More about Machine Translation

SIU Sai Cheong
School of Translation, Hang Seng Management College

MT Evaluation

Automatic Evaluation

The use of Bilingual Evaluation Understudy (BLEU) metric (Papineni, Roukos, Ward, & Zhu, 2002), a measure of n-gram precision with a value between 0 and 1, comparing a machine translation output with a reference:

$$BLEU = \min(1, e^{1-\frac{r}{c}}) \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right)$$

More about BLEU

$$BLEU = \min(1, e^{1-\frac{r}{c}}) \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right)$$

Given T = MT output and T_R = reference translation,

$$Precision = \frac{C(T \cap T_R)}{|T|}$$

Example

T = "Chinese officials responsibility of airport safety"

T_R = "Chinese officials are responsible for airport security"

$$Precision = \frac{3}{6} = 0.5$$

For more information, visit https://www.aclweb.org/anthology/P02-1040.pdf

More about SMT

Statistical Machine Translation

$$P(T|S) = \frac{P(S|T) \times P(T)}{P(S)}$$

$$\begin{aligned} T' &= \arg\max_{T} P(T|S) \\ &= \arg\max_{T} \underbrace{P(S|T)}_{\text{Translation Model}} \times \underbrace{P(T)}_{\text{Language Model}} \end{aligned}$$

SMT(1): Language Model

Language Model

Do you prefer fish ?
Elephants love her .
Give me fish .
Dogs love swimming .
We love dogs .

$$P(T)$$

Monolingual Corpus in the
Target Language

$P(T)$

- $= P(t_1 t_2 \dots t_n)$
- $= P\left(t_1\right) \prod_{i=2}^{n} P\left(t_i | t_1 t_2 \dots t_{i-1}\right)$

N-gram Model

- We consider *n* words at a time (i.e., the word *w* and *n*-1 words in the context history *h*).

$$
\begin{aligned}
& P(W) \\
= \; & P(w_1 w_2 \ldots w_n w_{n+1}) \\
= \; & \prod_{i=1}^{n+1} P(w_i \mid w_{i-n+1} w_{i-n+2} \ldots w_{i-1}) \\
= \; & \prod_{i=1}^{n+1} \frac{C(w_{i-n+1} w_{i-n+2} \ldots w_{i-1} w_i)}{C(w_{i-n+1} w_{i-n+2} \ldots w_{i-1})}
\end{aligned}
$$

n = 1

$$P(w_i) = \frac{C(w_i)}{|\text{corpus}|}$$

n > 1

$$P(w_i|h) = \frac{C(h+w_i)}{C(h)},$$

where $h = w_{i-n+1} w_{i-n+2} \ldots w_{i-1}$

where $n=|W|$, "<BOS>"$=w_{-n+2}, w_{-n+3}, \ldots, w_0$ and "<EOS>"$=w_{n+1}$

Unigram Model (*n*=1)

$$
\begin{aligned}
& P(W) \\
= \; & P(w_1 w_2 \ldots w_\mathrm{n} w_{\mathrm{n+1}}) \\
= \; & P(w_1) \times P(w_2) \times \ldots \times P(w_\mathrm{n}) \times P(w_{\mathrm{n+1}}) \\
= \; & \prod_{i=1}^{n+1} P(w_\mathrm{i})
\end{aligned}
$$

where $n = |W|$ and w_{n+1}="<EOS>"

Unigram Model

- P("How much is this book?")

$$
\begin{aligned}
&= P(\text{“how”}) \\
&\times P(\text{“much”}) \\
&\times P(\text{“is”}) \\
&\times P(\text{“this”}) \\
&\times P(\text{“book”}) \\
&\times P(\text{“?”}) \\
&\times P(\text{“<EOS>”})
\end{aligned}
$$

```
<BOS> What is this ? <EOS>
<BOS> This is a ball . <EOS>
<BOS> I bought a book . <EOS>
<BOS> That sounds good . <EOS>
<BOS> How much is it ? <EOS>
```

Bigram Model (*n*=2)

$$
\begin{aligned}
& P(W) \\
= \;& P(w_1 w_2 \ldots w_n w_{n+1}) \\
= \;& P(w_1|w_0) \times P(w_2|w_1) \times P(w_3|w_2) \times \ldots \times P(w_n|w_{n-1}) \times P(w_{n+1}|w_n) \\
= \;& \prod_{i=1}^{n+1} P(\mathrm{w_i}|\mathrm{w_{i-1}})
\end{aligned}
$$

where $n=|W|$, w_0="<BOS>" and w_{n+1}="<EOS>"

Bigram Model

- P(“How much is this book?”)

$$
\begin{aligned}
&= P(\text{“how”}|\text{“<BOS>”}) \\
&\times P(\text{“much”}|\text{“how”}) \\
&\times P(\text{“is”}|\text{“much”}) \\
&\times P(\text{“this”}|\text{“is”}) \\
&\times P(\text{“book”}|\text{“this”}) \\
&\times P(\text{“?”}|\text{“book”}) \\
&\times P(\text{“<EOS>”}|\text{“?”})
\end{aligned}
$$

```
<BOS> What is this ? <EOS>
<BOS> This is a ball . <EOS>
<BOS> I bought a book . <EOS>
<BOS> That sounds good . <EOS>
<BOS> How much is it ? <EOS>
```

Trigram Model (*n*=3)

$$
\begin{aligned}
& P(W) \\
= \; & P(w_1 w_2 \ldots w_n w_{n+1}) \\
= \; & P(w_1|w_{-1} w_0) \times P(w_2|w_0 w_1) \times P(w_3|w_1 w_2) \times \ldots \times P(w_n|w_{n-2} w_{n-1}) \times P(w_{n+1}|w_{n-1} w_n) \\
= \; & \prod_{i=1}^{n+1} P(\mathrm{w}_i|w_{i-2} w_{i-1})
\end{aligned}
$$

where $n=|W|$, $w_{-1}=w_0=$“<BOS>” and $w_{n+1}=$“<EOS>”

Trigram Model

- P(“How much is this book?”)

= *P*(“how”|“<BOS><BOS>”)
× *P*(“much”|“<BOS> how”)
× *P*(“is”|“how much”)
× *P*(“this”|“much is”)
× *P*(“book”|“is this”)
× *P*(“?”|“this book”)
× *P*(“<EOS>”|“book ?”

```
<BOS> What is this ? <EOS>
<BOS> This is a ball . <EOS>
<BOS> I bought a book . <EOS>
<BOS> That sounds good . <EOS>
<BOS> How much is it ? <EOS>
```

SMT(2): Translation Model

Translation Model

Source Language
我喜歡貓 。大象鍾意我 。人們喜歡狗 。他們嗜魚 。
我們喜歡猴子 。你喜歡豬 。

Target Language
I love cats. Elephants love me. People love dogs.
They prefer fish. We love monkeys. You prefer pigs.

$$P(S|T)$$

Bilingual Corpus

Translation Model

Source Language
我喜歡貓。大象鍾意我。人們喜歡狗。
他們嗜魚。我們喜歡猴子。你喜歡豬。

Target Language
I love cats. Elephants love me. People love dogs. They prefer fish. We love monkeys. You prefer pigs.

Bilingual Corpus

$$P(S \mid T)$$

Sentence Alignment

我 喜歡 貓 。 → I love cats .
大象 鍾意 我 。 → Elephants love me .
人們 喜歡 狗 。 → People love dogs .
他們 嗜 魚 。 → They prefer fish .
我們 喜歡 猴子 。 → We love monkeys .
你 喜歡 豬 。 → You prefer pigs .

. → 。 cats → 貓 dogs → 狗
elephants → 大象 fish → 魚 I → 我
prefer → 喜歡 prefer → 嗜
love → 喜歡 love → 鍾意
monkeys → 猴子 pigs → 豬

Word Alignment

Translation Model $P(S|T)$

If there are n word/phrase pairs

$$\{(s_1, t_1), (s_2, t_2), \ldots, (s_n, t_n)\},$$

and they form the sentence pair

$$S = s_1 s_2 \ldots s_n,\ T = t_1 t_2 \ldots t_n,$$

we have

$$P(S|T) = P(s_1|t_1) \times P(s_2|t_2) \times \ldots \times P(s_n|t_n)$$
$$= \prod_{i=1}^{n} P(s_n|t_n)$$

$$P(s_n|t_n) = \frac{C(s_n \cap t_n)}{C(t_n)}$$

Translation Model

Given the following:

S = “我去遠足。”
T = “I go hiking.”
a set of phrase pairs
= {(“我”,“I”),(“去”,“go”),(“遠足”,“hiking”),(“。”,“.”)}

What is $P(S|T)$ (i.e., the Probability of “我去遠足。” as the source of “I go hiking.”)?

Translation Model

Divide S and T by referring to the phrase pairs provided

P(“我去遠足。”|“I go hiking.”)

$= P$(“我”|“I”)

$\times\, P$(“去”|“go”)

$\times\, P$(“遠足”|“hiking”)

$\times\, P$(“。”|“.”)

SMT(3): Training

Training (1): Language Model

Monolingual Corpus

```
<BOS> Do you prefer fish ? <EOS>
<BOS> Elephants love her . <EOS>
<BOS> Give me fish . <EOS>
<BOS> They love elephants . <EOS>
<BOS> Dogs love swimming . <EOS>
<BOS> We love dogs . <EOS>
```

Monolingual Table

<table>
<tr><th>s</th><th>$w_i | w_{i-1}$</th><th>$C(s)$</th><th>$C(w_{i-1})$</th><th>$P(w_i | w_{i-1})$</th><th>Log
$P(w_i | w_{i-1})$</th></tr>
<tr><td>. <EOS></td><td><EOS> | .</td><td>5</td><td>5</td><td>1.0000</td><td>0.000</td></tr>
<tr><td>? <EOS></td><td><EOS> | ?</td><td>1</td><td>1</td><td>1.0000</td><td>0.000</td></tr>
<tr><td><BOS> do</td><td>do | <BOS></td><td>1</td><td>6</td><td>0.1667</td><td>-0.778</td></tr>
<tr><td><BOS> dogs</td><td>dogs | <BOS></td><td>1</td><td>6</td><td>0.1667</td><td>-0.778</td></tr>
<tr><td><BOS> elephants</td><td>elephants | <BOS></td><td>1</td><td>6</td><td>0.1667</td><td>-0.778</td></tr>
<tr><td><BOS> give</td><td>give | <BOS></td><td>1</td><td>6</td><td>0.1667</td><td>-0.778</td></tr>
<tr><td><BOS> they</td><td>they | <BOS></td><td>1</td><td>6</td><td>0.1667</td><td>-0.778</td></tr>
<tr><td><BOS> we</td><td>we | <BOS></td><td>1</td><td>6</td><td>0.1667</td><td>-0.778</td></tr>
<tr><td>do you</td><td>you | do</td><td>1</td><td>1</td><td>1.0000</td><td>0.000</td></tr>
<tr><td>dogs .</td><td>. | dogs</td><td>1</td><td>2</td><td>0.5000</td><td>-0.301</td></tr>
<tr><td>dogs love</td><td>love | dogs</td><td>1</td><td>2</td><td>0.5000</td><td>-0.301</td></tr>
<tr><td>elephants .</td><td>. | elephants</td><td>1</td><td>2</td><td>0.5000</td><td>-0.301</td></tr>
<tr><td>elephants love</td><td>love | elephants</td><td>1</td><td>2</td><td>0.5000</td><td>-0.301</td></tr>
<tr><td>fish .</td><td>. | fish</td><td>1</td><td>2</td><td>0.5000</td><td>-0.301</td></tr>
<tr><td>fish ?</td><td>? | fish</td><td>1</td><td>2</td><td>0.5000</td><td>-0.301</td></tr>
<tr><td>give me</td><td>me | give</td><td>1</td><td>1</td><td>1.0000</td><td>0.000</td></tr>
<tr><td>her .</td><td>. | her</td><td>1</td><td>1</td><td>1.0000</td><td>0.000</td></tr>
<tr><td>love dogs</td><td>dogs | love</td><td>1</td><td>4</td><td>0.2500</td><td>-0.602</td></tr>
<tr><td>love elephants</td><td>elephants | love</td><td>1</td><td>4</td><td>0.2500</td><td>-0.602</td></tr>
<tr><td>love her</td><td>her | love</td><td>1</td><td>4</td><td>0.2500</td><td>-0.602</td></tr>
<tr><td>love swimming</td><td>swimming | love</td><td>1</td><td>4</td><td>0.2500</td><td>-0.602</td></tr>
<tr><td>me fish</td><td>fish | me</td><td>1</td><td>1</td><td>1.0000</td><td>0.000</td></tr>
<tr><td>prefer fish</td><td>fish | prefer</td><td>1</td><td>1</td><td>1.0000</td><td>0.000</td></tr>
<tr><td>swimming .</td><td>. | swimming</td><td>1</td><td>1</td><td>1.0000</td><td>0.000</td></tr>
<tr><td>they love</td><td>love | they</td><td>1</td><td>1</td><td>1.0000</td><td>0.000</td></tr>
<tr><td>we love</td><td>love | we</td><td>1</td><td>1</td><td>1.0000</td><td>0.000</td></tr>
<tr><td>you prefer</td><td>prefer | you</td><td>1</td><td>1</td><td>1.0000</td><td>0.000</td></tr>
<tr><td colspan="4">Other Expressions</td><td>0.0000</td><td>-99.000</td></tr>
</table>

Training (1): Language Model

s	$w_i \mid w_{i-1}$	$C(s)$	$C(w_{i-1})$	$P(w_i \mid w_{i-1})$	$\log P(w_i \mid w_{i-1})$
. <EOS>	<EOS> | .	5	5	1.0000	0.000
? <EOS>	<EOS> | ?	1	1	1.0000	0.000
<BOS> do	do | <BOS>	1	6	0.1667	-0.778
<BOS> dogs	dogs | <BOS>	1	6	0.1667	-0.778

Bigrams

Probability

Log Probability

$$P(w|w_{i-1}) = \frac{C(w_{i-1}w_i)}{C(w_{i-1})} = \frac{C(s)}{C(w_{i-1})}$$

Training (2): Translation Model

Bilingual Corpus

Bilingual Table

我 喜歡 貓 。 ➔ I love cats .
大象 鍾意 我 。 ➔ Elephants love me .
人們 喜歡 狗 。 ➔ People love dogs .
他們 嗜 魚 。 ➔ They prefer fish .
我們 喜歡 猴子 。 ➔ We love monkeys .
你 喜歡 豬 。 ➔ You prefer pigs .

t	*s*	*C(s and t)*	*C(t)*	*P(s|t)*	*Log P(s|t)*
.	。	6	6	1.0000	0.000
cats	貓	1	1	1.0000	0.000
dogs	狗	1	1	1.0000	0.000
elephants	大象	1	1	1.0000	0.000
fish	魚	1	1	1.0000	0.000
I	我	1	1	1.0000	0.000
prefer	喜歡	1	2	0.5000	-0.301
prefer	嗜	1	2	0.5000	-0.301
love	喜歡	3	4	0.7500	-0.125
love	鍾意	1	4	0.2500	-0.602
me	我	1	1	1.0000	0.000
monkeys	猴子	1	1	1.0000	0.000
people	人們	1	1	1.0000	0.000
pigs	豬	1	1	1.0000	0.000
they	他們	1	1	1.0000	0.000
we	我們	1	1	1.0000	0.000
you	你	1	1	1.0000	0.000
Other word/phrase pairs				0.0000	-99.000

Training (2): Translation Model

t	*s*	*C*(*s* and *t*)	*C*(*t*)	*P*(*s*|*t*)	log *P*(*s*|*t*)
.	。	6	6	1.0000	0.000
cats	貓	1	1	1.0000	0.000
dogs	狗	1	1	1.0000	0.000
elephants	大象	1	1	1.0000	0.000

s and *t*

Probability

Log Probability

$$P(s|t) = \frac{C(s \text{ as the source of } t)}{C(t)} = \frac{C(s \text{ and } t)}{C(t)}$$

SMT(4): Decoding

Decoding

Noisy-channel Model

Log-linear Model

$$\arg\max_T P(S \mid T) \times P(T)$$

$$\arg\max_T \log P(S \mid T) + \log P(T)$$

Translation Model

Language Model

Translation Model

Language Model

Decoding (1): Extracting TL Expressions

狗喜歡大象。

t	s	$C(s \text{ and } t)$	$C(t)$	$P(s\|t)$	$\log P(s\|t)$
.	。	6	6	1.0000	0.000
cats	貓	1	1	1.0000	0.000
dogs	狗	1	1	1.0000	0.000
elephants	大象	1	1	1.0000	0.000
fish	魚	1	1	1.0000	0.000
I	我	1	1	1.0000	0.000
prefer	喜歡	1	2	0.5000	-0.301
prefer	嗜	1	2	0.5000	-0.301
love	喜歡	3	4	0.7500	-0.125
love	鍾意	1	4	0.2500	-0.602
me	我	1	1	1.0000	0.000
monkeys	猴子	1	1	1.0000	0.000
people	人們	1	1	1.0000	0.000
pigs	豬	1	1	1.0000	0.000
they	他們	1	1	1.0000	0.000
we	我們	1	1	1.0000	0.000
you	你	1	1	1.0000	0.000
Other word/phrase pairs				0.0000	-99.000

Decoding (1): Extracting TL Expressions

```
{(“狗”,“dogs”),
(“大象
”,“elephants”),
(“喜歡”,“prefer”),
(“喜歡”,“love”),
(“。”,“.”)}
```

T_1 = dogs elephants love.
T_2 = dogs love elephants.
T_3 = elephants dogs love.
T_4 = elephants love dogs.
T_5 = love elephant dogs.
T_6 = love dog elephants.
T_7 = dogs elephants prefer.
T_8 = dogs prefer elephants.
T_9 = elephants dogs prefer.
T_{10} = elephants prefer dogs.
T_{11} = prefer elephant dogs.
T_{12} = prefer dog elephants.

Decoding (2):
Selecting the Best Combination

{(“狗”,“dogs”),
(“大象”,“elephants”),
(“喜歡”,“prefer”),
(“喜歡”,“love”),
(“。”,“.”)}

~~T_1 = dogs elephants love.~~
T_2 = dogs love elephants.
~~T_3 = elephants dogs love.~~
~~T_4 = elephants love dogs.~~
~~T_5 = love elephant dogs.~~
~~T_6 = love dog elephants.~~
~~T_7 = dogs elephants prefer.~~
T_8 = dogs prefer elephants.
~~T_9 = elephants dogs prefer.~~
~~T_{10} = elephants prefer dogs.~~
~~T_{11} = prefer elephant dogs.~~
~~T_{12} = prefer dog elephants.~~

Decoding (2):
Selecting the Best Combination

Decoding Method 1: Noisy-channel Model

Final Score for a TT Candidate
$= P(S|T) \times P(T)$

Details of Noisy-channel Model

Consider T_2

$P(T_2)$

$= P(\text{“dogs”}|\text{“<BOS>”}) \times P(\text{“love”}|\text{“dogs”}) \times P(\text{“elephants”}|\text{“love”}) \times$
$P(\text{“.”}|\text{“elephants”}) \times P(\text{“<EOS>”}|\text{“.”})$

$= 0.1667 \times 0.5000 \times 0.2500 \times 0.5000 \times 1.0000$

$= 0.01041875$

$P(S|T_2)$

$= P(\text{“狗”}|\text{“dogs”}) \times P(\text{“喜歡”}|\text{“love”}) \times P(\text{“大象”}|\text{“elephants”}) \times P(\text{“。”}|\text{“.”})$

$= 1.0000 \times 0.7500 \times 1.0000 \times 1.0000$

$= 0.75$

Final Score for T_2

$= P(T_2) \times P(S|T_2)$

$= 0.01041875 \times 0.75$

$= 0.0078140625$

T_2 = dogs love elephants.

Details of Noisy-channel Model

Consider T_8

$P(T_8)$

= P(“dogs”|“<BOS>”) × P(“prefer”|“dogs”) × P(“elephants”|“prefer”) ×
P(“.”|“elephants”) × P(“<EOS>”|“.”)

= 0.1667 × 0.0000 × 0.0000 × 0.5000 × 1.0000

= 0

$P(S|T_8)$

= P(“狗”|“dogs”)×P(“喜歡”|“prefer”)×P(“大象”|“elephants”)×P(“。”|“.”)

= 1.0000 × 0.5000 × 1.0000 × 1.0000

= 0.5

Final Score for T_8

= $P(T_8) \times P(S|T_8)$

= 0 × 0.5

= 0

T_8 = dogs prefer elephants.

Decoding (2): Selecting the Best Combination

Decoding Method 2: Log-linear Model / Maximum Entropy Model (Simplified)

Final Score for a TT Candidate

$= \log P(S|T) + \log P(T)$

Details of Log-linear Model

Consider T_2

log $P(T_2)$

= *log* (P(“dogs”|“<BOS>”) × P(“love”|“dogs”) × P(“elephants”|“love”) ×
P(“.”|“elephants”) × P(“<EOS>”|“.”))

= *log* P(“dogs”|“<BOS>”) + *log* P(“love”|“dogs”) + *log* P(“elephants”|“love”) +
log P(“.”|“elephants”) + *log* P(“<EOS>”|“.”)

= -0.778 + (-0.301) + (-0.602) + (-0.301) + (0.000)

= -1.982

log $P(S|T_2)$

= *log* (P(“狗”|“dogs”)×P(“喜歡”|“love”)×P(“大象”|“elephants”)×P(“。”|“.”))

= *log* P(“狗”|“dogs”)+*log* P(“喜歡”|“love”)
+*log* P(“大象”|“elephants”)+*log* P(“。”|“.”)

= 0.000 + 0.000 + (-0.125) + 0.000

= -0.125

Final Score for T_2

= *log* $P(T_2)$ + *log* $P(S|T_2)$

= -2.107

T_2 = dogs love elephants.

Details of Log-linear Model

Consider T_8

log $P(T_8)$

= *log* (P(“dogs”|“<BOS>”) × P(“prefer”|“dogs”) × P(“elephants”|“prefer”) × P(“.”|“elephants”) × P(“<EOS>”|“.”))

= *log* P(“dogs”|“<BOS>”) + *log* P(“prefer”|“dogs”) + *log* P(“elephants”|“prefer”) + *log* P(“.”|“elephants”) + *log* P(“<EOS>”|“.”)

= -0.778 + (-99.000) + (-99.000) + (-0.301) + (0.000)

= -199.079

log $P(S|T_8)$

= *log* (P(“狗”|“dogs”)×P(“喜歡”|“prefer”)×P(“大象”|“elephants”)×P(“。”|“.”))

= *log* P(“狗”|“dogs”)+*log* P(“喜歡”|“prefer”) +*log* P(“大象”|“elephants”)+*log* P(“。”|“.”)

= 0.000 + (-0.301) + 0.000 + 0.000

= -0.301

Final Score for T_8

= *log* $P(T_8)$ + *log* $P(S|T_8)$

= -199.38

T_8 = dogs prefer elephants.

More about NMT

NMT(1): Artificial Neural Networks

More about Artificial Neural Networks

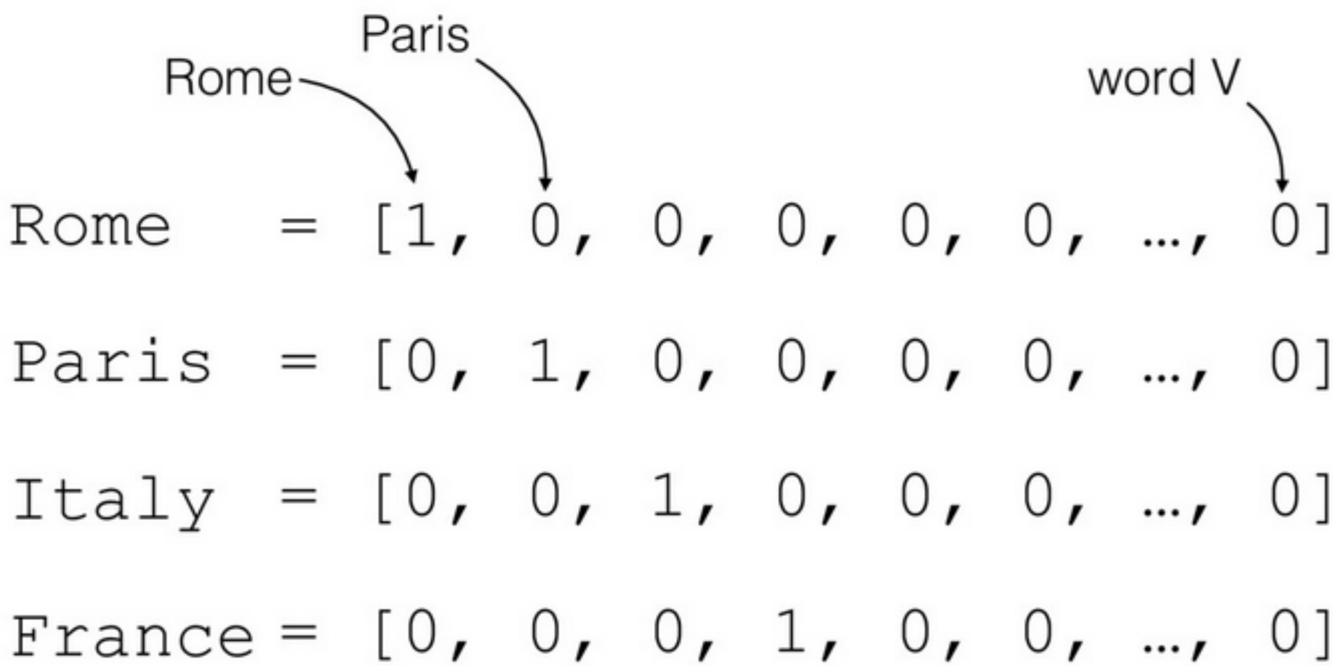
Neural networks and back-propagation explained in a simple way (https://medium.com/datathings/neural-networks-and-backpropagation-explained-in-a-simple-way-f540a3611f5e)

More about Artificial Neural Networks

Back-Propagation is very simple. Who made it Complicated? (https://medium.com/@14prakash/back-propagation-is-very-simple-who-made-it-complicated-97b794c97e5c)

NMT(2): Word Embedding

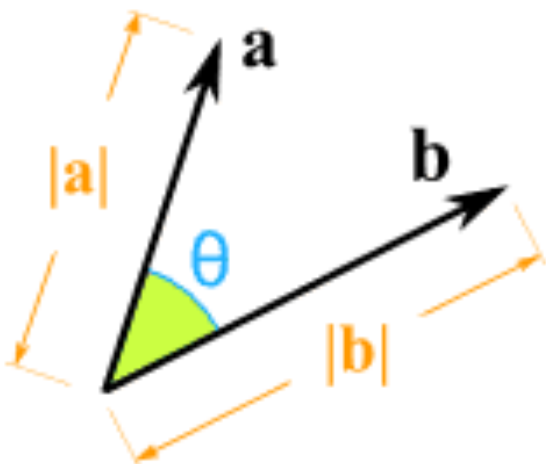
Word Embedding (1): One-hot Vector

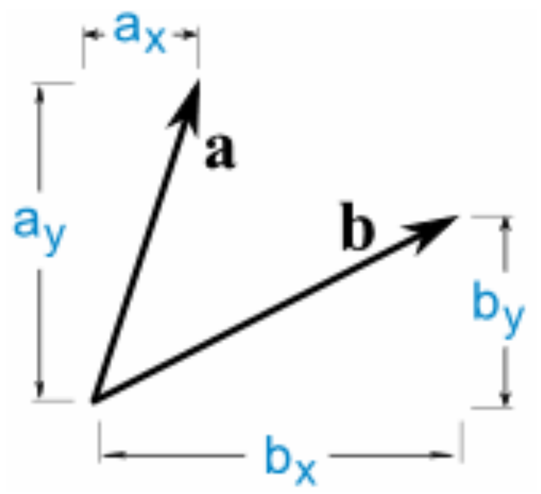


For more information, visit https://medium.com/@athif.shaffy/one-hot-encoding-of-text-b69124bef0a7

Problem

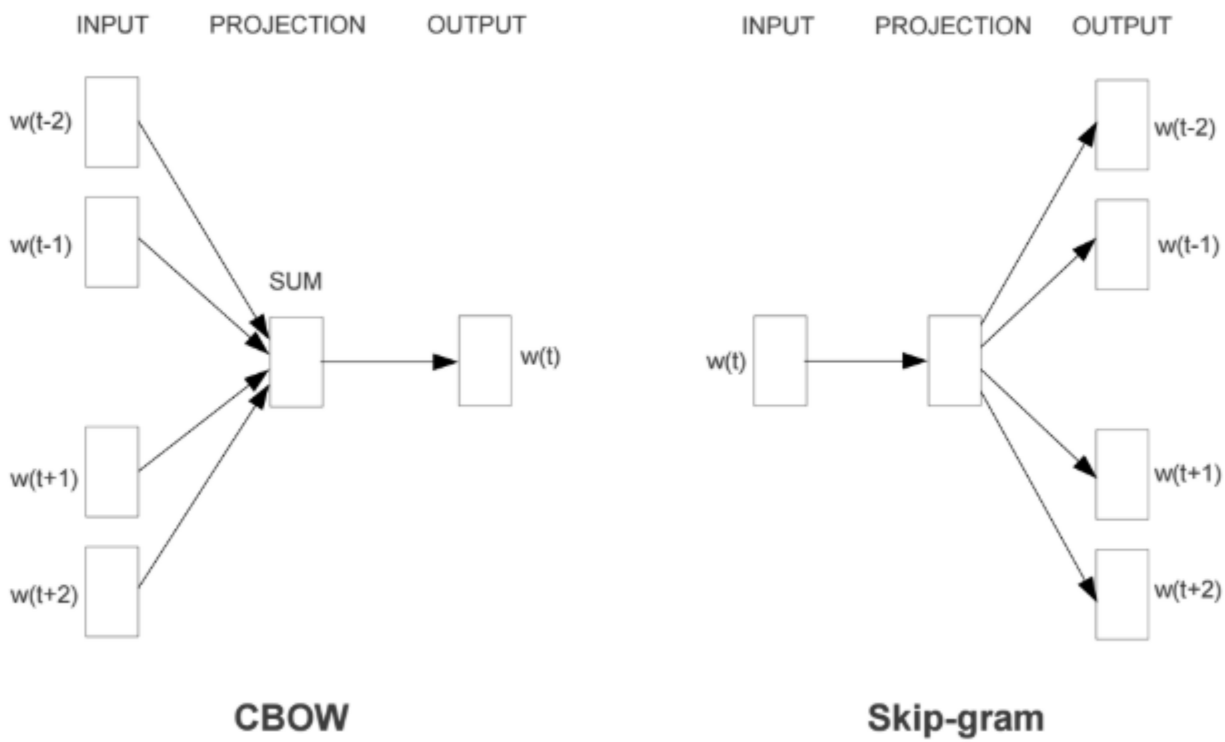
$$a \cdot b = a_x \times b_x + a_y \times b_y = |a||b|\cos(\theta)$$





For more information, visit https://www.mathsisfun.com/algebra/vectors-dot-product.html

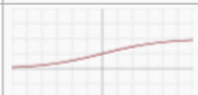
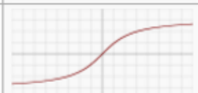
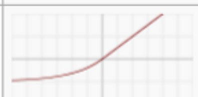
Word Embedding (2): Continuous Bag of Words and Skip-gram



For more information, visit
https://mubaris.com/2017/12/14/word2vec/

NMT(3): Activation functions

Activation functions

Name	Plot	Equation	Derivative
Identity		$f(x)=x$	$f'(x)=1$
Binary step		$f(x)=\begin{cases}0 & \text{for } x<0\\ 1 & \text{for } x\geq 0\end{cases}$	$f'(x)=\begin{cases}0 & \text{for } x\neq 0\\ ? & \text{for } x=0\end{cases}$
Logistic (a.k.a Soft step)		$f(x)=\frac{1}{1+e^{-x}}$	$f'(x)=f(x)(1-f(x))$
TanH		$f(x)=\tanh(x)=\frac{2}{1+e^{-2x}}-1$	$f'(x)=1-f(x)^2$
ArcTan		$f(x)=\tan^{-1}(x)$	$f'(x)=\frac{1}{x^2+1}$
Rectified Linear Unit (ReLU)		$f(x)=\begin{cases}0 & \text{for } x<0\\ x & \text{for } x\geq 0\end{cases}$	$f'(x)=\begin{cases}0 & \text{for } x<0\\ 1 & \text{for } x\geq 0\end{cases}$
Parameteric Rectified Linear Unit (PReLU)[2]		$f(x)=\begin{cases}\alpha x & \text{for } x<0\\ x & \text{for } x\geq 0\end{cases}$	$f'(x)=\begin{cases}\alpha & \text{for } x<0\\ 1 & \text{for } x\geq 0\end{cases}$
Exponential Linear Unit (ELU)[3]		$f(x)=\begin{cases}\alpha(e^x-1) & \text{for } x<0\\ x & \text{for } x\geq 0\end{cases}$	$f'(x)=\begin{cases}f(x)+\alpha & \text{for } x<0\\ 1 & \text{for } x\geq 0\end{cases}$
SoftPlus		$f(x)=\log_e(1+e^x)$	$f'(x)=\frac{1}{1+e^{-x}}$

For more information, visit
https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6

Softmax for Multi-class Classification (1)

The softmax function is a generalization of the logistic function that "squashes" a K-dimensional vector **z** of arbitrary real values to a K-dimensional vector of real values in the range [0, 1] that add up to 1.

$$\text{logits} \quad \begin{bmatrix} z_0 \\ z_1 \\ \vdots \\ z_n \end{bmatrix} \xrightarrow{\text{Sigmoid}} \begin{bmatrix} \frac{1}{1+\exp(-z_0)} \\ \frac{1}{1+\exp(-z_1)} \\ \vdots \\ \frac{1}{1+\exp(-z_n)} \end{bmatrix} \quad \text{independent probabilites}$$

[1.0,
2.0,
3.0,
4.0,
1.0]

[0.031062774127550943,
0.0844373744524495,
0.22952458061688552,
0.623912496675563,
0.031062774127550943]

For more information, visit
https://www.depends-on-the-definition.com/guide-to-multi-label-classification-with-neural-networks/

Softmax for Multi-class Classification (2)

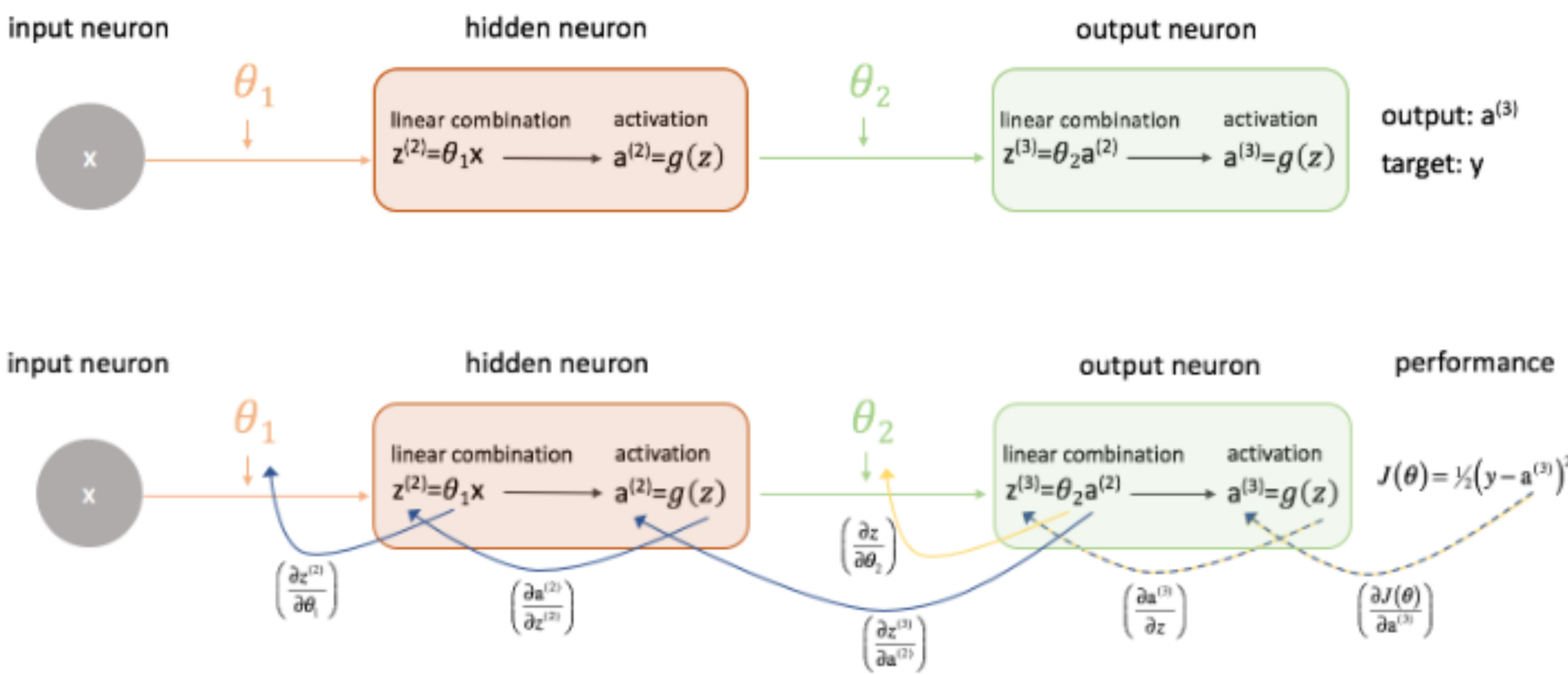
Softmax function takes an N-dimensional vector of real numbers and transforms it into a vector of real number in range (0,1) which add up to 1.

$$\begin{aligned} p_j &= \frac{e^{a_i}}{\sum_{k=1}^{N} e^{a_k}} \\ &= \frac{Ce^{a_i}}{C\sum_{k=1}^{N} e^{a_k}} \\ &= \frac{e^{a_i+\log(C)}}{\sum_{k=1}^{N} e^{a_k+\log(C)}} \end{aligned}$$

For more information, visit
https://deepnotes.io/softmax-crossentropy

NMT(4): Training and Backpropagation

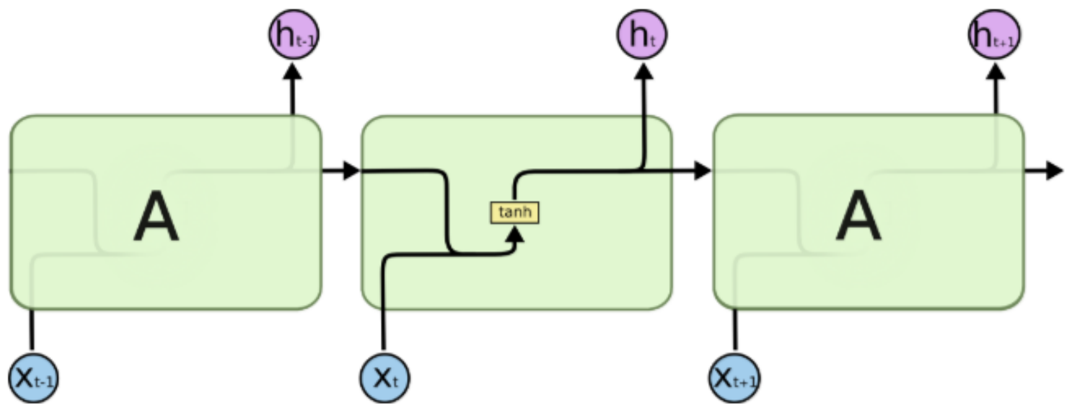
Training and Backpropagation

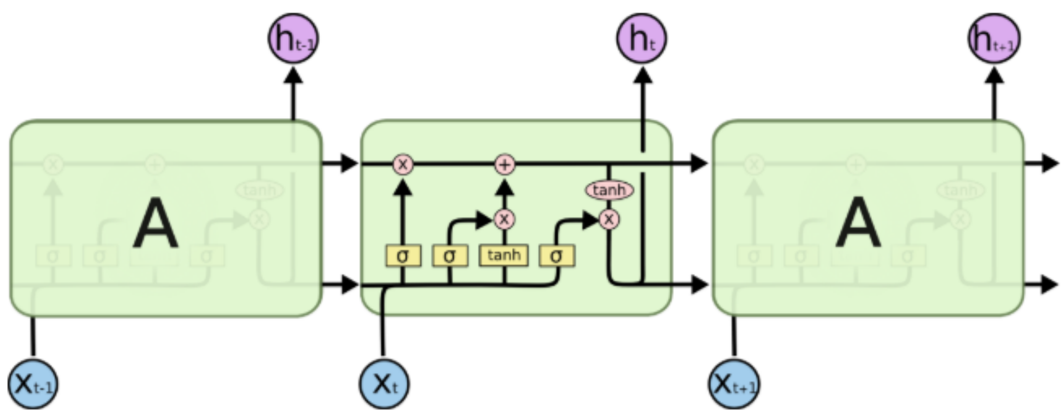


For more information, visit
https://www.jeremyjordan.me/neural-networks-training/

NMT(5): RNN and LSTM

Standard RNN VS Long Short-term Memory





For more information, visit
http://colah.github.io/posts/2015-08-Understanding-LSTMs/

NMT(6): Types of NMT

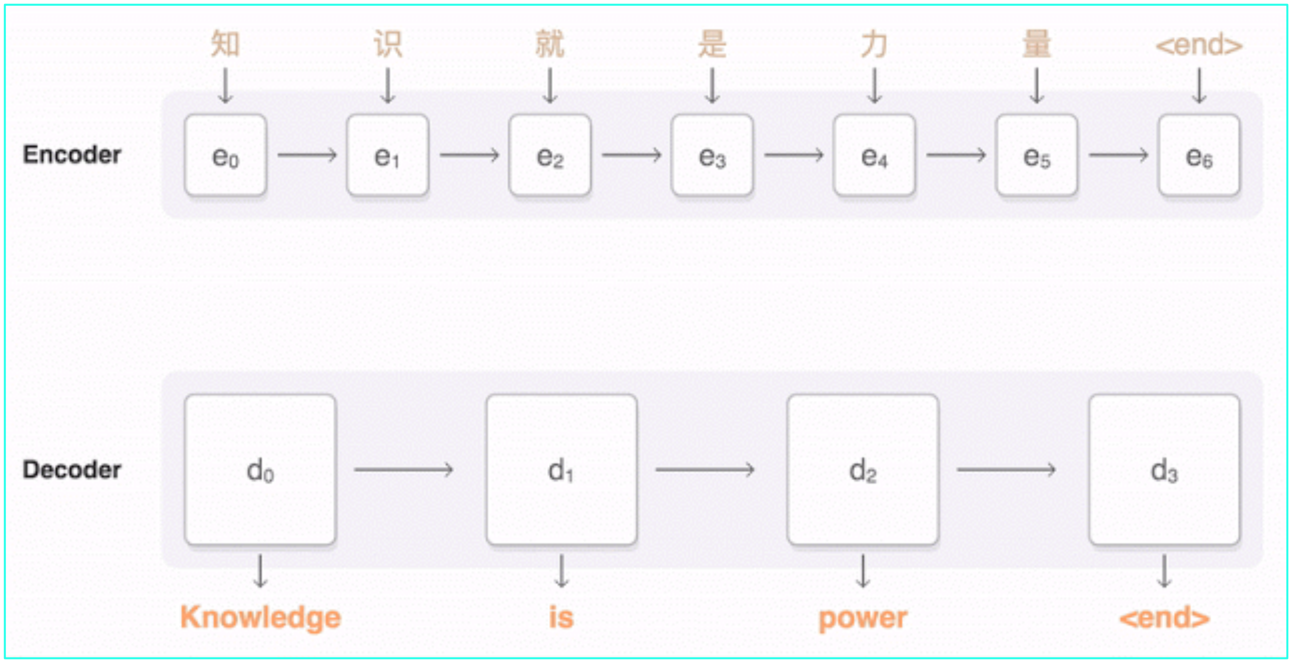
Attention Mechanism

Given **[I]** / [go] / [to] / [school] / [by] / [bus] / [.], what is the next word?
我
[I] / [go] / [to] / [school] / **[by]** / [bus] / [.] / [我] ⇒ ?
乘
[I] / [go] / [to] / [school] / [by] / **[bus]** / [.] / [我] / [乘] ⇒ ?
公車
[I] / **[go]** / **[to]** / [school] / [by] / [bus] / [.] / [我] / [乘] / [公車] ⇒ ?
去
[I] / [go] / [to] / **[school]** / [by] / [bus] / [.] / [我] / [乘] / [公車] / [去] ⇒ ?
學校
[I] / [go] / [to] / [school] / [by] / [bus] / **[.]** / [我] / [乘] / [公車] / [去] / [學校] / ⇒ ?
。

Translation: 我乘公車去學校。

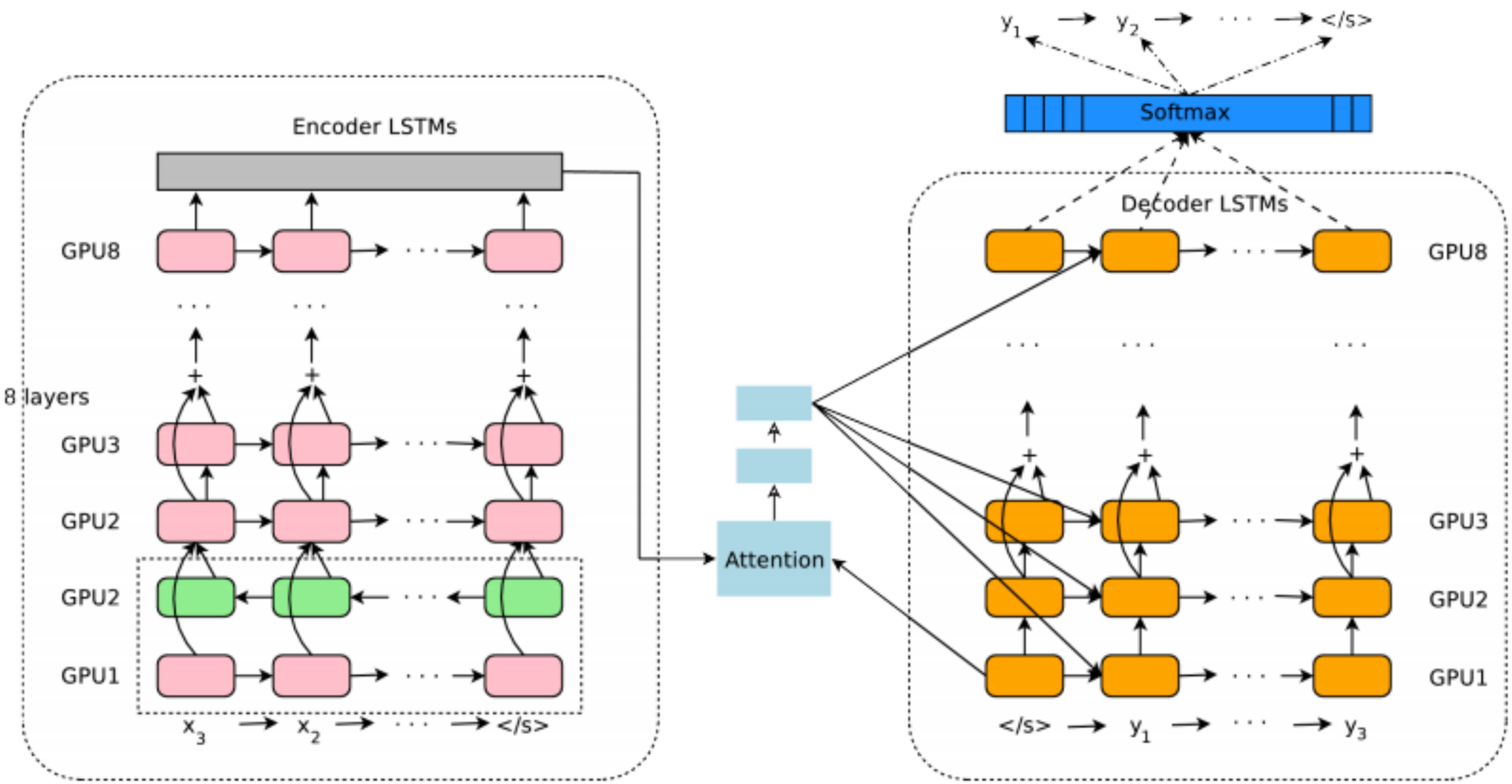
Note: [ ] refers to internal representation

RNN Encoder-decoder with Attention



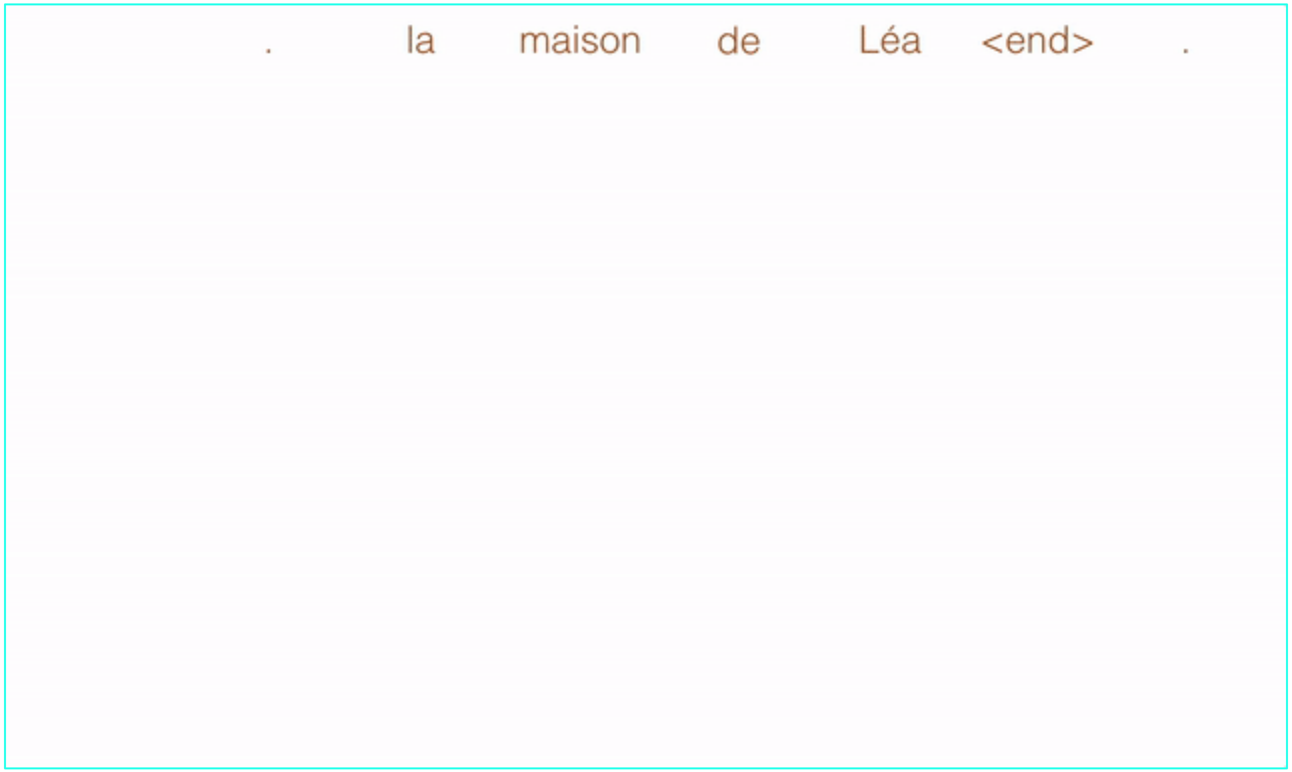
For more information, visit
https://ai.googleblog.com/2016/09/a-neural-network-for-machine.html

RNN Encoder-decoder with Attention



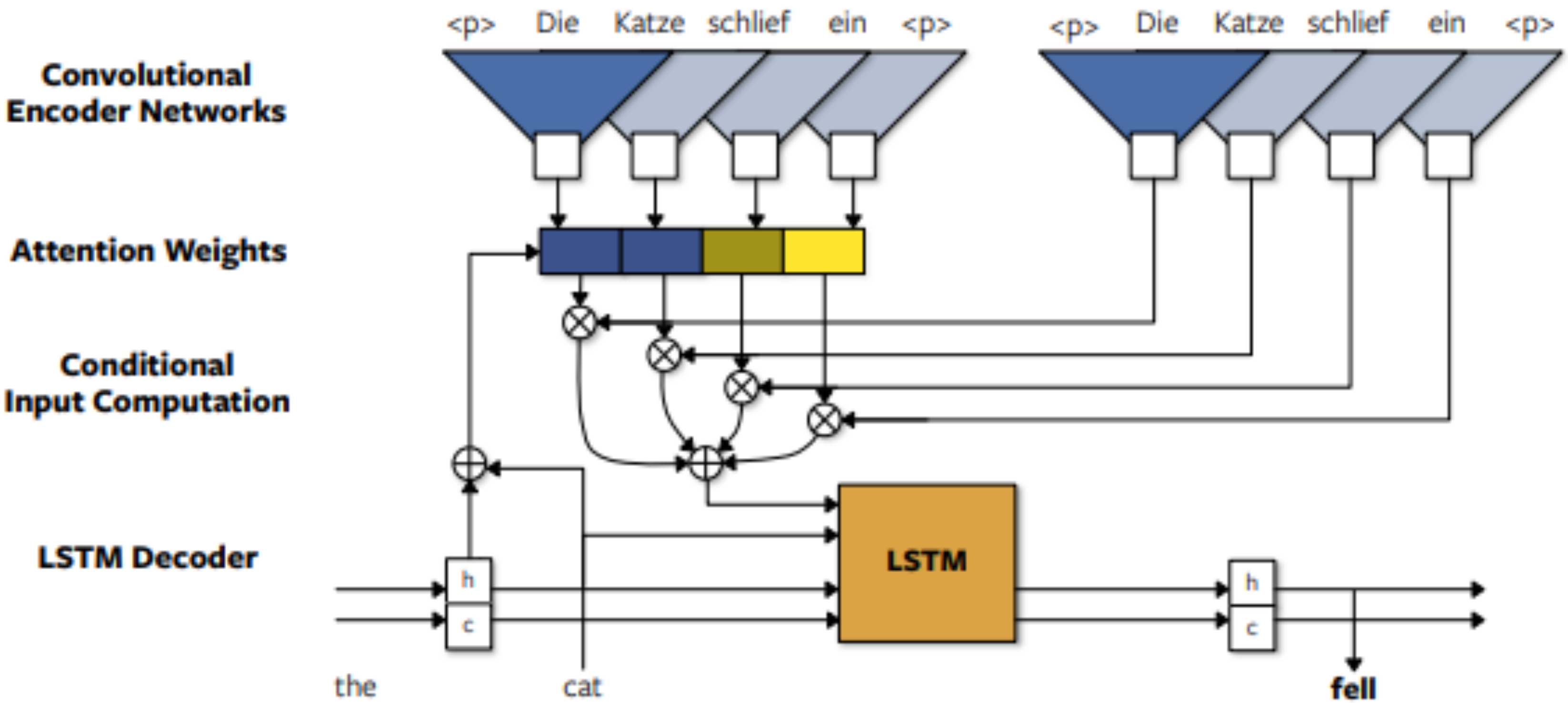
For more information, visit
https://arxiv.org/pdf/1609.08144.pdf

Convolutional NMT



For more information, visit https://code.facebook.com/posts/1978007565818999/a-novel-approach-to-neural-machine-translation/

Convolutional NMT

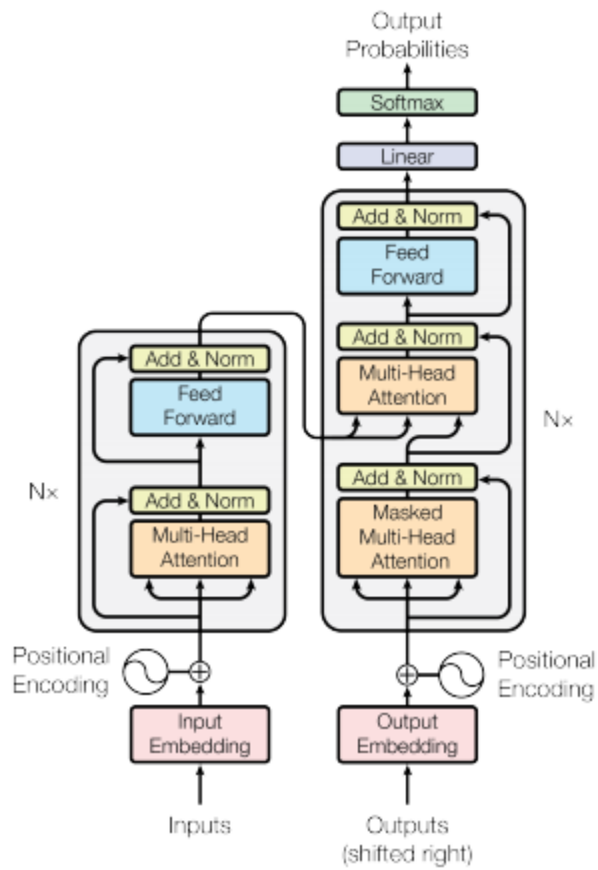


For more information, visit
https://arxiv.org/pdf/1611.02344.pdf

Self-attentional Transformer

For more information, visit
https://ai.googleblog.com/2017/08/transformer-novel-neural-network.html

Self-attentional Transformer



For more information, visit
https://arxiv.org/pdf/1706.03762.pdf