

# NLP Applications (I)

## Machine Translation

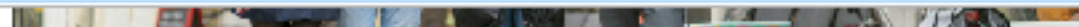
**Arianna Bisazza**

University of Groningen, The Netherlands

AthNLP2019



23 Sept 2019



In Groningen everything is within cycling distance and there are always acquaintances in the neighborhood. Students live criss-crossed the city, the buildings of the University of Groningen can be found throughout the center. In the winter the many bars and cafes are full of fellow students, in the summer you can find them in Noorderplantsoen, in the Stadspark or swimming in the water of the Stadsstrand. Our students live, live and study throughout the city:

[#TheCityIsOurCampus](#) .

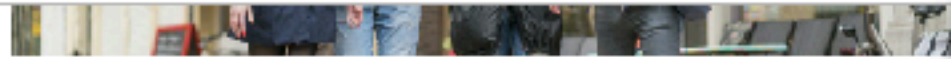
## Culture, shopping and going out



Prepare for a busy social life, because there is always something to do in Groningen. From large festivals such as Eurosonic Noorderslag and Noorderzon to performances in the local neighborhood pub, lectures, concerts, films or dance performances: it is actually impossible to get bored in Groningen.

Shop fans can also get lucky: big chains, promising new fashion brands, small boutiques, artisan shops, we have it all. It is not for nothing that our streets have been voted Best Shopping Street in the Netherlands several times.

Going out in Groningen is fun for everyone because of the diverse range of cafés, bars, karaoke bars and discos. And: our pubs have no closing times. People party, laugh and dance until the early hours. Not only in the pub, by the way. Also at the



In Groningen everything is within cycling distance. You can find acquaintances in the neighborhood. Student buildings of the University of Groningen can be found in the winter the many bars and cafes are full of people. You can find them in Noorderplantsoen, in the Stadsstrand. Our students live, live and study throughout the city: [#TheCityIsOurCampus](#).

Really?  
Wow! 🤔

Translated

Show Original Options

- ✓ Always Translate Dutch
- Never Translate Dutch
- Never Translate This Site
- Change Languages

### Culture, shopping and going out



Prepare for a busy social life, because there is always something to do in Groningen. From large festivals such as Eurosonic Noorderslag and Noorderzon to performances in the local neighborhood pub, lectures, concerts, films or dance performances: it is actually impossible to get bored in Groningen.

Shop fans can also get lucky: big chains, promising new fashion brands, small boutiques, artisan shops, we have it all. It is not for nothing that our streets have been voted Best Shopping Street in the Netherlands several times.

Going out in Groningen is fun for everyone because of the diverse range of cafés, bars, karaoke bars and discos. And: our pubs have no closing times. People party, laugh and dance until the early hours. Not only in the pub, by the way. Also at the

# Introduction

---

- 2019: Machine translation is a pervasive and reliable\* technology
  - free high-quality online systems
  - important part of professional translation workflow



\*well, mostly reliable

- Core problem of NLP, long history and variety of approaches:
  - rule-based MT: manually created lexicons, parsing and translation rules
  - statistical MT (SMT): based on information theory
  - neural MT: a conditional language model based on deep neural networks

# A brief history

---

1947: Weaver's memorandum

- frames MT as code deciphering problem (link to advances in cryptography)
- highlights limitations of early word-by-word translation approaches

1990's: **first SMT** systems (IBM models)

- noisy-channel formulation
- tightly connected to advances in speech recognition

1997: **early NMT** approaches proposed by two Spanish groups ("connectionist MT")

- cannot scale => abandoned

2000's: multiple advances in SMT

- **phrase-based SMT** (open-source platform Moses since 2006)
- **syntax-based SMT**: exploits parsers to bridge across languages
- **tree-based SMT**: synchronous grammars learnt from parallel data

2014-15: **seq2seq NMT** (RNN-based + attention)

2015-16: NMT beats phrase-based SMT, adopted by large online MT providers

2017: **Transformer** (fully attention based network) quickly becomes state-of-the-art

# Today's Lecture

---

- Before NMT: Phrase-based SMT
- NMT architectures
  - RNN-based seq2seq
  - RNN-based seq2seq + Attention
  - Transformer
- NMT decoding & Word segmentation
- Evaluation
- Human parity? and open issues
- Useful links

## Why Phrase-Based SMT??

- history is fun
- Greek like history
- it wasn't so long ago after all!
- MT history can give us a good grasp of how the NLP field, in general, evolved

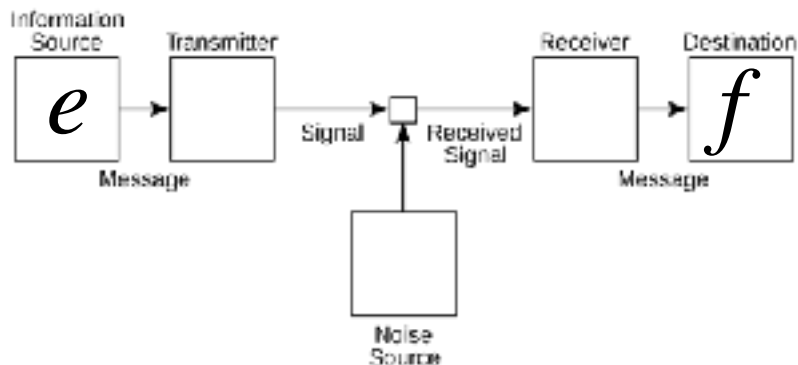


Before NMT:

# PHRASE-BASED SMT

# Fundamentals of Statistical MT

- Early SMT approaches adopted the *noisy channel* model:



$e$  : English (the *original* message)  
 $f$  : foreign (the *distorted* message)  
 $e^*$ : translation (the *recovered* message)

$$e^* = \arg \max_e p(e | f)$$

$$e^* = \arg \max_e \frac{p(f | e) p(e)}{p(f)}$$

$$e^* = \arg \max_e p(f | e) p(e)$$

**translation  
model**

**target language  
model**



# Fundamentals of Statistical MT (II)

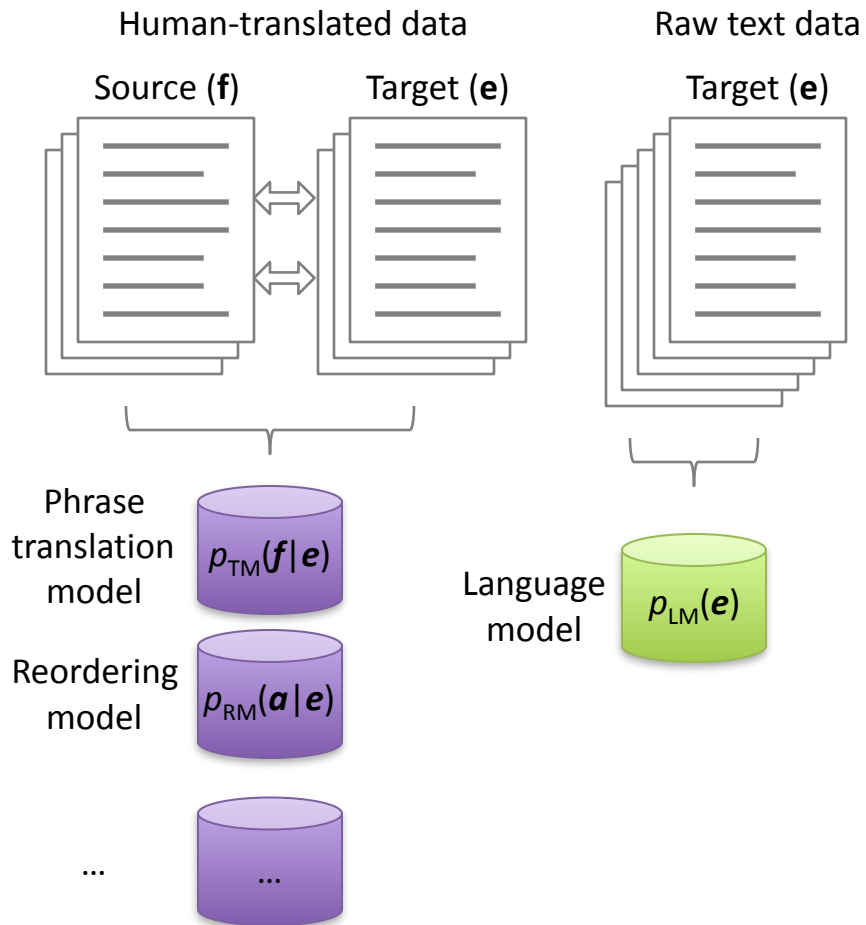
---

- Phrase-based SMT (discriminative approach):
  - linear combination of feature functions
  - introduce hidden variable  $a$ : phrase alignment

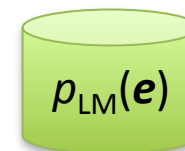
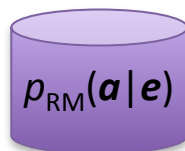
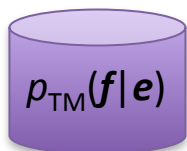
$$e^* = \arg \max_e \max_a \exp \left[ \sum_{i=1}^I \lambda_i h_i(f, e, a) \right]$$

- permits to add any kind of translation feature (“submodels”) like:
  - phrase translation probabilities  $p(\bar{f} | \bar{e})$  and  $p(\bar{e} | \bar{f})$
  - reordering probabilities  $p(\sigma(a))$
  - word translation probabilities, length penalties, ...
- feature weights can be uniform or *tuned* to maximize some measure of accuracy on devset

# SMT framework overview



# SMT framework overview



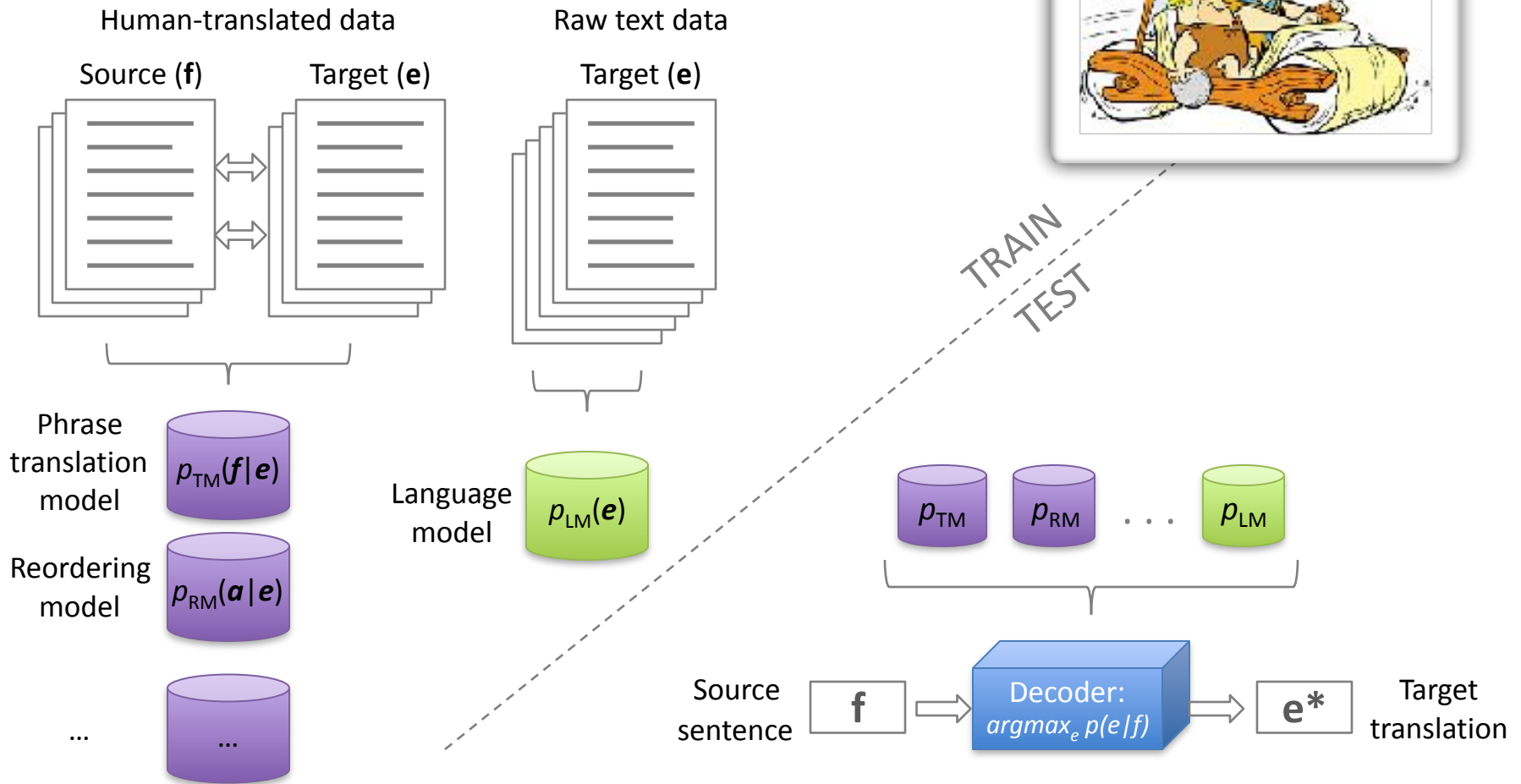
<b>freedom</b>	
liberta'	0.31
la liberta' di	0.22
mobilita'	0.08
...	...
<b>must</b>	
devono	0.50
deve	0.18
avrebbe dovuto	0.05
...	...

Translate phrase after the ??? source phrase			
	previous:	following:	other:
<i>free</i>	0.55	0.05	0.40
<i>freedom</i>	0.30	0.65	0.05
...			
<i>must</i>	0.90	0.02	0.08
<i>must be</i>	0.95	0.02	0.03
...	...		

$P(w_i/w_{i-2} w_{i-1})$	
liberta' di ...	
movimento	0.16
parola	0.09
fare	0.05
...	...
dev' essere ...	
il	0.22
la	0.18
dato	0.02
...	...

Source language (**f**): English  
Target language (**e**): Italian

# SMT framework overview



Log-linear combination:

$$e^* = \arg \max_e \max_a \exp \left[ \sum_{i=1}^I \lambda_i h_i(f, e, a) \right]$$

$$h_1 = \log P_{LM}(e)$$

$$h_2 = \log P_{RM}(a|e)$$

$$h_3 = \log P_{RM}(f|a, e) \quad 12$$

# PHRASE-BASED SMT DECODING

# Phrase-based SMT Decoding

---

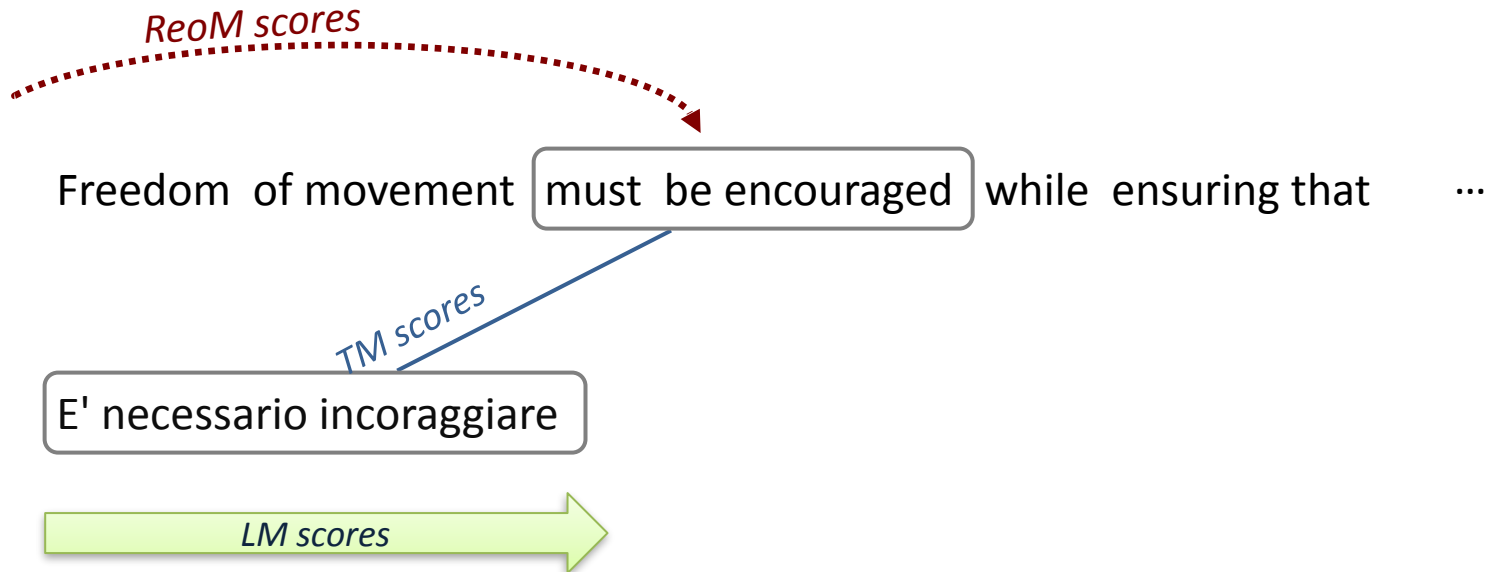
At translation time, search for the most probable translation according to the learned models



Freedom of movement must be encouraged while ensuring that ...

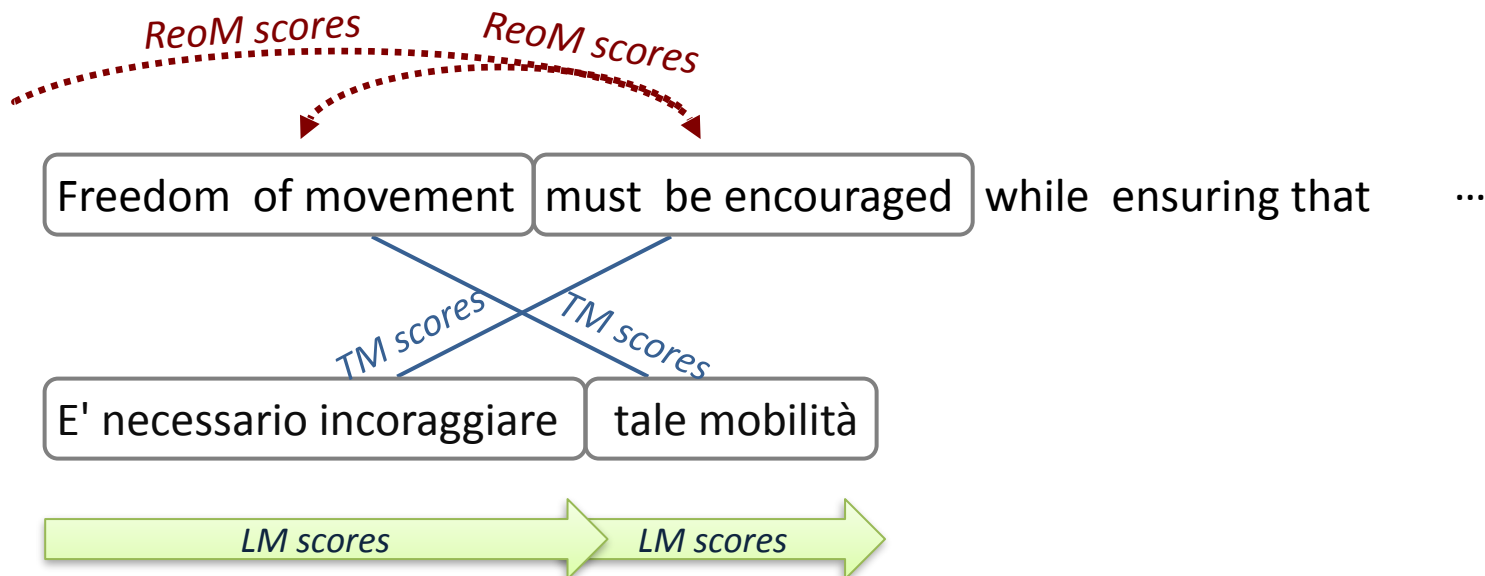
# Phrase-based SMT Decoding

At translation time, search for the most probable translation according to the learned models



# Phrase-based SMT Decoding

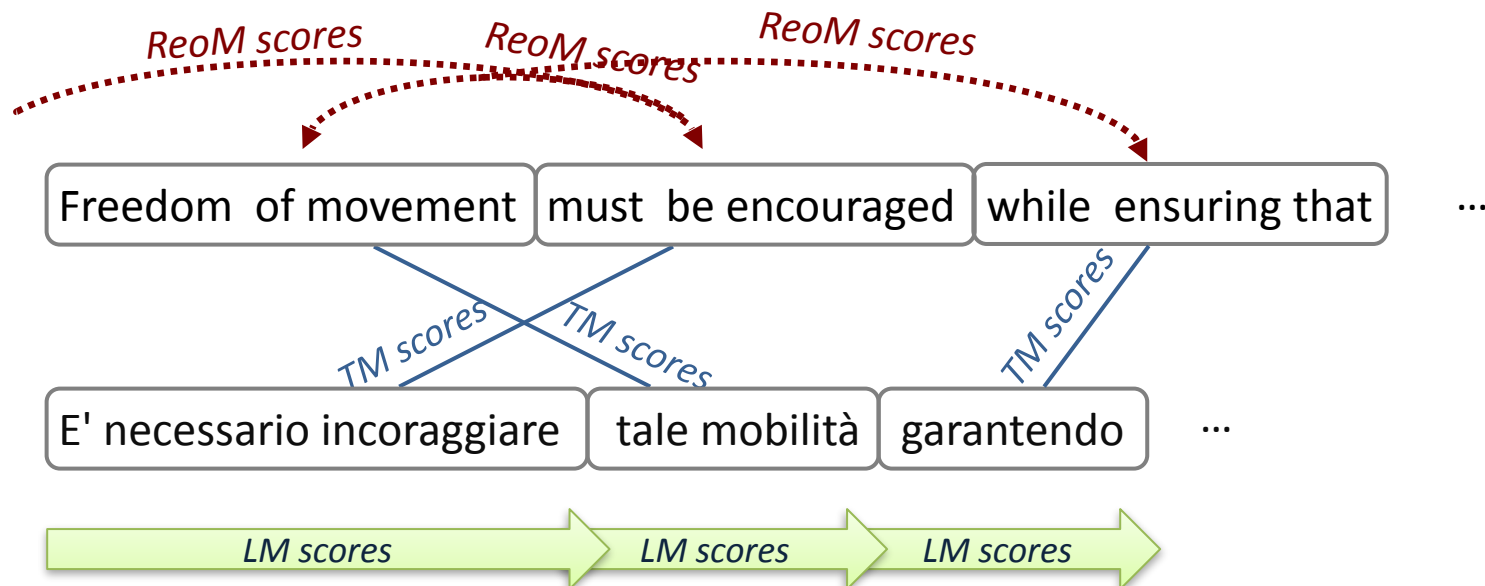
At translation time, search for the most probable translation according to the learned models





# Phrase-based SMT Decoding

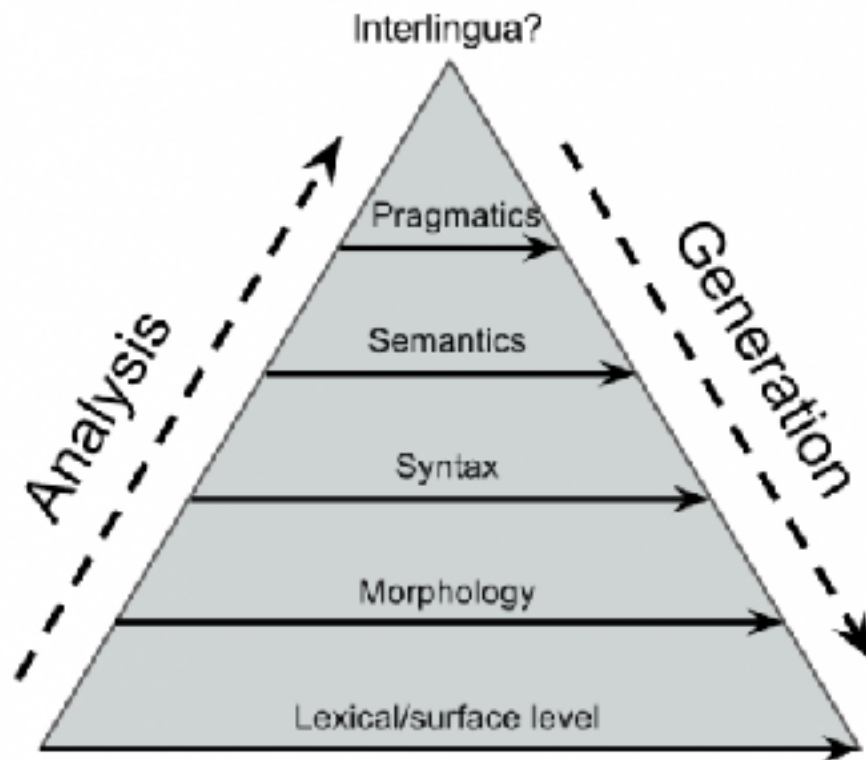
At translation time, search for the most probable translation according to the learned models



# **SMT: BEYOND PHRASE-BASED**

# The Vauquois Triangle

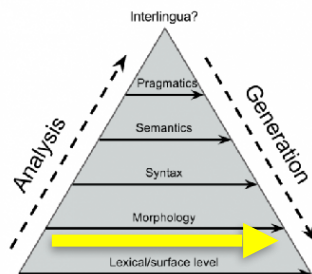
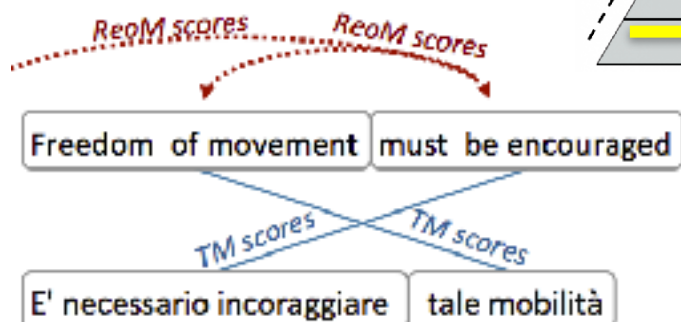
---



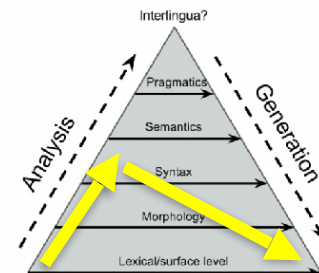
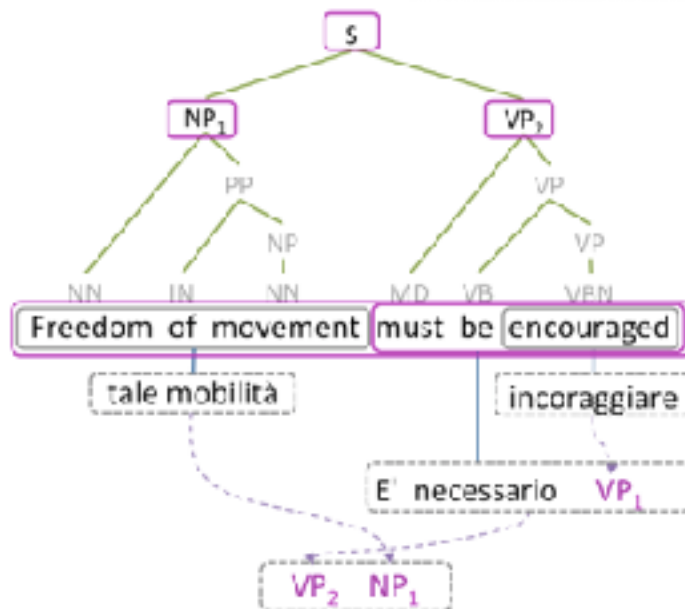
Bernard Vauquois  
1929-1985

# Phrase-based VS Tree-based SMT

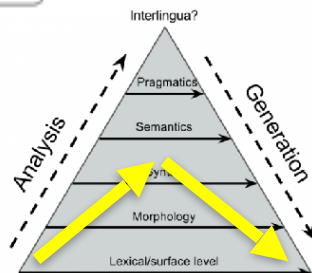
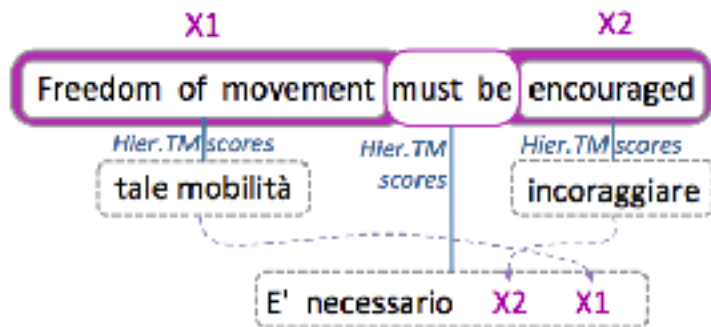
Phrase-based:



Syntax-based (tree-to-string):

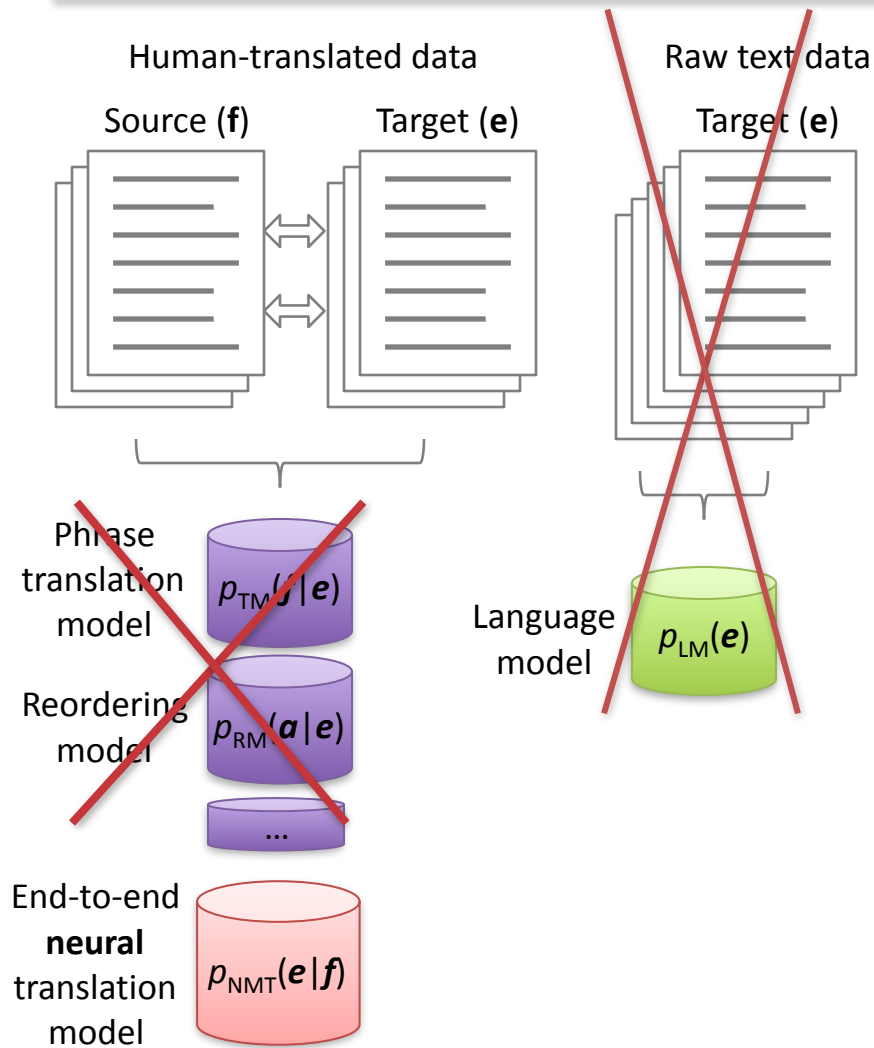


Hierarchical phrase-based:

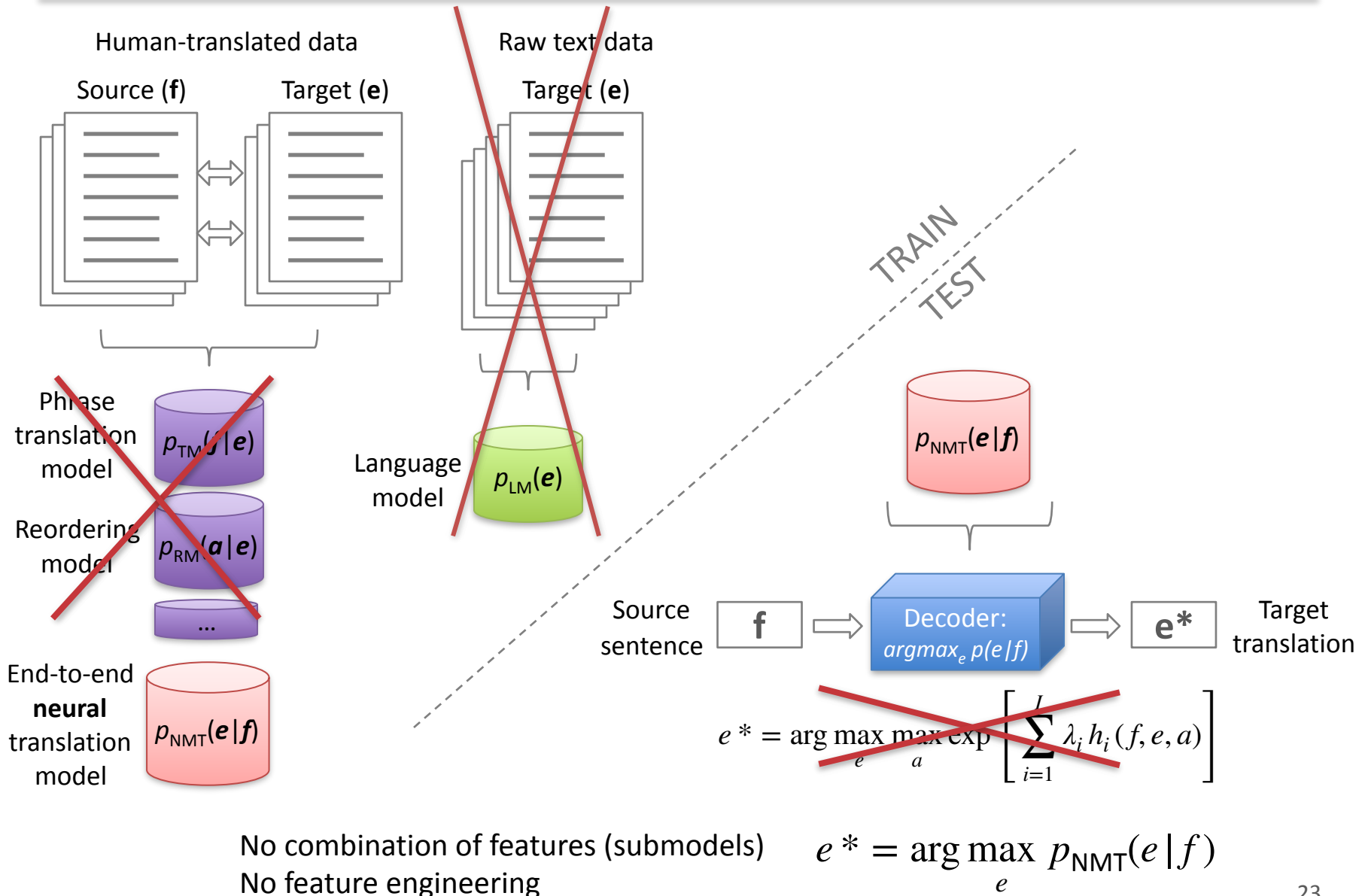


# NEURAL MACHINE TRANSLATION

# NMT framework overview

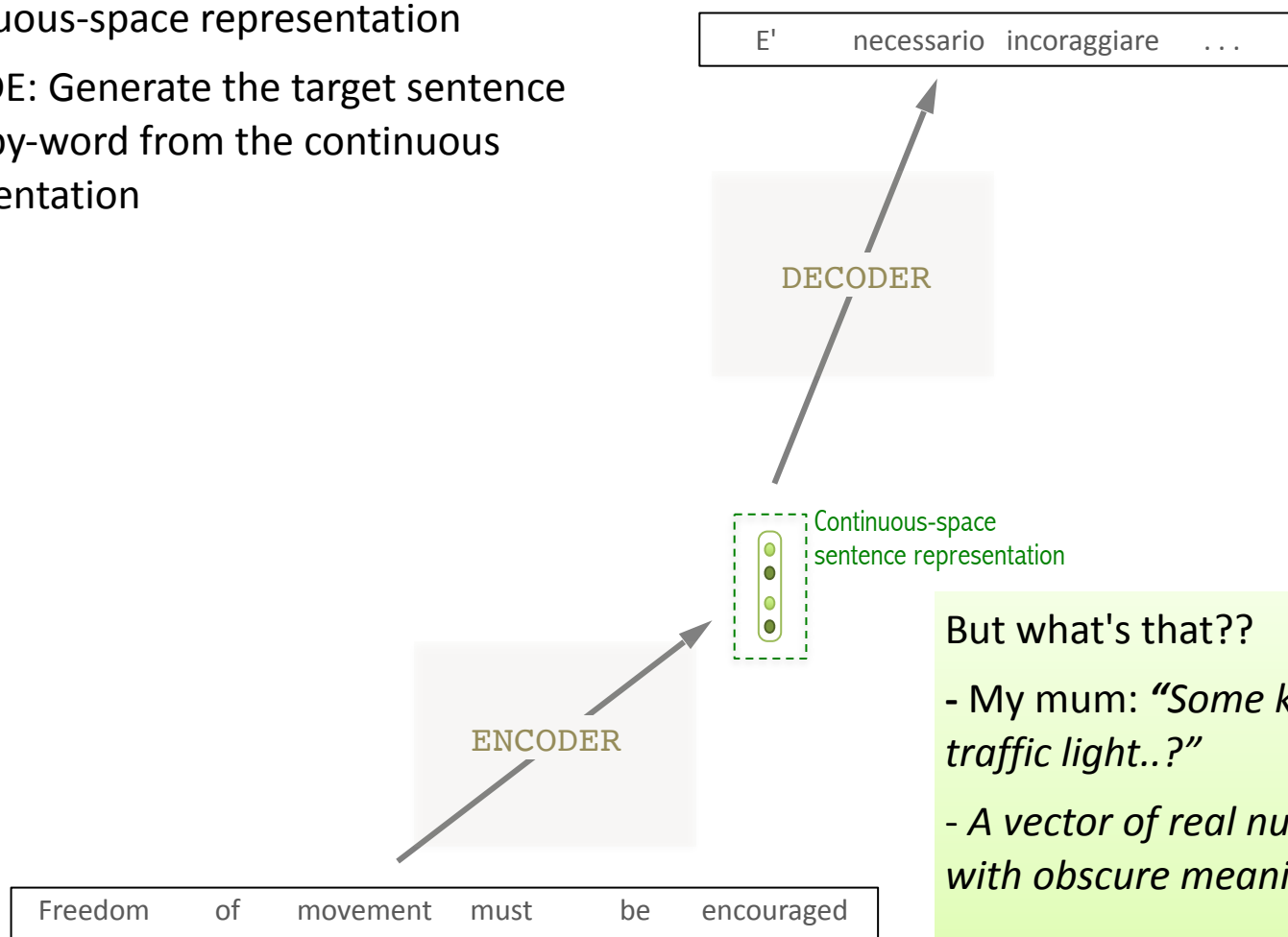



# NMT framework overview



# Sequence-to-sequence NMT

- ENCODE: Project input sentence into a continuous-space representation
- DECODE: Generate the target sentence word-by-word from the continuous representation



But what's that?? 

- My mum: *"Some kind of traffic light..?"*

- *A vector of real numbers with obscure meaning...*

**Let's take a step back!**

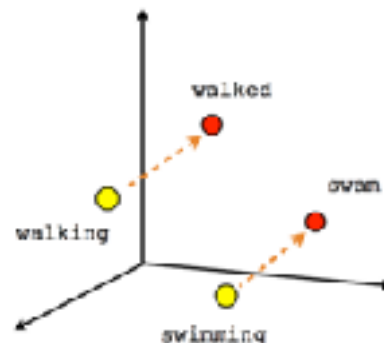
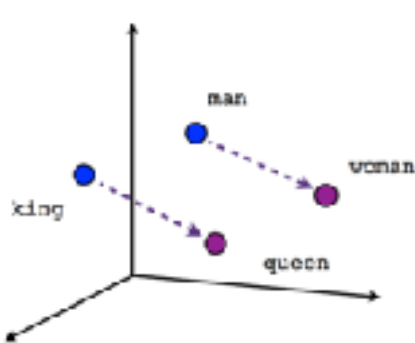


# Continuous-space representations

Get the intuition from **word** representations (a.k.a word embeddings):

- each word in a vocabulary is a vector, or a point in a  $n$ -dimensional space
- dimensions have no pre-defined meaning
- dimensions capture features of the input that are useful for the modeled task

abacus	0.01	-0.23	0.17	...	0.12
...					
man	0.21	-0.54	0.02	...	-0.21
...					
queen	0.22	0.65	0.02	...	-0.01
...					
walked	0.43	0.01	0.87	...	0.01
walking	0.39	0.01	-0.32	...	0.03
woman	0.38	0.81	0.02	...	-0.01

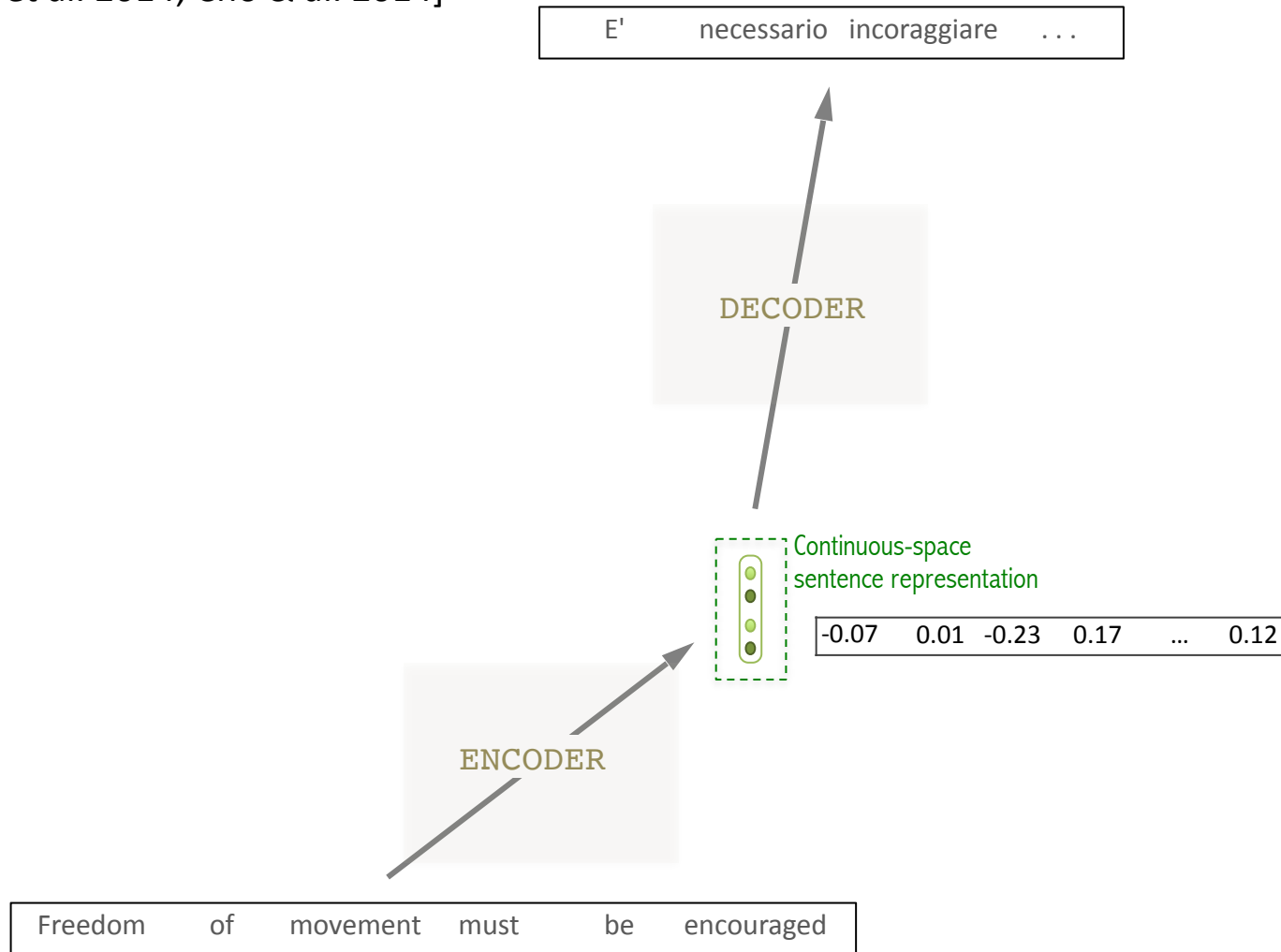


Male-Female

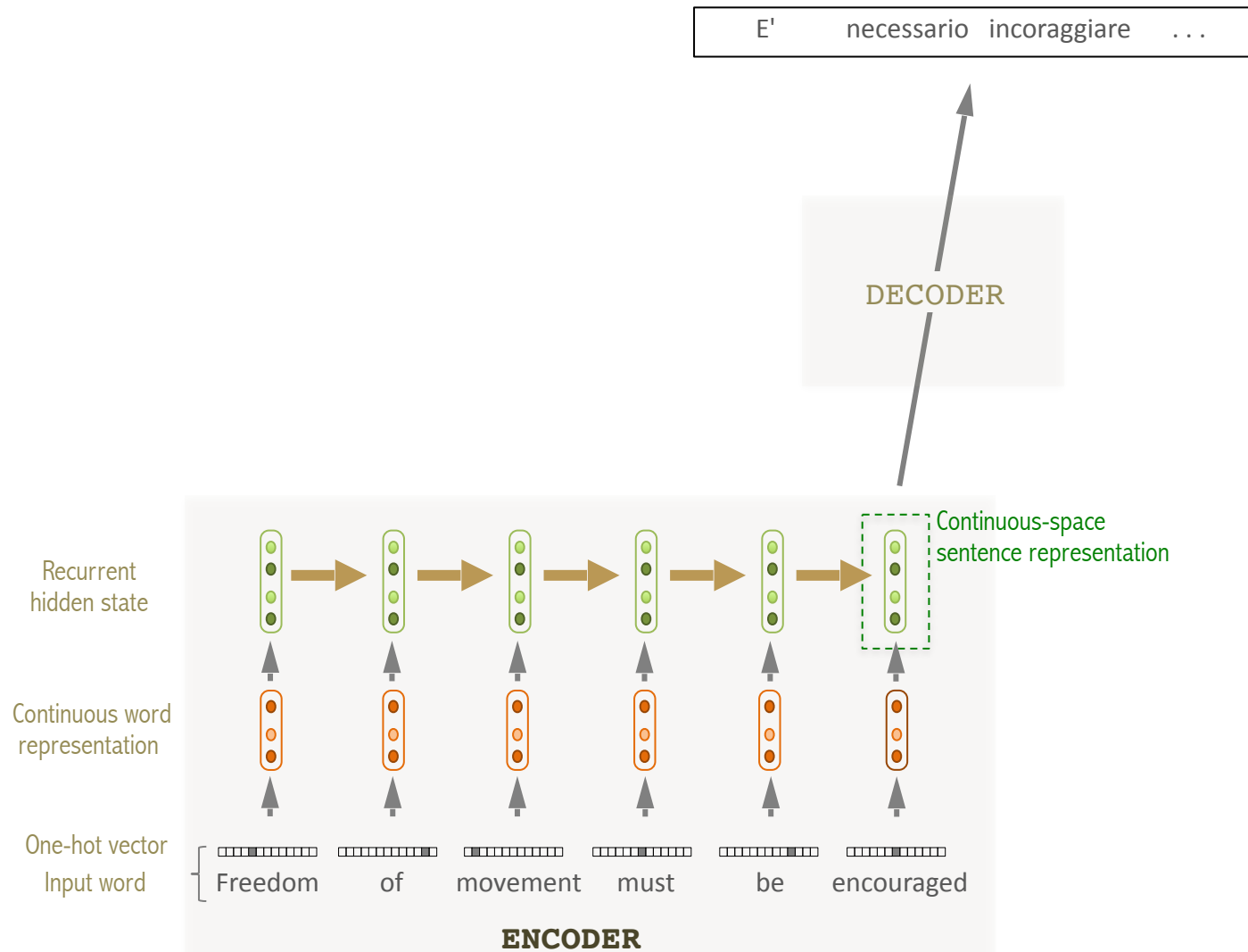
Verb tense

# RNN-based Seq2seq NMT

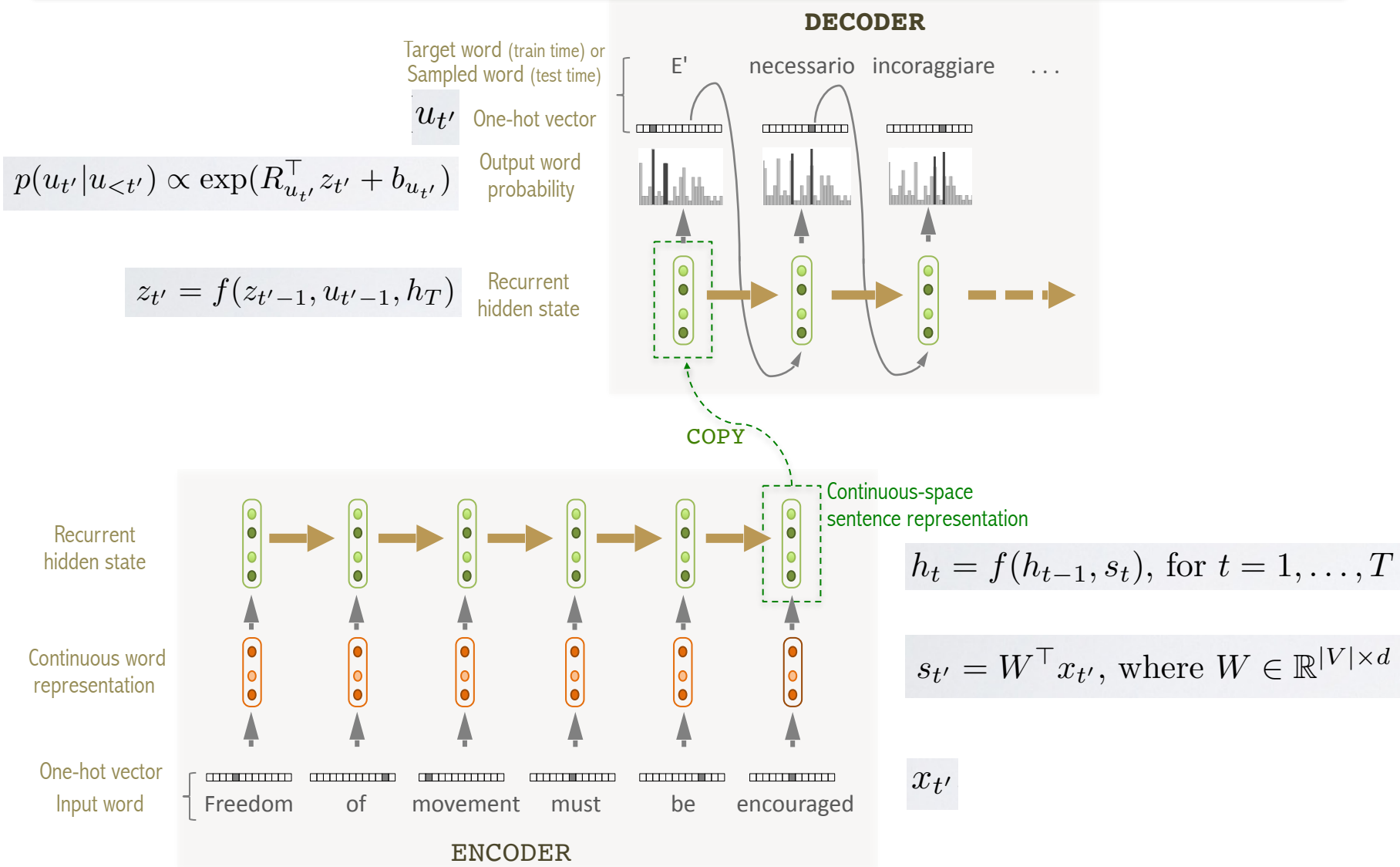
[Sutskever et al. 2014; Cho & al. 2014]



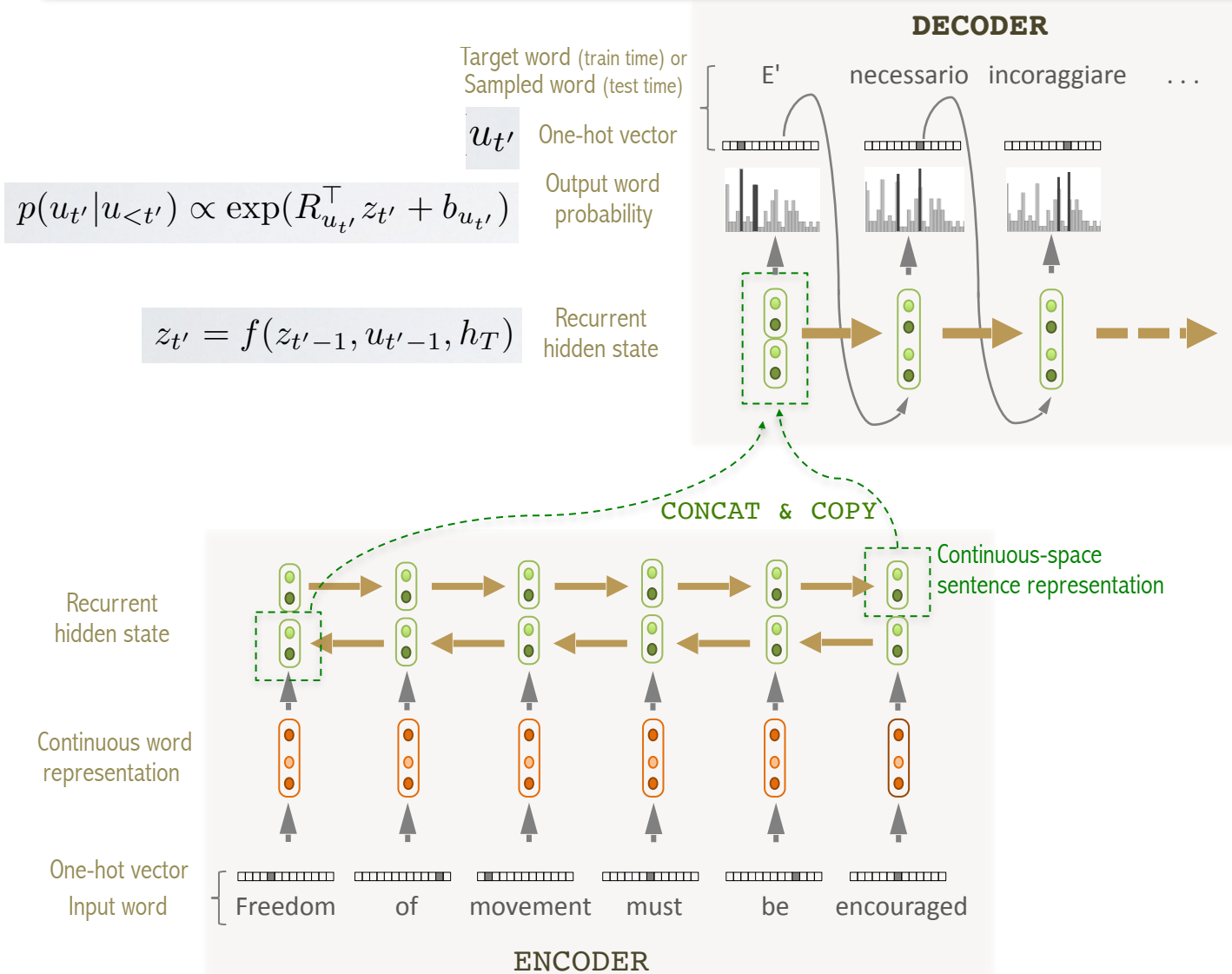
# RNN-based Seq2seq NMT



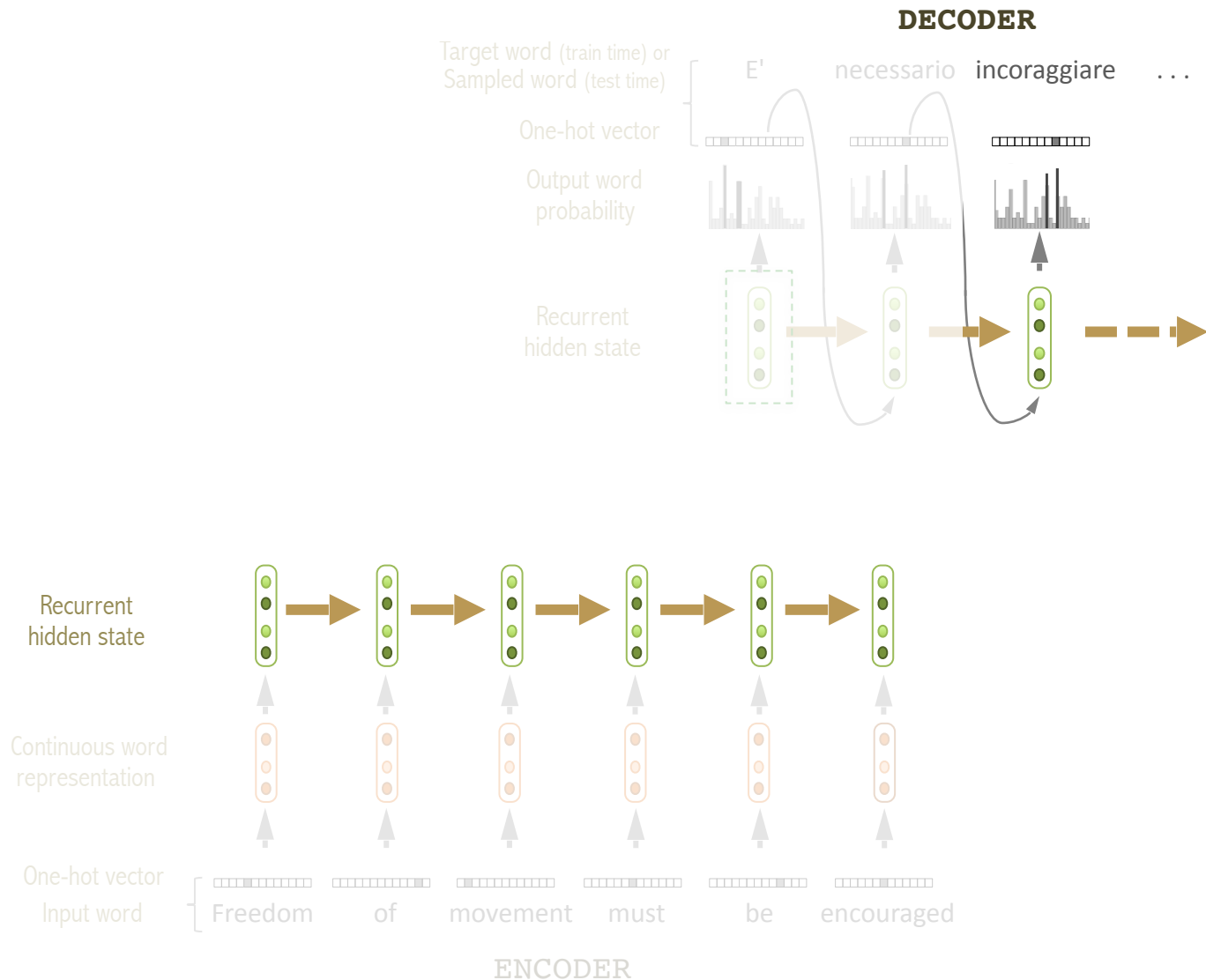
# RNN-based Seq2seq NMT



# RNN-based Seq2seq NMT + Bidirectional Encoder



# RNN-based Seq2seq NMT + Attention

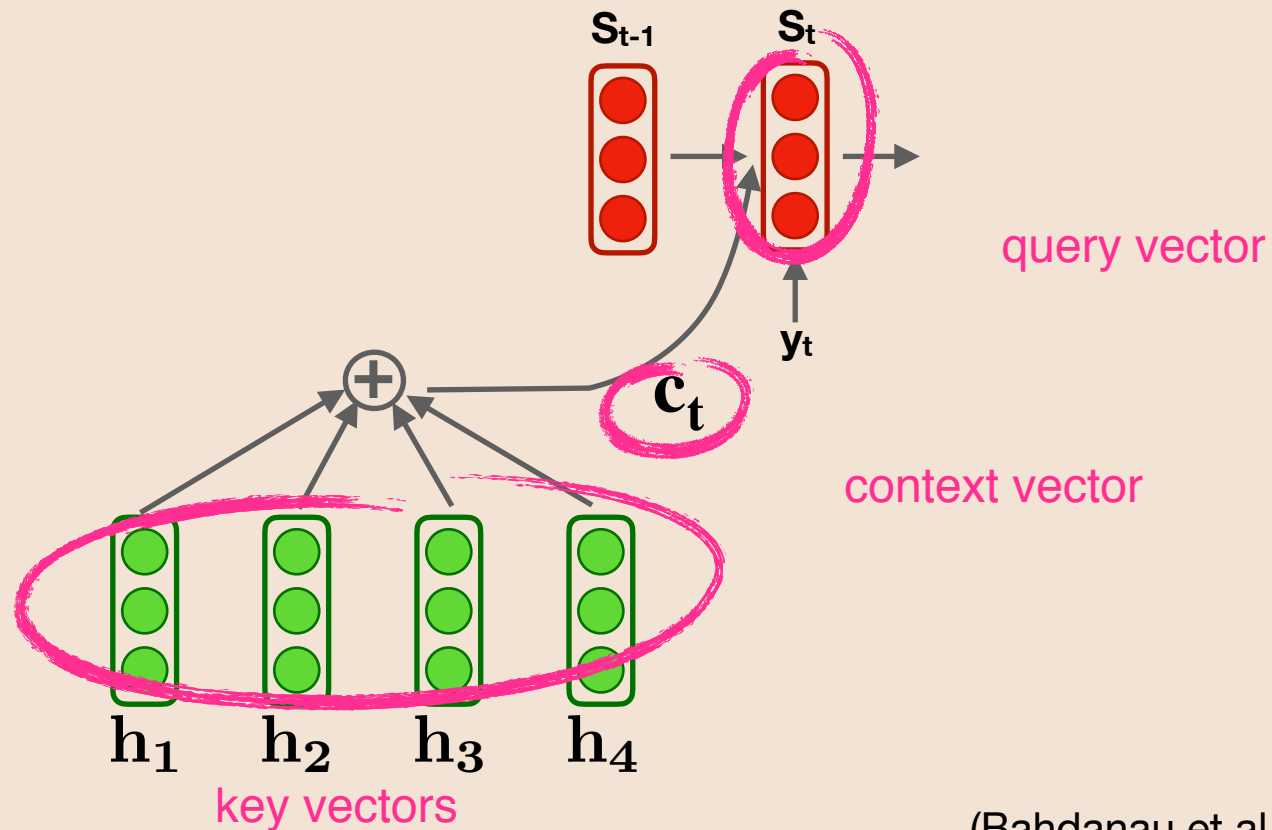




Slide by Barbara Plank

# Attention: Core Idea

- ▶ When decoding, perform a linear combination of the encoded input vectors, weighted by “attention weights”



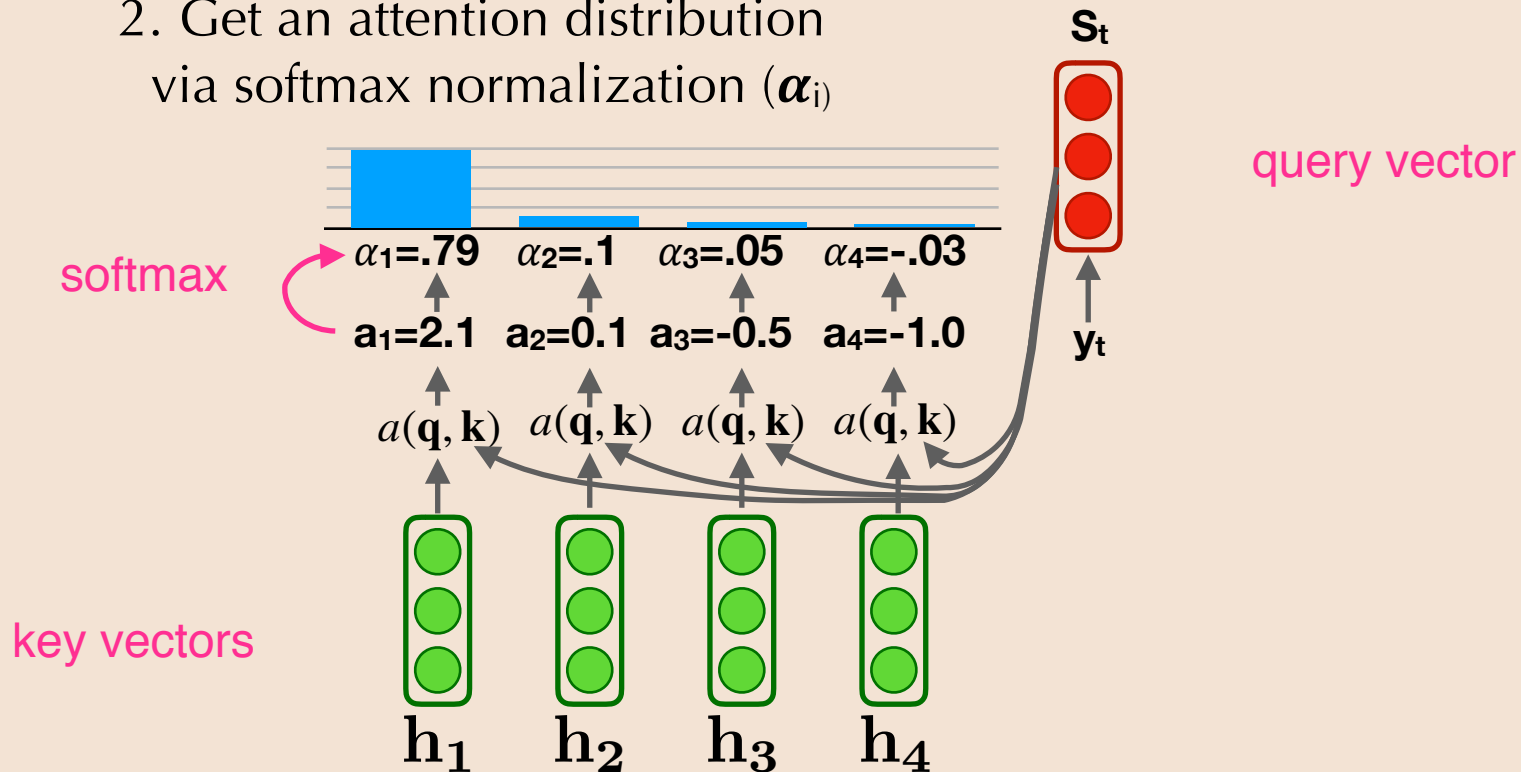
(Bahdanau et al., 2015)

# Calculating attention (1/2): Attention weights $\alpha$



Slide by Barbara Plank

1. For each query-key pair, calculate an **attention score** ( $a_i$ )
2. Get an attention distribution via softmax normalization ( $\alpha_i$ )



(Bahdanau et al., 2015)

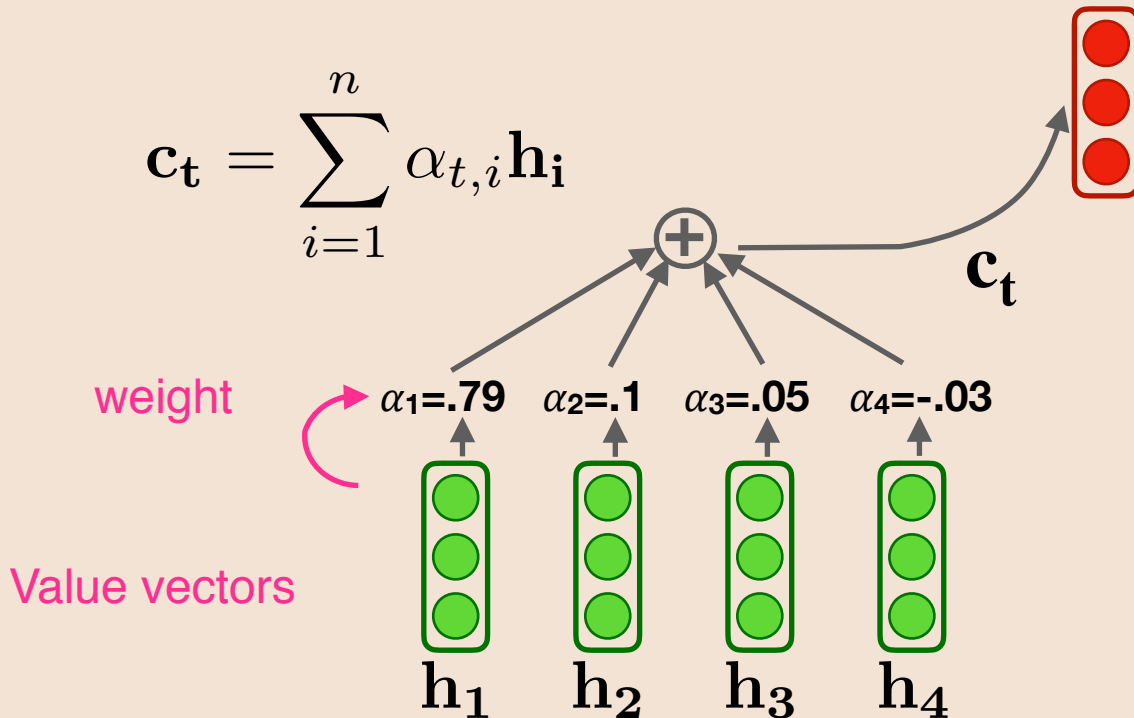


# Calculating attention (2/2): Attention weights $\alpha$



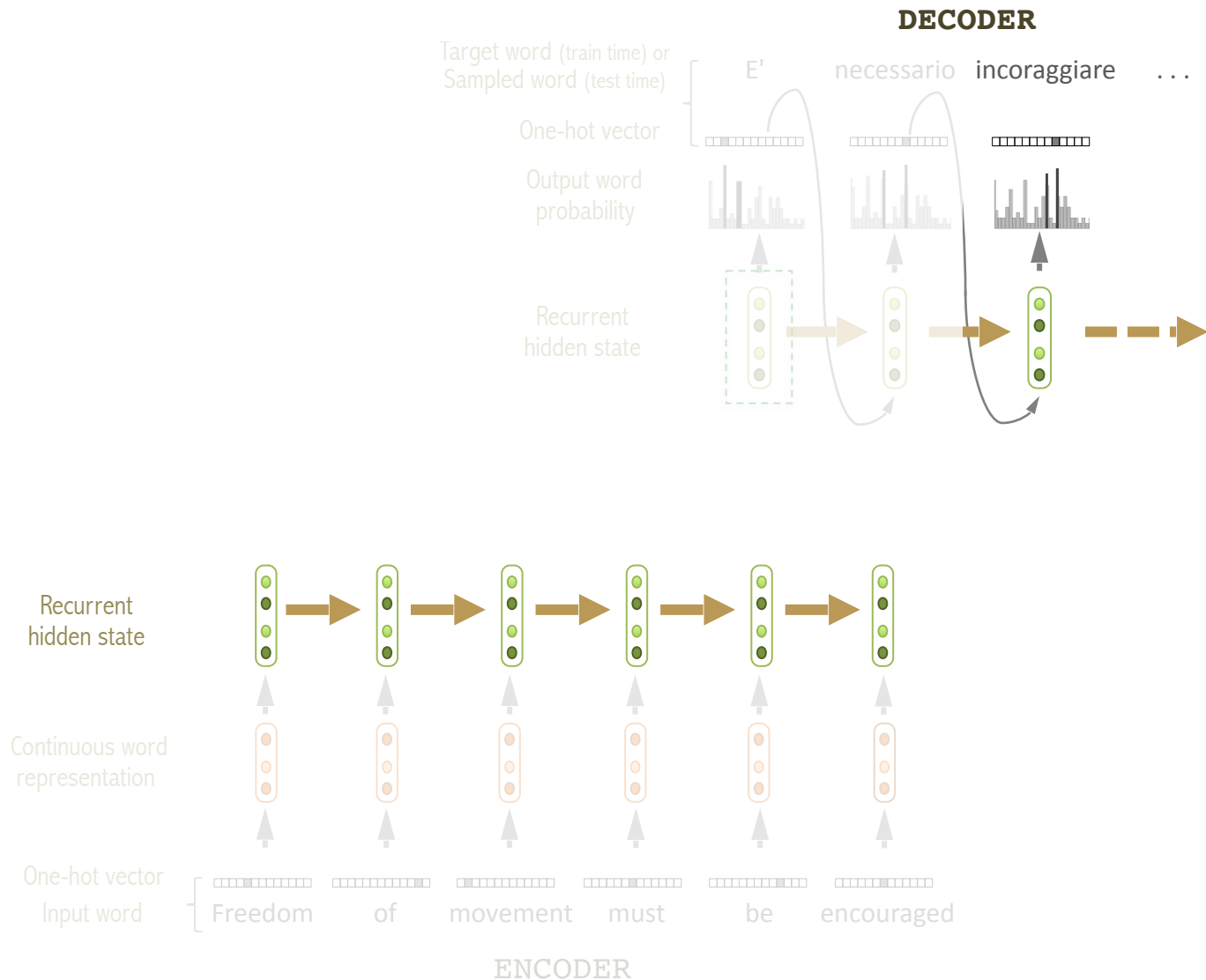
Slide by Barbara Plank

3. Combine together **value vectors** (can be the encoder states, like the key vectors) by taking the weighted sum to get  $\mathbf{c}$

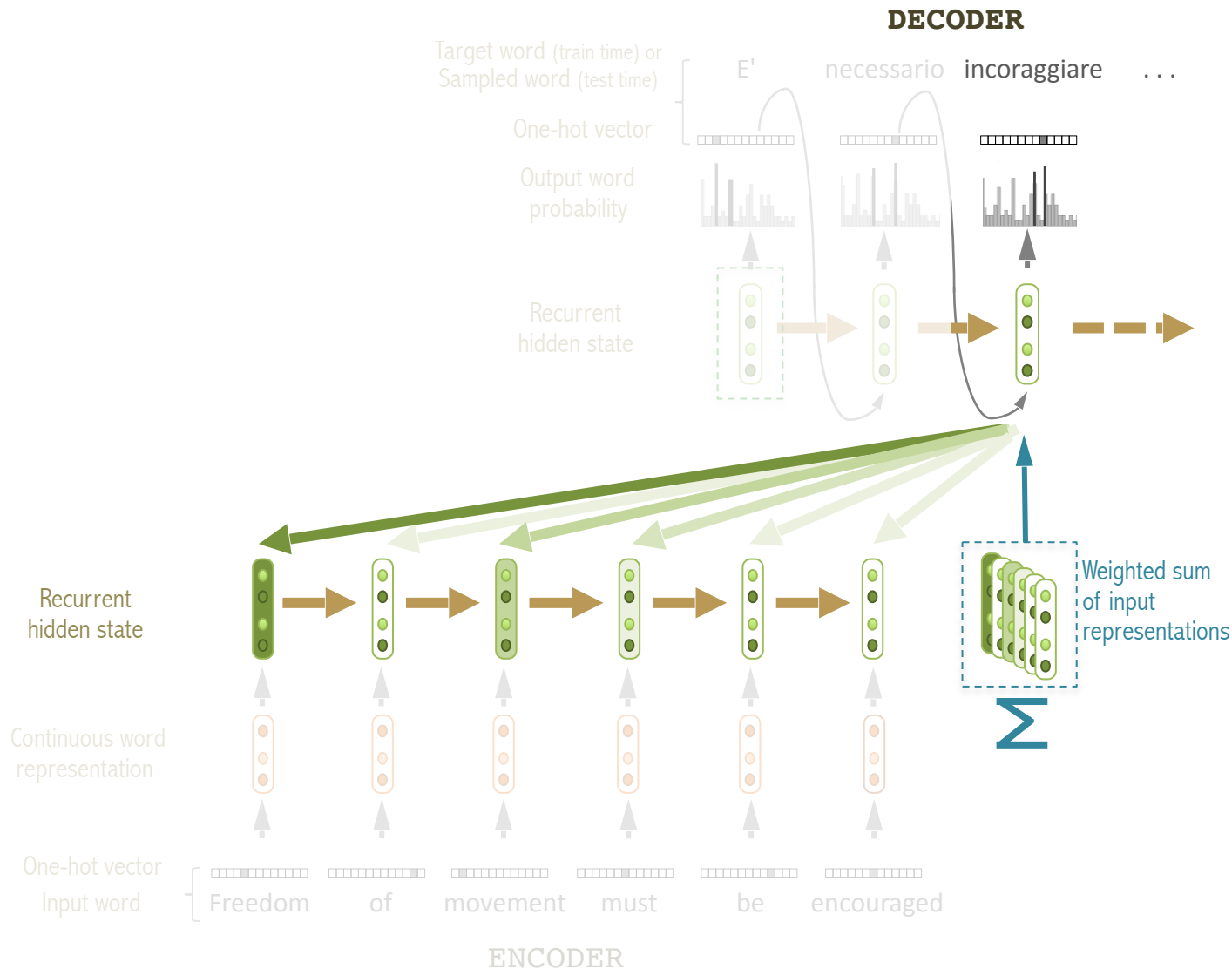


(Bahdanau et al., 2015)

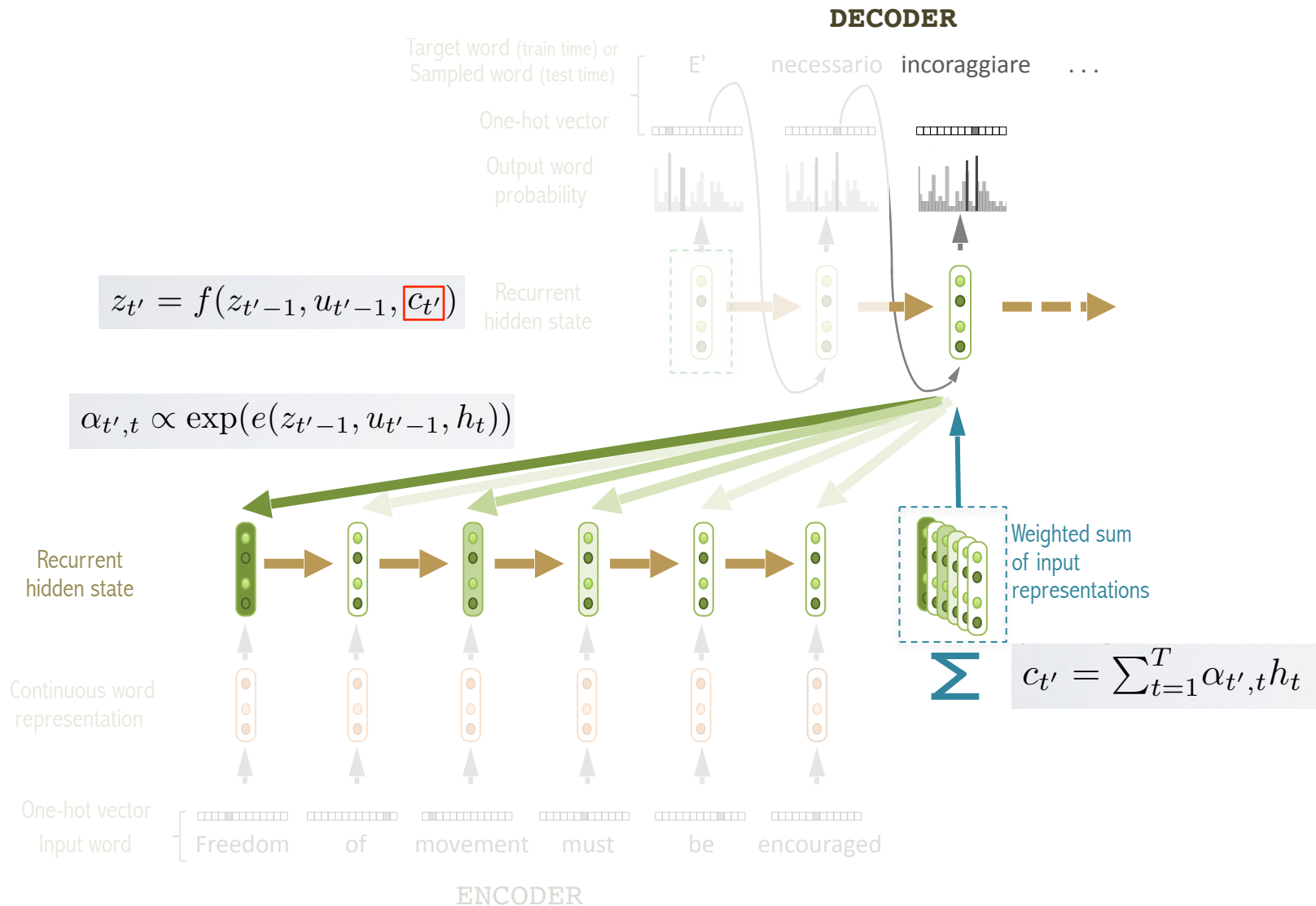
# RNN-based Seq2seq NMT + Attention



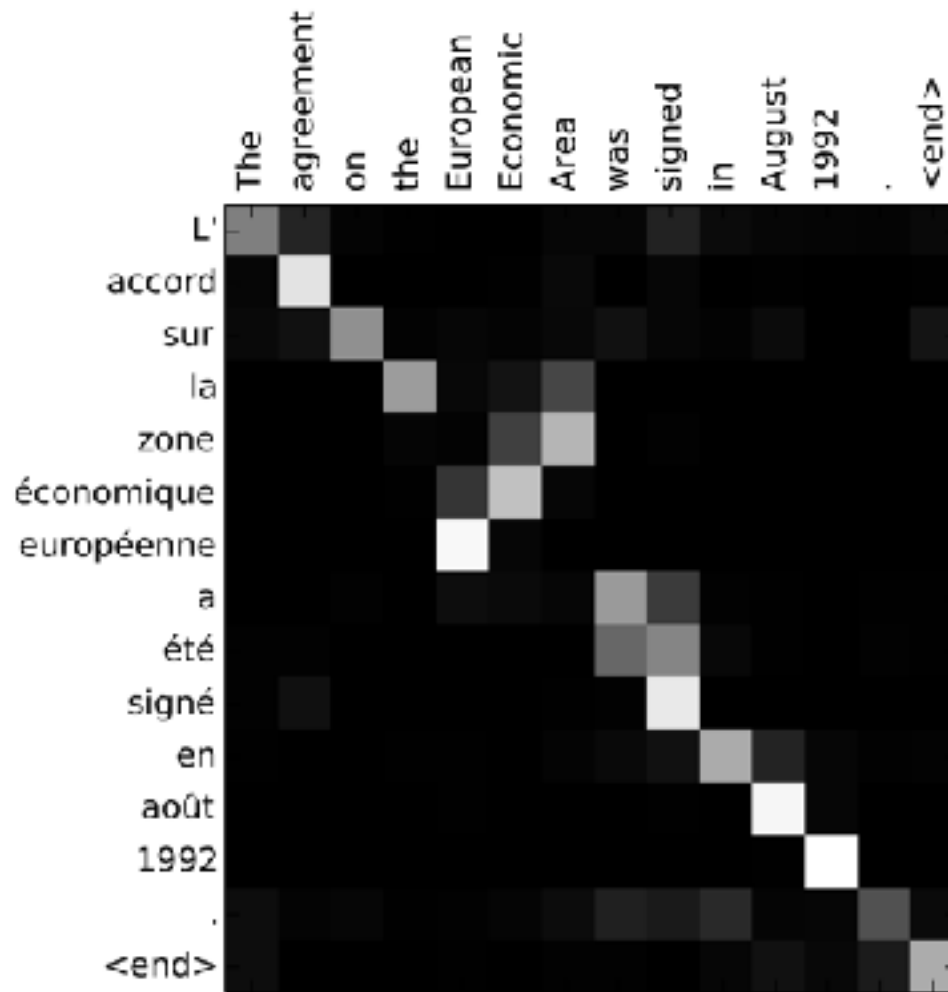
# RNN-based Seq2seq NMT + Attention



# RNN-based Seq2seq NMT + Attention

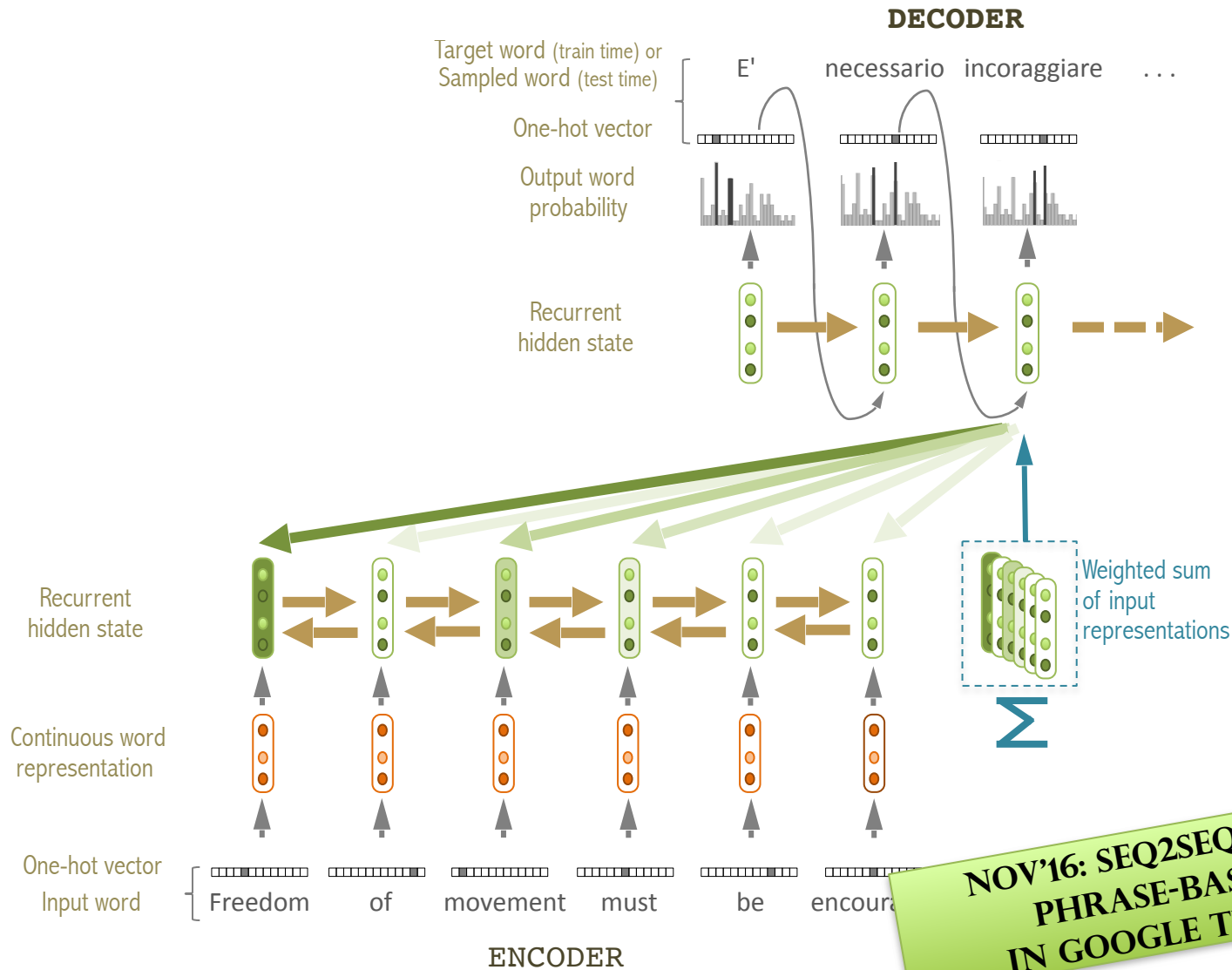


# Attention visualization



Taken from (Bahdanau et al. 2015)

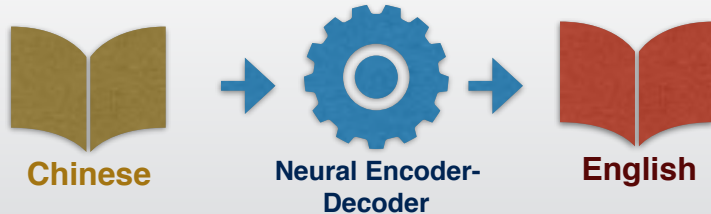
# RNN-based Seq2seq NMT + Attention



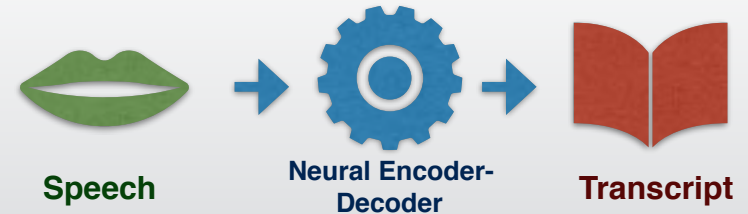
**NOV'16: SEQ2SEQ REPLACED PHRASE-BASED SMT IN GOOGLE TRANSLATE**

# One architecture, many applications!

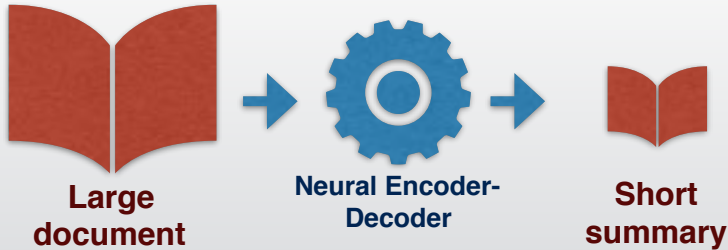
**machine translation:**



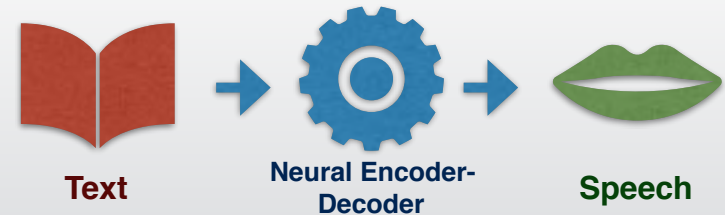
**speech recognition:**



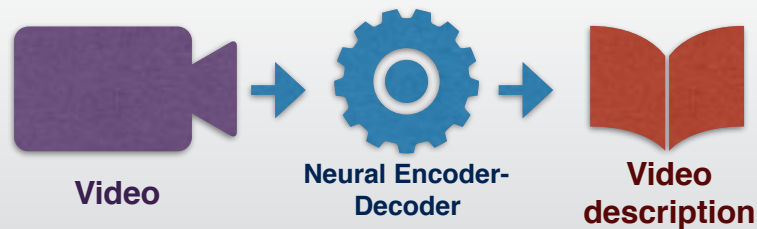
**text summarization:**



**speech synthesis:**



**video captioning:**



# NMT vs SMT

---

What has been solved (or extremely improved) by NMT\*:

✓ models capture distributional semantics of words and phrases

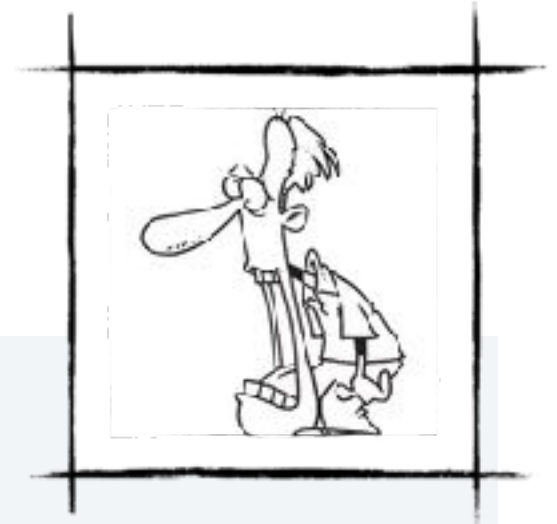
✓ overall grammaticality of output sentences

in particular: word reordering, long dependencies



# Back in 2014...

Montreal's first NMT online demo:

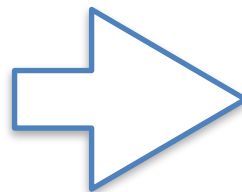


Type text here:

The Budapest Prosecutor's Office **has initiated** an investigation on the accident.

Translation:

Die Budapester Staatsanwaltschaft **hat ihre** Ermittlungen zum Vorfall **eingeleitet**.



# Today's Lecture

---

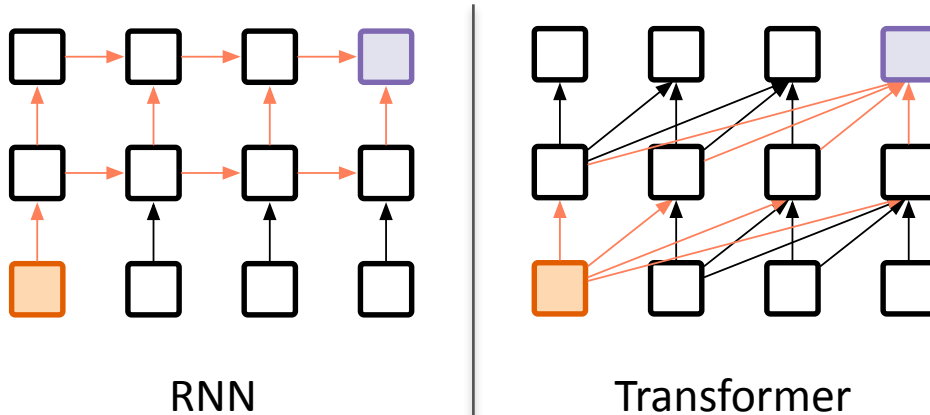
- Before NMT: Phrase-based SMT
- NMT architectures
  - RNN-based seq2seq
  - RNN-based seq2seq + Attention
  - Transformer
- NMT decoding & Word segmentation
- Evaluation
- Human parity? and open issues
- Useful links

# **FULLY ATTENTIONAL NETWORKS (A.K.A. TRANSFORMER)**

# Core idea: Attention is All You Need (Vaswani et al. 2017)

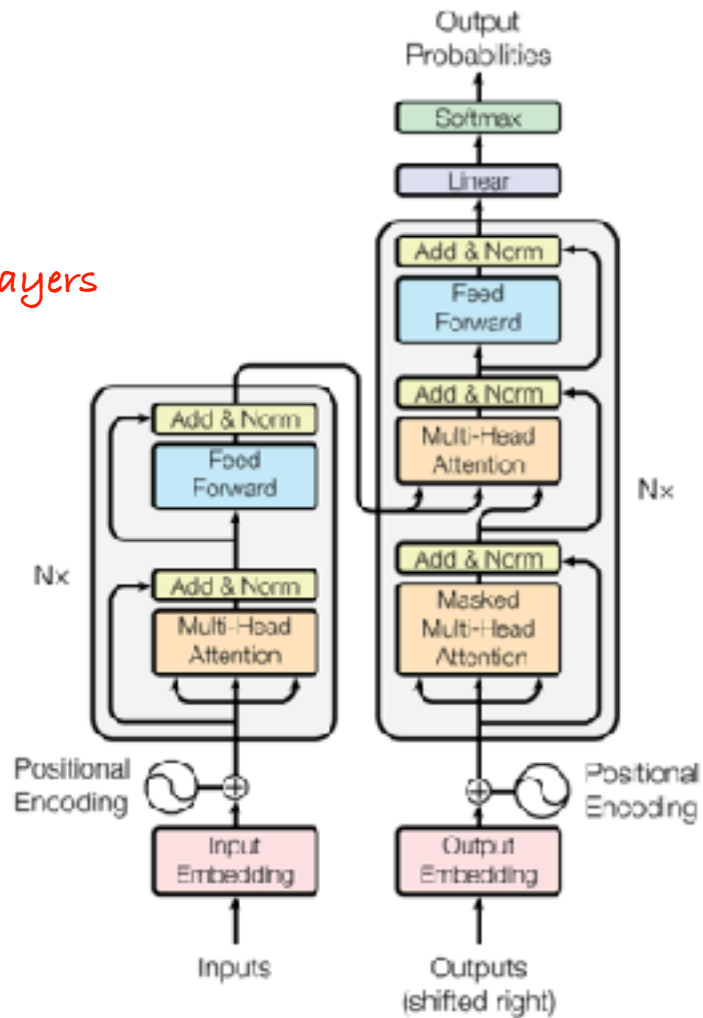
---

- Attention has major impact on seq2seq performance
  - Recurrency is an obstacle to parallelization
- => Can we build a fully attentional seq2seq model without recurrency?



# A scary beast

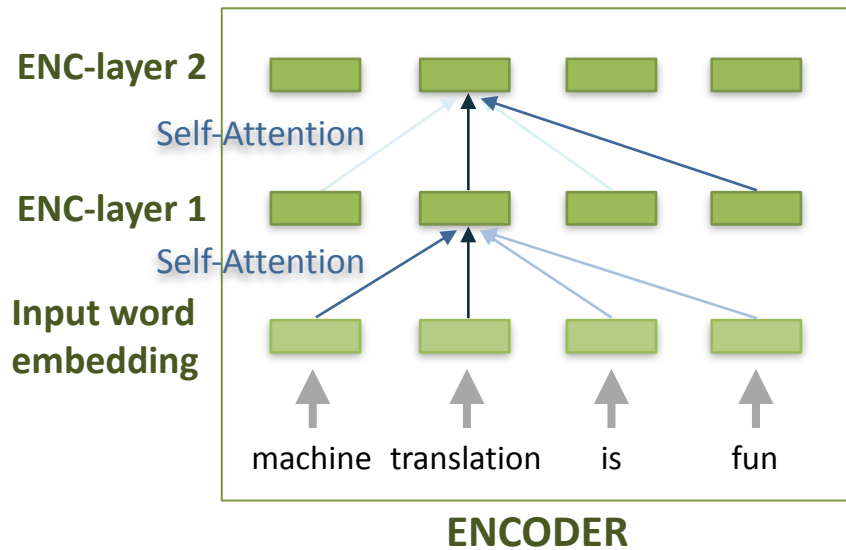
$N = \# \text{layers}$



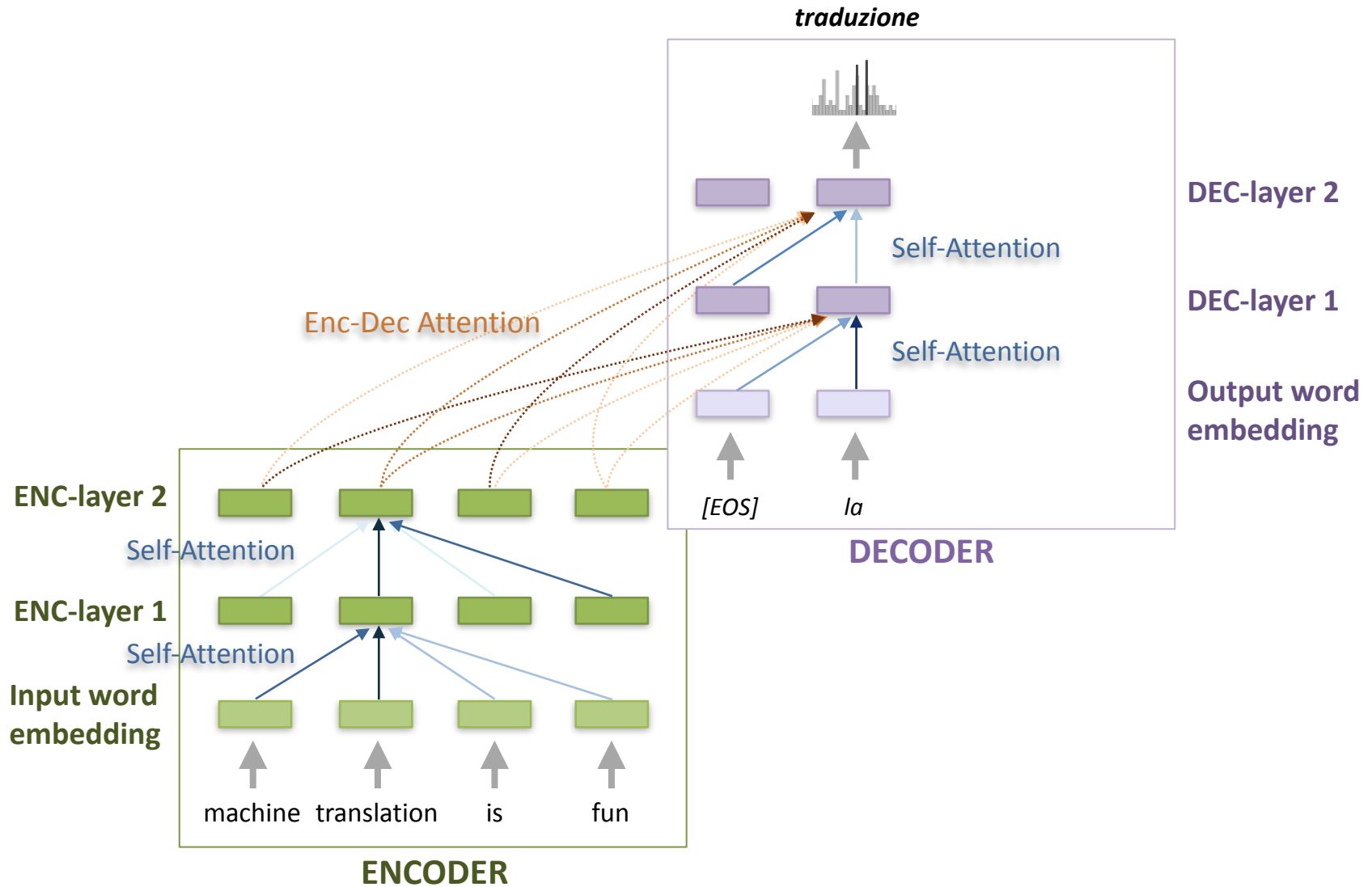
# **TRANSFORMER ARCHITECTURE OVERVIEW**

# Transformer Architecture Overview

---

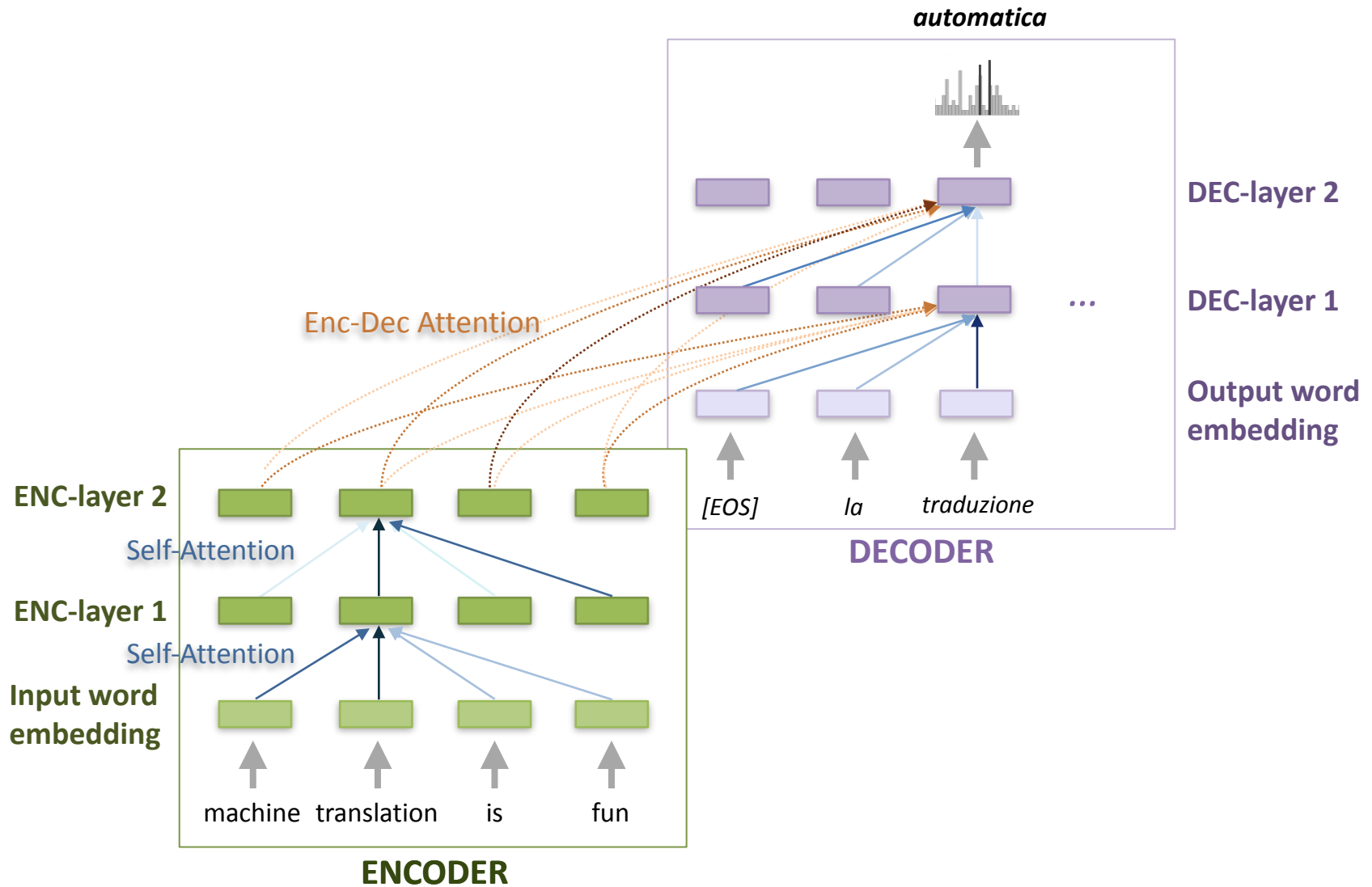


# Transformer Architecture Overview



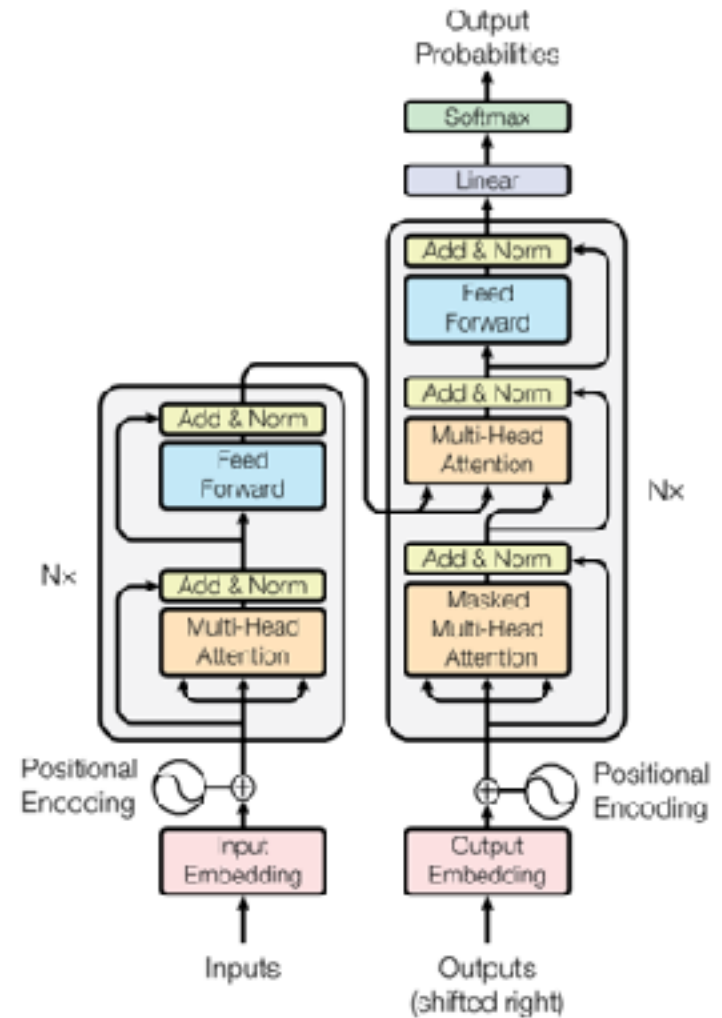


# Transformer Architecture Overview



Let's take a closer look:

# TRANSFORMER'S BUILDING BLOCKS



# Scaled Dot-Product Attention

To compute attention we need a scoring function

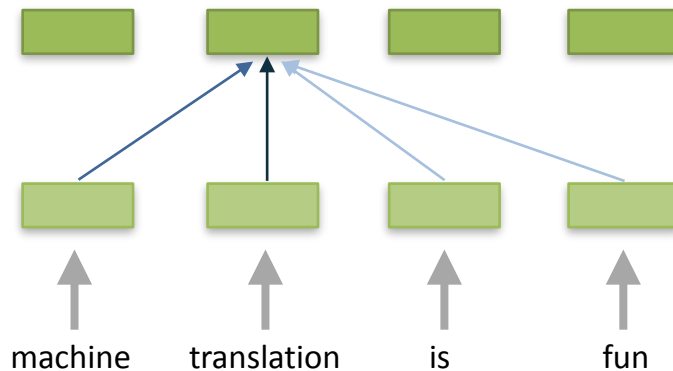
- **Dot-product** is simple and fast to compute\*
- Rationale: measure similarity of two (word-)vectors

Problem: for high-dimensional vectors, softmax gets very peaked and gradients small

=> Solution: scale the result of dot product

$$\text{score}(q_t, k_i) = q_t^\top k_i$$

$$\text{score}(q_t, k_i) = \frac{q_t^\top k_i}{\sqrt{d}}$$



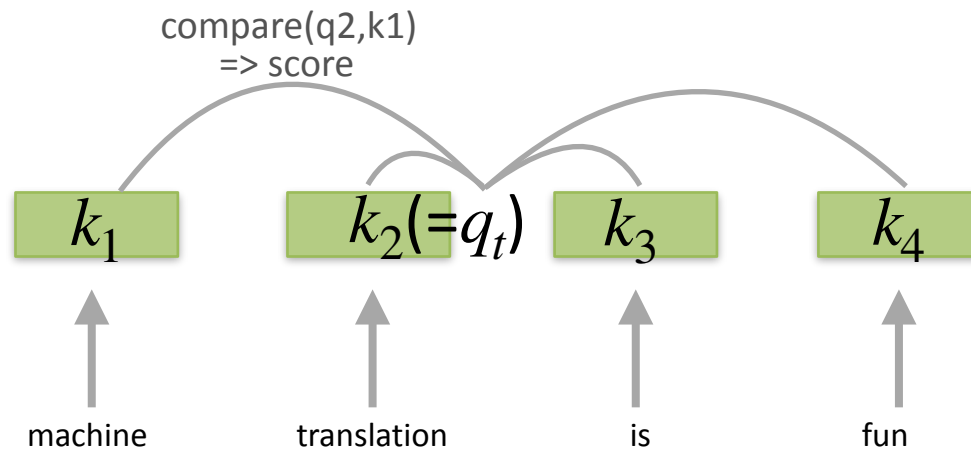
\*For a nice overview of different Attention Scoring Functions see:  
<https://towardsdatascience.com/attn-illustrated-attention-5ec4ad276ee3#ba24>

# Query-Key-Value

Now, where do  $q$  and  $k$  come from?

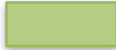
We could simply use the word vector   and compare it to all vectors in the sentence (including itself)

$$\text{score}(q_t, k_i) = \frac{q_t^\top k_i}{\sqrt{d}}$$



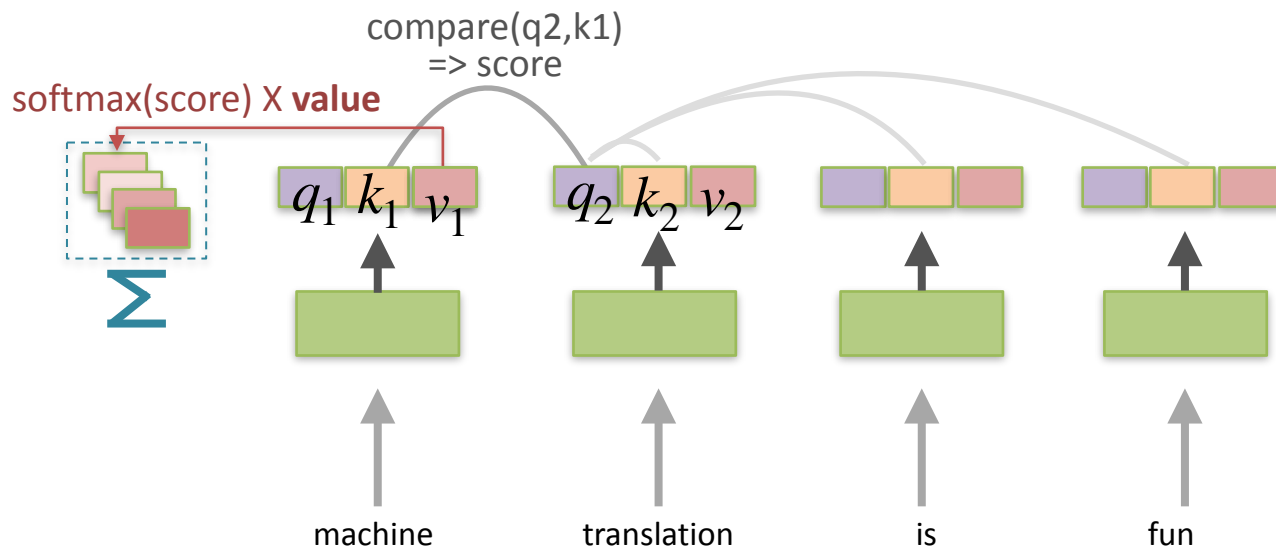
# Query-Key-Value

Now, where do  $q$  and  $k$  come from?

We could simply use the word vector  and compare it to all vectors in the sentence (including itself)


$$\text{score}(q_t, k_i) = \frac{q_t^\top k_i}{\sqrt{d}}$$

A better idea: Learn multiple 'views' of  to use as **query**, **key** and **value**



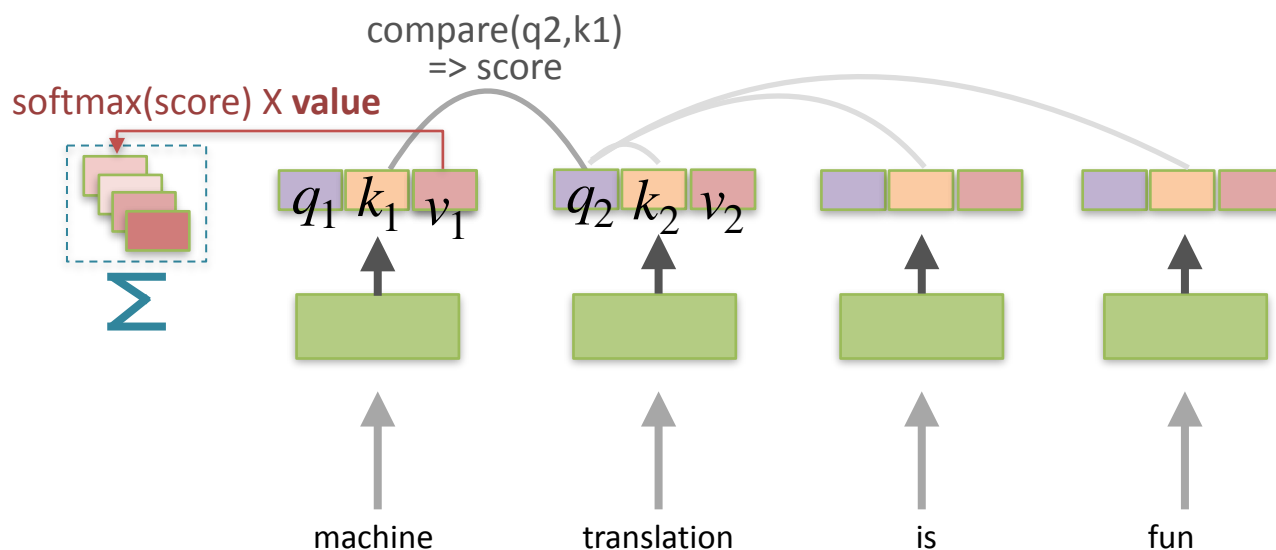
# Query-Key-Value

Now, where do  $q$  and  $k$  come from?

We could simply use the word vector  and compare it to all vectors in the sentence (including itself)

$$\text{score}(q_t, k_i) = \frac{q_t^\top k_i}{\sqrt{d}}$$

A better idea: Learn multiple 'views' of  to use as **query, key and value**



$$\text{Attention}(\hat{Q}, \hat{K}, \hat{V}) = \text{softmax}\left(\frac{\hat{Q}\hat{K}^\top}{\sqrt{d}}\right)\hat{V}$$

$$\begin{aligned} \hat{Q}, \hat{K}, \hat{V} &= QW^Q, KW^K, VW^V \\ &= XW^Q, XW^K, XW^V \quad (\text{self-attention}) \end{aligned}$$

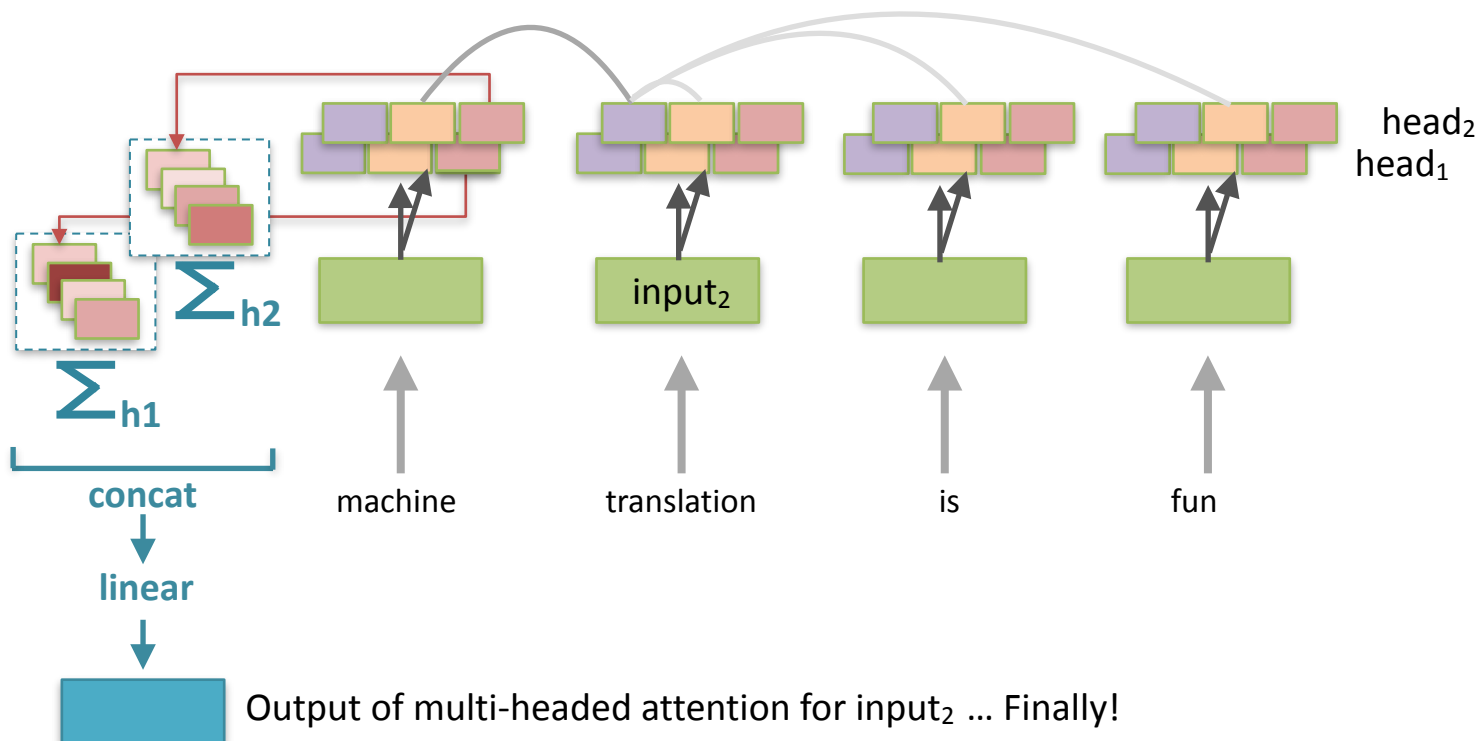
**We are not done yet ...**

# Multi-Head Attention

Words can interact with each other in different ways.

One attention distribution may not be enough to capture: coreference effects, topic cohesion, other syntactic/semantic relationships, etc.

Multi-Head gives the attention layer multiple *representation subspaces*

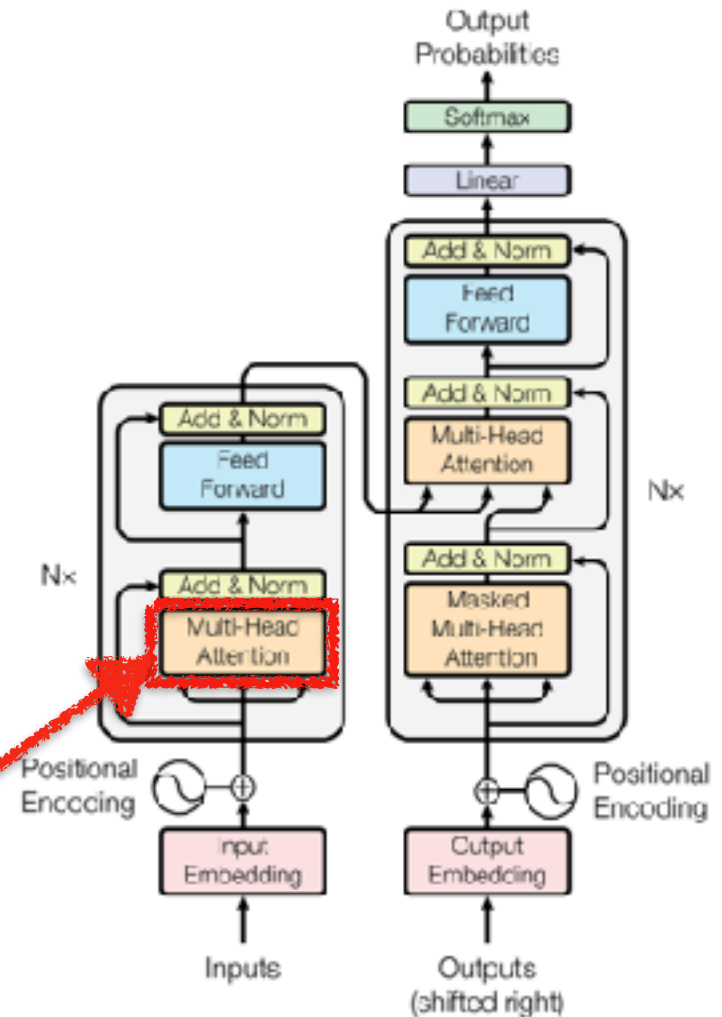
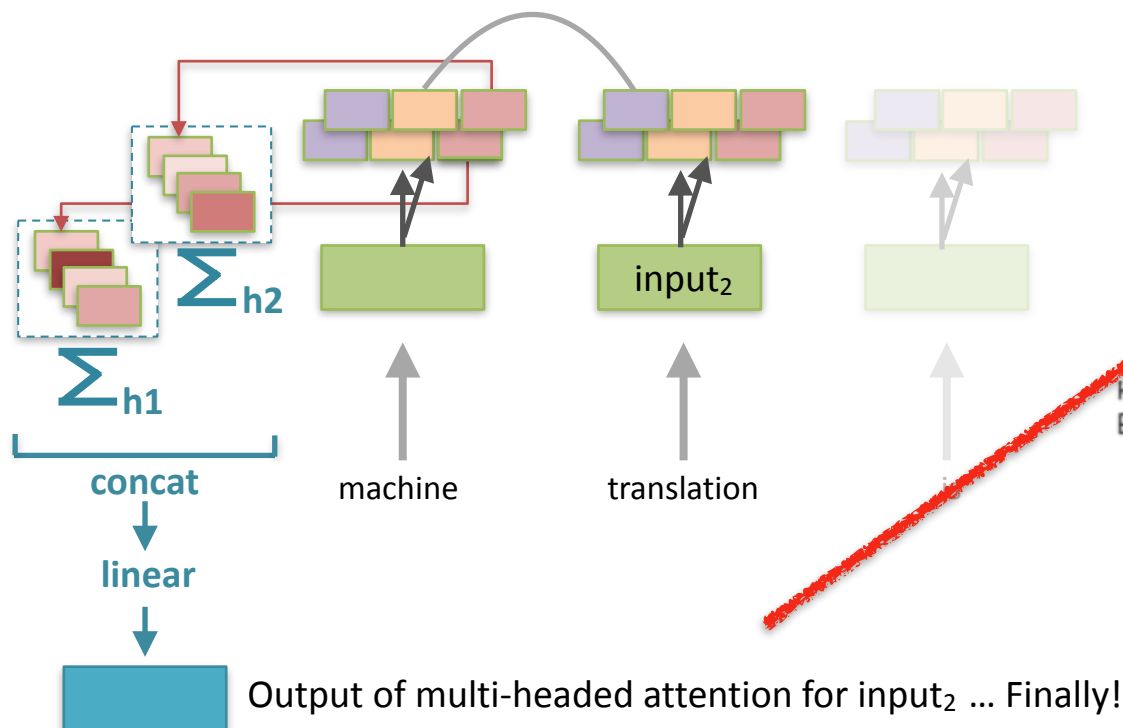




# Multi-Head Attention

$$\text{MultiHead}(Q, K, V) = [\text{head}_1; \dots; \text{head}_h] W^O$$

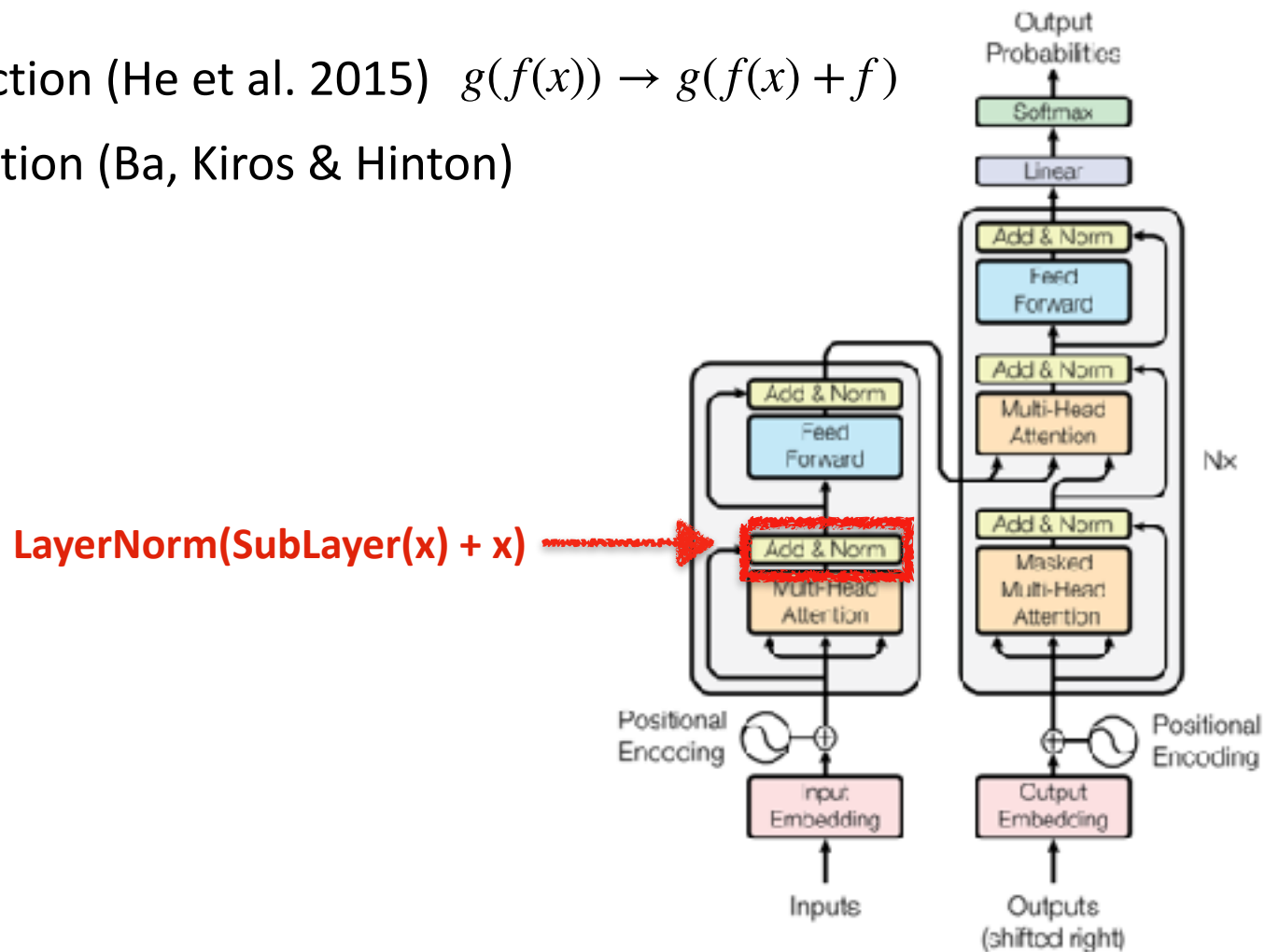
where  $\text{head}_i = \text{Attention}(\hat{Q}_i, \hat{K}_i, \hat{V}_i)$   
 $= \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$



# Add & Norm

Last ingredients:

- Residual connection (He et al. 2015)  $g(f(x)) \rightarrow g(f(x) + f)$
- Layer normalization (Ba, Kiros & Hinton)

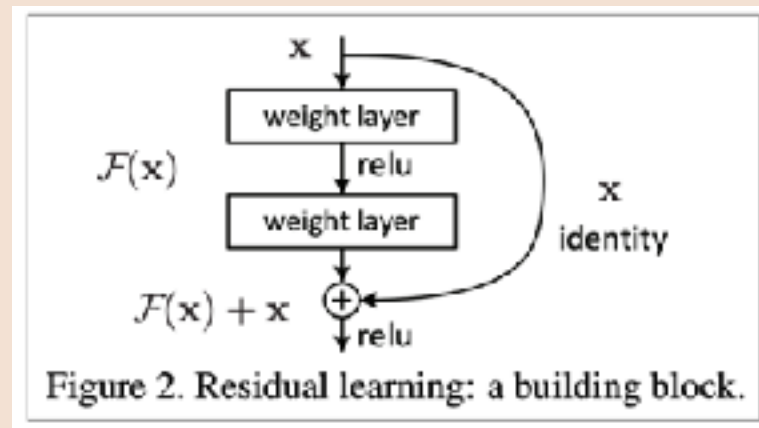


# Residual connections



Slide by Barbara Plank

- ▶ Is the vanishing gradient problem specific to RNNs?
  - ▶ No! Also for deep FFNN and ConvNets
- ▶ **Solution:** add direct “skip” connections (ResNet, residual connections) - proposed by He et al., (2015)
  - ▶ i.e. add  $F(x) + x$ , instead of  $F(x)$
  - ▶ allows for training deeper models



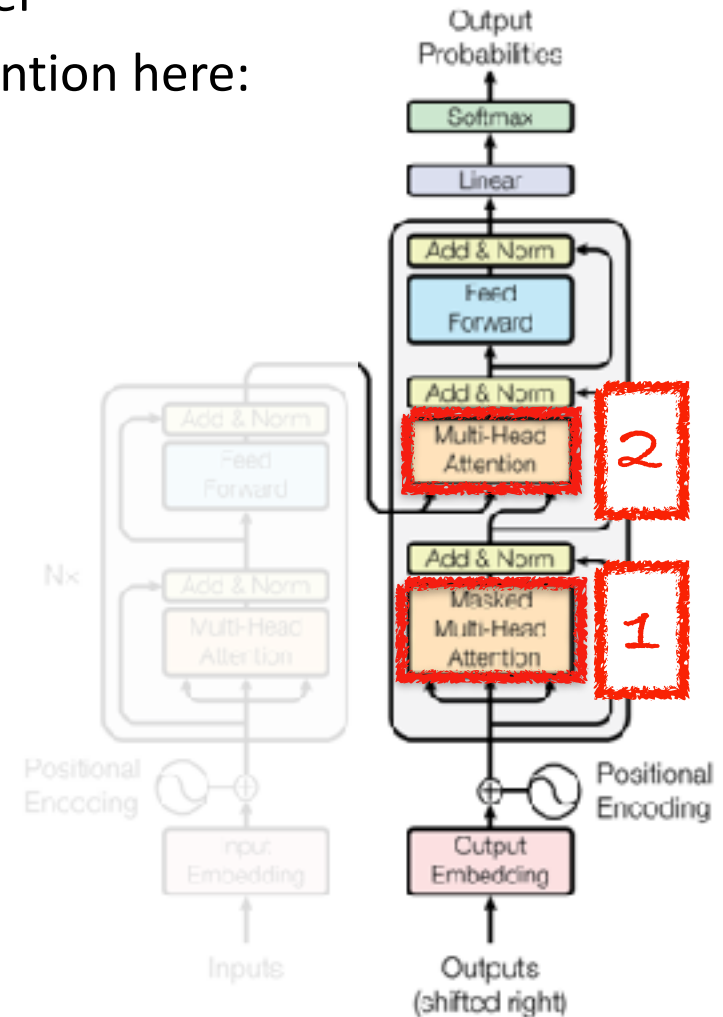
# Attention in the Decoder

We have looked at self-attention in the encoder

Now moving to the decoder => 2 types of attention here:

- 1** Masked Self Attention:
- captures target-side context
  - same as before, but can only look at positions before the current word (*masked*)

- 2** Encoder→Decoder Attention:
- captures src-trg translation equivalences
  - Query comes from target (decoder), Key & Value from source (encoder)



# Attention in the Decoder

We have looked at self-attention in the encoder

Now moving to the decoder => 2 types of attention here:

1

Masked Self Attention:

- captures target-side context
- same as before, but can only look at positions before the current word (*masked*)

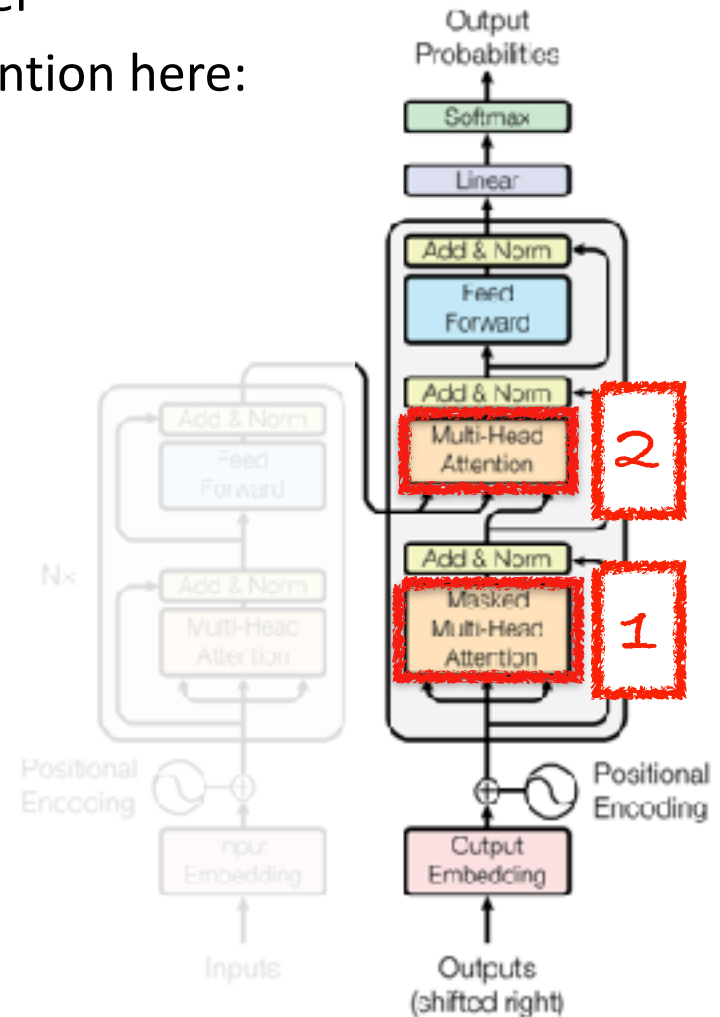
2

Encoder → Decoder Attention:

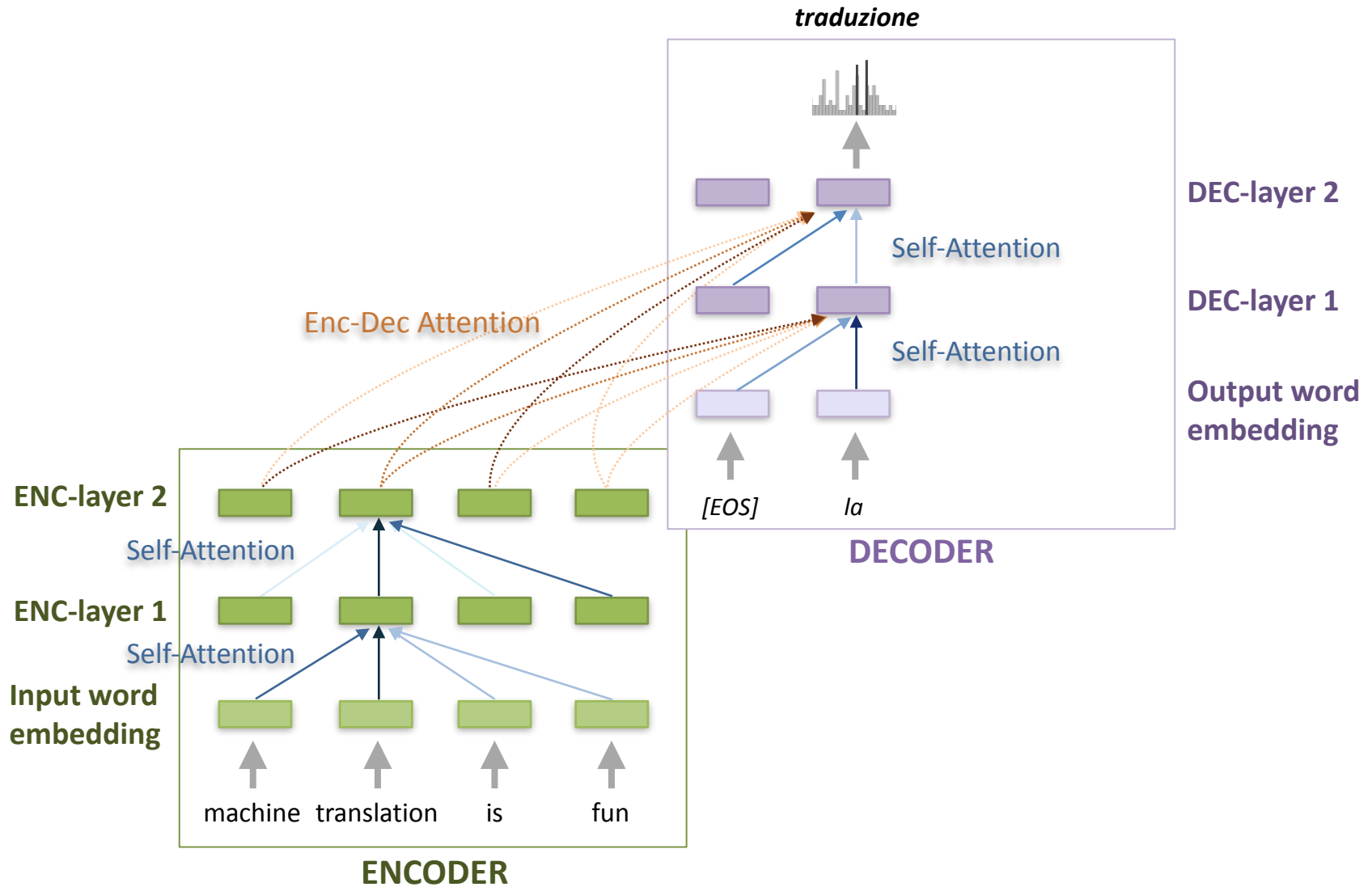
- captures src-trg translation equivalences
- Query comes from target (decoder), Key & Value from source (encoder)

$$\hat{Q}, \hat{K}, \hat{V} = QW^Q, KW^K, VW^V$$

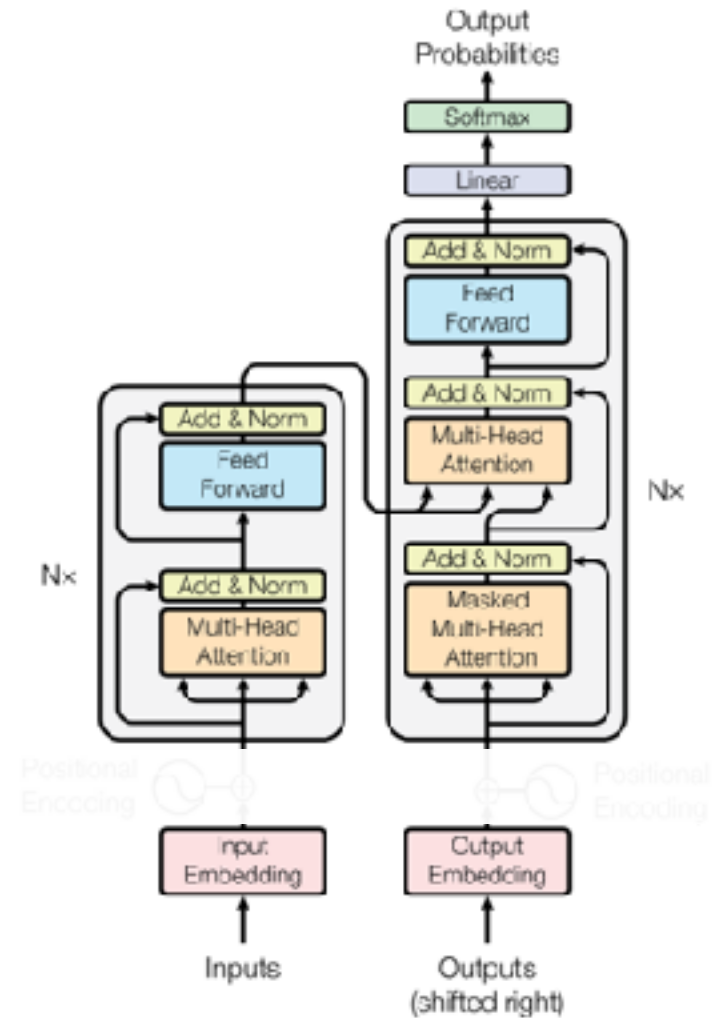
$$= \begin{cases} \mathbf{X}W^Q, \mathbf{X}W^K, \mathbf{X}W^V & (\text{self attention}) \\ \mathbf{X}W^Q, \mathbf{Y}W^K, \mathbf{Y}W^V & (\text{enc} \rightarrow \text{dec attention}) \end{cases}$$



# Transformer Architecture Overview



Are we missing anything?



# Positional embeddings

Recurrency naturally represents the order of words in a sentence:

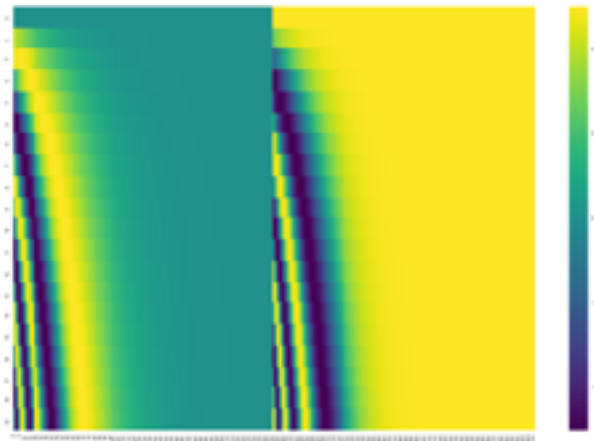
*w<sub>3</sub> comes after w<sub>2</sub> which comes after w<sub>1</sub>...*

Transformer needs an explicit way to represent a word's position

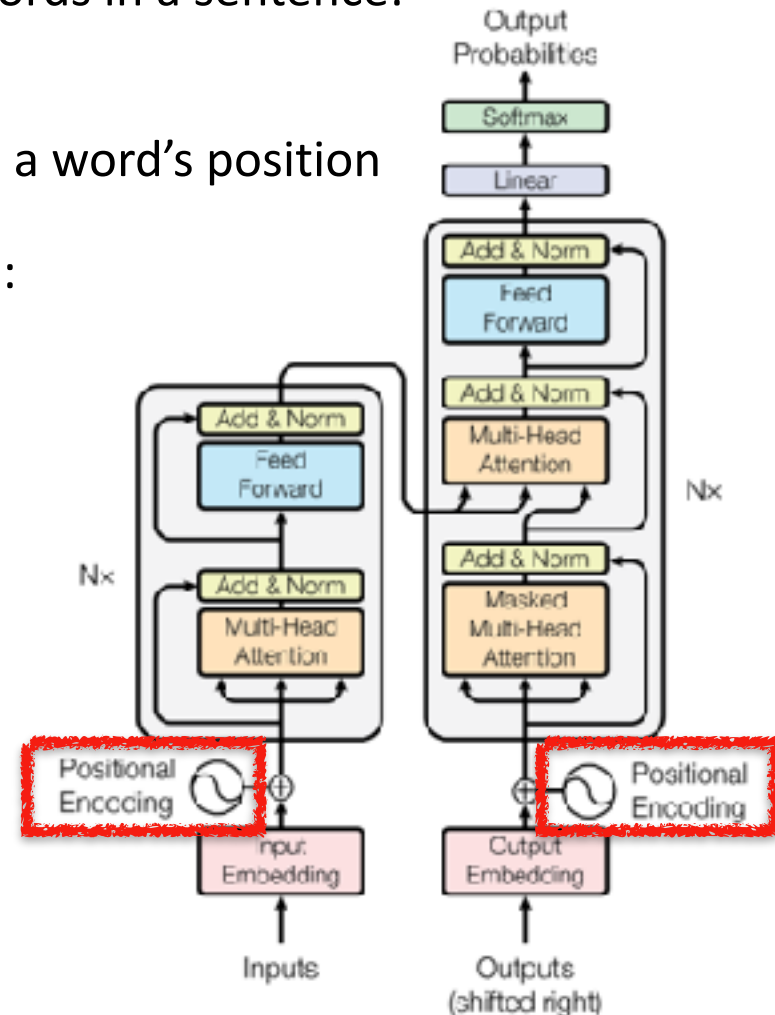
The original architecture employs this function:

$$PE_{(pos,2i)} = \sin(pos/10000^{2i/d_{model}})$$

$$PE_{(pos,2i+1)} = \cos(pos/10000^{2i/d_{model}})$$



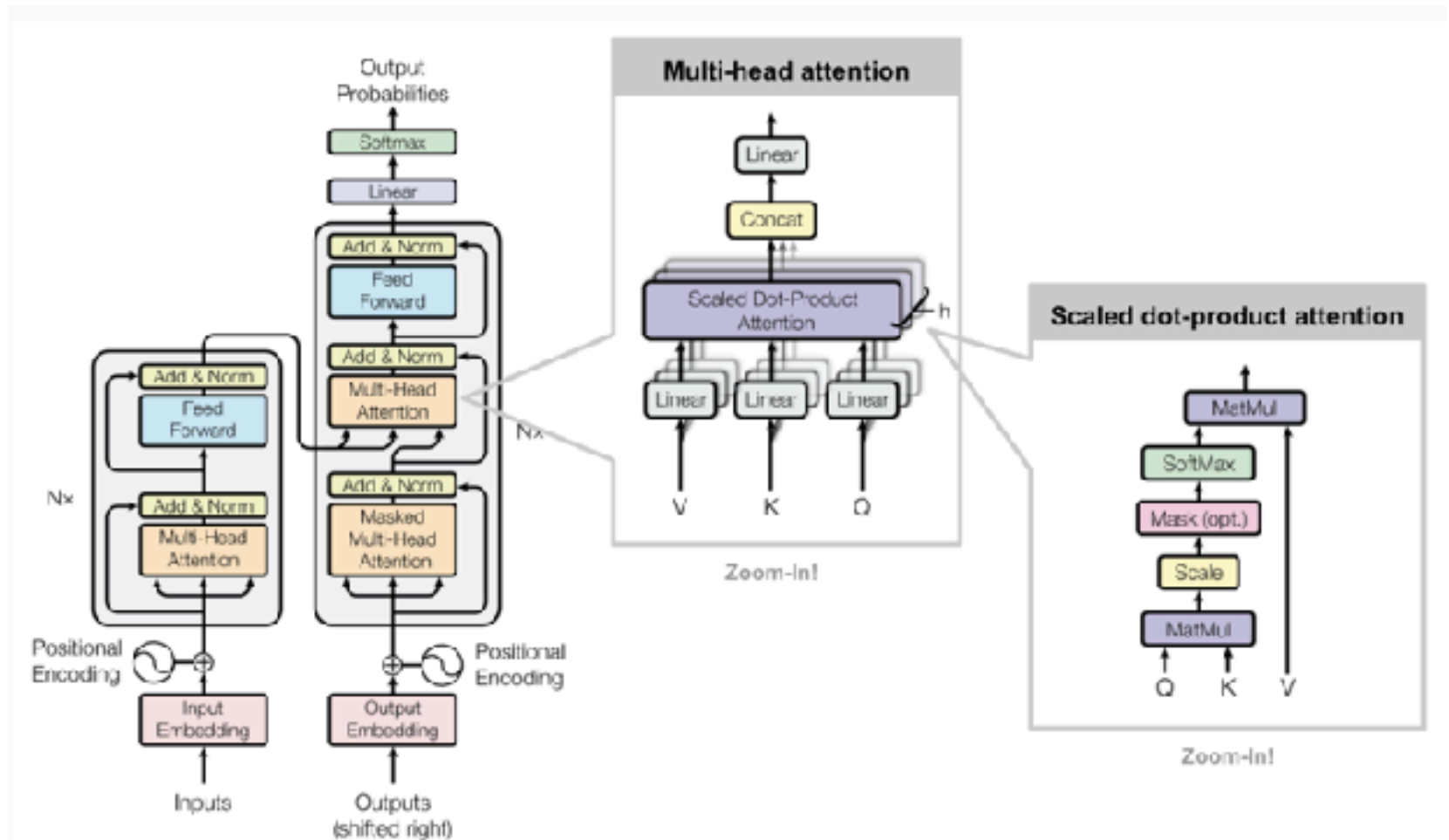
<http://jalamar.github.io/illustrated-transformer/>



**Why?** Because for any fixed offset  $k$ ,  $PE_{pos+k}$  can be represented as a linear function of  $PE_{pos}$

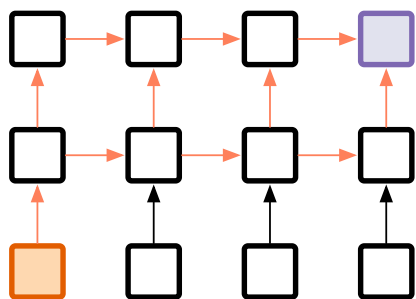


# Putting it altogether

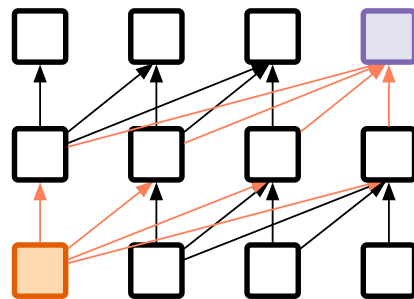


# RECURRENT SEQ-TO-SEQ VS TRANSFORMER

# RNN-seq2seq vs Transformer



RNN



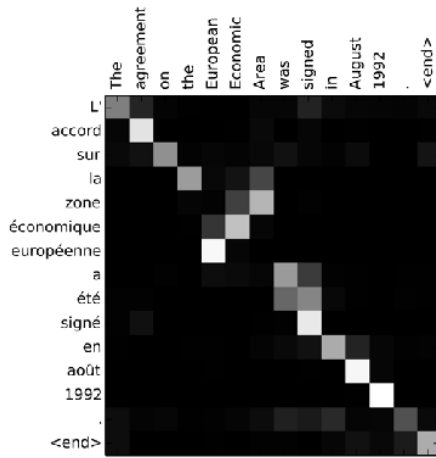
Transformer

- ✓ Much more parallelizable
- ✓ Lower complexity
- ✓ Shorter path among any input positions

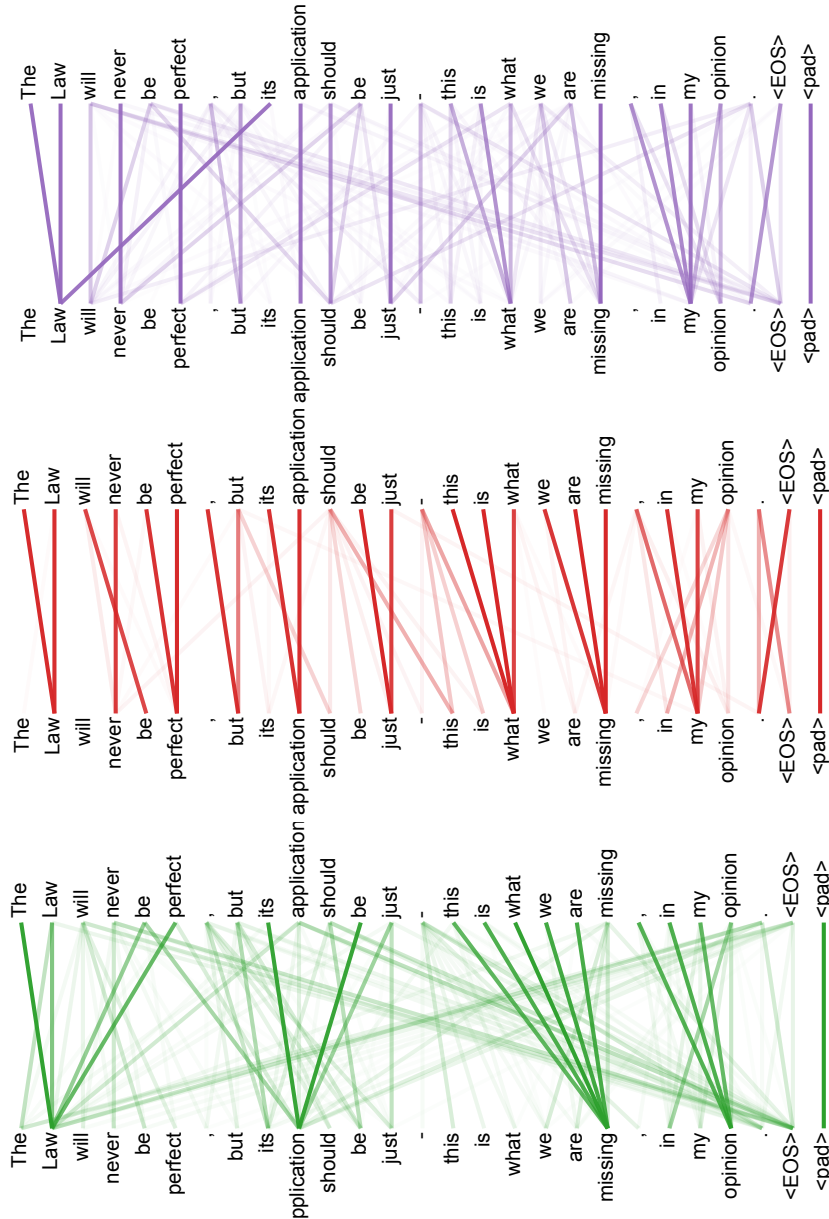
- RNNs (esp. LSTM) are cognitively inspired: represent memory constraints
- Transformer = result of clever engineering & brute-force architecture search
- Does it matter for MT quality? Maybe not
- In fact Transformer is state-of-the-art in MT (and beyond!)
- Ability of RNN/Transformer to model language structure is hot debate topic

Note: This lecture did not cover Convolutional Neural Networks. These also work quite well for MT but are limited to capture dependencies that fall within a chosen kernel size.

# Interpretability



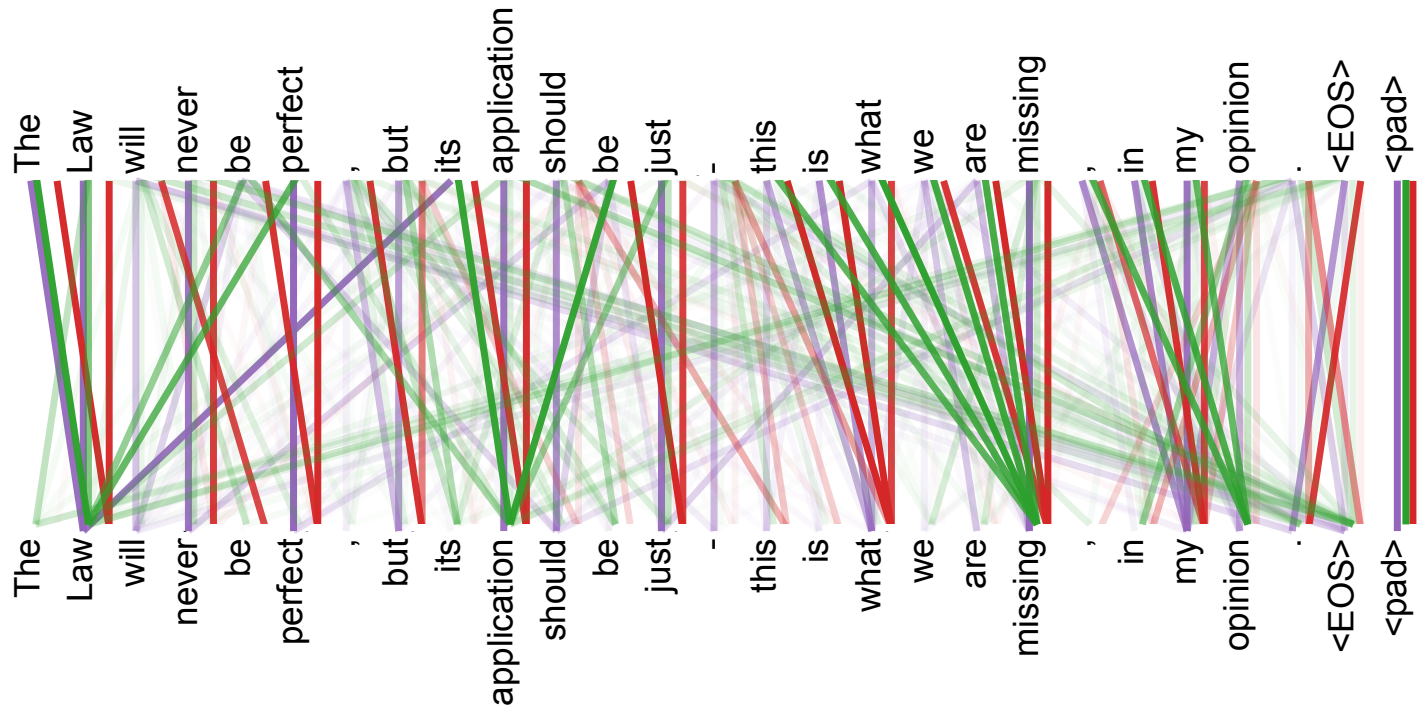
Taken from (Bahdanau et al. 2015)



Taken from (Vaswani et al. 2017)

# Interpretability

---

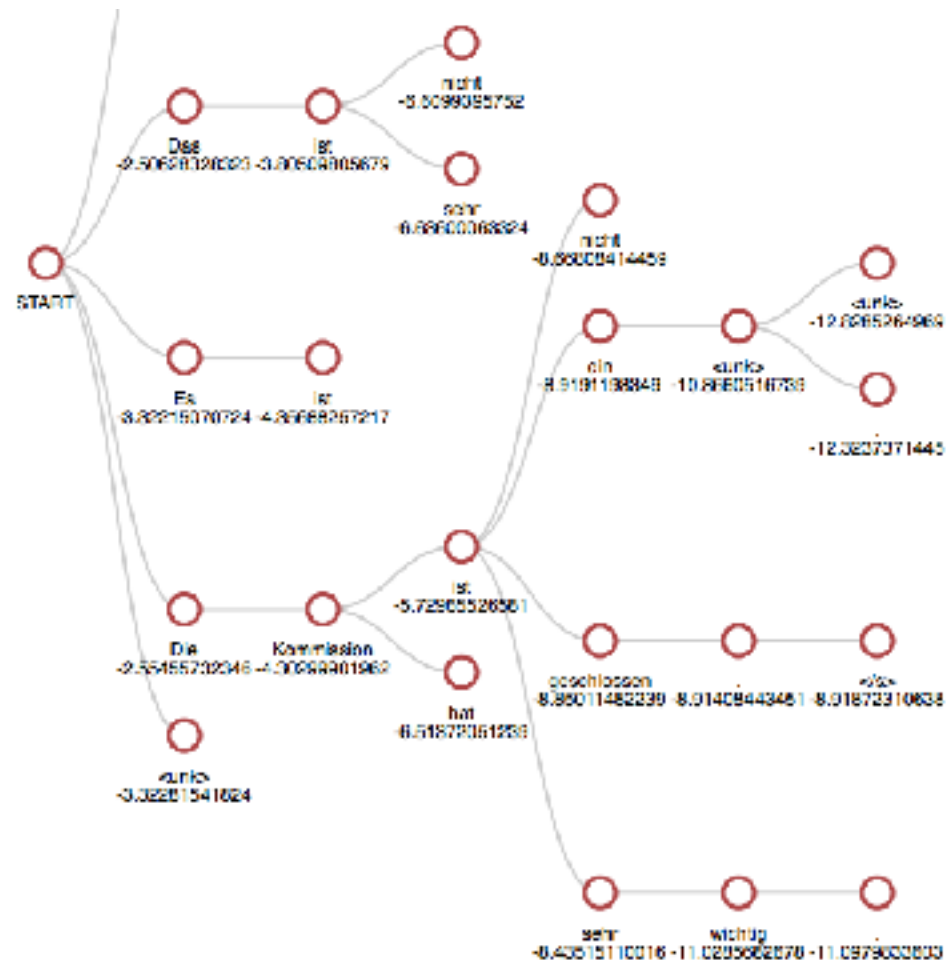


**Is this really more interpretable..?**

# **NMT INFERENCE (DECODING) & WORD SEGMENTATION**

# NMT Decoding

- A simple beam-search procedure is usually sufficient to produce high-quality translations
- A typical beam size: 5
  - Works better than 1 (greedy decoding)
  - Larger beam does not mean better results *in general*



# Word Segmentation

---

- Early NMT models were trained at level of words (space-delimited tokens):
  - Vocabulary was limited at the top N frequent words
  - Rare words mapped to <unk>
- Character-level NMT has also been widely studied:
  - Suitable for morphologically rich languages
  - Sequences become longer → capturing dependencies more difficult
  - Character/word hybrid modeling strategies exist but are typically complex and expensive
- A practical compromise: Subword segmentation
  - Simple data-driven segmentation models (BPE, Sentence Piece) work quite well\*. Idea: Only segment less frequent substrings
  - New words can always be segmented → no more <unk> tokens

*SRC health research institutes*

*REF Gesundheitsforschungsinstitute*

*NMT Gesundheits|forsch|ungsin|stitute*

\*Finding optimal segmentation techniques for MT is an open research topic (see e.g. Ataman & Federico, 2018)



We need to talk about

**EVALUATION**

# MT Evaluation

---

- Evaluating MT is almost as hard as MT itself!
- Potentially infinite ways to translate the same sentence correctly

*IT Sono venuta ad Atene per tenere questa lezione.*

*EN In order to give this lecture I have come to Athens.*

*To teach this class I have come to Athens.*

*To give this lecture I have traveled to Athens.*

*I have come to Athens in order to give this lecture.*

...

Typical solution:

- Collect  $n$  reference translations
- Compare MT output to references
- More overlap → Better translation

Everything has changed in MT, but the most widely used metric is still...

**BLEU!**

# BLEU (Papineni & al. 2002)

---

A modified average of n-gram precisions (usually with  $n$  in [1..4])

$$\text{BLEU-}n = \underbrace{\min \left( 1, \frac{\text{output-length}}{\text{reference-length}} \right)}_{\text{Brevity penalty}} \times \underbrace{\prod_{i=1}^n \text{precision}_i^{\left(\frac{1}{n}\right)}}_{\text{Geometric mean of n-gram precisions}}$$

$$\text{precision}_i = \frac{\# \text{correct-ngrams}_i^*}{\# \text{total-ngrams}_i}$$

\*Nb of correct ngram X is 'clipped' to max count of X in any reference (see paper for details)

- Computed over the whole test corpus to avoid zero counts
- Recall cannot be trivially computed, therefore Brevity Penalty is used to penalize short outputs

# BLEU (Papineni & al. 2002)

A modified average of n-gram precisions (usually with  $n$  in [1..4])

$$\text{BLEU-}n = \underbrace{\min \left( 1, \frac{\text{output-length}}{\text{reference-length}} \right)}_{\text{Brevity penalty}} \times \underbrace{\prod_{i=1}^n \text{precision}_i^{\left(\frac{1}{n}\right)}}_{\text{Geometric mean of n-gram precisions}}$$

$$\text{precision}_i = \frac{\#\text{correct-ngrams}_i^*}{\#\text{total-ngrams}_i}$$

Example (with  $n=2$ )

REF1 *in order to give this lecture I have traveled to Athens* ( $\ell=11$ )

REF2 *I came to Athens in order to give this lecture* ( $\ell=10$ )

MT to teach this class I have come to Athens ( $\ell=9$ )

$$\text{BP} = 9/10 = 0.9$$

$$\text{precision}_1 = 6/9 = 0.67$$

$$\text{precision}_2 = 2/8 = 0.25$$

$$\text{BLEU-2} = 0.9 \times \left( 0.67^{\left(\frac{1}{2}\right)} \times 0.25^{\left(\frac{1}{2}\right)} \right) = 0.37$$

# BLEU: Issues

---

- Only exact lexical matches count
- Synonyms, paraphrases, or morphological variants don't count
- Most of the time only 1 reference is available :(

Example (with $n=2$ )		
<del>REF1</del>	<del>in order to give this lecture I have traveled to Athens</del>	<del>(<math>\ell=11</math>)</del>
REF2	I came to Athens in order to give this lecture	( $\ell=10$ )
MT	<u>to</u> <u>teach</u> <u>this</u> <u>class</u> <u>I</u> <u>have</u> <u>come</u> <u>to</u> <u>Athens</u>	( $\ell=9$ )

- Similar issues affect evaluation of other generation tasks:

Translation:      SrcLang(meaning X)      → TrgLang(meaning X)  
Summarization:   Text(X)                      → ShortText(X)  
Paraphrasing:     Sentence(X)                      → Sentence'(X)

- Finding MT metrics that correlate well with human judgement is a research field on its own (with dedicated shared task at WMT)

# OPEN ISSUES IN MT

# MT: Human parity?

March 2018: Microsoft claims human parity on a very difficult language pair

## Achieving Human Parity on Automatic Chinese to English News Translation

Hany Hassan, Anthony Aue, Chang Chen, Vishal Chowdhary, Jonathan Clark, Christian Federmann, Xuedong Huang, Marcin Junczys-Dowmunt, William Lewis, Mu Li, Shujie Liu, Tie-Yan Liu, Renqian Luo, Arul Menezes, Tao Qin, Frank Seide, Xu Tan, Fei Tian, Lijun Wu, Shuangzhi Wu, Yingce Xia, Dongdong Zhang, Zhirui Zhang, and Ming Zhou

Microsoft AI & Research

### Abstract

Machine translation has made rapid advances in recent years. Millions of people are using

Yes: MT quality has greatly improved thanks to the neural revolution, **but...**

- translation quality considering (document-level) context is still shaky
- recent studies reveal biases and lack of sistematicity in NMT
- NMT is very data hungry!
- dealing with rich vocabularies (morphology) is still hard



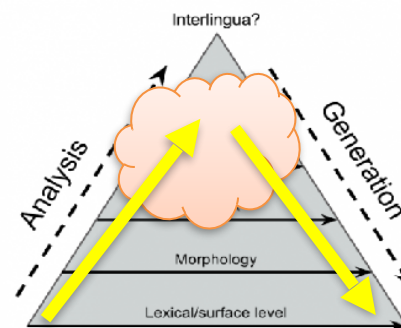
# NMT vs SMT

What has been solved (or extremely improved) by NMT\*:

- ✓ models capture distributional semantics of words and phrases
- ✓ overall grammaticality of output sentences
  - in particular: word reordering, long dependencies

New (or exacerbated) issues in NMT:

- how to make use of large monolingual data 
$$e^* = \arg \max_e p(f | e) p(e)$$
$$e^* = \arg \max_e p_{\text{NMT}}(e | f)$$
- learning representations of rare words
- poor model interpretability, makes it difficult to:
  - 'debug' translation errors
  - enhance models with expert knowledge (e.g. terminologies or morphological lexicons)



# REFERENCES & USEFUL LINKS

# An (incomplete) list of references (I)

---

## *- Statistical Machine Translation (Phrase-Based and Beyond):*

[Koehn, 2012] **Statistical Machine Translation**. Cambridge University Press.

[Bisazza & Federico, 2016] **A Survey of Word Reordering in SMT: Computational Models and Language Phenomena**. Computational Linguistics.

## *- Early attempts to NMT in the 90's:*

[Castano & Casacuberta, 1997] **A connectionist approach to MT**. In Proc. of EUROSPEECH.

[Forcada and Neco, 1997] **Recursive hetero-associative memories for translation**. In Proc. of IWANN.

## *- Foundations of modern NMT:*

[Kalchbrenner & Blunsom, 2013] **Recurrent Continuous Translation Models**

[Sutskever et al. 2014] **Sequence to Sequence Learning with Neural Networks**

[Cho & al. 2014] **Learning Phrase Representations using RNN Encoder-Decoder for Statistical MT**

[Bahdanau & al, 2015] **Neural Machine Translation by Jointly Learning to Align and Translate**

[Vaswani & al. 2017] **Attention Is All You Need**.

## *- Influential NMT system papers:*

[Wu & al. 2016] **Google's Neural MT System: Bridging the Gap between Human and MT**.

[Hassan & al. 2018] **Achieving Human Parity on Automatic Chinese to English News Translation**.

## *- Word segmentation:*

[Sennrich & al. 2016] **NMT of Rare Words with Subword Units**.

[Kudo & Richardson, 2018] **SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing**.

[Ataman & Federico , 2018] **An Evaluation of Two Vocabulary Reduction Methods for NMT**.

# An (incomplete) list of references (II)

---

## *- BLEU evaluation metric:*

[Papineni & al. 2002] **BLEU: a Method for Automatic Evaluation of Machine Translation**

[Callison-Burch et al. 2006] **Re-evaluating the Role of BLEU in Machine Translation Research**

[Federmann, 2011] [How can we measure machine translation quality?](#)

[Dorr, 2011] [Machine Translation Evaluation](#)

## *- SMT vs NMT:*

[Bentivogli & al. 2018] **Neural versus Phrase-based MT quality: An in-depth Analysis on English–German and English–French**

## *- Interpretability/Linguistic probing of NMT models:*

[Shi & al. 2016] **Does string-based neural MT learn source syntax?**

[Belinkov & al. 2017] **What do neural machine translation models learn about morphology?**

[Sennrich, 2017] **How Grammatical is Character-level NMT? Assessing MT Quality with Contrastive Translation Pairs**

## *- RNN-seq2seq vs Transformer:*

[Tran & al. 2018] **The importance of being recurrent for modeling hierarchical structure**

[Tang & al. 2018] **Why self-attention? a targeted evaluation of neural machine translation architectures**

# Useful Links

---

- Transformer paper annotated with code:

<http://nlp.seas.harvard.edu/2018/04/03/attention.html>

- The illustrated Transformer:

<http://jalammar.github.io/illustrated-transformer/>

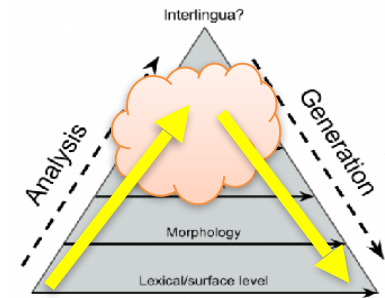
- Attention? Attention! (blogpost on the concept of attention and its variants):

<https://lilianweng.github.io/lil-log/2018/06/24/attention-attention.html>

- Choose your NMT toolkit:

[Joey NMT: A Minimalist NMT Toolkit for Novices](#) (Kreutzer, Bastings, Riezler, 2019)

The paper contains an extensive list of other toolkits and their features



THANKS FOR YOUR  
ATTENTION!

AthNLP2019



23 Sept 2019