

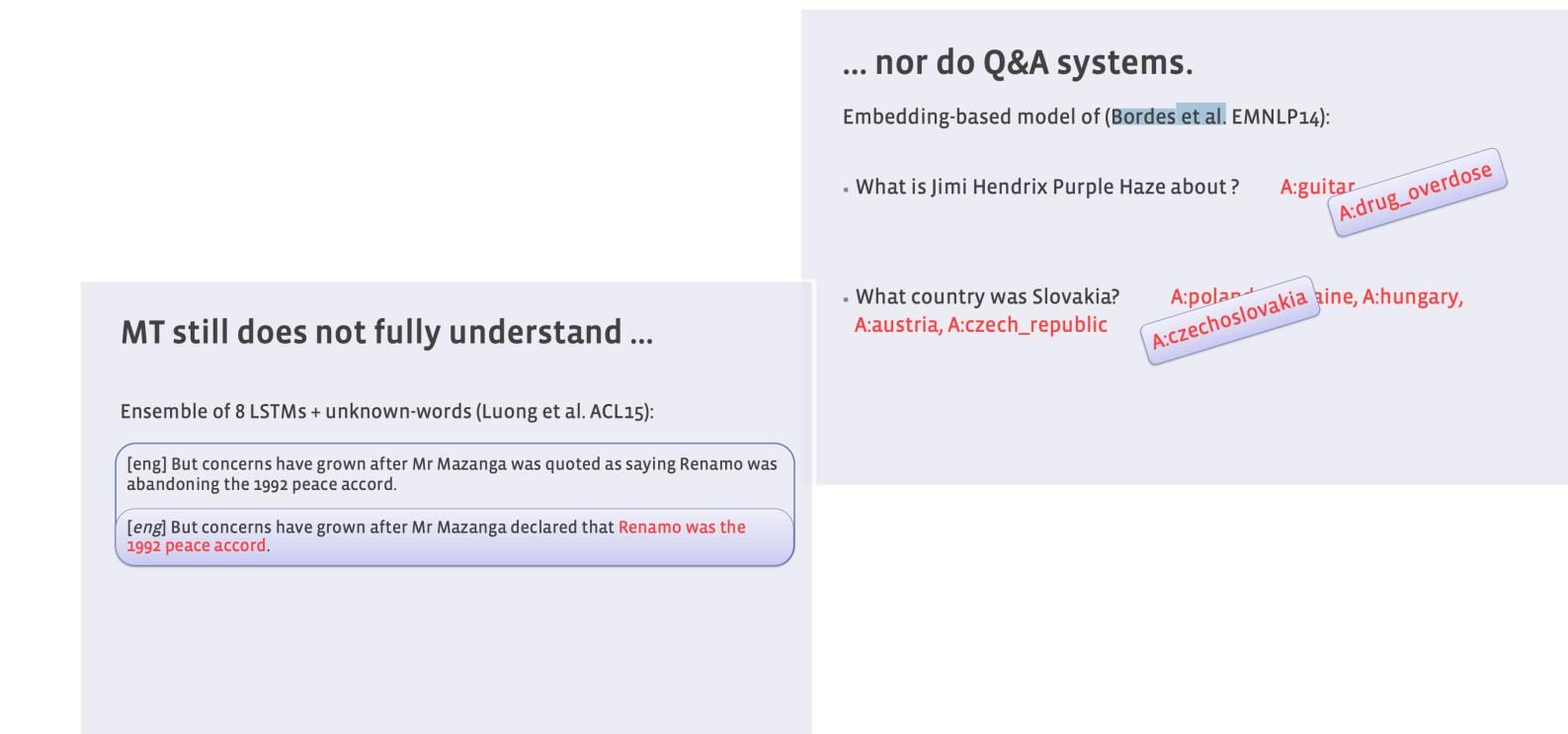
**SPRINT 24**

# 言語処理の応用タスク

翻訳、質問応答、文書分類、文書要約など

Deep learning関連技術において上記のような言語処理の応用タスクの研究が進められていますが、その中でもまずははじめに「翻訳」が発展しました。その重要な背景として、翻訳のためのデータセットの整備が進んでいたことが挙げられます。

2014年に登場したSequence to SequenceやAttentionによって、翻訳はある程度の精度を得られるようになりました。



その後の2015～2017年には、質問応答のためのデータセットの整備が進められました。2018年には一般言語理解のためのデータセットGLUEが登場しました。これは事前学習モデルにファインチューニングを施し、マルチタスクを解かせ成績を評価するベンチマークデータセットです。

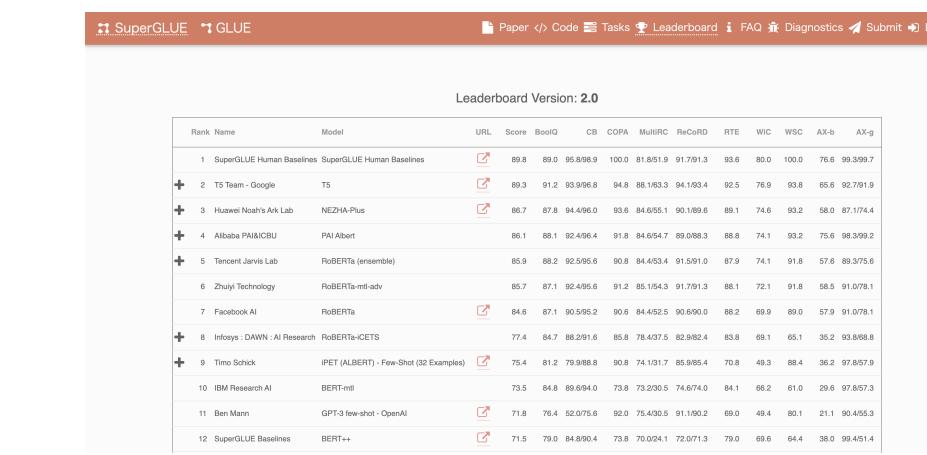
<http://www.thespermwhale.com/jaseweston/babi/abordes-ICLR.pdf>

# 言語処理のベンチマークデータセット

ImageNet を用いた 1,000 クラス画像分類タスクによって

事前学習 (教師あり学習) を行なった画像認識とは対照的に

言語処理における事前学習は Wikipedia などの文章を用いた教師なし学習である。



The screenshot shows the SuperGLUE leaderboard version 2.0. The table lists 12 models, each with their name, URL, and scores across 11 different metrics: CB, CQA, MultiRC, ReCoRD, RTE, WIC, WSC, AX-b, AX-g, and AX-s. The top-ranked model is SuperGLUE Human Baseline, followed by T5 and NEZHA-Plus.

Rank	Name	Model	URL	Score	BioQ	CB	CQA	MultiRC	ReCoRD	RTE	WIC	WSC	AX-b	AX-g
1	SuperGLUE Human Baseline	SuperGLUE Human Baseline	<a href="#">🔗</a>	89.8	89.0	95.989.9	100.0	91.651.9	91.719.3	89.4	80.0	100.0	76.6	93.990.7
2	T5 Team - Google	T5	<a href="#">🔗</a>	89.3	91.2	93.916.8	94.8	88.163.3	94.150.4	85.3	76.9	93.8	65.6	92.791.9
3	Huawei Noah's Ark Lab	NEZHA-Plus	<a href="#">🔗</a>	86.7	87.8	94.046.0	95.6	84.055.1	90.169.6	86.1	74.6	93.2	66.0	91.714.4
4	Abibab PAIR-LU	PALBERT	<a href="#">🔗</a>	86.1	86.1	92.494.6	91.8	84.654.7	89.008.3	86.4	74.1	93.2	75.6	89.399.2
5	Tencent Javis Lab	RoBERTS (ensemble)	<a href="#">🔗</a>	85.9	88.2	92.505.6	90.8	84.450.4	91.501.0	87.4	74.1	91.8	77.6	89.371.6
6	Zhiyu Technology	RoBERTS-mt-adv	<a href="#">🔗</a>	85.7	87.1	92.455.6	91.2	85.154.3	91.719.3	86.1	73.1	91.8	66.3	91.076.1
7	Facebook AI	RoBERTS	<a href="#">🔗</a>	84.6	87.1	93.505.2	90.8	84.450.2	90.000.0	86.2	69.9	89.2	77.9	91.070.1
8	Mitoya - DAWN-AI Research	RoBERTa-ICET5	<a href="#">🔗</a>	77.4	84.7	88.291.8	85.8	78.437.5	82.882.4	82.4	68.1	85.1	56.2	93.899.8
9	Timo Schick	IPET (ALBERT) - Few-Shot (22 Examples)	<a href="#">🔗</a>	75.4	81.2	79.988.8	90.8	74.101.7	85.864.4	70.8	68.3	86.4	56.2	97.857.9
10	BBM Research AI	BERT-mn	<a href="#">🔗</a>	73.5	84.8	89.694.0	73.8	73.203.5	74.674.0	84.1	68.2	81.0	29.6	97.857.3
11	Ben Mann	GPT-3 few-shot - OpenAI	<a href="#">🔗</a>	71.8	76.4	82.075.6	92.0	79.403.5	91.106.2	69.0	68.4	80.1	21.1	94.460.3
12	SuperGLUE Baseline	BERT++	<a href="#">🔗</a>	71.5	79.0	84.890.4	73.8	70.294.1	72.571.3	79.0	68.6	64.4	28.0	94.461.4

質問応答：

bAbI

<https://research.fb.com/downloads/babi/>

**RACE Reading Comprehension Dataset**

[http://www.qizhexie.com/data/RACE\\_leaderboard.html](http://www.qizhexie.com/data/RACE_leaderboard.html)

**SQuAD Stanford Question Answering Dataset**

<https://rajpurkar.github.io/SQuAD-explorer/>

一般言語理解：

**GLUE**

<https://gluebenchmark.com/leaderboard>

**Super GLUE**

<https://super.gluebenchmark.com/leaderboard>

# 最近のモデルのパラメータ数はどうなってる？

<https://arxiv.org/pdf/1910.01108.pdf>

<https://www.microsoft.com/en-us/research/blog/turing-nlg-a-17-billion-parameter-language-model-by-microsoft/>

<https://towardsdatascience.com/gpt-3-the-new-mighty-language-model-from-openai-a74ff35346fc>

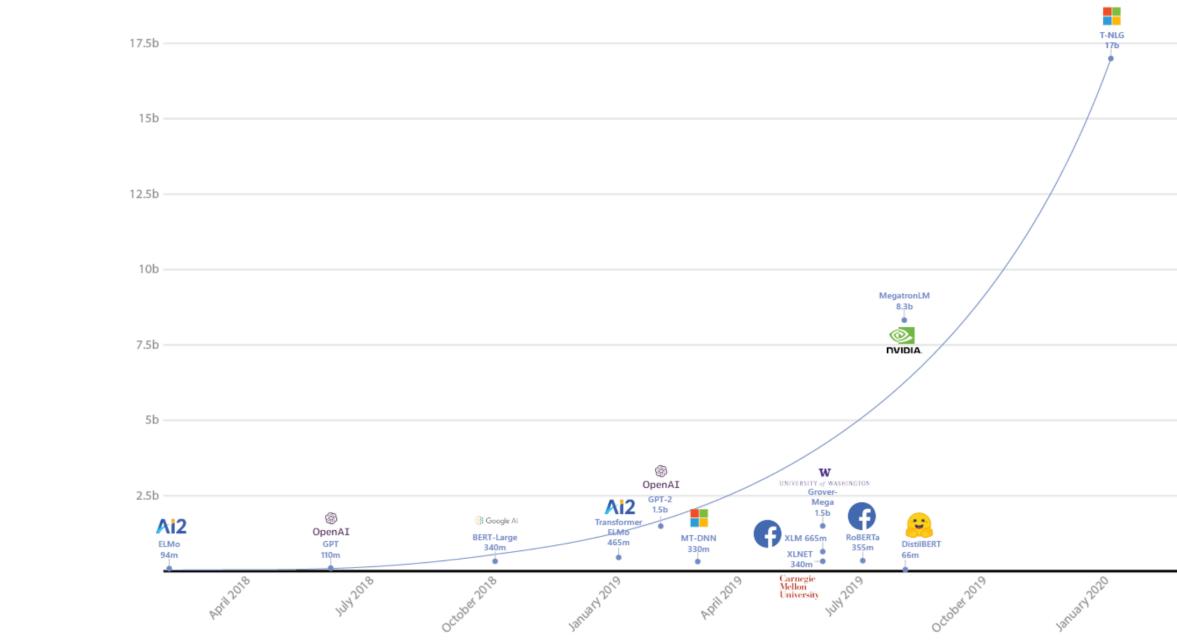
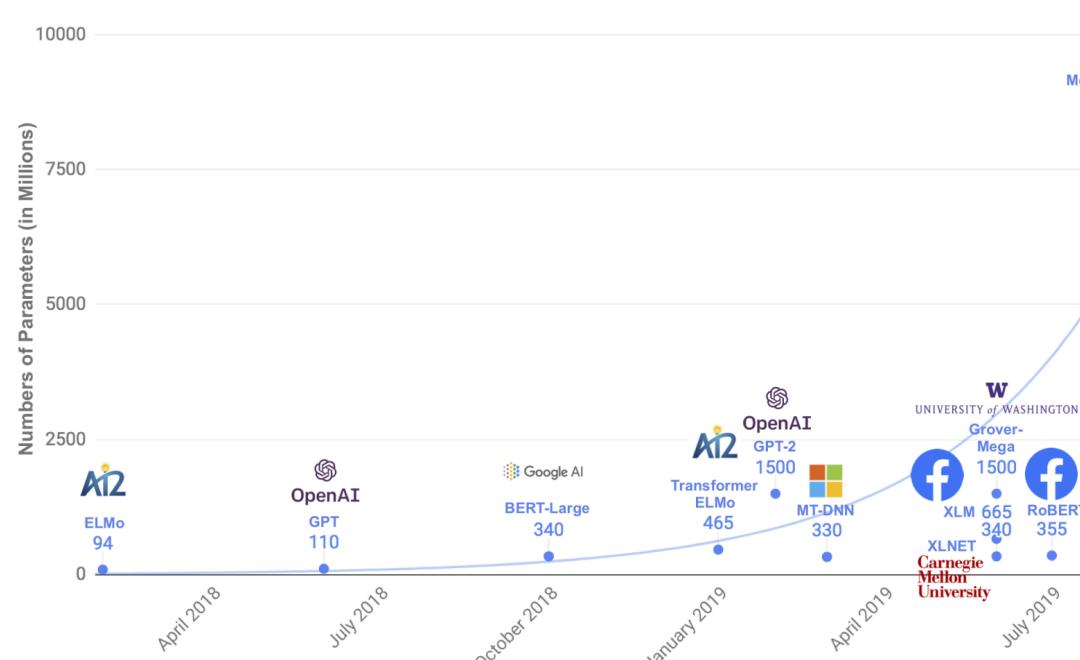
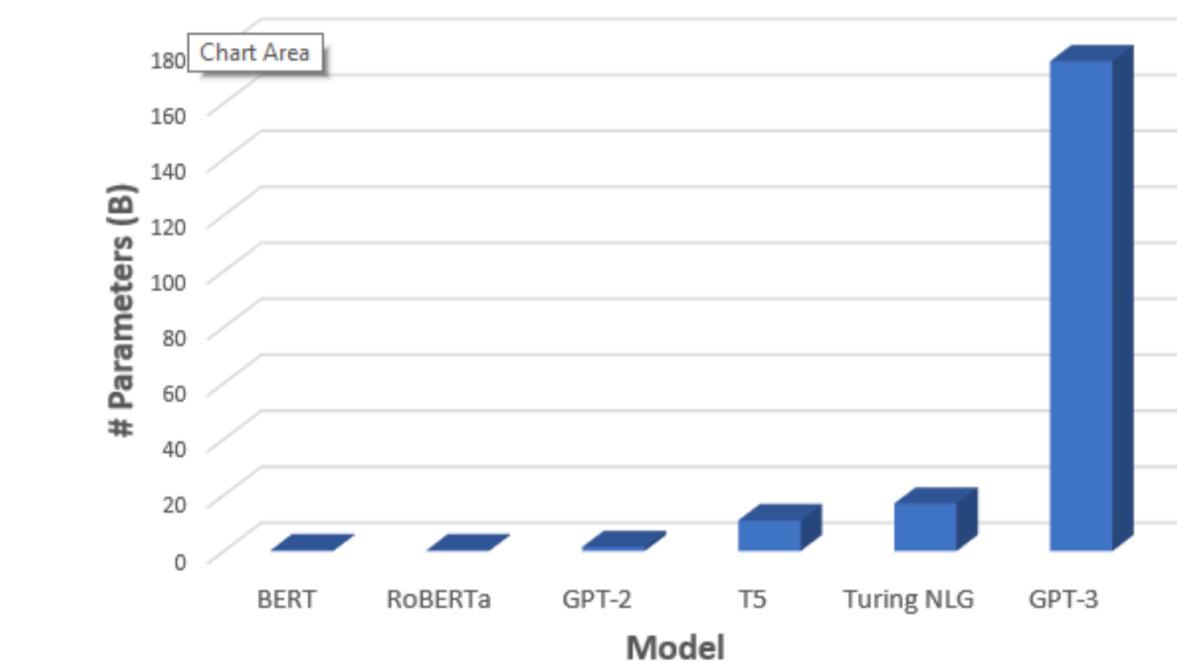
BERT - Large : 3.4億

RoBERTa : 3.5億

DistilBERT : 6600万

T - NLG : 170億

GPT - 3 : 1750億



# seq2seq

Sequence to Sequence

系列変換モデル

シーケンストゥシーケンス (seq2seq) モデル :

2014年発表

<https://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.pdf>

適用領域 :

機械翻訳、音声認識、テキスト要約など

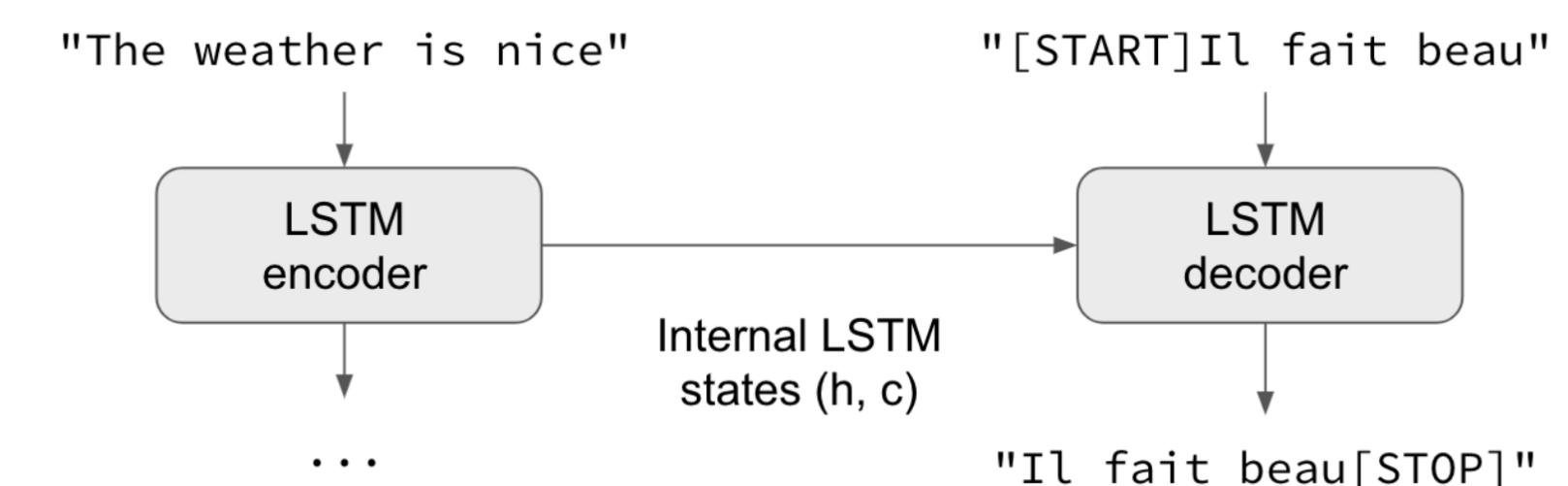
最初のタスク :

Neural Machine Translation (NMT)

※seq2seqの登場により、Google翻訳の精度が向上し、SMT（統計的機械翻訳）から NMT(ニューラル機械翻訳) に研究トレンドが変わった。

3つの主要コンポーネント (オートエンコーダ的) :

- エンコーダ
- 中間ベクトル
- デコーダ



<https://nextjournal.com/gkoehler/machine-translation-seq2seq-cpu>

[余談]

# オートエンコーダとは

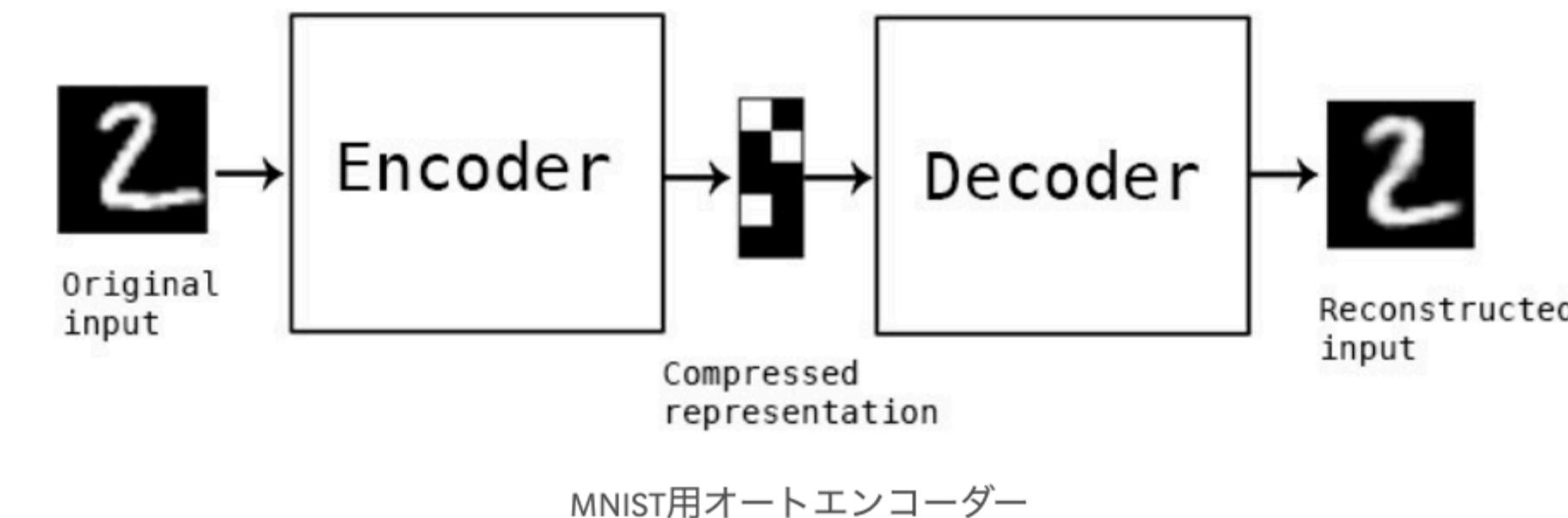
AutoEncoder (自己符号化器)

オートエンコーダのネットワークアーキテクチャに登場するコンポーネントは、エンコーダー（符号化器）とデコーダー（複合化器）と呼ばれる。

参考：

<https://towardsdatascience.com/auto-encoder-what-is-it-and-what-is-it-used-for-part-1-3e5c6f017726>

<https://cs.stanford.edu/people/karpathy/convnetjs/demo/autoencoder.html>



エンコーダー：入力次元を縮小し、入力データをエンコードされた表現に圧縮する方法をモデルが学習します。

ボトルネック：入力データの圧縮表現を含むレイヤーです。これは、入力データの最小可能次元です。

デコーダー：モデルが、可能な限り元の入力に近くなるように符号化表現からデータを再構成する方法を学習する。

# seq2seq

系列を受けとり、別の系列へ変換し、確率を出力するモデル

$$p(y_1, \dots, y_{T'} | x_1, \dots, x_T) = \prod_{t=1}^{T'} p(y_t | v, y_1, \dots, y_{t-1})$$

上の式はseq2seqの数式表現です。入力系列の  $x$  が与えられ

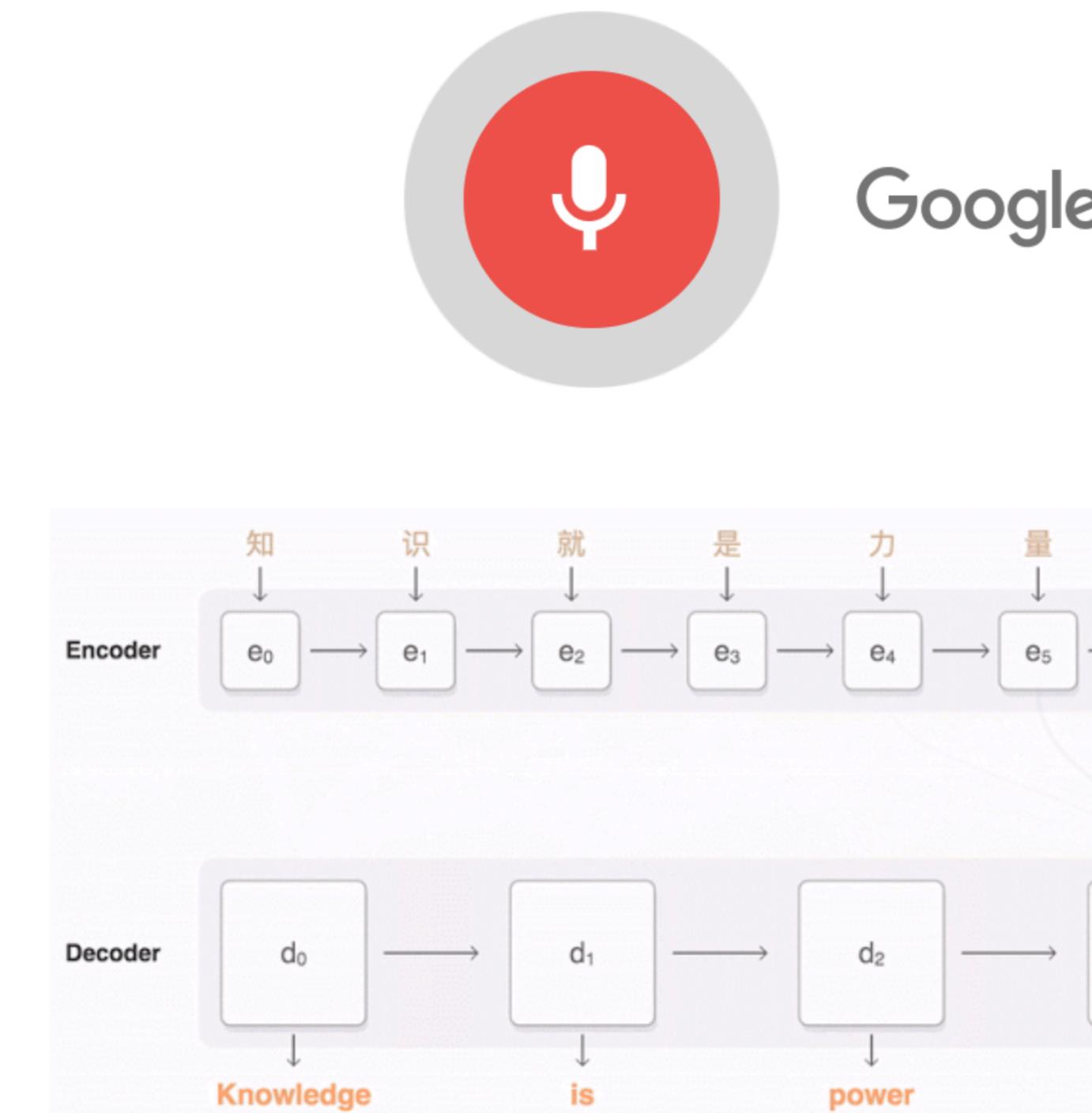
た上で（条件付き）、 $y$  が得られる確率をモデル化します。

$x$  を英語、 $y$  を日本語とすると、「和訳」というタスクをこ

なせます。

RNNの出力ベクトル：

$$\begin{aligned} h_t &= \text{sigm}(W^{\text{hx}}x_t + W^{\text{hh}}h_{t-1}) \\ y_t &= W^{\text{yh}}h_t \end{aligned}$$



適用例としては、Google翻訳、音声対応デバイス、オンラインチャットボットなどのアプリケーション、ビデオキャプションなどがあります。

<https://ai.googleblog.com/2016/09/a-neural-network-for-machine.html>

# seq2seq

ベンチマークテストはWMT-14を用い、英語からフランス語への翻訳タスクを解いた

[https://www.tensorflow.org/datasets/catalog/wmt14\\_translate](https://www.tensorflow.org/datasets/catalog/wmt14_translate)

seq2seqでは、入力系列と出力系列に異なる2つのLSTMを  
それぞれ**Encoder**と**Decoder**として用いました。

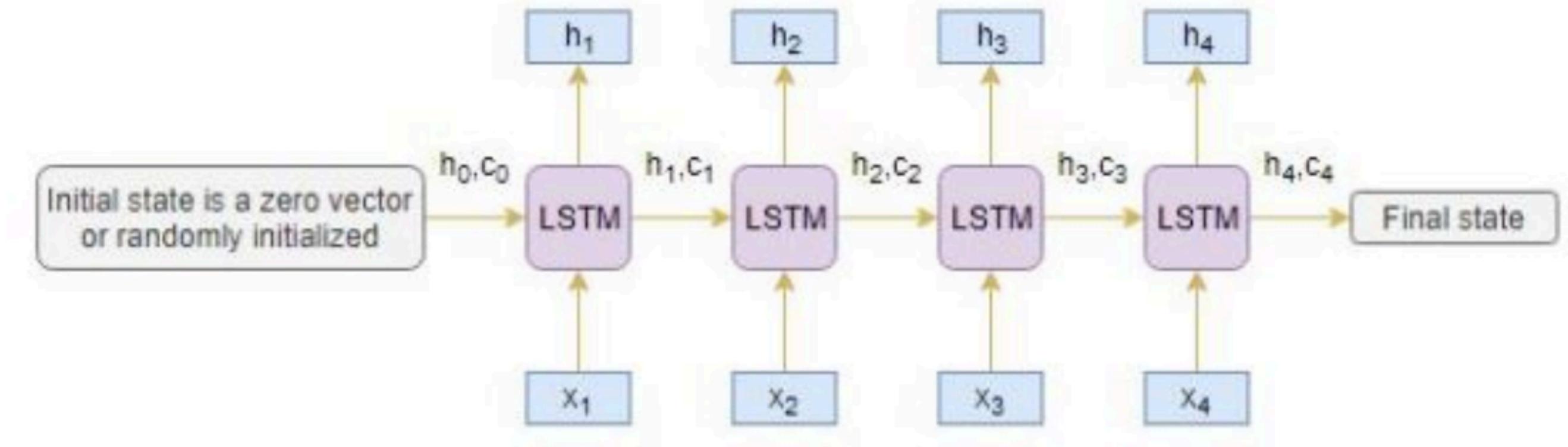
このアイデアにより、系列にまつわる問題、すなわち**入力と出力のサイズとカテゴリが異なる問題**を解決してくれました。

例えば、“What are you doing today?” を英語から中国語に翻訳したい場合は、5つの単語の入力と7つの記号の出力（今天你在做什麼？）が必要となります。

この場合、単一のLSTMネットワークを使用して各単語を英語の文から中国語の文にマッピングすることはできません。

# seq2seqのEncoder

$P(w_2|w_1) \ P(w_3|w_1w_2) \ P(w_4|w_1 \dots w_3) \ P(w_5|w_1 \dots w_4)$



入力ベクトル：

(機械翻訳ならば、元の言語の単語のベクトル)

出力ベクトル：

(隠れ状態 & セル状態)

最後のタイムステップの隠れ状態 ( $h_i$ ) およびセル状態 ( $c_i$ ) は、Decoderの短期記憶・長期記憶を初期化するために使用される。

# seq2seqのDecoder

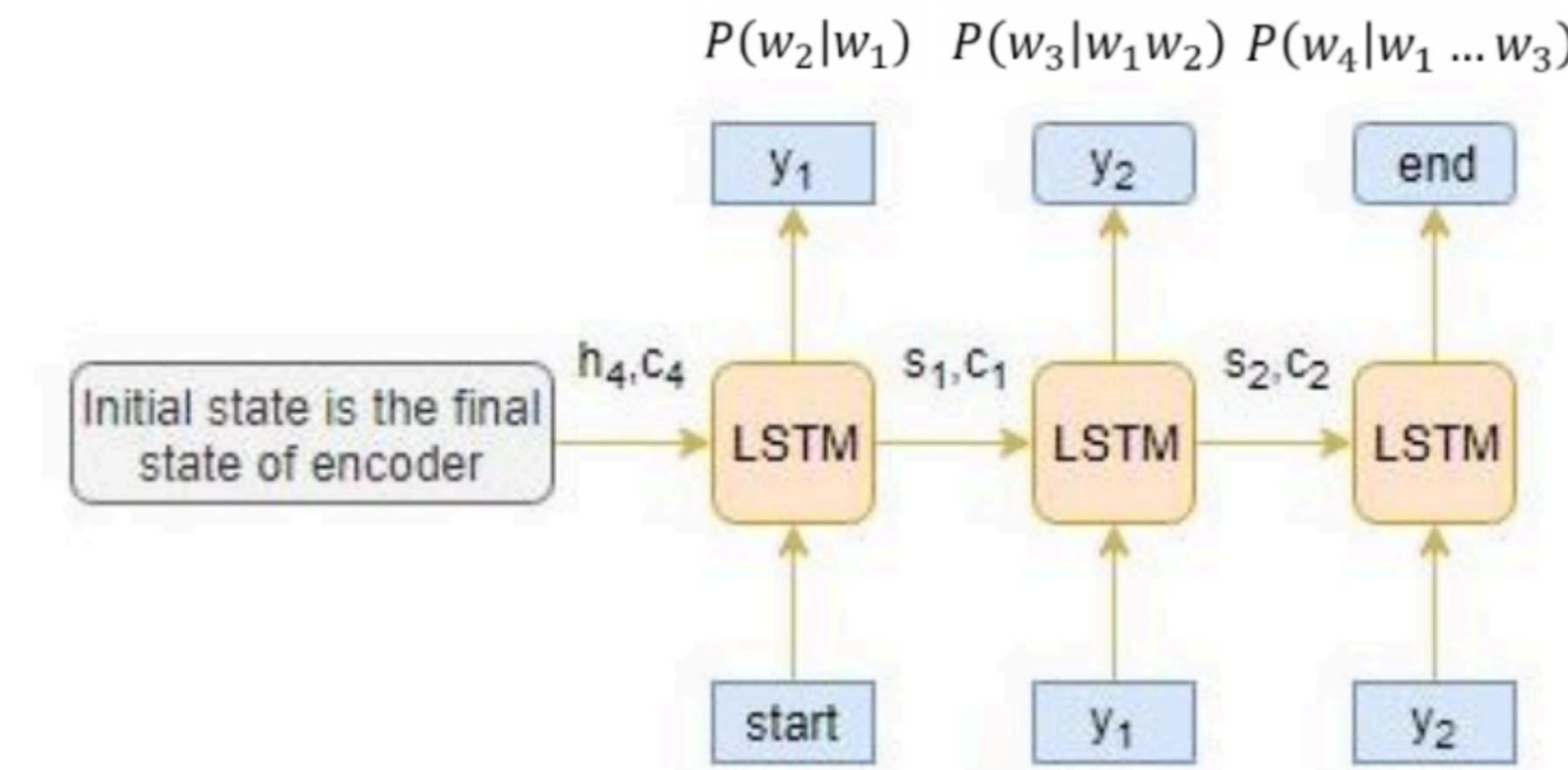
\*訓練のとき

入力ベクトル1：

(機械翻訳ならば、<EOS>+翻訳先の言語の単語のベクトル)

入力ベクトル2：

(Encoderの隠れ状態 & セル状態)



正解ラベル：

(機械翻訳ならば、翻訳先の言語の単語のベクトル  
+<EOS>)

出力ベクトル：

(機械翻訳ならば、翻訳先の言語のシーケンスの次の単語を予測する確率ベクトル)

[余談]

# 特殊記号について

1. <EOS>は、Decoderにとって、文の始まり/終わりを意味する。モデルが最初に学習することが多い。いくつかの単語の後にこれが出現したら出力（単語のサンプリング）を停止することを学習する。トレーニングデータにこの信号がない場合、ほとんどの場合、モデルはランダムな単語で文を埋めようとする。
2. <UNK>これは、それらの単語が単語として存在しないことを示すプレースホルダーである。
3. <PAD>ほとんどのシーケンスは可変長であるため、モデルがバッチ計算を実行できるように、シーケンスを固定長にパディングする。

[https://paulx-cn.github.io/blog/4th\\_Blog/](https://paulx-cn.github.io/blog/4th_Blog/)

# seq2seqのDecoder

\* 推論のとき

訓練時とちがい「教師強制」なしで

Decoderの予測をDecoderに再入力する

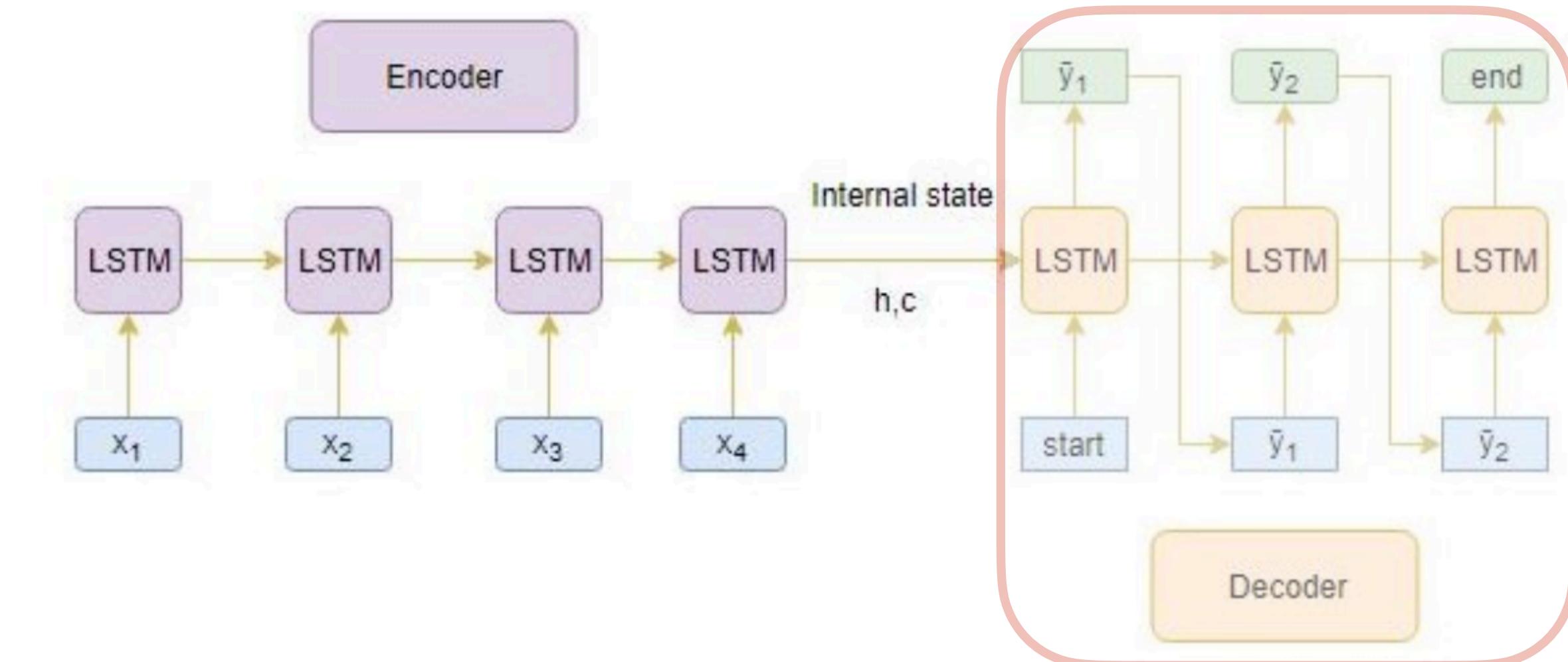
入力ベクトル1：

最初のステップは、<EOS>をDecoder LSTMの入力とする。

次のステップからは、Decoder LSTMの出力ベクトルを次の時点のDecoder LSTMの入力とする。

入力ベクトル2：

(Encoderの隠れ状態 & セル状態)



[余談]

# 教師強制 (Teacher Forcing)

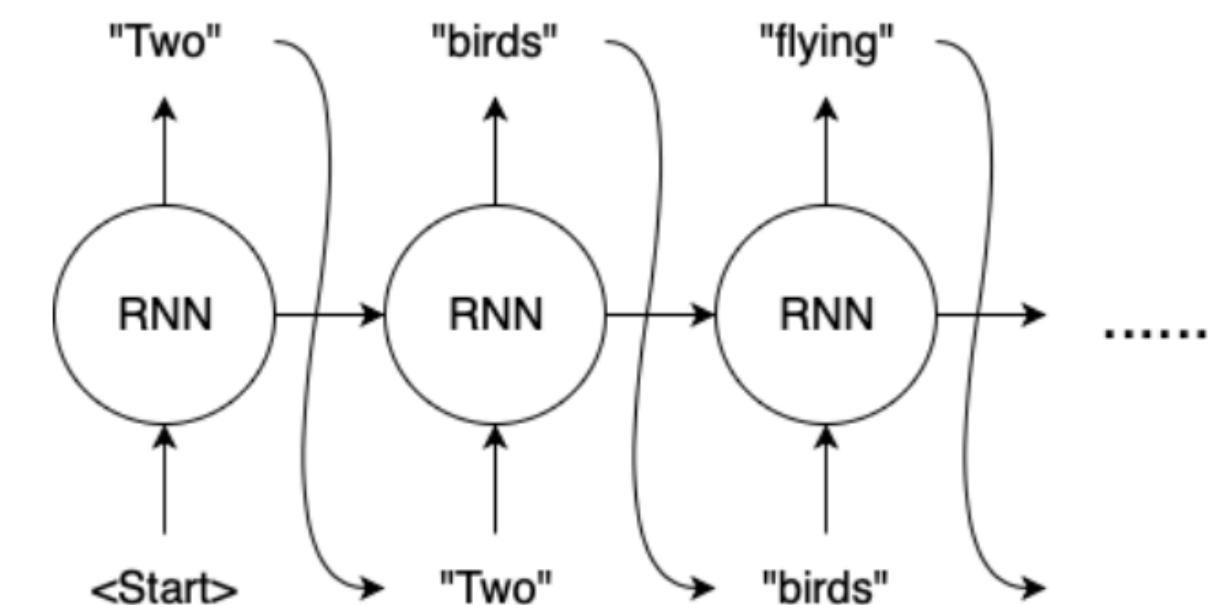
訓練時に、タイムステップごとにモデルに対して教師データを与えること。

<https://blog.keras.io/a-ten-minute-introduction-to-sequence-to-sequence-learning-in-keras.html>

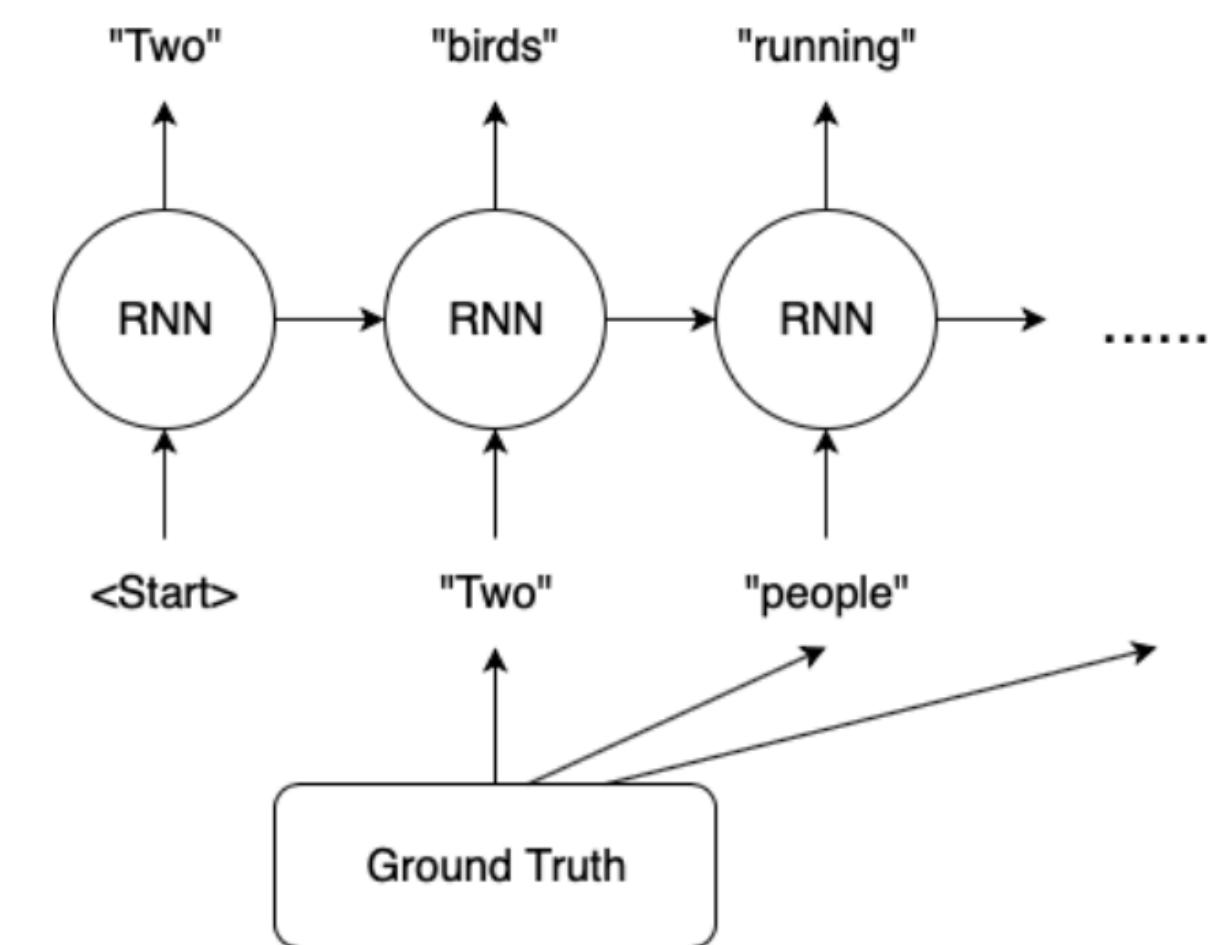
Teacher Forcingに対して、モデルが出力したデータを利用する手法をFree Learningと呼びます。

**Exposure Bias**の問題：

学習時の時系列入力はground truthが与えられるが、推論時は自身の予測トークンを入力に入れる必要があるため、学習時の時のような動作への保証がありません。この問題は長いテキスト生成の時に特に問題になります。

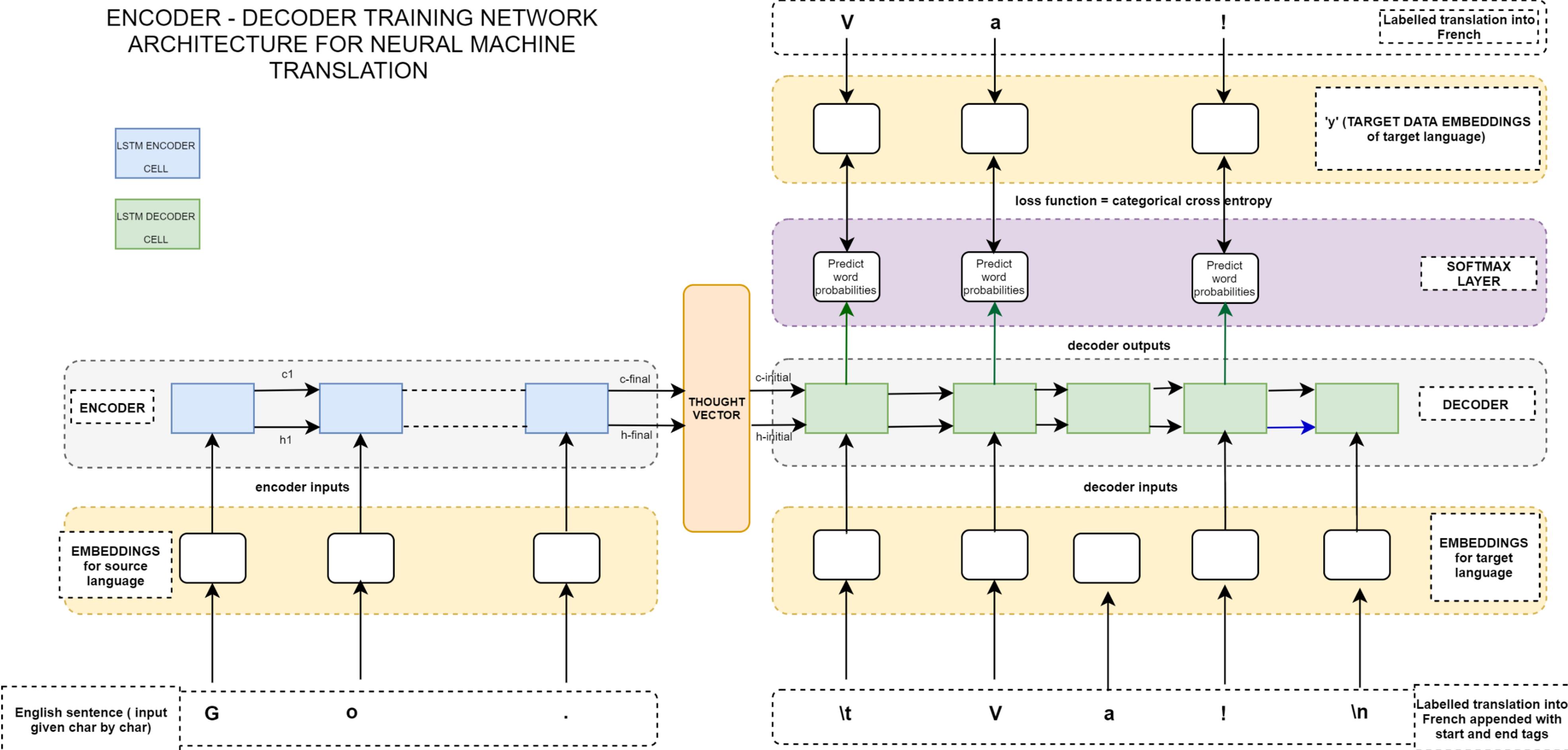


Without Teacher Forcing



With Teacher Forcing

# システム全体像



[発展]

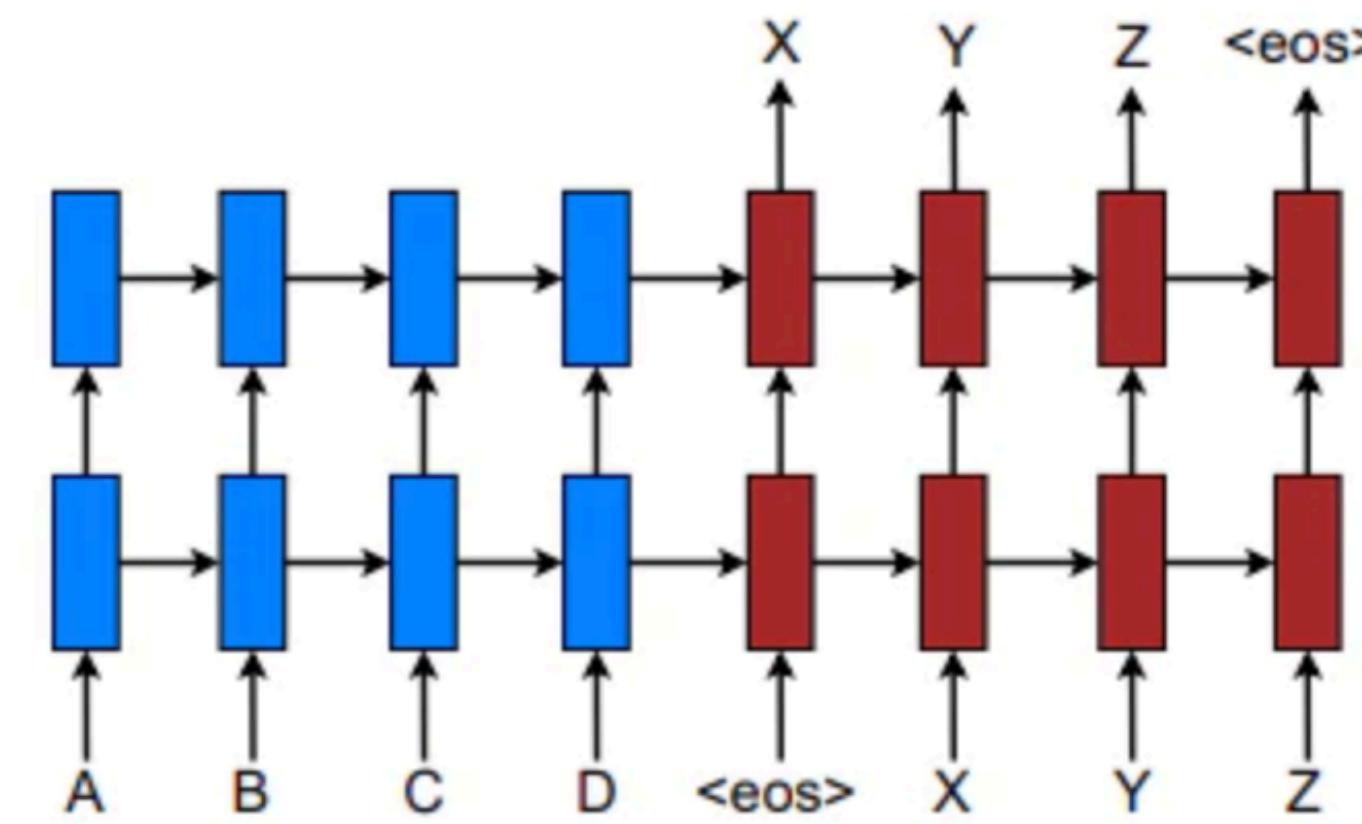
# Seq2Seq + Attention

Seq2Seqを用いても長期依存問題（次単語予測が長期記憶に依存すること）を解消することは難しかった。

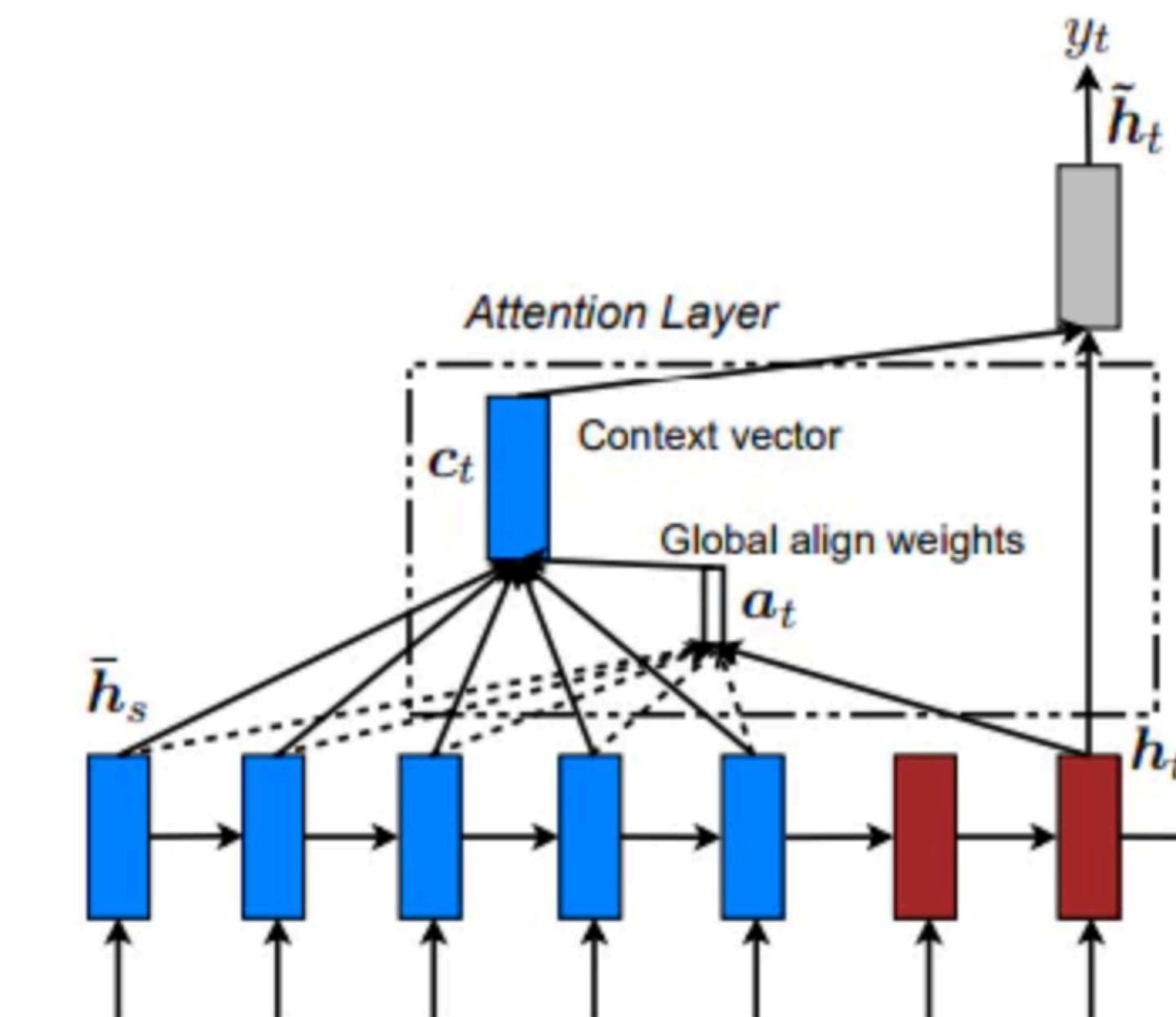
入力系列が長くなると、入力系列の情報をDecoderに伝えるのが困難になる。

そこで、過去の情報を重み付けして再利用するAttention機構をSeq2Seqに追加する手法が提案された。

<https://arxiv.org/pdf/1509.01025.pdf>



Seq2Seq



Seq2Seq + Attention