+Back-translation improves results significantly
- No clear benefit from BPE (perhaps hurts embedding learning?)

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) + 100k parallel	21.81	21.74	15.24	10.95
Supervised	6. Comparable NMT	20.48	19.89	15.04	11.05
	7. GNMT (Wu et al., 2016)	-	38.95	-	24.61

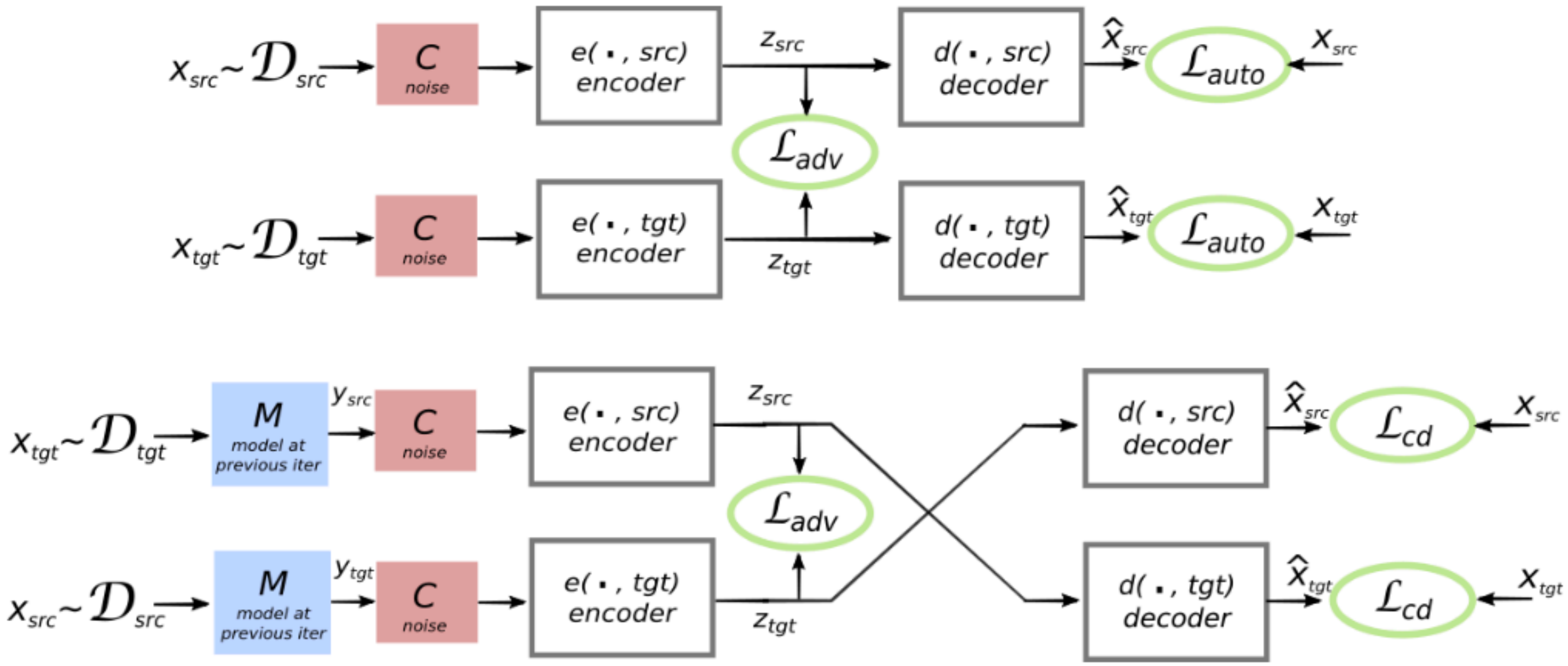
- Denoising alone degrades performance of embeddings nearest-neighbor
- Denoising+Back-translation improves results significantly
- No clear benefit from BPE (perhaps hurts embedding learning?)
- Semi supervised learning can also use this framework with notable gains

Results

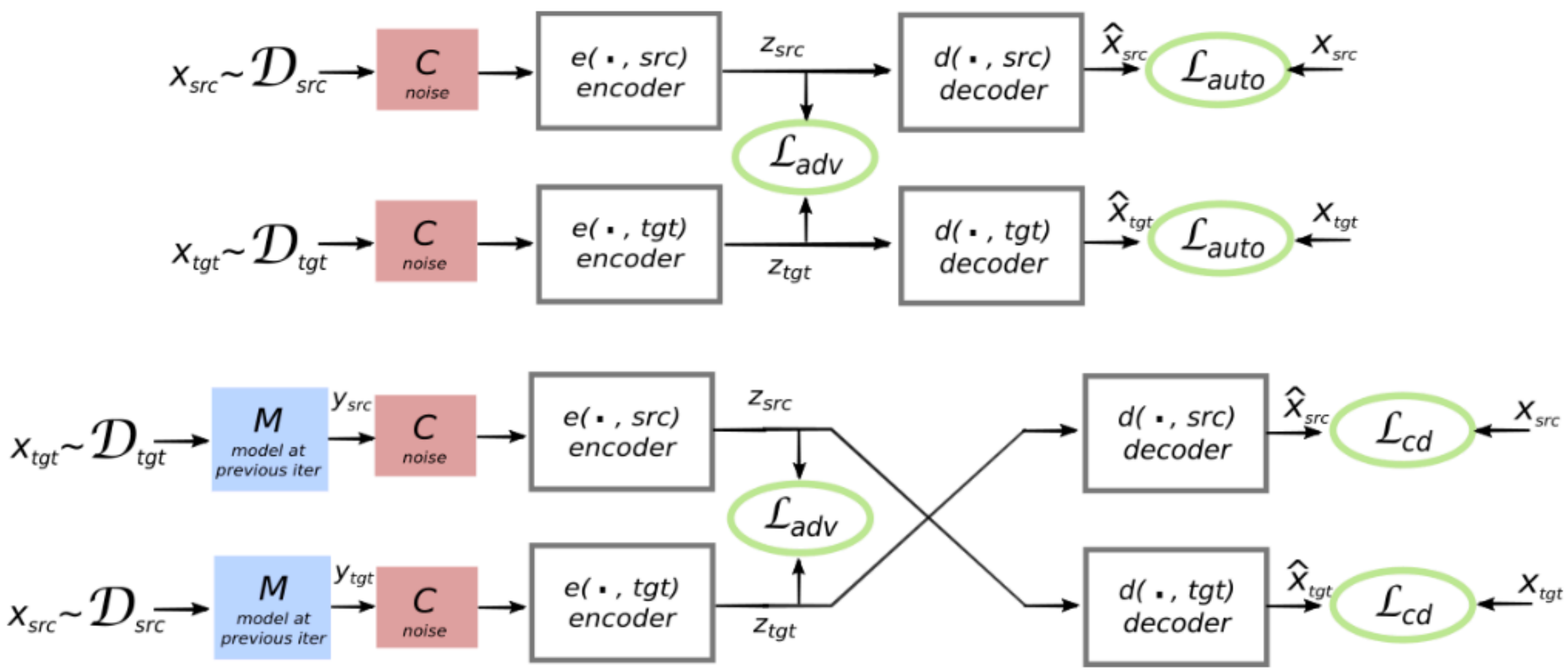
		FR-EN	EN-FR	DE-EN	EN-DE
Unsupervised	1. Baseline (emb. nearest neighbor)	9.98	6.25	7.07	4.39
	2. Proposed (denoising)	7.28	5.33	3.64	2.40
	3. Proposed (+ backtranslation)	15.56	15.13	10.21	6.55
	4. Proposed (+ BPE)	15.56	14.36	10.16	6.89
Semi-supervised	5. Proposed (full) + 100k parallel	21.81	21.74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

- Denoising alone degrades performance of embeddings nearest-neighbor
- Denoising+Back-translation improves results significantly
- No clear benefit from BPE (perhaps hurts embedding learning?)
- Semi supervised learning can also use this framework with notable gains
- **Still a very large gap from the supervised approach (but a nice start nonetheless)**

Paper II: Lample, Denoyer & Ranzato

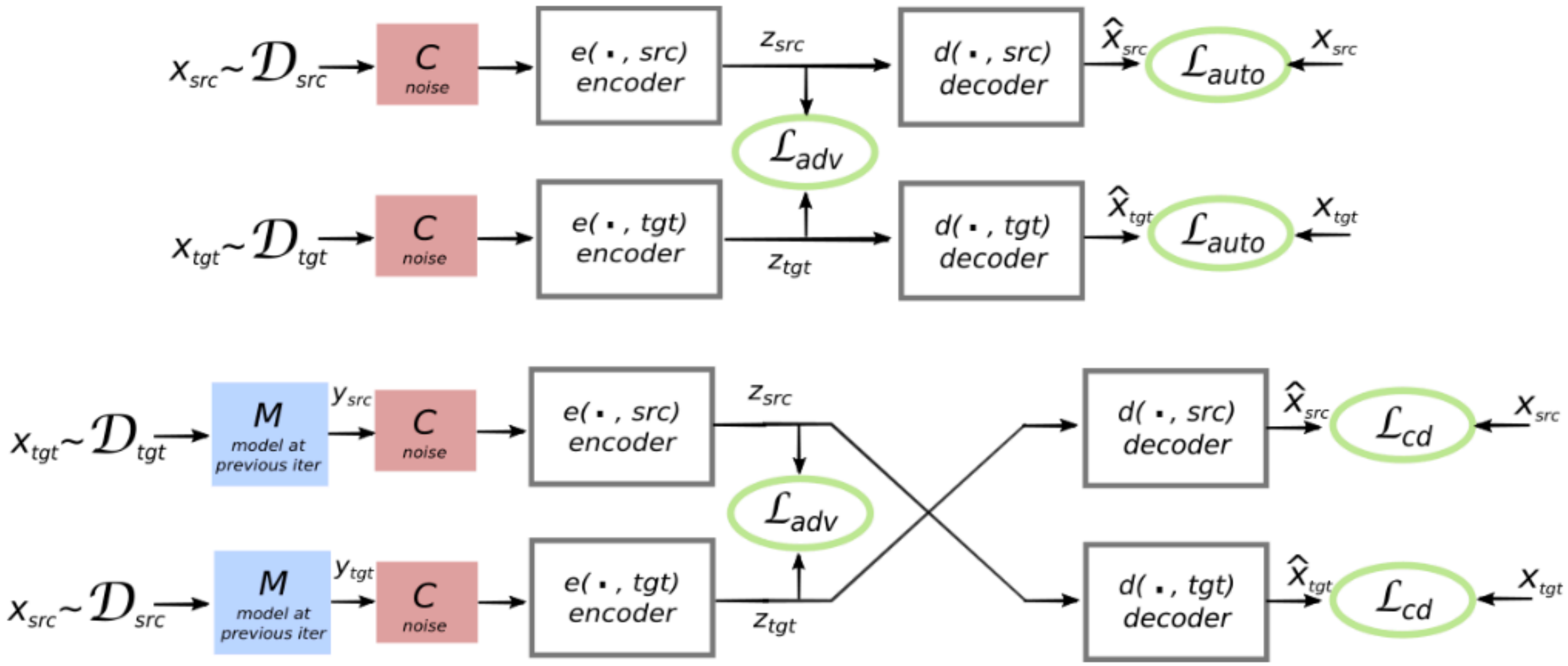


Paper II: Lample, Denoyer & Ranzato



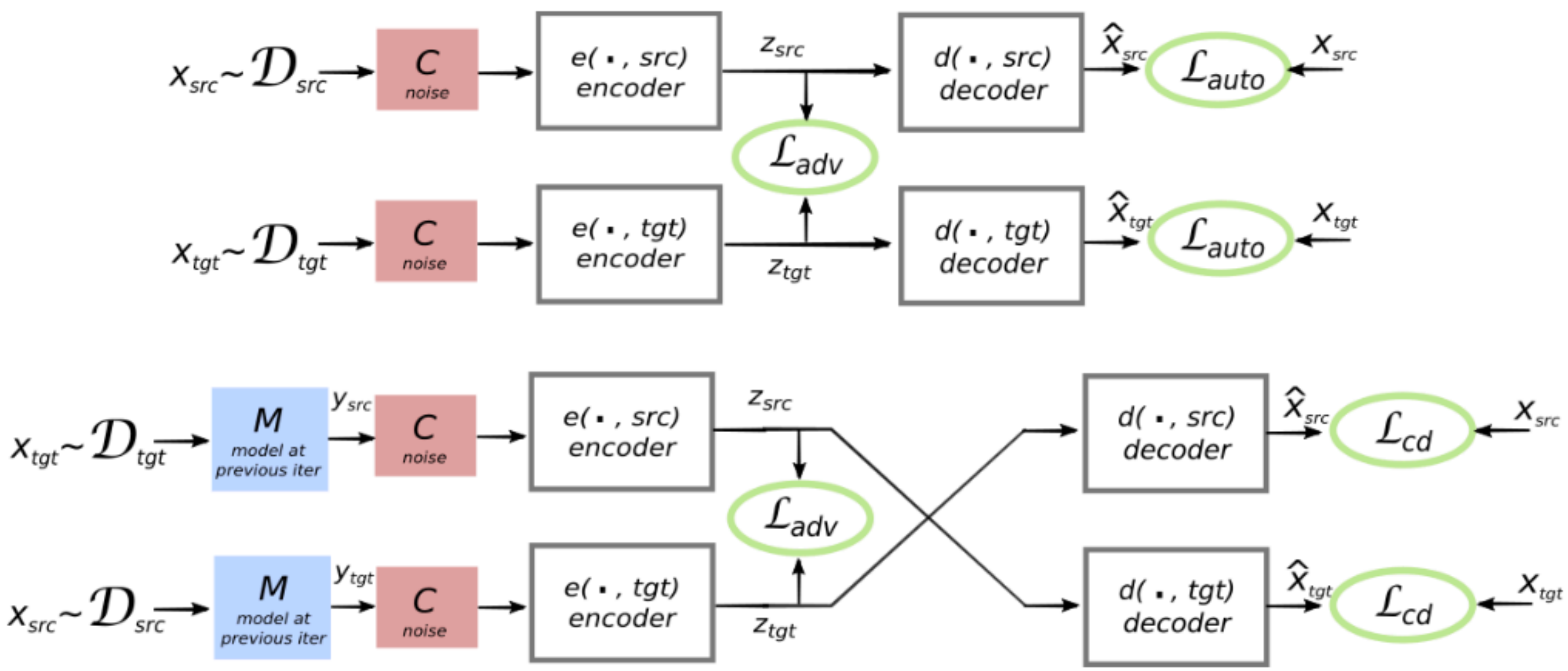
- **Model Architecture:**

Paper II: Lample, Denoyer & Ranzato



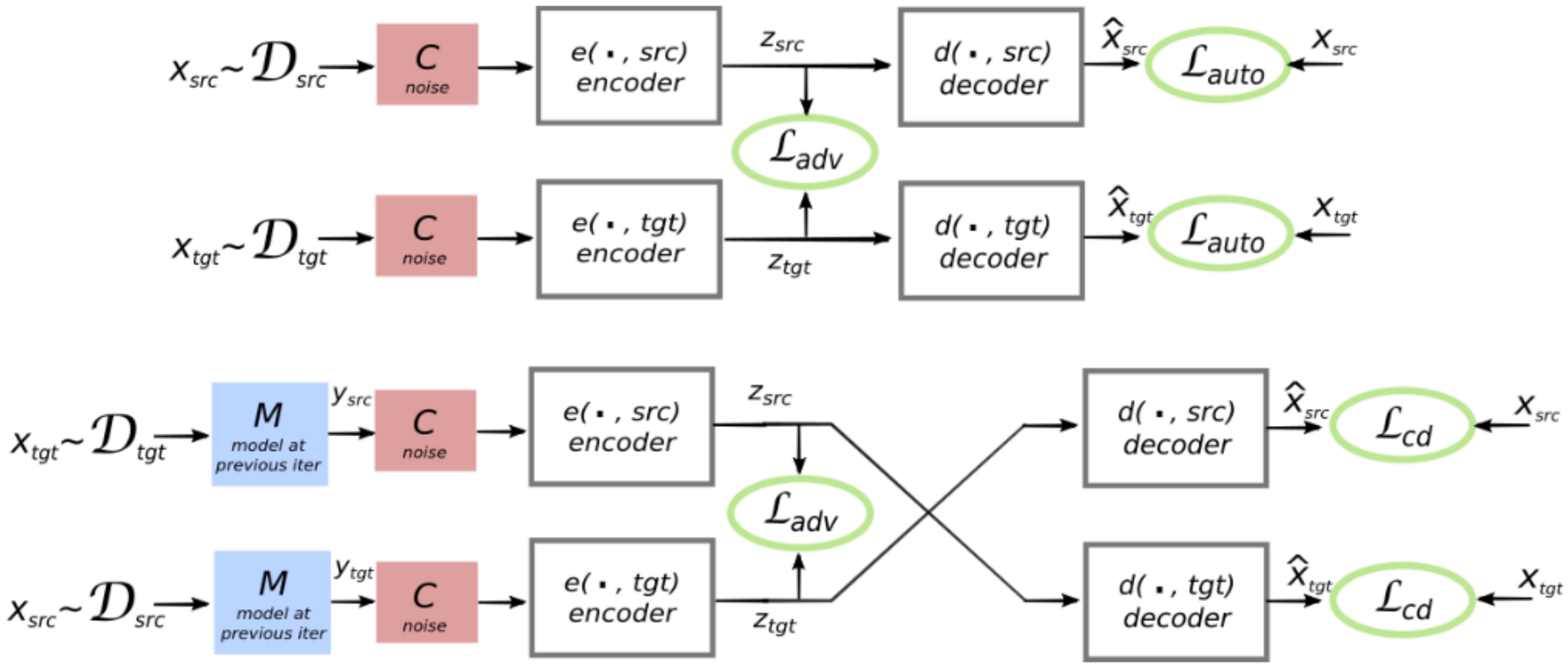
- **Model Architecture:**
  - **Shared** GRU encoder, **Shared** GRU decoder

Paper II: Lample, Denoyer & Ranzato



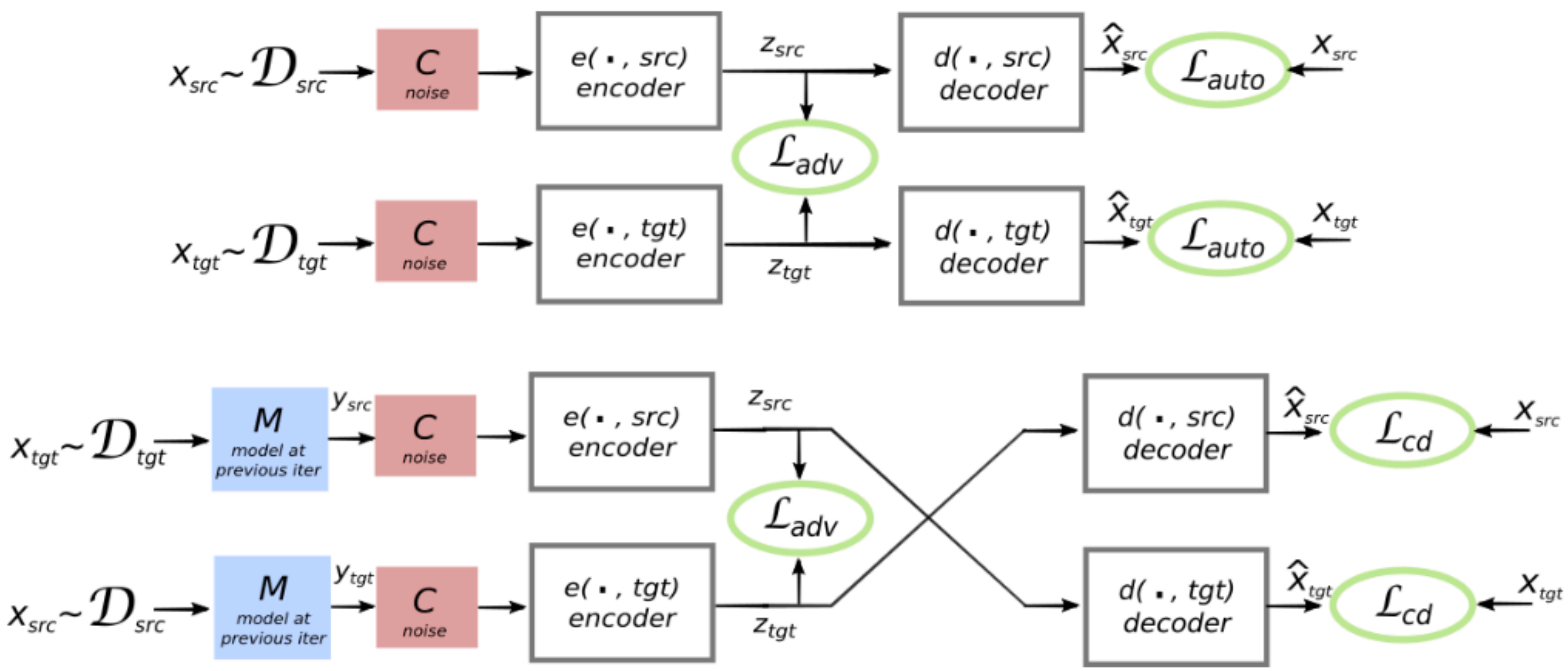
- **Model Architecture:**
  - **Shared** GRU encoder, **Shared** GRU decoder
  - Attention

Paper II: Lample, Denoyer & Ranzato



- **Model Architecture:**
  - **Shared** GRU encoder, **Shared** GRU decoder
  - Attention
- **Main “Tricks”:**

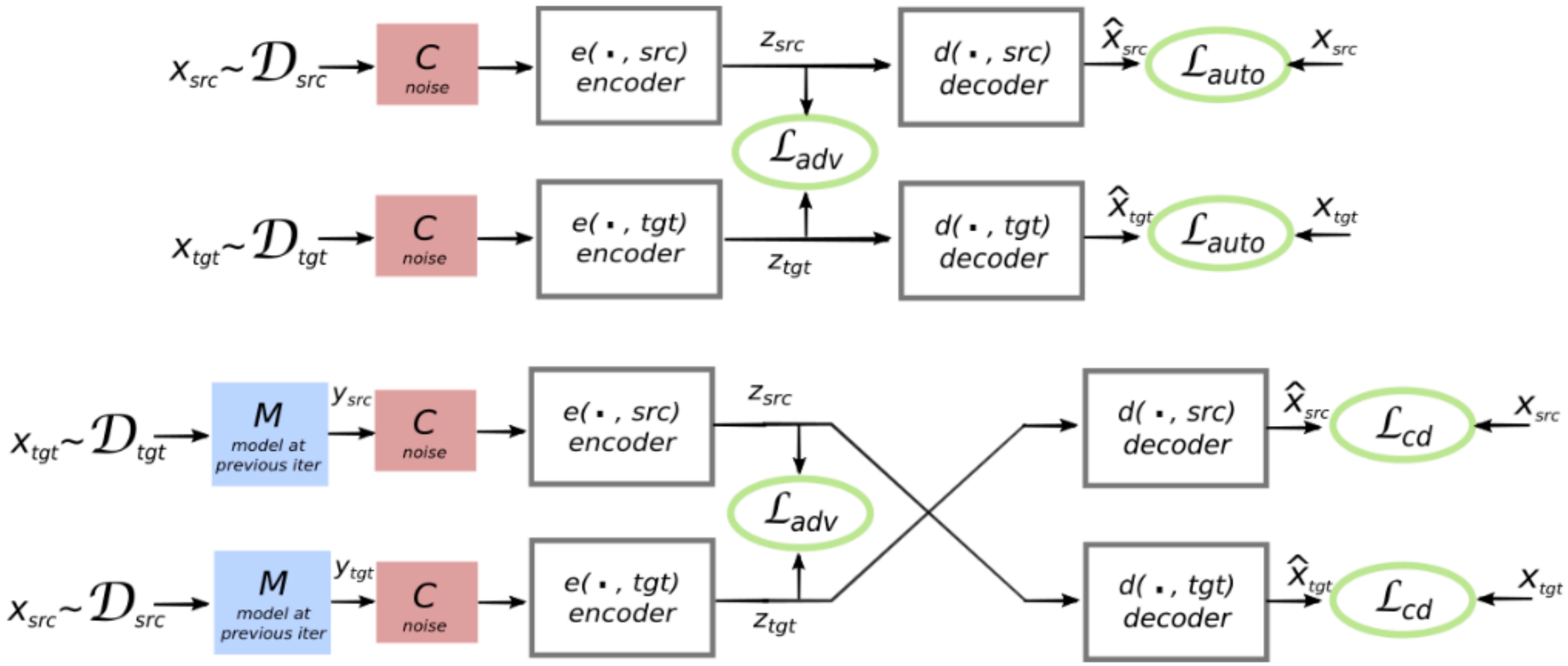
Paper II: Lample, Denoyer & Ranzato



- **Model Architecture:**
  - **Shared** GRU encoder, **Shared** GRU decoder
  - Attention

- **Main “Tricks”:**
  - **Changing, adversarially trained** unsupervised cross-lingual embeddings (**Adequacy**)

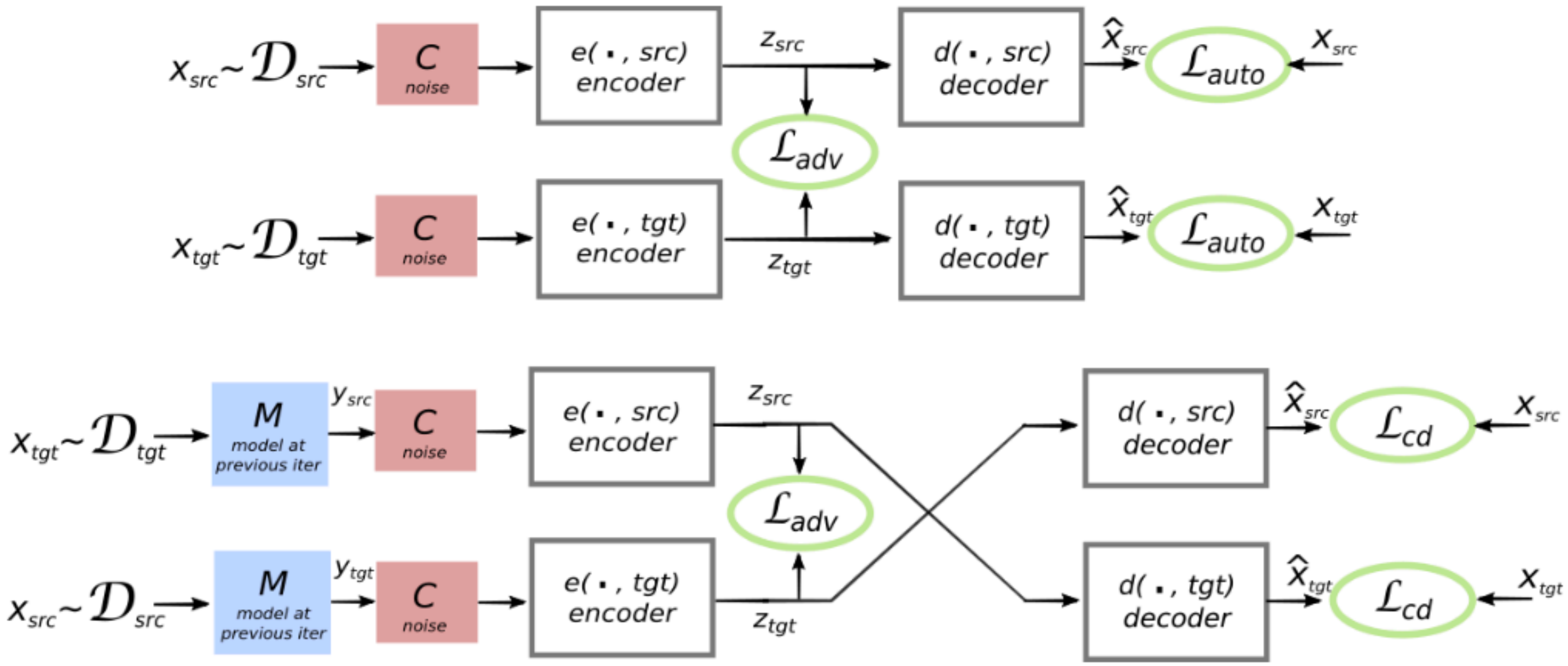
Paper II: Lample, Denoyer & Ranzato



- **Model Architecture:**
  - **Shared** GRU encoder, **Shared** GRU decoder
  - Attention

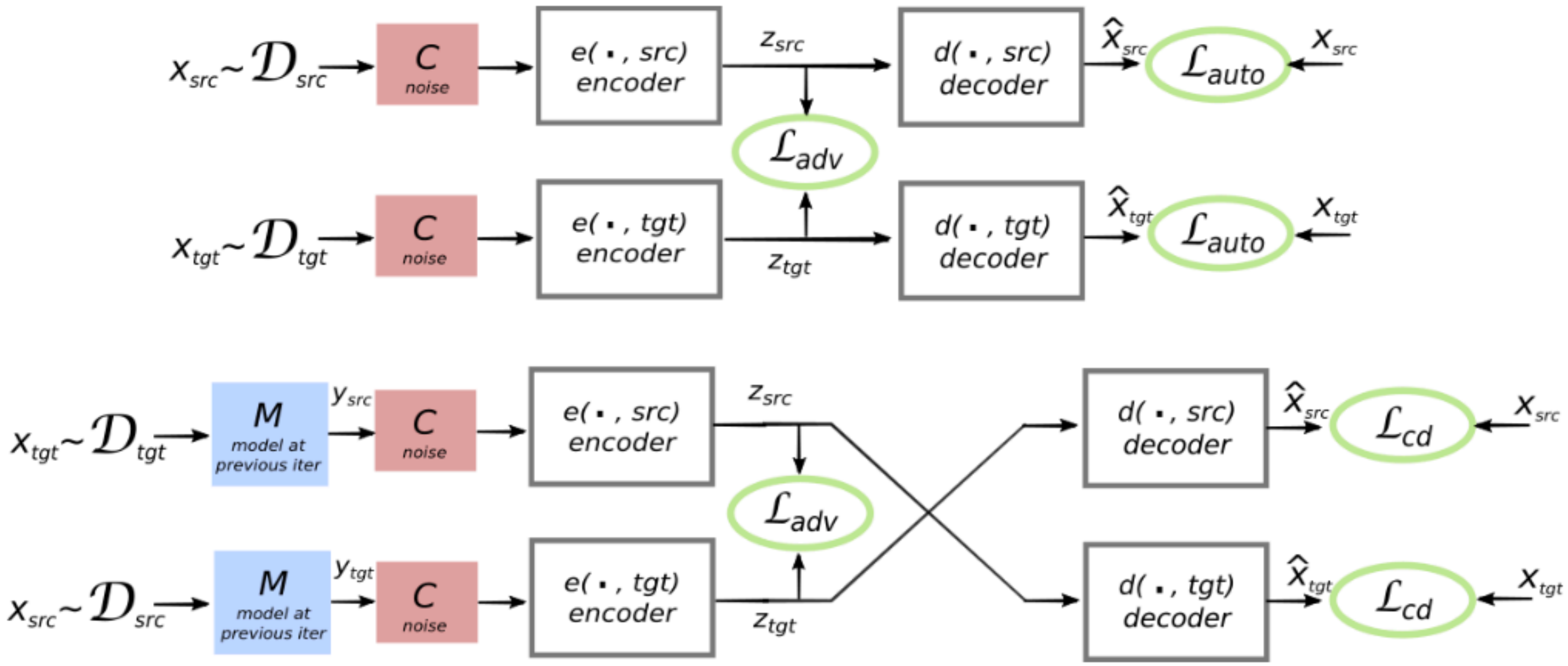
- **Main “Tricks”:**
  - **Changing, adversarially trained** unsupervised cross-lingual embeddings (**Adequacy**)
  - Backtranslation loss (**Adequacy**)

Paper II: Lample, Denoyer & Ranzato



- **Model Architecture:**
  - **Shared** GRU encoder, **Shared** GRU decoder
  - Attention
- **Main "Tricks":**
  - **Changing, adversarially trained** unsupervised cross-lingual embeddings (**Adequacy**)
  - Backtranslation loss (**Adequacy**)
  - Denoising auto-encoder loss (**Fluency**)

Paper II: Lample, Denoyer & Ranzato



- **Model Architecture:**
  - **Shared** GRU encoder, **Shared** GRU decoder
  - Attention

- **Main "Tricks":**
  - **Changing, adversarially trained** unsupervised cross-lingual embeddings (**Adequacy**)
  - Backtranslation loss (**Adequacy**)
  - Denoising auto-encoder loss (**Fluency**)
  - **Adversarial loss**

Adversarial Learning

Adversarial Learning

- Introduced by Ganin et al., 2016 for domain adaption in computer vision

Adversarial Learning

- Introduced by Ganin et al., 2016 for domain adaption in computer vision
- The general idea: force the model to “unlearn” a specific objective to make it learn better representation for the target objective

Adversarial Learning

- Introduced by Ganin et al., 2016 for domain adaption in computer vision
- The general idea: force the model to “unlearn” a specific objective to make it learn better representation for the target objective
- Used twice here:

Adversarial Learning

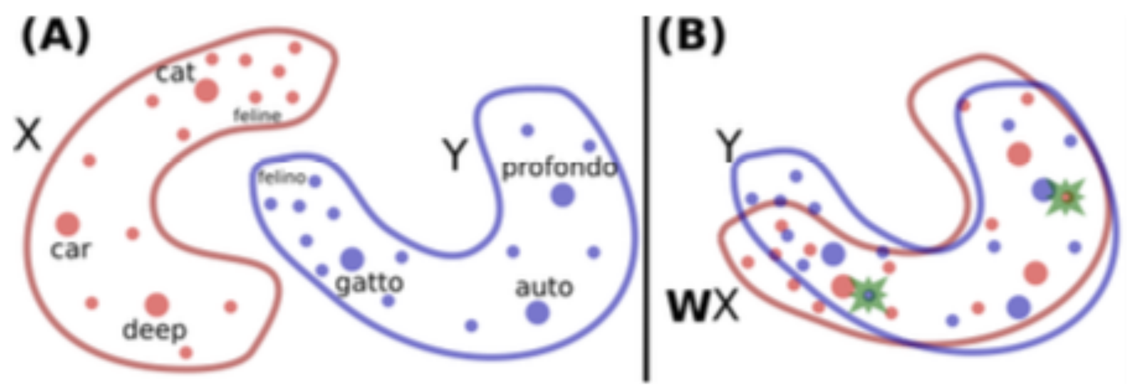
- Introduced by Ganin et al., 2016 for domain adaption in computer vision

- The general idea: force the model to “unlearn” a specific objective to make it learn better representation for the target objective

- Used twice here:

    - In the cross-lingual embedding learning - to learn a mapping from one embedding space to the other:

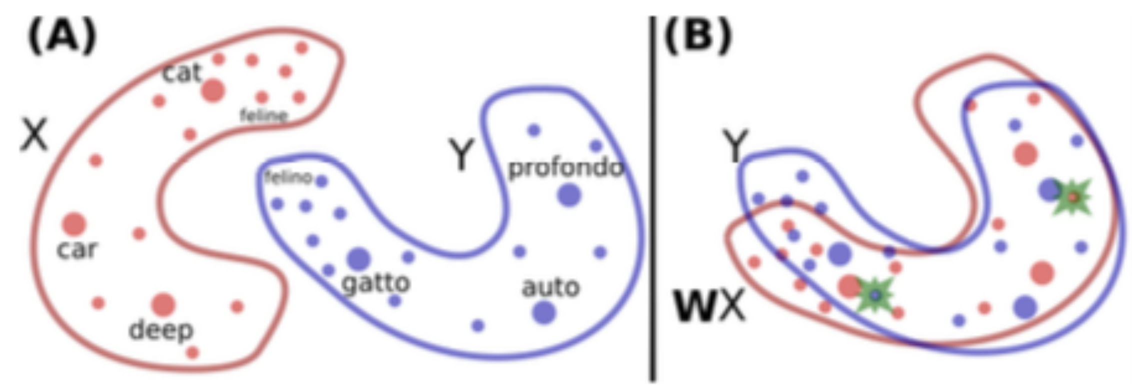
Conneau et al. 2017



Adversarial Learning

- Introduced by Ganin et al., 2016 for domain adaption in computer vision
- The general idea: force the model to “unlearn” a specific objective to make it learn better representation for the target objective
- Used twice here:
    - In the cross-lingual embedding learning - to learn a mapping from one embedding space to the other:

Conneau et al. 2017



    - In the NMT training - to “push” the representations from the two languages to a shared “semantic” space

$$p_D(l|z_1, ..., z_m) \propto \prod_{j=1}^{m} p_D(\ell|z_j),$$

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

Unsupervised Model Selection Criterion

Unsupervised Model Selection Criterion

- When do we stop training without a validation set? can we do better than fixed amount of updates?

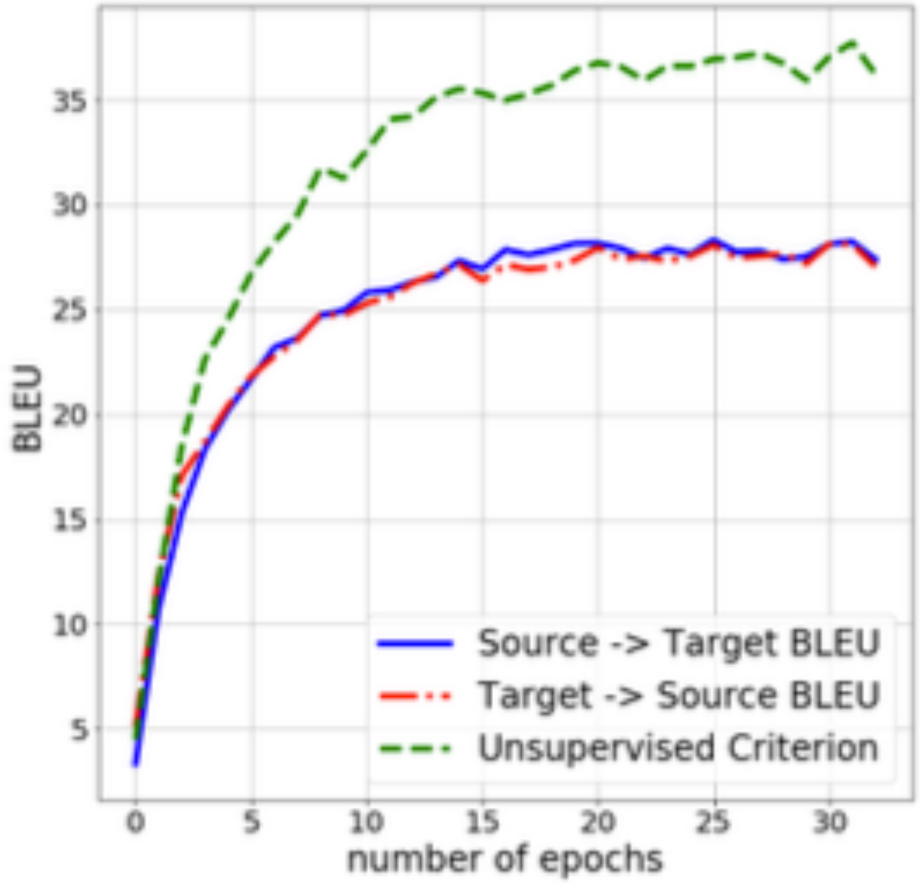
Unsupervised Model Selection Criterion

- When do we stop training without a validation set? can we do better than fixed amount of updates?
- Measure “corruption” when translating a sentence back and forth using the model (in both directions), using BLEU

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

Unsupervised Model Selection Criterion

- When do we stop training without a validation set? can we do better than fixed amount of updates?
- Measure “corruption” when translating a sentence back and forth using the model (in both directions), using BLEU
- Correlates well with “supervised” BLEU, no need for parallel sentences



$$MS(e, d, \mathcal{D}_{src}, \mathcal{D}_{tgt}) =$$

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

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

Results

	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

Results

	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

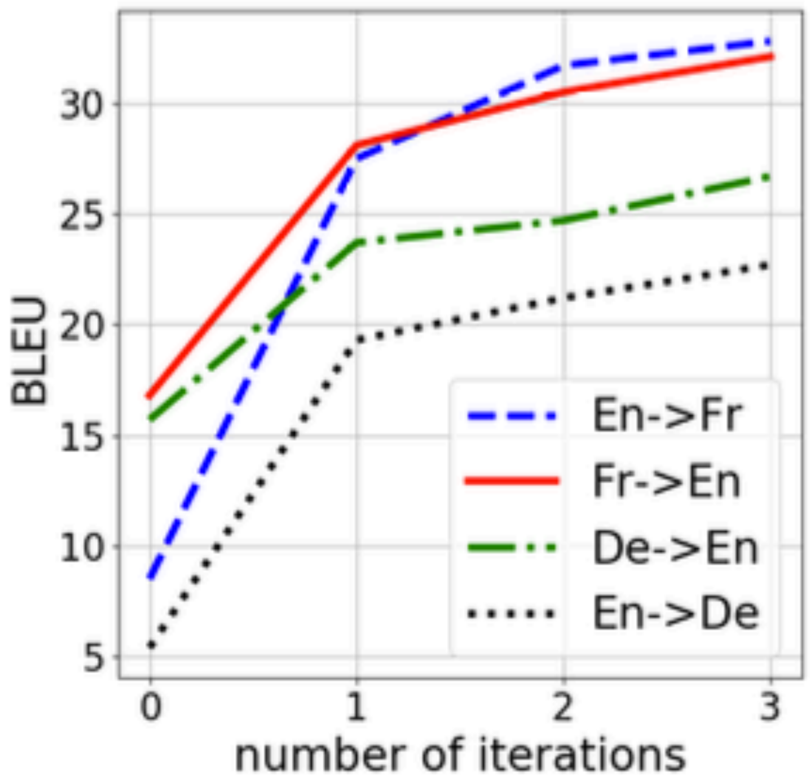
- Model significantly outperforms word-by-word baselines, showing the importance of the back-translation + denoising + adversarial approach

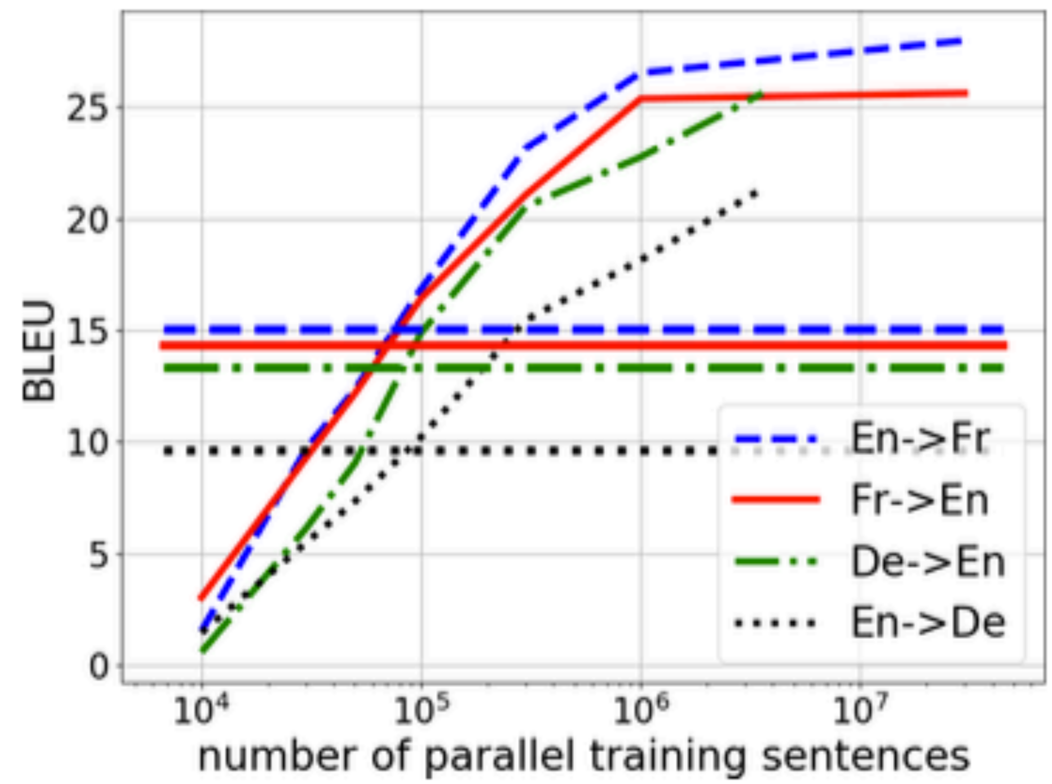
Results

	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

- Model significantly outperforms word-by-word baselines, showing the importance of the back-translation + denoising + adversarial approach
- Supervised models are still significantly better

Results





- Unsupervised models performance is equivalent to a supervised model with ~100k parallel sentences

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**

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**

- Back-translation, pre-trained word vectors and de-noising are crucial

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**

- Back-translation, pre-trained word vectors and de-noising are crucial
- Adversarial loss gives a nice boost of ~3-6 points

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**

- Back-translation, pre-trained word vectors and de-noising are crucial
- Adversarial loss gives a nice boost of ~3-6 points
- Best model obtained using all components together

Comparison

Comparison

- **Both models:**

Comparison

- **Both models:**
    - Heavily rely on bilingual word embeddings

Comparison

- **Both models:**
  - Heavily rely on bilingual word embeddings
  - Heavily rely on de-noising and back-translation using a shared encoder

Comparison

- **Both models:**
    - Heavily rely on bilingual word embeddings
    - Heavily rely on de-noising and back-translation using a shared encoder
- **Notable Differences:**

Comparison

- **Both models:**
  - Heavily rely on bilingual word embeddings
  - Heavily rely on de-noising and back-translation using a shared encoder
- **Notable Differences:**
  - Perform back-translation once per epoch (Lample et al.) vs. after every update (Artetxe et al.)

Comparison

- **Both models:**
  - Heavily rely on bilingual word embeddings
  - Heavily rely on de-noising and back-translation using a shared encoder
- **Notable Differences:**
  - Perform back-translation once per epoch (Lample et al.) vs. after every update (Artetxe et al.)
  - BPE and word-based modeling (Artetxe et al.) vs. word based alone (Lample et al.)

Comparison

- **Both models:**
  - Heavily rely on bilingual word embeddings
  - Heavily rely on de-noising and back-translation using a shared encoder
- **Notable Differences:**
  - Perform back-translation once per epoch (Lample et al.) vs. after every update (Artetxe et al.)
  - BPE and word-based modeling (Artetxe et al.) vs. word based alone (Lample et al.)
  - Fixed embeddings (Artetxe et al.) vs. changing (Lample et al.)

Comparison

- **Both models:**
  - Heavily rely on bilingual word embeddings
  - Heavily rely on de-noising and back-translation using a shared encoder
- **Notable Differences:**
  - Perform back-translation once per epoch (Lample et al.) vs. after every update (Artetxe et al.)
  - BPE and word-based modeling (Artetxe et al.) vs. word based alone (Lample et al.)
  - Fixed embeddings (Artetxe et al.) vs. changing (Lample et al.)
  - Different decoder per language (Artetxe et al.) vs. shared encoder (Lample et al.)

Comparison

- **Both models:**
  - Heavily rely on bilingual word embeddings
  - Heavily rely on de-noising and back-translation using a shared encoder
- **Notable Differences:**
  - Perform back-translation once per epoch (Lample et al.) vs. after every update (Artetxe et al.)
  - BPE and word-based modeling (Artetxe et al.) vs. word based alone (Lample et al.)
  - Fixed embeddings (Artetxe et al.) vs. changing (Lample et al.)
  - Different decoder per language (Artetxe et al.) vs. shared encoder (Lample et al.)
  - Adversarial training (Lample et al.)

Comparison

- **Both models:**
  - Heavily rely on bilingual word embeddings
  - Heavily rely on de-noising and back-translation using a shared encoder
- **Notable Differences:**
  - Perform back-translation once per epoch (Lample et al.) vs. after every update (Artetxe et al.)
  - BPE and word-based modeling (Artetxe et al.) vs. word based alone (Lample et al.)
  - Fixed embeddings (Artetxe et al.) vs. changing (Lample et al.)
  - Different decoder per language (Artetxe et al.) vs. shared encoder (Lample et al.)
  - Adversarial training (Lample et al.)
  - Slightly different noise method (Lample et al.) - swapping and dropping words, also adding noise before back-translation

Comparison

- **Both models:**
  - Heavily rely on bilingual word embeddings
  - Heavily rely on de-noising and back-translation using a shared encoder
- **Notable Differences:**
  - Perform back-translation once per epoch (Lample et al.) vs. after every update (Artetxe et al.)
  - BPE and word-based modeling (Artetxe et al.) vs. word based alone (Lample et al.)
  - Fixed embeddings (Artetxe et al.) vs. changing (Lample et al.)
  - Different decoder per language (Artetxe et al.) vs. shared encoder (Lample et al.)
  - Adversarial training (Lample et al.)
  - Slightly different noise method (Lample et al.) - swapping and dropping words, also adding noise before back-translation
  - Use unsupervised model selection criterion (Lample et al.) vs. fixed amount of updates

Comparison

- **Both models:**
    - Heavily rely on bilingual word embeddings
    - Heavily rely on de-noising and back-translation using a shared encoder
- **Notable Differences:**
    - Perform back-translation once per epoch (Lample et al.) vs. after every update (Artetxe et al.)
    - BPE and word-based modeling (Artetxe et al.) vs. word based alone (Lample et al.)
    - Fixed embeddings (Artetxe et al.) vs. changing (Lample et al.)
    - Different decoder per language (Artetxe et al.) vs. shared encoder (Lample et al.)
    - Adversarial training (Lample et al.)
    - Slightly different noise method (Lample et al.) - swapping and dropping words, also adding noise before back-translation
    - Use unsupervised model selection criterion (Lample et al.) vs. fixed amount of updates
    - Initialize back-translation using nearest-neighbor word-by-word translation (Lample et al.)

Conclusions

Conclusions

- **Still a long way to go!**

Conclusions

- **Still a long way to go!**
    - Results are still very weak in comparison to the supervised approach

Conclusions

- **Still a long way to go!**
    - Results are still very weak in comparison to the supervised approach
    - First two papers to tackle the task (in the neural context)

Conclusions

- **Still a long way to go!**
  - Results are still very weak in comparison to the supervised approach
  - First two papers to tackle the task (in the neural context)
- **New avenue for future research**

Conclusions

- **Still a long way to go!**
    - Results are still very weak in comparison to the supervised approach
    - First two papers to tackle the task (in the neural context)
- **New avenue for future research**
    - Character level modeling

Conclusions

- **Still a long way to go!**
  - Results are still very weak in comparison to the supervised approach
  - First two papers to tackle the task (in the neural context)
- **New avenue for future research**
  - Character level modeling
  - Better semi-supervised learning

Conclusions

- **Still a long way to go!**
  - Results are still very weak in comparison to the supervised approach
  - First two papers to tackle the task (in the neural context)
- **New avenue for future research**
  - Character level modeling
  - Better semi-supervised learning
  - Multilingual setting - more than 2 languages

Conclusions

- **Still a long way to go!**
  - Results are still very weak in comparison to the supervised approach
  - First two papers to tackle the task (in the neural context)
- **New avenue for future research**
  - Character level modeling
  - Better semi-supervised learning
  - Multilingual setting - more than 2 languages
  - Other sequence to sequence tasks with scarce parallel data

Thanks!

References

- Mikolov, Le & Sutskever, 2013 - “Exploiting Similarities among Languages for Machine Translation”
- Sustekever, Vinyals & Le, 2014 - “Sequence to Sequence Learning with Neural Networks”
- Bahdanau, Cho & Bengio, 2014 - Neural Machine Translation by Jointly Learning to Align and Translate
- Conneau, Lample, Ranzato, Denoyer & Jegou, 2017 - "Word Translation Without Parallel Data"
- Artetxe, Labaka, Agirre & Cho, 2017 - "Unsupervised Neural Machine Translation”
- Lample, Denoyer & Ranzato, 2017 - “Unsupervised Machine Translation Using Monolingual Corpora Only”
- Koehn & Knowles, 2017 - “Six Challenges for Neural Machine Translation”
- Artetxe, Labaka & Agirre, 2017 - “Learning bilingual word embeddings with (almost) no bilingual data”