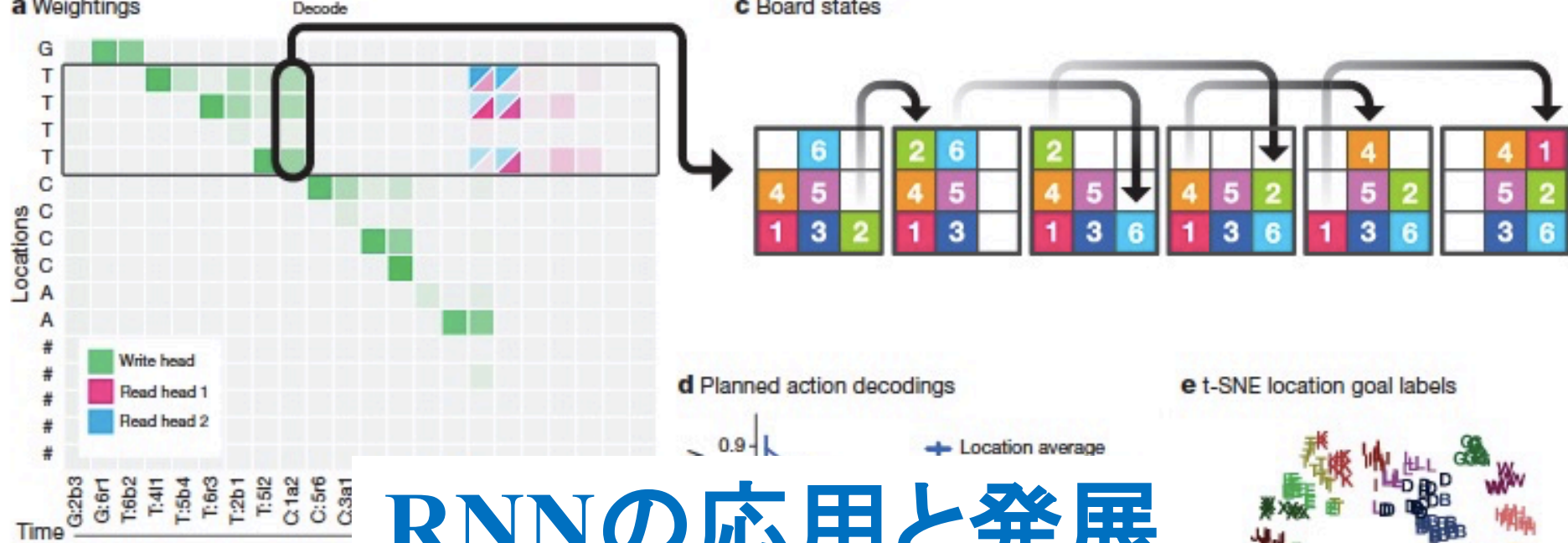




RNNの応用と発展

-- Google翻訳とDNC --

丸山不二夫



Je tiens l'affaire! (I've got it!)

-- Champollion 1822/9/14

and then fainted away from his excitement.

The bond between the signifier and the signified is arbitrary. Since I mean by sign the whole that results from the associating of the signifier with the signified, I can simply say: the linguistic sign is arbitrary.

-- Saussure

The language faculty provides the means to construct a digitally infinite array of structured expressions, each of which has a semantic interpretation expressing a thought, and each of which can be externalized by means of some sensory modality. The infinite set of semantically interpreted objects constitutes what has sometimes been called a "language of thought"

-- Chomsky

We describe visualizations of the new system in action, which provide early evidence of shared semantic representations (interlingua) between languages.

-- Google

はじめに

- 2012年から始まったディープラーニングのブームは、5年経った今また、新しい展開を見せはじめている。
- 一つは、昨年11月に製品版のサービスが始まったGoogleのニューラル機械翻訳が、素晴らしい成長を見せたことだ。翻訳の分野での成果だが、自然言語処理の飛躍的な発展が起こるかもしれないと期待させるものだ。
- もう一つは、同じく、Googleが昨年末に発表した、DNC (Differentiable Neural Computer) だ。視覚を中心とした感覚・運動系の処理が得意なCNNや、言語等のシーケンス処理が得意なRNNとは、違う分野(論理的推論)に違うアーキテクチャーで挑戦しようとしている。アーキテクチャー的には、ディープラーニング技術の発展形ではあるのだが、ノイマン型コンピューターの拡張のようにも見える。

はじめに

- 筆者は、人工知能研究は、以前にも述べたように、今後、次のような領域に拡大するものと考えている。
 - 人間固有の言語能力の機械による実現。
 - 人間の数学的・論理的な推論能力に関わるもの。
 - こうした展望から見ると、昨今の動きは、とても興味ふかいものである。ただ、こうした未来の人工知能が、現在のディープラーニング技術から直接発展していくものなのかについては、いろいろな可能性があることを意識したほうがいいとも考えている。
 - 今回のセミナーでは、時間と内容の両面で、両方を取り上げるのは難しいと判断した。今回は、Google翻訳を中心におこうと思う。DNCについては、次回のマルレクのテーマにしたいと思っている。
-

AI技術を構成する複数の流れの中で、なぜ、ニューラル・ネットワークが重要なのか？

- A) 既知のデータの統計的分析をもとに、数値予測・クラス分けを行うもの。機械学習技術。
- B) 経験的に構築された知識・推論・対話システム。パーソナル・アシスタンス・システム etc。
- C) ニューラルネットワークの手法を用いて、生物の感覚・運動系の能力の相同物を機械上で実現しようとするもの。ディープ・ラーニング。
- D) 人間固有の言語能力の機械による実現。
- E) 人間の数学的・論理的な推論能力に関わるもの。

ニューラル・ネットワークは、 新しいAI技術の**現在**の技術的中核だから

A) 既知のデータの統計的分析をもとに、数値予測・クラス分けを行うもの。機械学習技術。

NNは、同等の能力を持つ

B) 経験的に構築された知識・推論・対話システム。パーソナル・アシスタンス・システム etc。

Speech2Text
Text2Speech
Human Interface

C) ニューラルネットワークの手法を用いて、生物の感覚・運動系の能力の相同物を機械上で実現しようとするもの。ディープ・ラーニング。

CNN+RNN
機械翻訳
Knowledge Database

D) 人間固有の言語能力の機械による実現。

E) 人間の数学的・論理的な推論能力に関わるもの。

未来の、
「人工知能」

Agenda Part I

Google翻訳

- Googleニューラル機械翻訳に先行したもの
 - ルールベース機械翻訳モデル
 - 統計的機械翻訳モデル
 - 統計的機械翻訳モデルの成功と限界
 - パラレル・コーパスの現在
 - ニューラル確率言語モデル -- Bengioの「次元の呪い」
 - 語の「意味ベクトル」 -- Word2Vec
 - RNNの文法認識能力
-

Agenda Part I

Google翻訳

- Googleニューラル機械翻訳
 - システムの概観
 - Encoder-Decoder
 - Attention Mechanism
 - Wordpiece
 - Google多言語ニューラル機械翻訳
 - 多言語翻訳を単一システムで
 - 実験結果
 - ゼロ・ショット翻訳
 - インターリンガの存在
-

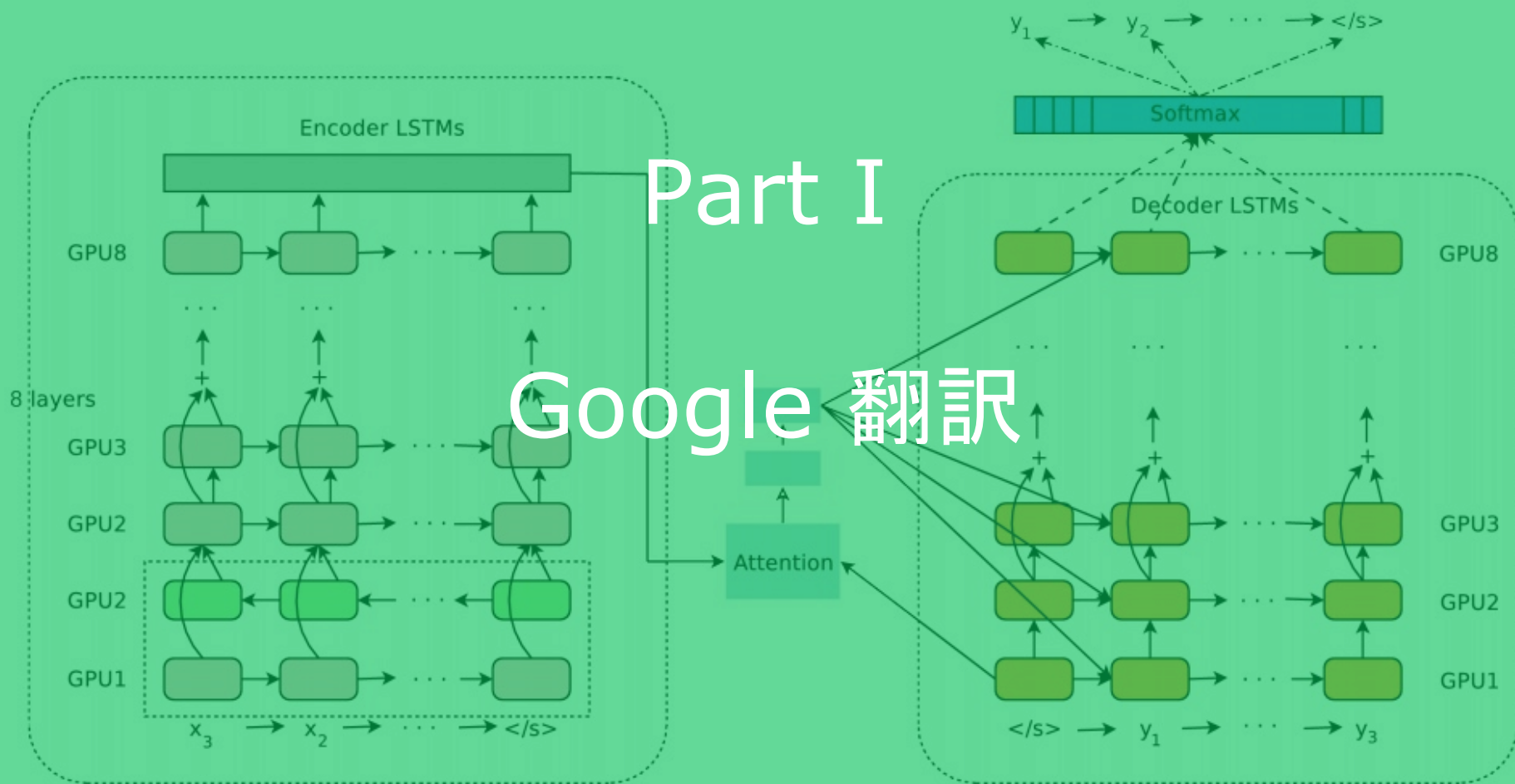
Agenda Part II

可微分ニューラルコンピューター

- 可微分ニューラルコンピューター (DNC)
 - DNCシステム概観
 - ヘッドとメモリーの相互関係
 - DNCは、どんな課題に取り組んでいるか？
-

Part I

Google 翻訳



ニューラル機械翻訳に先行したもの

新しいGoogle翻訳は、「ニューラル機械翻訳モデル」に基づいている。それに対して、それ以前のGoogle翻訳は、「統計的機械翻訳モデル」に基づいたものだ。

ここでは、まず、新しいGoogle翻訳以前の代表的な機械翻訳モデルを振り返り、Googleの新しいニューラル機械翻訳モデルが、どのような技術の系譜と問題意識から生まれたのかを考えてみようと思う。

ルール・ベース機械翻訳モデル (RBMT)

ルール・ベースの機械翻訳モデル (Rule-Based Machine Translation) は、古典的なものだ。日本でも、このモデルに基づく多くの翻訳システムが、かつては作られていた。(当時の日本は、AIでも機械翻訳でも、先進国だった。)

現在の機械翻訳モデルの主流からは、外れている。

□ 人間が(ここでは日本人としよう)、英語の文章を日本語の文章に翻訳する場合を考えよう。その時、最低限必要になるのは、次の三つだ。

1. 英語・日本語の辞書
2. 英語の文法の知識
3. 日本語の文法の知識(暗黙の知識を含めて)

□ ただ、これで十分かというと、そうではない。英語の構文を日本語の構文に変換する知識が必要となる。

-
- ルール・ベースの機械翻訳は、基本的には、こうした人間の翻訳作業と同じことを、機械にさせようとする。原理はわかりやすい。機械の翻訳のルールは、人間が手で書くので、誤訳があっても、その原因を突き止めることは容易だ。
 - ただ、電子的に利用可能ないい辞書があるかは分野による。と言って、ゼロから辞書を作るのは手間がかかる。その上、文法構造の解析は、意外と難しい。様々の付加的情報を追加する必要がある。ただ、我々は、母語にせよ外国語にせよ、自分の持つ文法知識を、たいていの場合、ルールとして全て書き下せるわけではないのだ。
 - 重要なことは、ルールベースのアプローチが、「意味」を、論理的な述語で記述しようとしていることだと思う。この方向の探求は、今でも有効であると思っている。
-

Rule-based MT

- Parse SL to some more abstract form: **the meaning?**

The dog chases a cat

→ $\text{chase}_1(\text{dog}_1: [\text{def}], \text{cat}_1: [\text{indef}])$

- Transfer to the target language abstract form

→ $\text{追う}_1(\text{犬}_1: [\text{def}], \text{猫}_1: [\text{indef}])$

- Generate from this

→ 犬 が 猫 を 追う
inu ga neko wo ou
dog NOM cat ACC chase

Combinatory Categorical Grammars

square	blue	or	round	yellow	pillow
ADJ	ADJ	C	ADJ	ADJ	N
$\lambda x.square(x)$	$\lambda x.blue(x)$	$disj$	$\lambda x.round(x)$	$\lambda x.yellow(x)$	$\lambda x.pillow(x)$
N/N	N/N		N/N	N/N	
$\lambda f.\lambda x.f(x) \wedge square(x)$	$\lambda f.\lambda x.f(x) \wedge blue(x)$		$\lambda f.\lambda x.f(x) \wedge round(x)$	$\lambda f.\lambda x.f(x) \wedge yellow(x)$	
N/N	N/N		N/N	N/N	
$\lambda f.\lambda x.f(x) \wedge square(x) \wedge blue(x)$			$\lambda f.\lambda x.f(x) \wedge round(x) \wedge yellow(x)$		
$<\Phi>$					
N/N					
$\lambda f.\lambda x.f(x) \wedge ((square(x) \wedge blue(x)) \vee (round(x) \wedge yellow(x)))$					
$>$					
N					
$\lambda x.pillow(x) \wedge ((square(x) \wedge blue(x)) \vee (round(x) \wedge yellow(x)))$					

ある意味で、ルールベース翻訳の現代版 Chomsky の言語理論とも接点がある。

<https://goo.gl/PZE9oA>

show	me	flights	to	Boston
S/N		N	PP/NP	NP
$\lambda f.f$		$\lambda x.flight(x)$	$\lambda y.\lambda x.to(x, y)$	$BOSTON$
			$\xrightarrow{PP}$	
			$\lambda x.to(x, BOSTON)$	
			$\xrightarrow{N \setminus N}$	
			$\lambda f.\lambda x.f(x) \wedge to(x, BOSTON)$	
			$\xleftarrow{N}$	
			$\lambda x.flight(x) \wedge to(x, BOSTON)$	
			$\xrightarrow{S}$	
			$\lambda x.flight(x) \wedge to(x, BOSTON)$	

行の上は、Syntax
 行の下は、Semantics

SyntaxとSemanticsを
 同時に扱うことができる

go	to	the	chair
S	AP/NP	NP/N	N
$\lambda a.move(a)$	$\lambda x.\lambda a.to(a, x)$	$\lambda f.\iota x.f(x)$	$\lambda x.chair(x)$
		$\xrightarrow{\quad} NP$ $\iota x.chair(x)$	
		$\xrightarrow{\quad} AP$ $\lambda a.to(a, \iota x.chair(x))$	
		$\xrightarrow{\quad} S \backslash S$ $\lambda f.\lambda a.f(a) \wedge to(a, \iota x.chair(x))$	
		$\xrightarrow{\quad} S$ $\lambda a.move(a) \wedge to(a, \iota x.chair(x))$	

行の上は、Syntax
行の下は、Semantics

SyntaxとSemanticsを
同時に扱うことができる

**Curry-Howard-(Lambek)
correspondence**

<https://goo.gl/PZE9oA>

統計的機械翻訳モデル (SMT)

ルール・ベースの機械翻訳モデルが、その精度を上げるのに伸び悩んでいる中、新しい機械翻訳のモデルに注目が集まる。それが統計的機械翻訳モデル (Statistical Machine Translation) である。

「ニューラル機械翻訳」という新しいアプローチも、広い意味では、このカテゴリーに属する。

1990年 Peter F. Brown et al. "A Statistical Approach To Machine Translation" <https://goo.gl/UnTIxo>

「機械翻訳の分野は、現代のデジタルコンピュータとほとんど同じ程古いものだ。1949年に、ウォーレン・ウィーバー (Warren Weaver) は、この問題が統計的方法と情報理論から得られるアイデアで攻略できると提案していた。情報理論の領域は、当時、彼やクロード・シャノン (Claude Shannon) やその他の人たちが開発していたものだ。このアプローチは、多くの理論上の反対意見を集め、研究者はすぐにこのアプローチを放棄したが、我々は、本当の障害は、利用可能なコンピュータの相対的な無能力と、こうした攻略には不可欠な、そこから統計情報を収集する機械で可読なテキストの欠如にあったと考えている。今日では、コンピュータは、1950年より5桁ほど早くなり、数百メガバイトのストレージを備えている。機械可読なコーパスも利用可能である。.... 我々は、今こそ、これらを、機械翻訳に生かすべき時だと感じている。」

SMTの基本的アイデア

- ニヶ国語間の翻訳を考える。翻訳されるべき文 S (ソース Sourceの S だと思えばいい)と翻訳結果の文 T (ターゲット Targetの T である)のすべてのペア (S, T) に対して、ある確率を割り当てる。
 - 確率 $\Pr(T|S)$ は、ソース言語に文 S が現れる時、翻訳がターゲット言語で文 T を生み出す確率と解釈される。
 - 例えば、英語から日本語への翻訳なら、 $\Pr(\text{"リンカーン大統領はいい弁護士だった"} \mid \text{"President Lincoln was a good lawyer"})$ は、高い確率をマークするだろうが、 $\Pr(\text{"僕は今朝歯磨きをした"} \mid \text{"President Lincoln was a good lawyer"})$ の確率は低くなるように確率を与える。
-

検索としての翻訳

- こうした見方をすると、機械翻訳は、次のような問題だと考えることができる。
 - ターゲット言語の文Tが与えられた時、我々は、翻訳が文Tを生成した文Sを検索する。与えられたTに対して、もっともありそうなSを選ぶことで、翻訳のエラーは最小化されるのは明らかなので、 $\text{Pr}(S|T)$ が最大になるようにS を選べばいいことになる。
 - 単純化すれば、膨大な二国語コーパスから、適当な翻訳を「検索」というアプローチなのだが、それでも、先行した「ルールベース機械翻訳モデル」より、高い翻訳精度を達成した。
 - IBMの旧Watsonの「成功」も、「知識」ベースだが、検索と文理解の結合の取り組みと考えることは、出来るのだ。
-

-
- 次の関係は簡単にわかる。(ベイズの公式)

$$\Pr(S|T) = \frac{\Pr(S) \Pr(T|S)}{\Pr(T)}$$

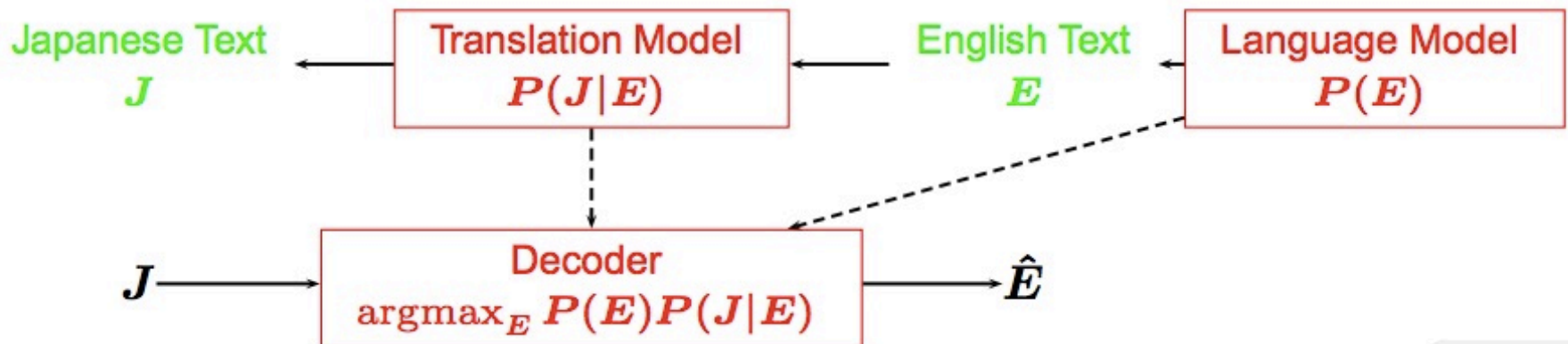
- 右辺の分母の $\Pr(T)$ は、 S には依存しないので、 $\Pr(S|T)$ を最大にするには、分子の $\Pr(S) \Pr(T|S)$ を最大にすれば十分である。この最初の項 $\Pr(S)$ を「この言語モデルでの S の確率」と言い、二番目の項 $\Pr(T|S)$ を「与えられた S に対する T の翻訳の確率」という。
-

統計的機械翻訳モデルの概念図

Statistical Machine Translation

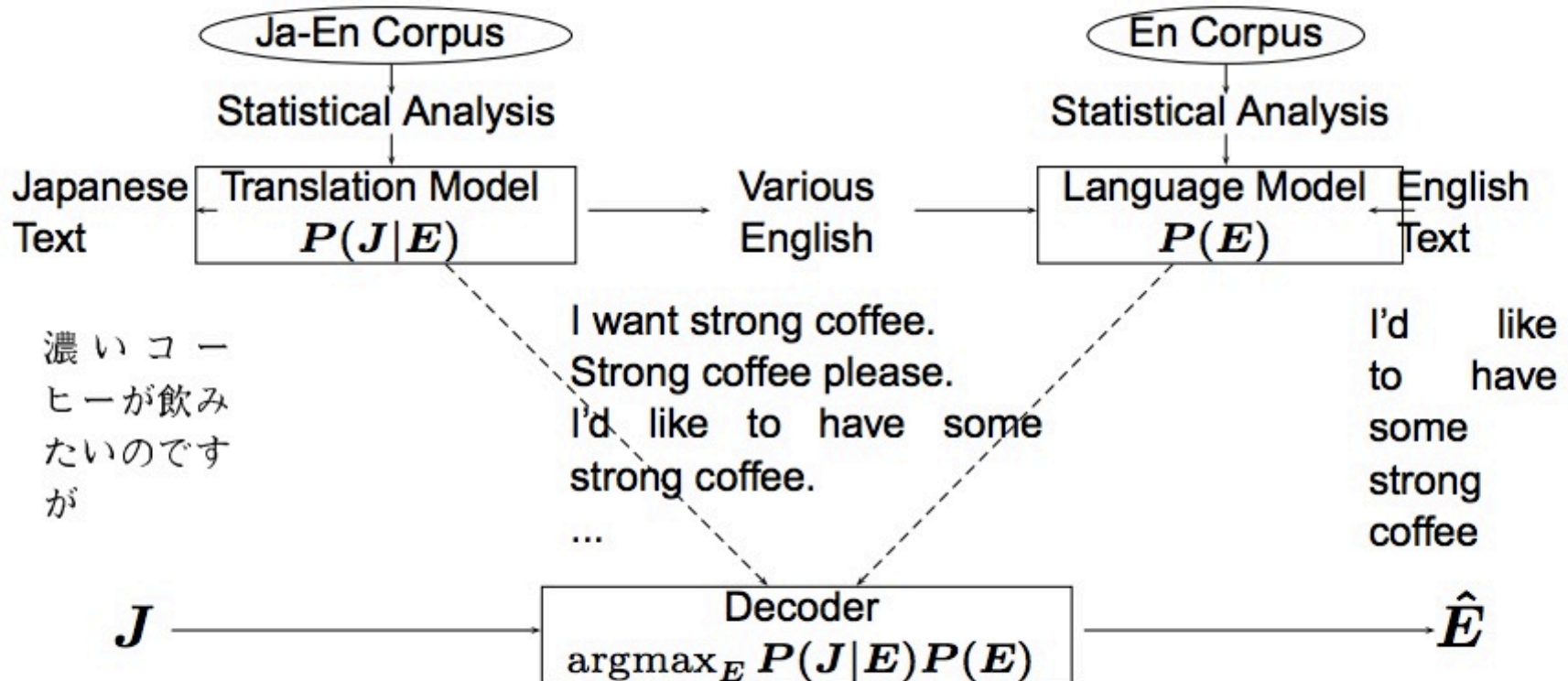
- The Basic Idea (Brown *et al* 1990)
Find the most probable English sentence given a foreign language sentence

$$\hat{E} = \operatorname{argmax}_E P(E|J)$$

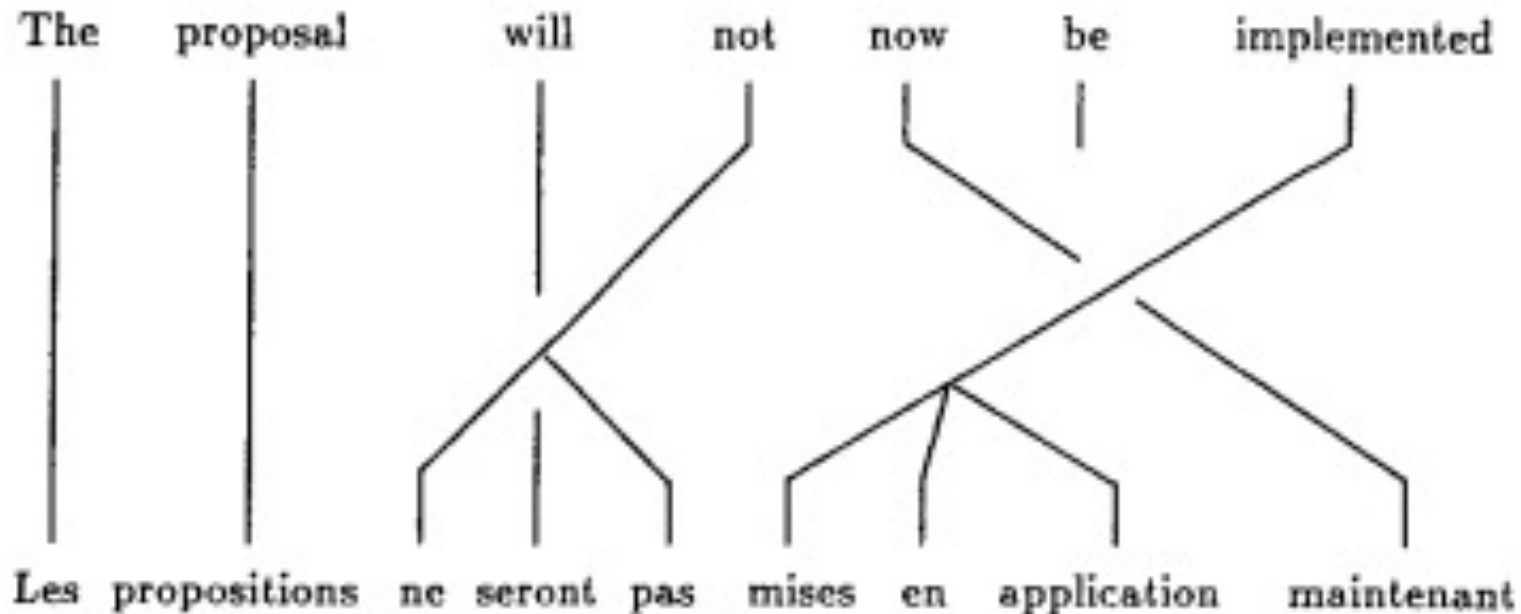


統計的機械翻訳モデルの概念図

Statistical MT Framework



Alignment (1対1とは限らない)



Fertility

1対多関係の表現

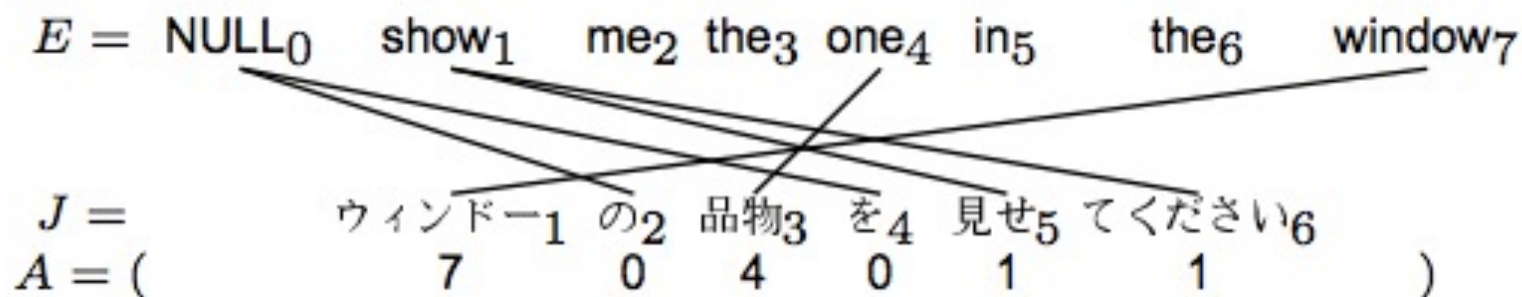
$$\begin{aligned} & \Pr(\text{fertility} = 1 | \text{John}) \times \Pr(\text{Jean} | \text{John}) \times \\ & \Pr(\text{fertility} = 0 | \text{does}) \times \\ & \Pr(\text{fertility} = 2 | \text{beat}) \times \Pr(\text{est} | \text{beat}) \Pr(\text{battu} | \text{beat}) \times \\ & \Pr(\text{fertility} = 1 | \text{the}) \times \Pr(\text{Le} | \text{the}) \times \\ & \Pr(\text{fertility} = 1 | \text{dog}) \times \Pr(\text{chien} | \text{dog}) \times \\ & \Pr(\text{fertility} = 1 | \langle \text{null} \rangle) \times \Pr(\text{par} | \langle \text{null} \rangle). \end{aligned}$$

Aligning Text

こうした、25年以上前の取り組みが、NMTで復活している。Attentionという、新しい名前です。

本来は、「統計」の問題ではなく、「文法」の問題として、扱うべきなのだと思うのだが。

	show ₁	me ₂	the ₃	one ₄	in ₅	the ₆	window ₇
ウィンドー ₁	☐	☐	☐	☐	☐	☒	☒
の ₂	☐	☐	☐	☐	☐	☐	☐
品物 ₃	☐	☐	☐	☒	☐	☐	☐
を ₄	☐	☐	☐	☐	☐	☐	☐
見せ ₅	☒	☒	☐	☐	☐	☐	☐
てください ₆	☒	☒	☐	☐	☐	☐	☐



統計的機械翻訳モデル (SMT)の 成功と限界

こうした、統計的機械翻訳モデル (SMT)の成功は、翻訳モデルの質にではなく、コーパスの量に対する関心を向けさせることになった。

しかし、後で見るように、その限界に対する洞察も進む。

一つは、文の理解には、文が持つ固有の情報を取り込むような統計的方法が必要だという認識である。それが、ニューラル機械翻訳への道を開く。それが「意味」への注目であるとするれば、もう一つは、ニューラルネットが、様々な「文法」を理解出来ているという発見である。

Very Very Large Corpora

- 2001年のBankoらの論文 “Scaling to Very Very Large Corpora for Natural Language Disambiguation” のタイトルにも、そうした傾向が如実に表れている。
<http://research.microsoft.com/pubs/66840/acl2001.pdf>
 - ただ、Bankoらのモデルでは、“For breakfast I ate _____ eggs.” の空白部分に入るものを、{to, two, too} の三つの中から選べという、極めて単純な問題に答えるのに、10億語のコーパスの学習が必要だった。
-

Googleの以前の機械翻訳システム

- Googleの以前の機械翻訳に対するアプローチも、スペル訂正アルゴリズムの成功に気を良くして、"Power of Data"を前面に押し出して、"Yes! Size Matters" と叫んでいた。「コーパスが二倍になれば、翻訳の質は、0.5%上がる」と。
 - ただ、今回の新しいGoogle翻訳による翻訳精度の改善は、単純なコーパスの量の拡大によるものではない。Googleは、翻訳のアルゴリズムを見直し、機械翻訳のモデルを「統計的機械翻訳モデル」から「ニューラル機械翻訳モデル」に変えたのである。
-

勝つのは誰？

- Andrew NGは、この論文を取り上げ、次のように述べる。
<http://bit.ly/1qKh8ro>

「勝つのは、最良のアルゴリズムを持っている人ではなく、もっとも多くのデータを持っている人である。」

“It’s not who has the best algorithm that win. It’s who has the most data.”

- これは、皮肉なのだろうか？
-

CERNは、2016年4月に、300TBにも及ぶ、
加速器LHCの観測データを公開している

“It’s not who has the best algorithm that win.
It’s who has the most data.” だろうか

<http://goo.gl/UA8z70>

パラレル・コーパスの現在

ここでは、少し視点を変えて、SMTにとってもNMTにとっても本質的に重要な、コーパスの現在を見てみよう。驚くほどの大規模化が進んでいる。

「勝つのは、最良のアルゴリズムを持っている人ではなく、もっとも多くのデータを持っている人である。」という主張が、強い説得力を持つことは疑いない。ただ一点、「最良のアルゴリズム」の持つ、破壊的なインパクトを軽視していることを除けば。

コーパスの例(英仏)

English	French
According to our survey, 1988 sales of mineral water and soft drinks were much higher than in 1987, reflecting the growing popularity of these products. Cola drink manufacturers in particular achieved above-average growth rates. The higher turnover was largely due to an increase in the sales volume. Employment and investment levels also climbed. Following a two-year transitional period, the new Foodstuffs Ordinance for Mineral Water came into effect on April 1, 1988. Specifically, it contains more stringent requirements regarding quality consistency and purity guarantees.	Quant aux eaux minérales et aux limonades, elles rencontrent toujours plus d'adeptes. En effet, notre sondage fait ressortir des ventes nettement supérieures à celles de 1987, pour les boissons à base de cola notamment. La progression des chiffres d'affaires résulte en grande partie de l'accroissement du volume des ventes. L'emploi et les investissements ont également augmenté. La nouvelle ordonnance fédérale sur les denrées alimentaires concernant entre autres les eaux minérales, entrée en vigueur le 1er avril 1988 après une période transitoire de deux ans, exige surtout une plus grande constance dans la qualité et une garantie de la pureté.

Gale & Church "A Program for Aligning Sentences in Bilingual Corpora"
<http://www.aclweb.org/anthology/J93-1004.pdf>から。

コーパスの例(英仏)

English	French
According to our survey, 1988 sales of mineral water and soft drinks were much higher than in 1987, reflecting the growing popularity of these products. Cola drink manufacturers in particular achieved above-average growth rates.	Quant aux eaux minérales et aux limonades, elles rencontrent toujours plus d'adeptes. En effet, notre sondage fait ressortir des ventes nettement supérieures à celles de 1987, pour les boissons à base de cola notamment.
The higher turnover was largely due to an increase in the sales volume.	La progression des chiffres d'affaires résulte en grande partie de l'accroissement du volume des ventes.
Employment and investment levels also climbed.	L'emploi et les investissements ont également augmenté.
Following a two-year transitional period, the new Foodstuffs Ordinance for Mineral Water came into effect on April 1, 1988. Specifically, it contains more stringent requirements regarding quality consistency and purity guarantees.	La nouvelle ordonnance fédérale sur les denrées alimentaires concernant entre autres les eaux minérales, entrée en vigueur le 1er avril 1988 après une période transitoire de deux ans, exige surtout une plus grande constance dans la qualité et une garantie de la pureté.

Gale & Church "A Program for Aligning Sentences in Bilingual Corpora"
<http://www.aclweb.org/anthology/J93-1004.pdf>から。

機械翻訳に利用される、現代のコーパスの規模

- WMT'14の英語(En) <-> フランス語(Fr)データセットには、3,600万の文のペアが含まれている。
 - WMT'14の英語(En) <-> ドイツ語(De)データセットには、500万の文のペアが含まれている。
 - Googleは、内部に、英語 <-> 日本語(Ja)、英語 <-> 韓国語(Ko)、英語 <-> スペイン語(Es)、英語 <-> ポルトガル語(Pt) 等々の多くのデータセットを持っているが、**その規模は、先のWMTのデータセットより、2～3桁大きいという。**
 - **Googleニューラル機械翻訳では、GPU100個を使って、フルトレーニングには最大1,000万ステップ、収束までには3週間かかることがあるという。**
-

最大の電子的コーパス

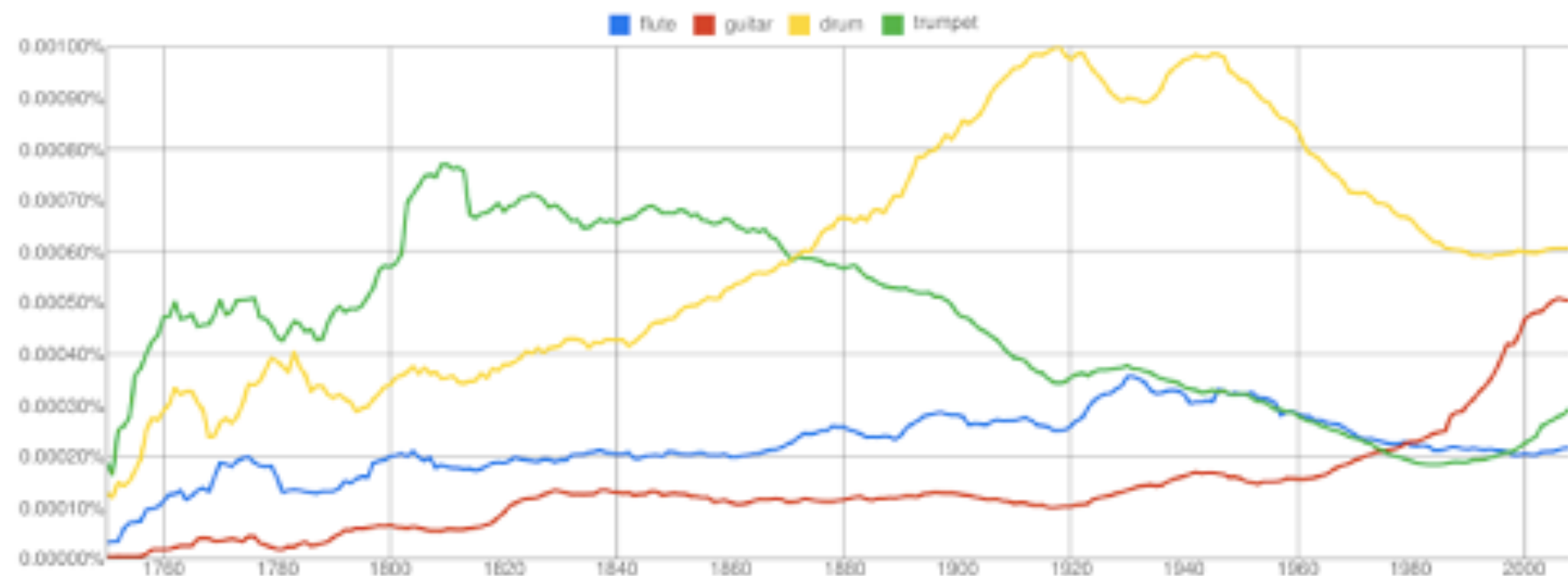
Googleの書籍のデジタル化 2004年～

- Googleとハーバード大を始めとする全世界の40の大学図書館の共同作業。
- 2010年 Science誌 J. B. Mitchel et al. “Quantitative Analysis of Culture Using Millions of Digitized Books” <https://goo.gl/kitoq>
- 現在まで出版された書籍の約4%に当たる、**5,195,769 冊**をデジタル化。収録された語彙は、英語(361 billion), フランス語 (45B), スペイン語 (45B), ドイツ語 (37B), 中国語 (13B), ロシア語 (35B), ヘブライ語 (2B)
- それらのデータにアクセスするツールが公開されている。” Find out what’s in a word, or five, with the **Google Books Ngram Viewer**” <https://goo.gl/KmGZ3j>
(うまく、アクセスできない！)

Graph these case-sensitive comma-separated phrases:

between and from the corpus with smoothing of .

[Search lots of books](#)



Search in Google Books:

1750 - 1793	1794 - 1923	1924 - 1940	1941 - 1981	1982 - 2008	flute
1750 - 1825	1826 - 1979	1980 - 1992	1993 - 2001	2002 - 2008	guitar
1750 - 1777	1778 - 1801	1802 - 1817	1818 - 1952	1953 - 2008	trumpet
1750 - 1801	1802 - 1911	1912 - 1927	1928 - 1983	1984 - 2008	drum

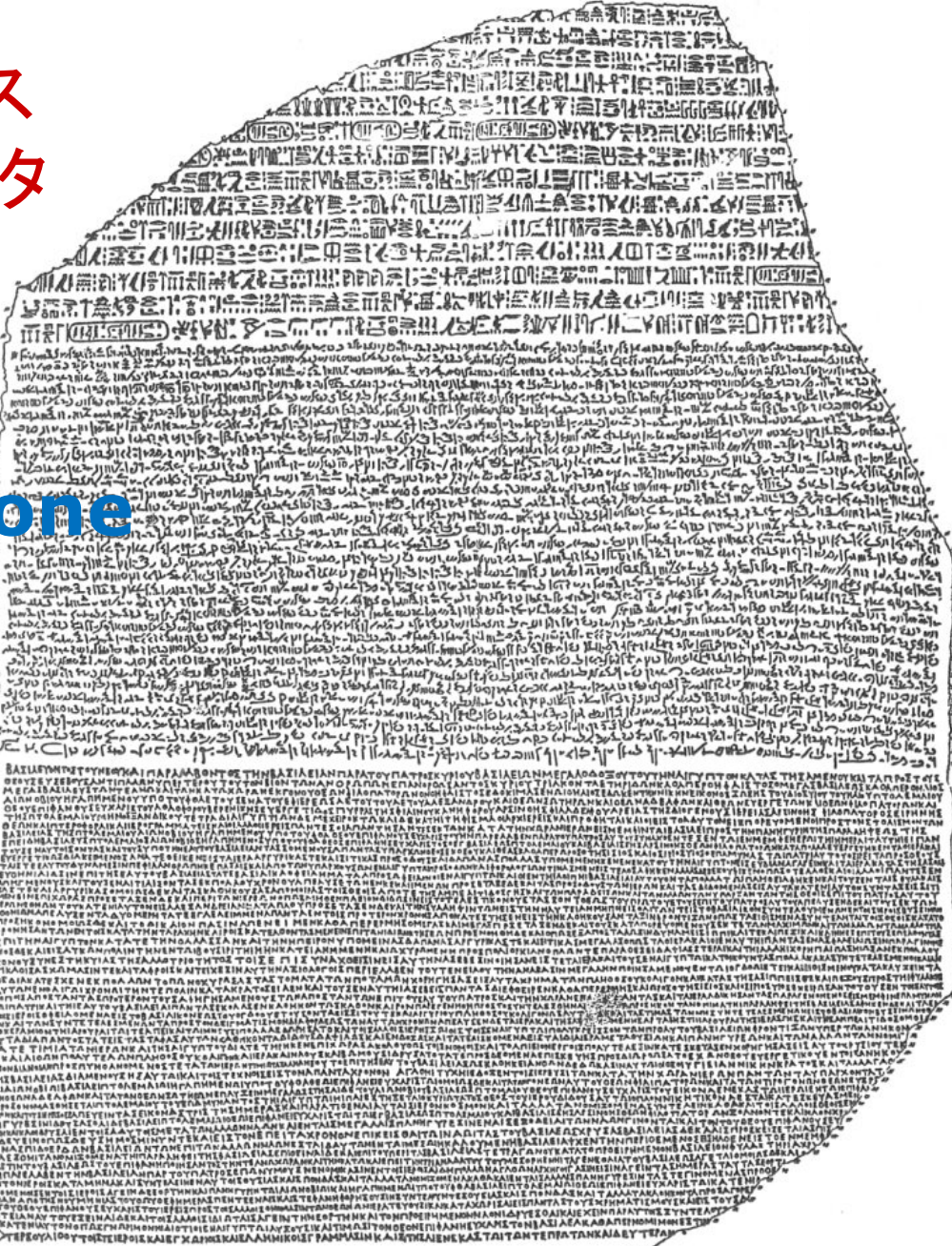
Run your own experiment! Raw data is available for download [here](#).

最古のコーパス パラレル・データ

Rosetta Stone

BC 196年

1822年
シャンポリオンが解読



神聖文字

民衆文字

ギリシャ
文字

最古の平行・データ

ベヒストウン碑文 楔形文字の解読

- コーパス: ベヒストウン碑文 BC522~
(エラム語、古代ペルシア語、アッカド語
の三つの言語で書かれている)
解読者: **ローリンソンとヒンクス**
1846-1851



線文字Bの解読

- ❑ 紀元前1450年から紀元前1375年頃までミュケナイ時代に、ギリシャ本土からエーゲ海諸島の王宮で用いられていた文字。
- ❑ 発見者であるイギリスの考古学者アーサー・エヴァンズにより線文字Bと命名された。
- ❑ 1953年、イギリスの建築家マイケル・ヴェントリスと言語学者ジョン・チャドウィックによりギリシア語として解読された。



<https://goo.gl/4SZhkn>

Phaistos diskの解読 2014年

- でも、それとは真逆の取り組みも存在する。クレタのGareth Owensは、たった一つの粘土板にきざまれた、45種類の「文字」で書かれた241文字の「文」の解読に成功したという。

<https://goo.gl/4Ye6Be>



ニューラル確率言語モデル Bengioの「次元の呪い」

統計的機械翻訳モデルに代わる、ニューラル機械翻訳モデルを提案したのは、次のBengioの論文である。

2003年 Yoshua Bengio et al.

“A Neural Probabilistic Language Model”

<http://goo.gl/977AQp>

Benjioの「次元の呪い」

- Bengioは、早くから、統計的機械翻訳モデルに現れる組み合わせの数の爆発を意識していた一人である。彼は、それを「次元の呪い」Curse of Dimensionality と呼んだ。
 - 26文字のアルファベット15文字以内で構成される語の数は、高々、 26^{15} である。スペル訂正の次元は、その程度のオーダーである。ただし、語彙が10万個ある言語での10個の語からなる文は、 $100000^{10} = 10^{50}$ 種類もある！
 - この₁ 文₂ は₃ 10^4 個₅ の₆ 語₇ から₈ できて₉ いる₁₀
 - このように、10語文というのは、そんなに長い文章ではない。が、 10^{50} というのは、とてつもなく巨大な数である。
-

2003年 Yoshua Bengio et al.

“A Neural Probabilistic Language Model” <http://goo.gl/977AQp>

- 「統計的言語モデルの目標は、ある言語における語のシーケンスの結合確率関数を学習することである。ただ、これは、次元の呪いのために本質的に困難である。その上でモデルが試される語のシーケンスは、モデルが学習中に会った全ての語のシーケンスとは異なっている可能性が高い。」
- 「伝統的だが非常に成功したn-gramベースのアプローチは、訓練データ中に現れた非常に短いシーケンスの重なりを連結することで、一般化の能力を得ている。それぞれの訓練用データ中の文が、モデルに、**指数関数的な数のその文と意味的に近い文の情報を伝えることを可能にするような、語の分散表現を学習することで、この次元の呪いと戦うことを提案する。**」

□ 具体的には、この論文で、彼は、次のような方法を提案する。

1. 語彙中のそれぞれの語に、 R_m に実数値の値を持つ、分散した語の特徴ベクトル(word feature vector)を対応づける。
2. 語の並びの結合確率関数を、この並びの中の語の特徴ベクトルで表現する。
3. 語の特徴ベクトルとこの確率関数のパラメーターを、同時に学習する。

□ 要は、統計的モデルのように、単に語の並びの統計的性質に依拠するだけでなく、それぞれの語に「特徴ベクトル」という語の「意味」の対応物を導入しようということだと僕は理解している。

Word2Vec

語の「意味ベクトル」

2013年に、GoogleにいたTomas Mikolovらは、語が埋め込まれたベクター空間が、言語学的に（文法的にも、意味論的にも）面白い性質を持っていることを発見する。それが、次の論文である。

Tomas Mikolov et al.

“Linguistic Regularities in Continuous Space

Word <http://goo.gl/j25y8s>

2013年 Tomas Mikolov et al.

“Linguistic Regularities in Continuous Space Word <http://goo.gl/j25y8s>

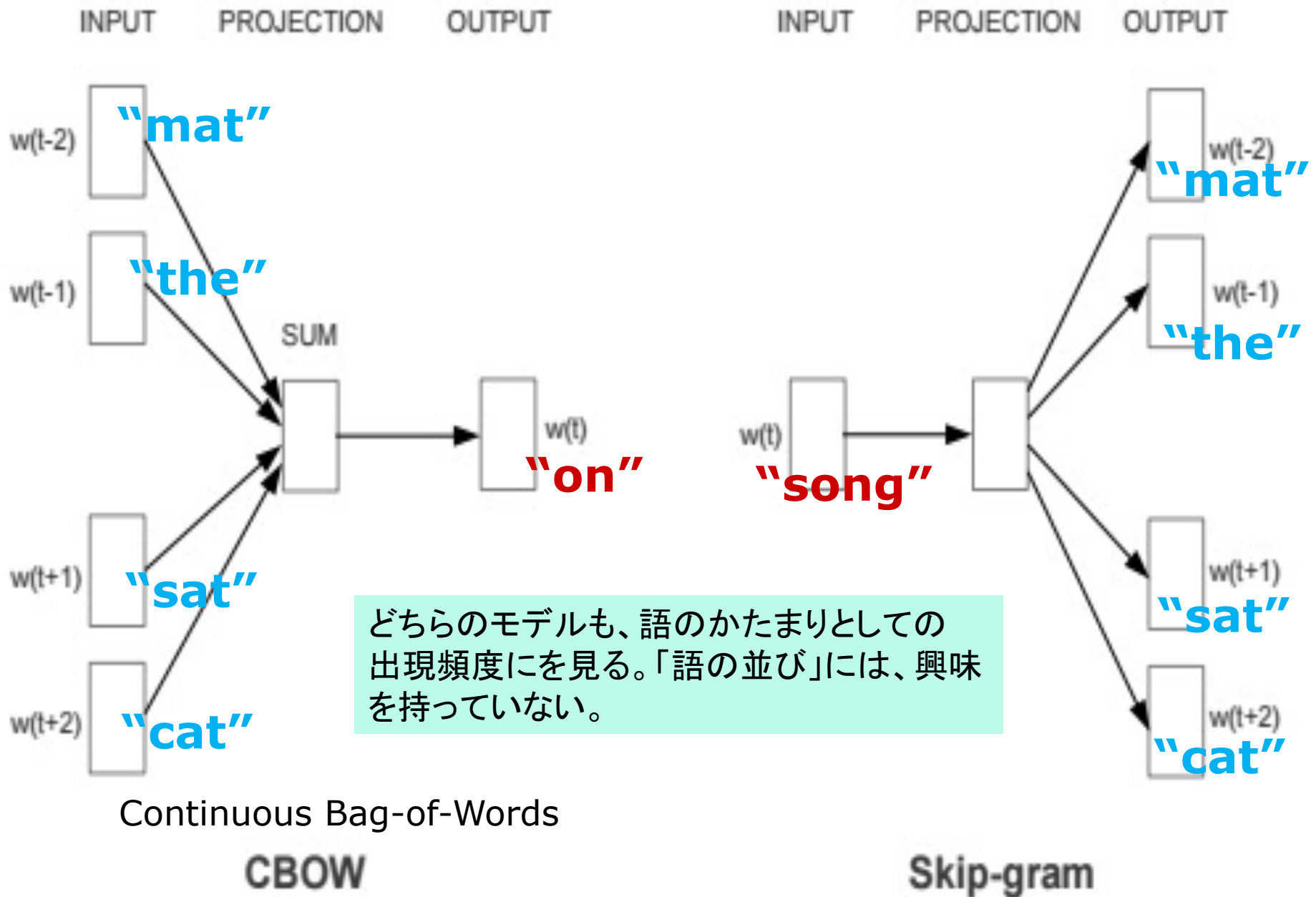
「連続スペース言語モデルは、最近、さまざまなタスクにわたって優れた結果を示している。本論文では、入力層の重みによって暗黙的に学習されたベクトル空間での単語表現を調べる。

これらの表現は、言語の文法規則と意味規則を把握する上で驚くほど優れており、それぞれの関係はその関係に特有のベクトルオフセットによって特徴付けられることがわかる。

これにより、単語間のオフセットに基づいて、ベクトル指向の推論が可能になる。例えば、男性/女性の関係性は自動的に学習され、そこから導かれたベクトル表現では、“KING – MAN + WOMAN” は、“QUEEN” に非常に近いベクトルになる。」

Word2Vecの登場

- Google Codeに、オープンソースとして公開され、
<https://code.google.com/p/word2vec/> 大きな関心を集める。
 - ここで使われているモデルは、次の二つ。
 - **CBOW(Continuous Bag-of-Word) モデル**
複数の語の集まりから、一緒に出現しそうな一つの語の確率を調べる。
 - **Skip-gram モデル**
一つの語が与えられた時、一緒に出現しそうな複数の語の確率を調べる。
-



どんな語が、与えられた語の近くに 埋め込まれるか？

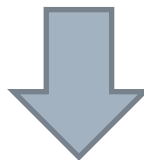
FRANCE	JESUS	XBOX	REDDISH	SCRATCHED	MEGABITS
AUSTRIA	GOD	AMIGA	GREENISH	NAILED	OCTETS
BELGIUM	SATI	PLAYSTATION	BLUISH	SMASHED	MB/S
GERMANY	CHRIST	MSX	PINKISH	PUNCHED	BIT/S
ITALY	SATAN	IPOD	PURPLISH	POPPED	BAUD
GREECE	KALI	SEGA	BROWNISH	CRIMPED	CARATS
SWEDEN	INDRA	PSNUMBER	GREYISH	SCRAPED	KBIT/S
NORWAY	VISHNU	HD	GRAYISH	SCREWED	MEGAHERTZ
EUROPE	ANANDA	DREAMCAST	WHITISH	SECTIONED	MEGAPIXELS
HUNGARY	PARVATI	GEFORCE	SILVERY	SLASHED	GBIT/S
SWITZERLAND	GRACE	CAPCOM	YELLOWISH	RIPPED	AMPERES

どんな語が、与えられた語の近くに埋め込まれるか？

- 似た意味を持つ言葉は、似たベクトルを持つ。
- 似た言葉で置き換えても、正しい文は、正しい文に変わる。

“a **few** people sing well”

正しい文



“a **couple** people sing well”

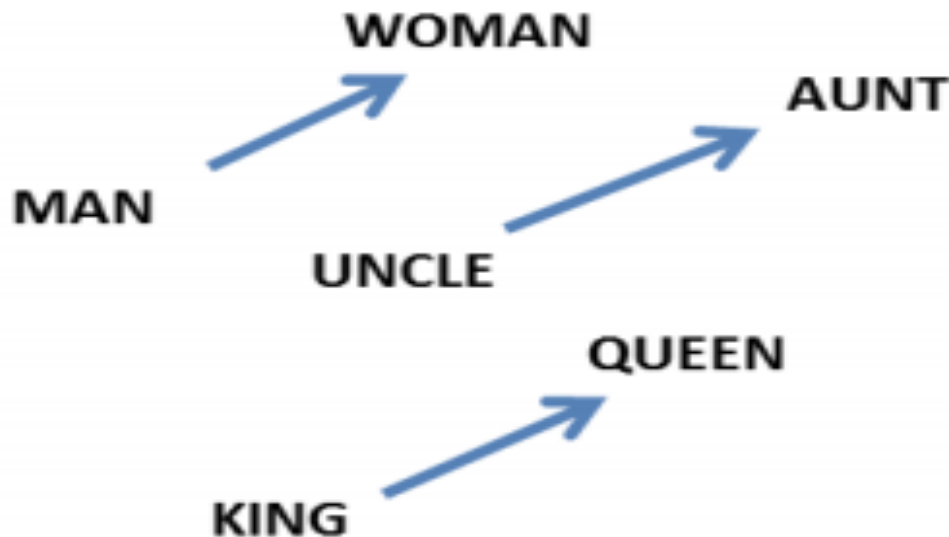
正しい文

- 意味が似ていなくても、同じクラスの言葉で置き換えても、正しい文は、正しい文に変わる。

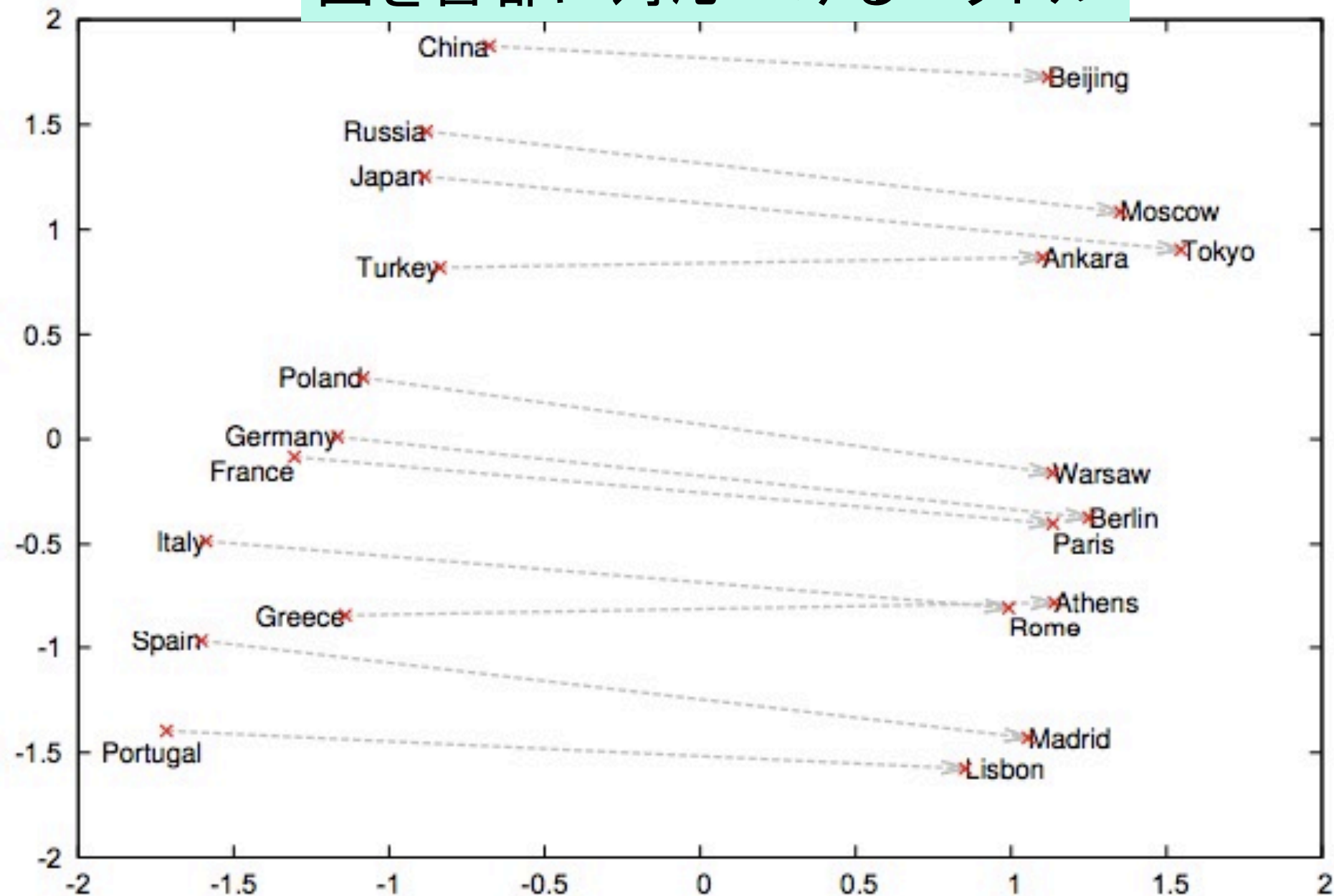
“the **wall** is **blue**” → “the **ceiling** is **red**”

意味を変換するベクトルは共通？

- Word Embeddingは、もっと面白い性質を持つ。下の図のように、男性から女性へのベクトルがあるように見える。
- $W(\text{"woman"}) - W(\text{"man"}) \approx W(\text{"aunt"}) - W(\text{"uncle"})$
 $W(\text{"woman"}) - W(\text{"man"}) \approx W(\text{"queen"}) - W(\text{"king"})$



国を首都に対応づけるベクトル



ベクトルで表現される関係の例

Relationship	Example 1	Example 2	Example 3
France - Paris	Italy: Rome	Japan: Tokyo	Florida: Tallahassee
big - bigger	small: larger	cold: colder	quick: quicker
Miami - Florida	Baltimore: Maryland	Dallas: Texas	Kona: Hawaii
Einstein - scientist	Messi: midfielder	Mozart: violinist	Picasso: painter
Sarkozy - France	Berlusconi: Italy	Merkel: Germany	Koizumi: Japan
copper - Cu	zinc: Zn	gold: Au	uranium: plutonium
Berlusconi - Silvio	Sarkozy: Nicolas	Putin: Medvedev	Obama: Barack
Microsoft - Windows	Google: Android	IBM: Linux	Apple: iPhone
Microsoft - Ballmer	Google: Yahoo	IBM: McNealy	Apple: Jobs
Japan - sushi	Germany: bratwurst	France: tapas	USA: pizza

文法的関係を表すベクトルも存在する

Type of relationship	Word Pair 1		Word Pair 2	
Common capital city	Athens	Greece	Oslo	Norway
All capital cities	Astana	Kazakhstan	Harare	Zimbabwe
Currency	Angola	kwanza	Iran	rial
City-in-state	Chicago	Illinois	Stockton	California
Man-Woman	brother	sister	grandson	granddaughter
Adjective to adverb	apparent	apparently	rapid	rapidly
Opposite	possibly	impossibly	ethical	unethical
Comparative	great	greater	tough	tougher
Superlative	easy	easiest	lucky	luckiest
Present Participle	think	thinking	read	reading
Nationality adjective	Switzerland	Swiss	Cambodia	Cambodian
Past tense	walking	walked	swimming	swam
Plural nouns	mouse	mice	dollar	dollars
Plural verbs	work	works	speak	speaks

RNNの文法認識能力

BengioからMikolovに至る流れは、文理解では、「意味」featureの取り込みが重要であることを示しているのだが、もう一方で、RNNが、高い文法理解能力を持つことの理解も進む。詳しくは、前回のマルレク「RNNの基礎」を参照されたい。

<https://goo.gl/cvDgQX>

ただ、そこには、いくつかの問題がある。

それについては「方法論上の問題」を参照してほしい。

RNNの能力について -- LSTM原論文を読む

"Long Short Term Memory"

Sepp Hochreiter et al.

<https://goo.gl/sDAq81>

1997年



RNNによる文の生成

"Generating Text with Recurrent Neural Networks"

Ilya Sutskever et al.

<http://goo.gl/vHRHSn>

2011年



RNNの驚くべき能力

*"The Unreasonable Effectiveness
of Recurrent Neural Networks"*

Andrej Karpathy

<http://goo.gl/mNqwCv>

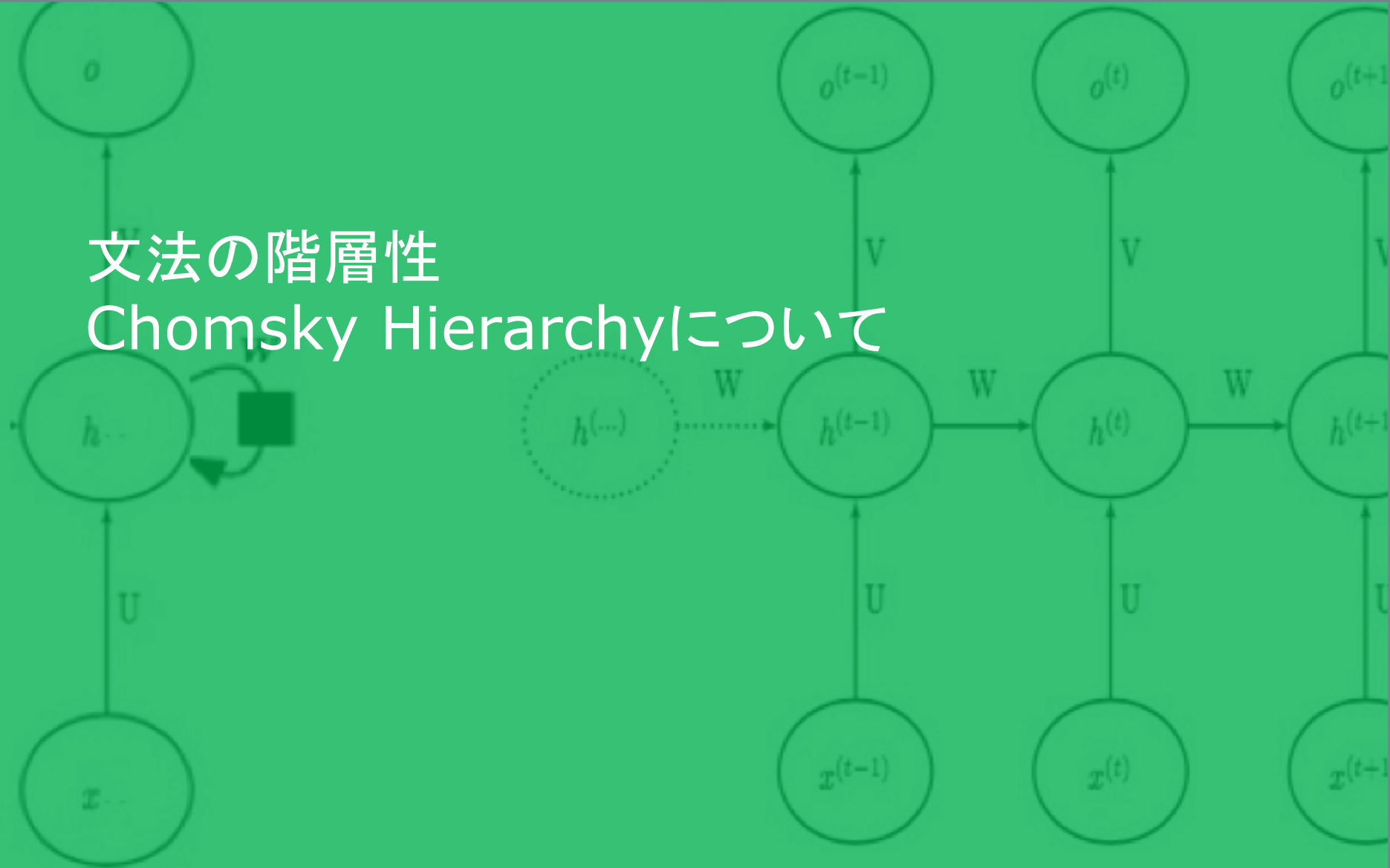
2015年



マシンは、簡単な文法を理解できる

- これらの取り組みは、マシンが、例えば、C言語の構文や、LaTeXの構文は、ほぼ完璧に学習していること示しており、興味深いものだ。ただし、自然言語の生成では、いくつかの破綻が見られる。
- このことは、プログラム言語の文法規則が、基本的には「文脈自由文法(レベル2)」で、自然言語の文法規則である「文脈依存文法(レベル1)」よりも単純であることの表れとして理解できる。より単純な、有限オートマトンで表わされる「正規文法(レベル3)」の構造を、その出力から推定する問題が、膨大な計算を必要とするように、ここでは、膨大な計算が行われている。
- もちろん、このアプローチの最大の問題は、文の「意味」を捉え損ねていることである。文字通り「意味がない」のだ。

文法の階層性 Chomsky Hierarchyについて



Chomsky Hierarchy

- 形式的言語の形式的文法は、次のような階層をなすことが知られている。これをChomsky Hierarchyと呼ぶ。
 - タイプ-0 文法は、全ての形式文法を包含する。
 - タイプ-1 文法は、文脈依存言語を生成する。
 - タイプ-2 文法は、文脈自由言語を生成する。
 - タイプ-3 文法は、正規言語を生成する。
- 機械が、この階層の中に位置付けられるどの形式的な文法を理解・学習できたかを考えることができる。

Chomsky Hierarchy

Grammar	Languages	Automaton	Production rules (constraints)
Type-0	Recursively enumerable	Turing machine	$\alpha \rightarrow \beta$ (no restrictions)
Type-1	Context-sensitive	Linear-bounded non-deterministic Turing machine	$\alpha A \beta \rightarrow \alpha \gamma \beta$
Type-2	Context-free	Non-deterministic pushdown automaton	$A \rightarrow \gamma$
Type-3	Regular	Finite state automaton	$A \rightarrow a$ and $A \rightarrow aB$

https://en.wikipedia.org/wiki/Chomsky_hierarchy

Chomsky Hierarchyと Deep Learningでの文法理解の取り組み

- 機械が、この階層の中に位置付けられるどの形式的な文法を理解・学習できたかを考えることができる。
 - Hochreiterの仕事は、機械が、有限オートマトンで表現される正規文法を、理解・学習できることを示唆している。
 - Karpathyの仕事は、機械が、BNF(Backus-Naur Form)で表現されるような、Latex, プログラム言語の文脈自由文法を、理解・学習できることを、強く示唆している。
-

Chomsky Hierarchyと Deep Learningでの文法理解の取り組み

	文法	例	Deep Learning
Type-0	帰納的可算	チューリングマシン	
Type-1	文脈依存文法	自然言語*	?
Type-2	文脈自由文法	プログラム言語	Karpathy 2015年
Type-3	正規文法	有限オートマトン	Hochreiter 1997年

自然言語の多くの文法は、Context-Freeで記述できる。
Mildly Context Sensitive Languages', proposed
by Aravind Joshi.

Chomsky Hierarchyと Deep Learningでの文法理解の取り組み

	文法	例	Deep Learning
Type-0	帰納的可算	チューリングマシン	
Type-1	文脈依存文法	自然言語	Google Neural Machine Translation System ? 2016年
Type-2	文脈自由文法	プログラム言語	Karpathy 2015年
Type-3	正規文法	有限オートマトン	Hochreiter 1997年

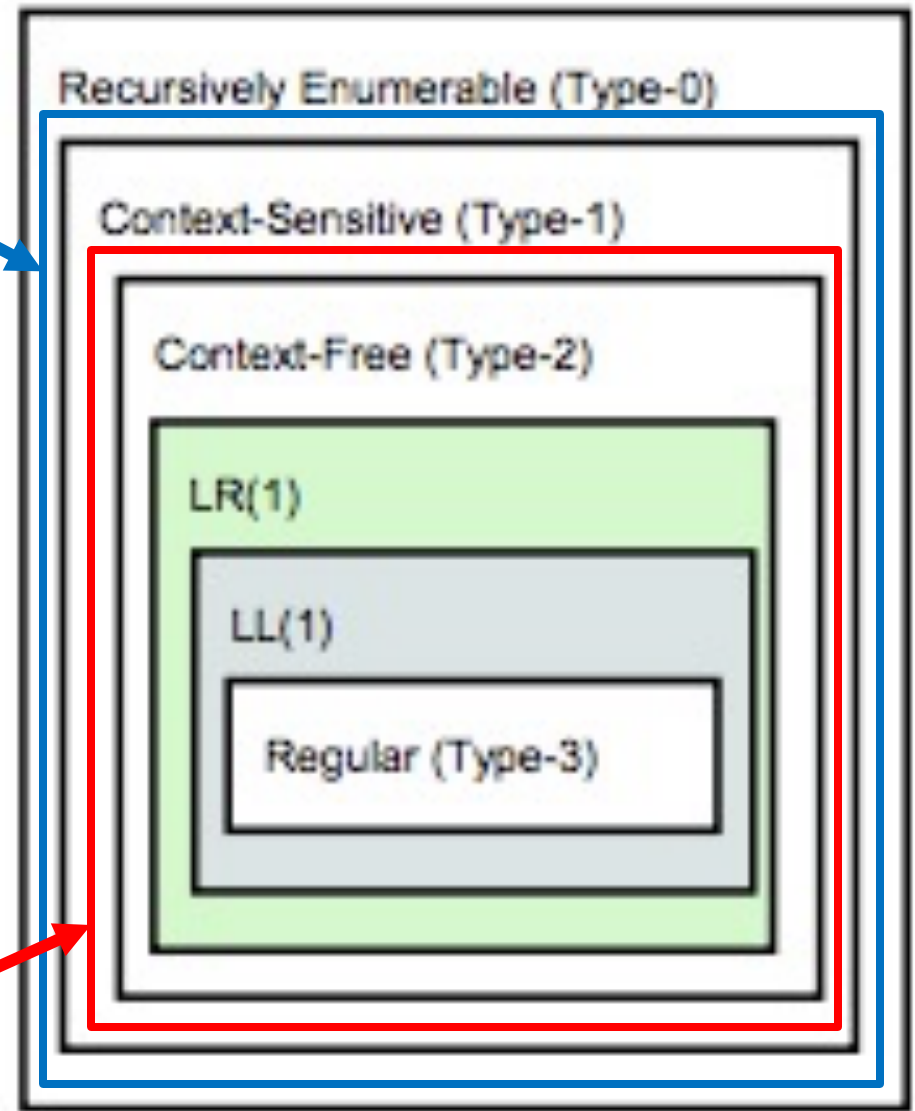
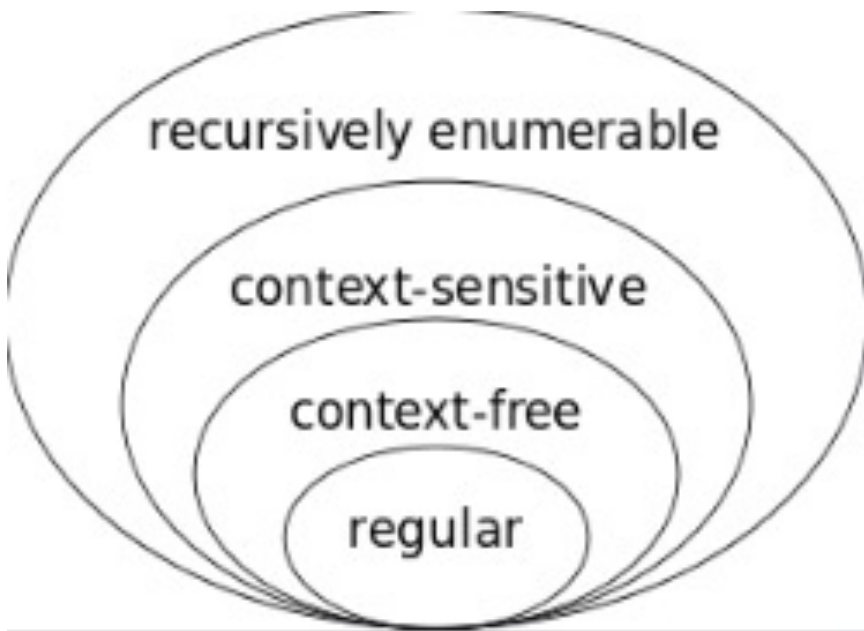
自然言語の文法は、この階層のどこに位置付けられるか？

- 問題は、自然言語の文法が、この階層のどこに位置付けられるかということである。
 - 元の4階層の区分は、基本的だが粗いものである。Context-Freeな文法の中にも、LRやLLがあるように。
 - かつてのChomskyは、英語は Regular 文法ではないと明確に述べたが、それが Context-Freeであるかどうかは、明確には判断を下していないようにも見える。ただ、自然言語は、Context-Free性を満たさないというのが、大方の意見のように見える。それは、Context-Freeを超えた、Mildly Context-Sensitive な文法だという。(次の図の、赤い四角の部分)
-

Minimalist ProgramとMerge

- 現在のChomskyのMinimalist Programでは、自然言語の文法を、文法の階層の中に位置付けようとする問題意識は、あまりないようにも見える。
 - ただ、Minimalist Programの中心的概念であるMergeについては、それがRecursiveであることが強調されている。Recursiveというのは、Recursive Enumerableの一部分で、決定可能なものである。(図の青い四角の部分)
-

**Recursive Language
"Merge" is recursive**



自然言語？

Mildly Context Sensitive Languages

方法論上の問題

機械が理解・学習できる文法のクラスを、Chomsky階層の中で位置付けようとする試みは、興味深いものだが、そこには、大きな方法論的な問題がある。

Hochreiterが、単純なRNNでは、有限オートマトンの生み出すReverse文法の認識が「できず」、LSTMなら、それが「できる」と主張したことを振り返ってみよう。

機械の能力の階層性？

- 第一。ここでは、単純なRNNとLSTMという、二つの機械の能力が問題になっている。機械の構成の違いが、機械の能力の違いを生み出しているのは確かなのだが、我々は、二つの機械の出力をみてその能力を判断しているのであって、その機械の構成から、その能力の予測ができているわけではない。
- Chomsky階層の各クラスの生成能力と相互の包含関係を、我々は、形式的に証明できる。(cf. Turing MachineとAutomatonの理論) ただ、ニューラル・ネットワークから構成される様々な機械のモデルの、それぞれの生成能力とその相互の関係を、我々はきちんと把握できてはいない。
- Google翻訳のLSTM8段重ねx2の構成を、すごいとは思いつつ、どうすごいのかをうまく語れないのだ。(結果から見て、すごいとは言えるのだが)

機械に「できること」「できないこと」の判断

- 第二。LSTMがRever文法の認識が「できる」というのは、相当数の学習の後で、正解率がほぼ100%になるということであり、他の機械に、それが「できない」というのは、相当数の学習を繰り返しても、正解率が向上しないということである。**それは、双方ともに、統計的・確率的な判断である。**それは避け難いことだし、そのことの意味をよく考える必要がある。ただし、それは先に見たように、機械の構成で基本的には、規定されている。
- 相当数といったが、どのくらい学習を繰り返すべきかは、暗黙のうちに了解されている前提がある。それは、10年とかではない。（当たり前だ。でも、人間が一人前になるには、10年の経験が必要という分野だってあることはある。話が横道にそれたが、だから、「たくさんのGPUと膨大なデータと十分な学習時間があれば、精度をいくらでもあげられる」という無邪気な楽観論に、僕は、懐疑的だ。問題は、どんなマシンを作るかだ。）

網羅的な検証は不可能

- 第三。Hochreiterのマシンは、Revere文法を認識したというが、それは正規文法を認識したことにはならない。実際に示されたのは、正規文法の一部であるRevere文法の認識能力である。
- 僕は、LSTMは正規文法を認識できると考えているのだが、与えられたデータ(例えそれがいかに巨大なものであろうと)で学習したニューラル・ネットの能力を、その出力で判断するというアプローチを取る限り(それが、普通のディープラーニングのアプローチだ)、機械の能力をChomsky Hierarchyにマッピングすることは難しいだろう。現実には不可能な網羅的な検証は、形式的・抽象的定義の飛翔に追いつくことはできないのだ。
- 正確に言えば、「Hochreiterのマシンで認識できない、~~Revere文法を満たす文は存在しないこと~~」を、我々は、証明してはいないのだ。

楽観的な展望

- いろいろ、悲観的なことを書いたが、そこで終わっていると、気持ちが悪い。
- 例えば、正規文法を認識するニューラル・ネットを、スクラッチから構成することは、難しくはない。状態の遷移図を、そのままニューロンの役割に置き換えればいいはずだ。
- 問題は、例えば、LSTMが、その学習をつうじて、こうした回路を内部に、試行錯誤を通じて近似的に構成する能力を持つことを、きちんといえてないことだ。そのことは、経験的には当然のようにも思えるが、詳しい過程やその背後の理論構造を、僕は知らない。
- そのことを明らかにできれば、文法の階層だけではなく、機械の構成の違いに基づく機械の階層について語るができるだろう。

Googleニューラル機械翻訳

Googleニューラル機械翻訳 GNMTの登場は、エポック・メイキングなものである。同時に、それは、ディープラーニングの自然言語処理技術の発展の自然な継承でもあり、その集大成でもある。(その意味では、こうした方向での更なる前進は、パラレル・データをさらに増やす以外は難しいということなのかもしれない。)

ここでは、GNMTに先行した技術との関連を重点的に見る。

人間と機械翻訳のギャップに橋をかける Googleのニューラル機械翻訳システム

***"Google's Neural Machine Translation
System: Bridging the Gap between Human
and Machine Translation"***

Yonghui Wu et al.

<https://goo.gl/YqIAAW>

2016年



Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi
yonghui,schuster,zhifengc,qvl,mnorouzi@google.com

Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey,
Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser,
Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens,
George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa,
Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, Jeffrey Dean

Abstract

Neural Machine Translation (NMT) is an end-to-end learning approach for automated translation, with the potential to overcome many of the weaknesses of conventional phrase-based translation systems. Unfortunately, NMT systems are known to be computationally expensive both in training and in translation inference – sometimes prohibitively so in the case of very large data sets and large models. Several authors have also charged that NMT systems lack robustness, particularly when input sentences contain rare words. These issues have hindered NMT's use in practical deployments and services, where both accuracy and speed are essential. In this work, we present GNMT, Google's Neural Machine Translation system, which attempts to address many of these issues. Our model consists of a deep LSTM network with 8 encoder and 8 decoder layers using residual connections as well as attention connections from the decoder network to the encoder. To improve parallelism and therefore decrease training time, our attention mechanism connects the bottom layer of the decoder to the top layer of the encoder. To accelerate the final translation

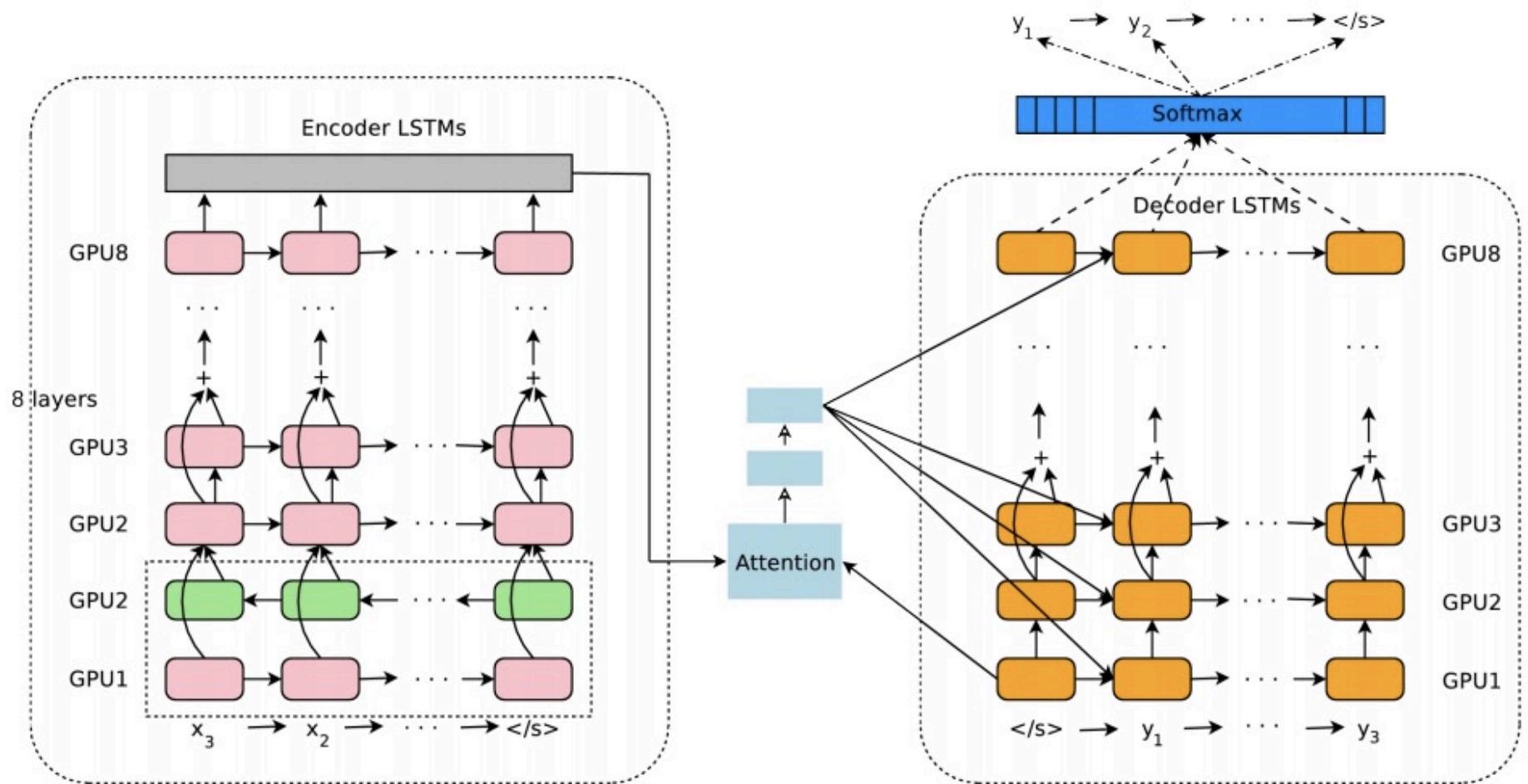
<https://arxiv.org/pdf/1609.08144.pdf>

Googleニューラル機械翻訳 システムの概観

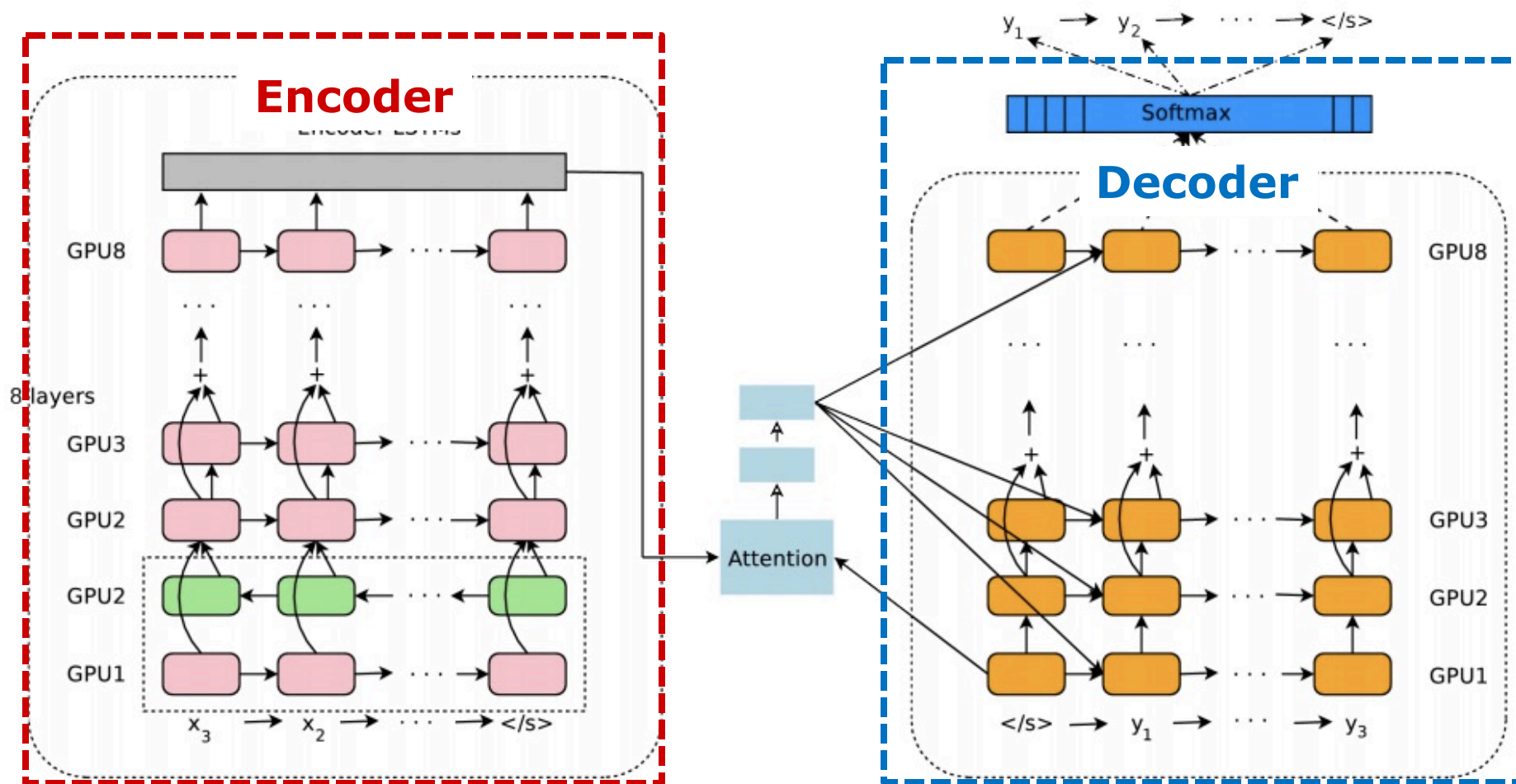
最初に、GNMTのシステムを概観する。

そのいくつかの要素、Residue Connection, Bi-Directional Encoder, Quantifization等については、小論では、詳しく述べることはできなかった。

GNMTのアーキテクチャの概念図

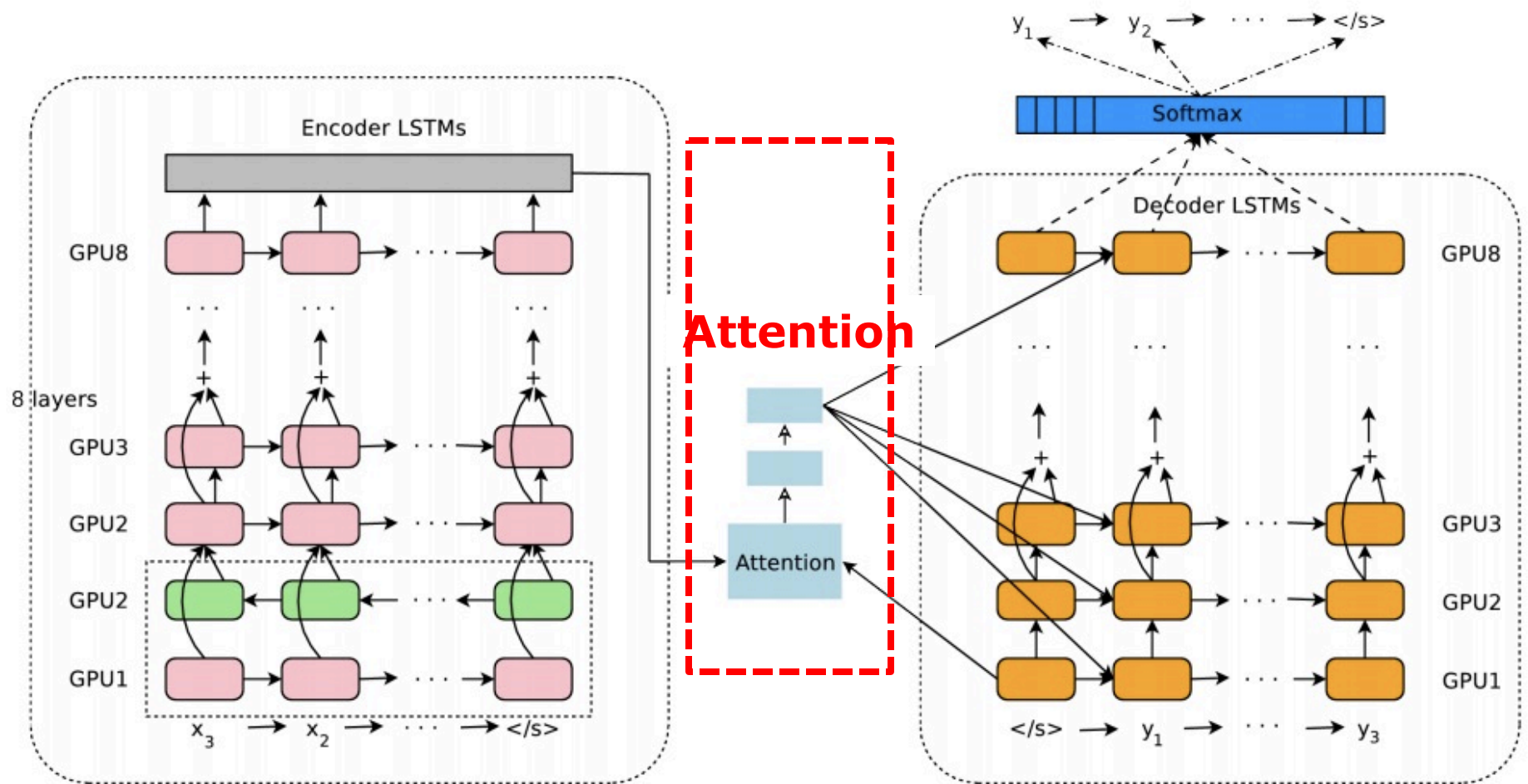


Encoder / Decoder



左側に、LSTMを8段重ねにした Encoder LSTMがあり、
右側には、同じくLSTMを8段重ねにした Decoder LSTMがある。

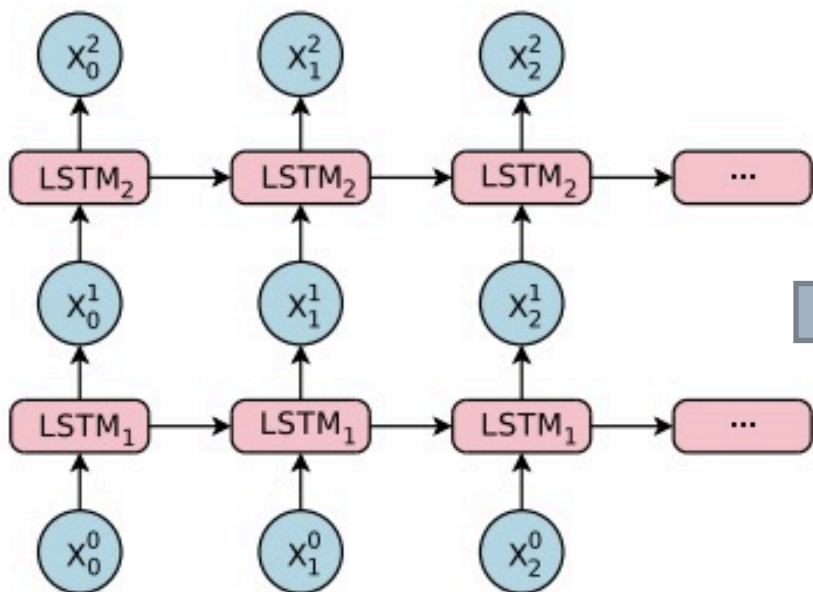
Attention Mechanism



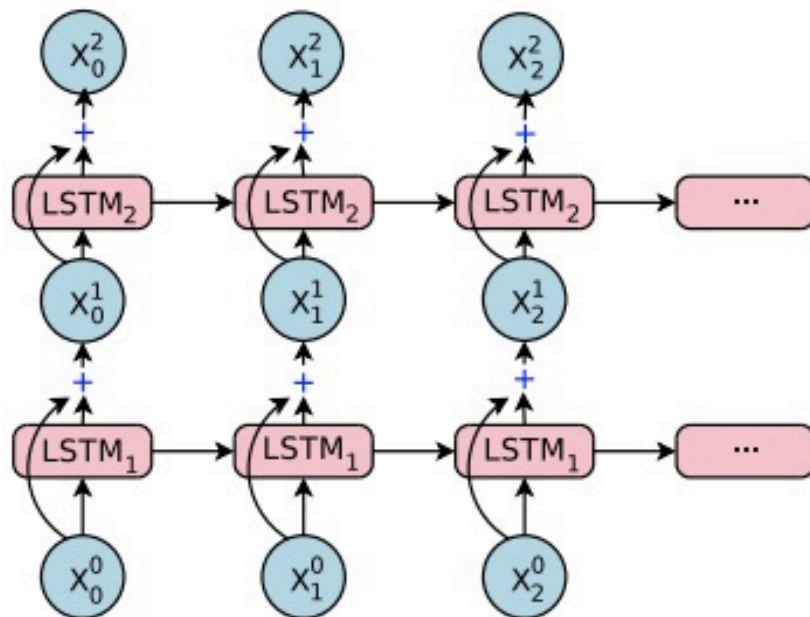
EncoderとDecoderの間に、Attentionと記された領域がある。
ここからの出力Attention Context は、Decoderのすべての
ノードに供給されている。

Residue Connection

通常のStacked LSTM

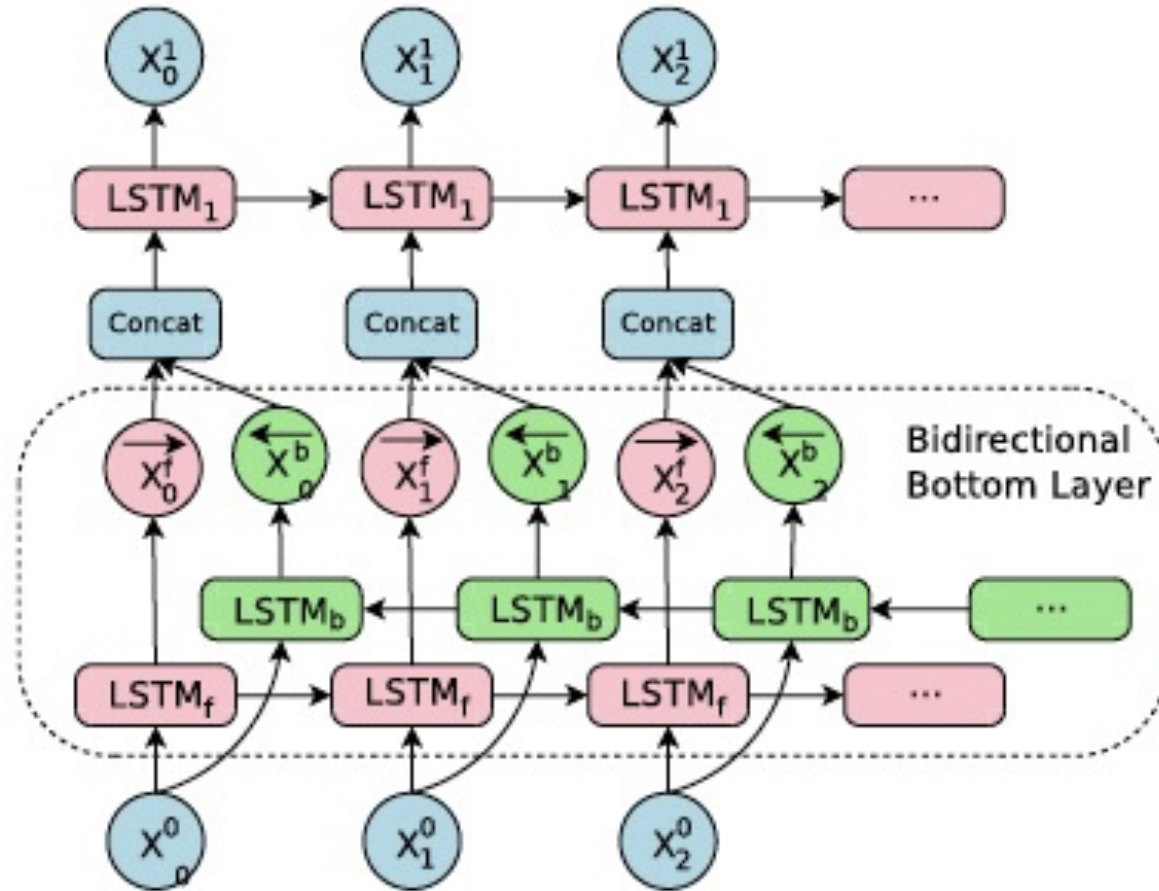


GNMT



細かく見ると、LSTMの段の積み重ねに、特徴があるのがわかる。積み重ねられたLSTMは、一つ下のLSTMからの出力を受け取るだけでなく、もう一つ下のLSTMからの出力をも受け取っている。これを Residue Connectionと呼ぶ。

Bi-directional Encoder for First Layer



Encoder側の一番下の方の二つのLSTMの処理が、逆向きに走っている。

Quantizable Model and Quantized Inference

LSTM

with Residual
connection

$$\mathbf{W} = [\mathbf{W}_1, \mathbf{W}_2, \mathbf{W}_3, \mathbf{W}_4, \mathbf{W}_5, \mathbf{W}_6, \mathbf{W}_7, \mathbf{W}_8]$$

$$\mathbf{i}_t = \text{sigmoid}(\mathbf{W}_1 \mathbf{x}_t + \mathbf{W}_2 \mathbf{m}_t)$$

$$\mathbf{i}'_t = \tanh(\mathbf{W}_3 \mathbf{x}_t + \mathbf{W}_4 \mathbf{m}_t)$$

$$\mathbf{f}_t = \text{sigmoid}(\mathbf{W}_5 \mathbf{x}_t + \mathbf{W}_6 \mathbf{m}_t)$$

$$\mathbf{o}_t = \text{sigmoid}(\mathbf{W}_7 \mathbf{x}_t + \mathbf{W}_8 \mathbf{m}_t)$$

$$\mathbf{c}_t = \mathbf{c}_{t-1} \odot \mathbf{f}_t + \mathbf{i}'_t \odot \mathbf{i}_t$$

$$\mathbf{m}_t = \mathbf{c}_t \odot \mathbf{o}_t$$

Input Gate

Forget Gate

Output Gate

$$s_i = \max(\text{abs}(\mathbf{W}[i, :]))$$

$$\mathbf{WQ}[i, j] = \text{round}(\mathbf{W}[i, j] / s_i \times 127.0)$$

All accumulator values ($\mathbf{c}_t^i$ and $\mathbf{x}_t^i$) are represented using 16-bit integers
All matrix multiplications are done using 8-bit integer multiplication
All other operations (Activation, elementwise operation) 16bit

Encoder / Decoder

HintonのAutoencoder

こうした特徴を持つGNMTのうち、ここでは、まず、Encoder/Decoderの枠組みに注目しよう。

ニューラル・ネットワークの中で、Encoder-Decoderの枠組みを提案した基本的な論文は、Hintonの 2006年の **”Reducing the Dimensionality of Data with Neural Networks”** である。<https://goo.gl/Ki41L8>

2006年 Hinton ”Reducing the Dimensionality of Data with Neural Networks” <https://goo.gl/Ki41L8>

「高次元の入力ベクターを再構成する小さな中央層を持つ多層のニューラル・ネットワークを訓練することで、高次元のデータを低次元のコードに変換することができる。

このようなAuto Encoderネットワークの重みを調整するのに勾配降下法を利用できる。ただし、それは、重みの初期値が、良好な解に近い場合にのみうまく働く。

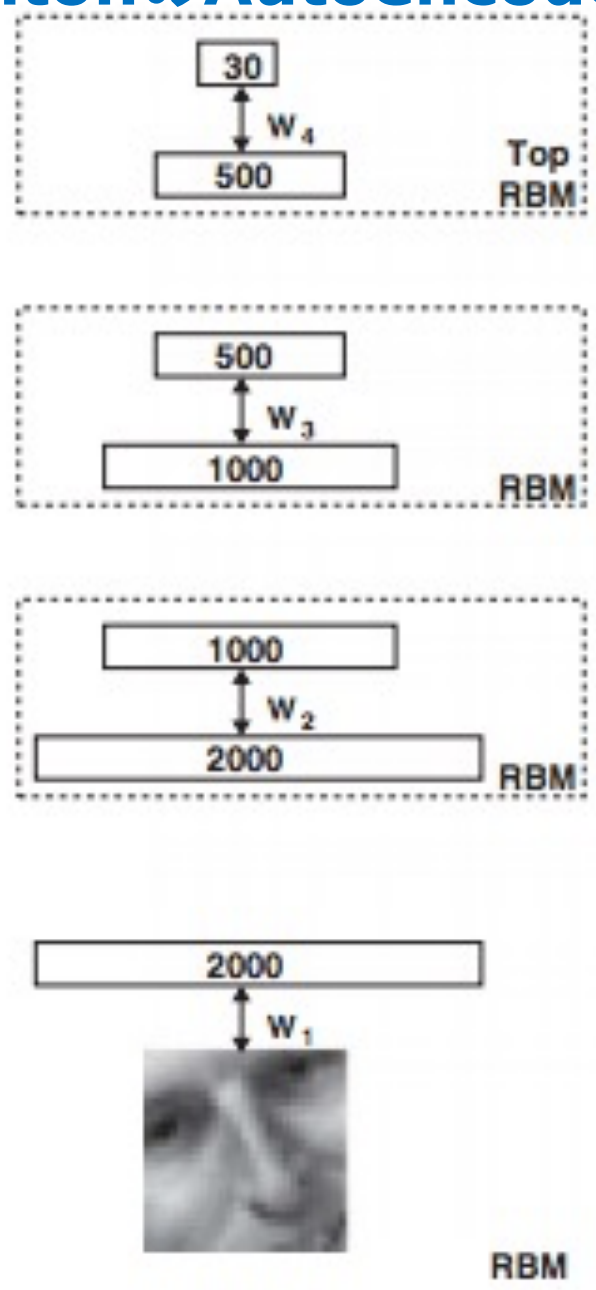
我々は、ディープAuto Encoderネットワークが、低次元のコードを学習することを可能にする重みの初期化の効率的な方法について述べる。この方法は、データの次元を下げるツールとしての主成分分析より、ずっと優れている。」

Autoencoder

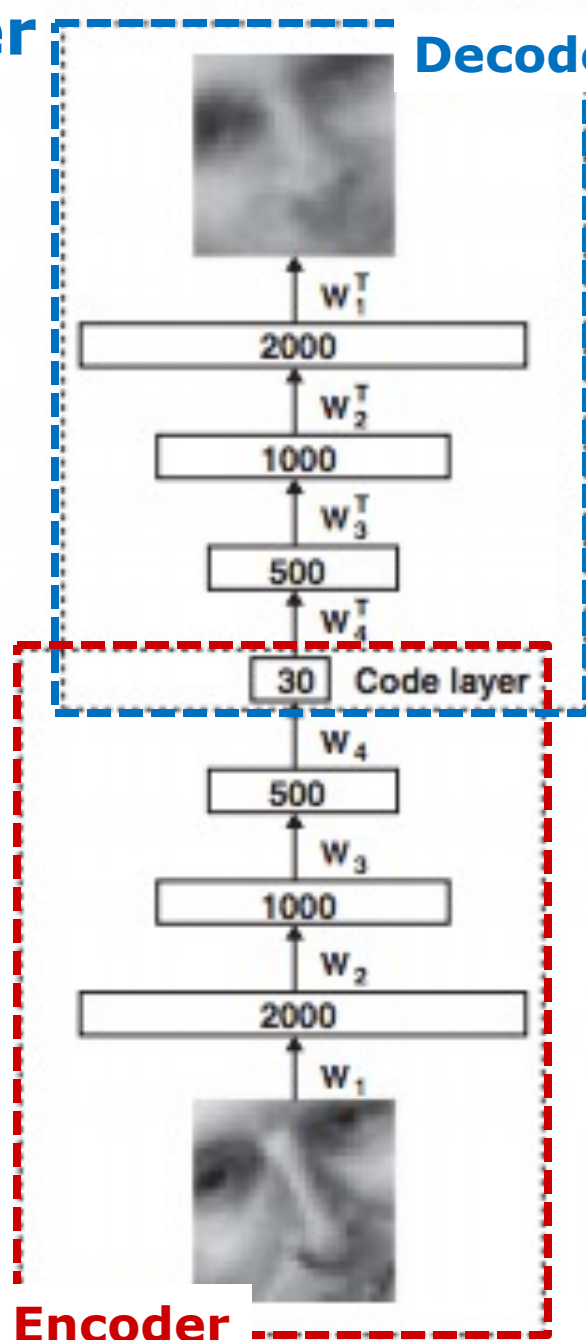
- 次の図の中央、下の方の赤い枠がEncoderである。Encoderは、2000次元のベクトル(2000 pixelの画像データ)を、30次元のベクトルに変える。上の方の青い枠のDecoderは、この30次元のベクトルから、2000次元のベクトルを生成する。(こうして、画像が復元される)論文では、このAutoencoderを微調整する方法が示されているのだが(下図の右側。画質が改善されている)、それについては割愛する。
 - 注目して欲しいのは、ここでは、入力に与えられたデータ自身が、教師用のデータの役割を果たすので、その意味では、ラベルづけられた教師用のデータを必要としないということ。Autoencoderとは、「自己エンコーダ」の意味である。
-

HintonのAutoencoder

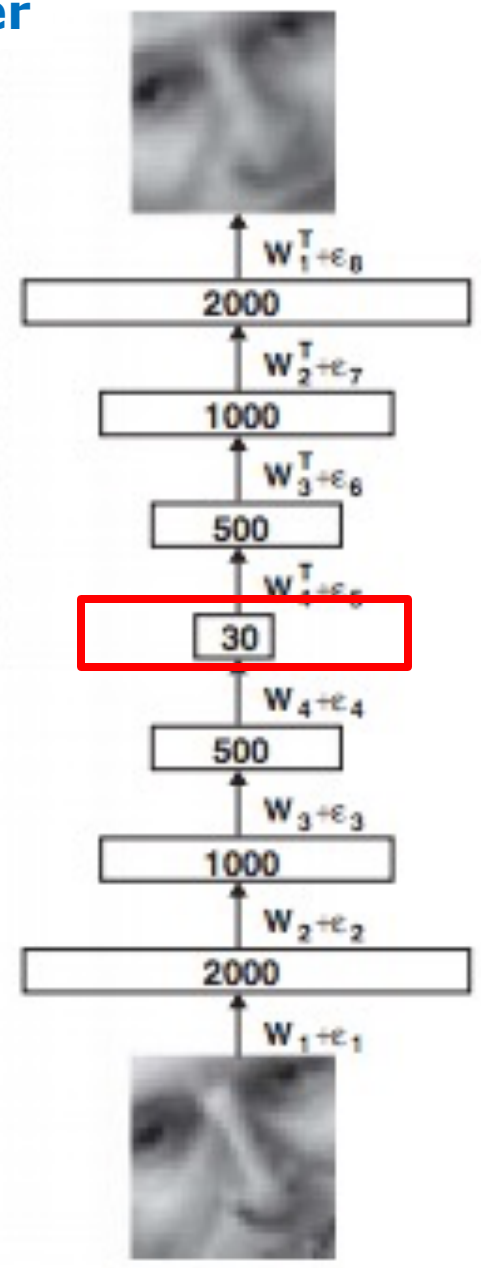
Decoder



Pretraining



Unrolling

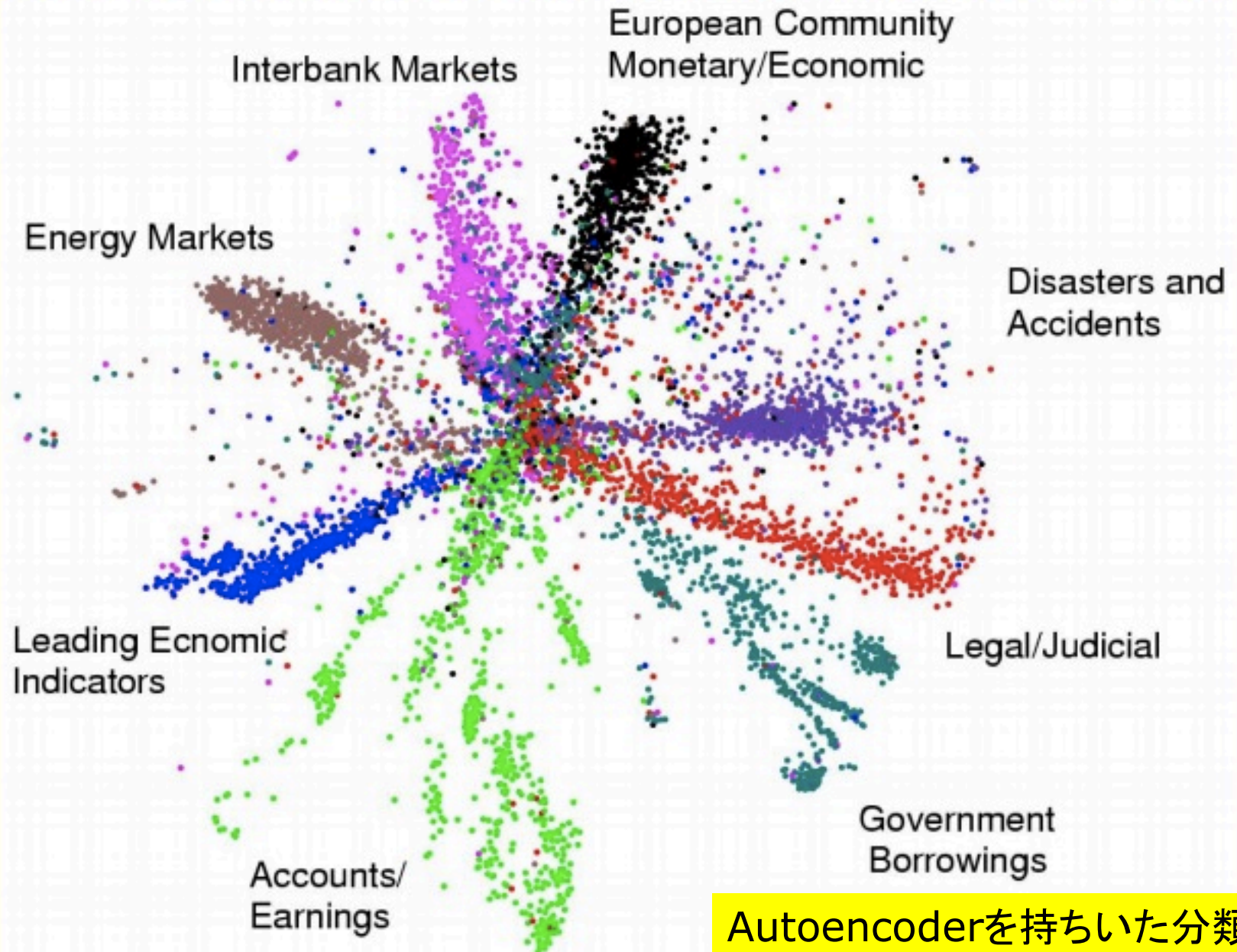


Fine-tuning

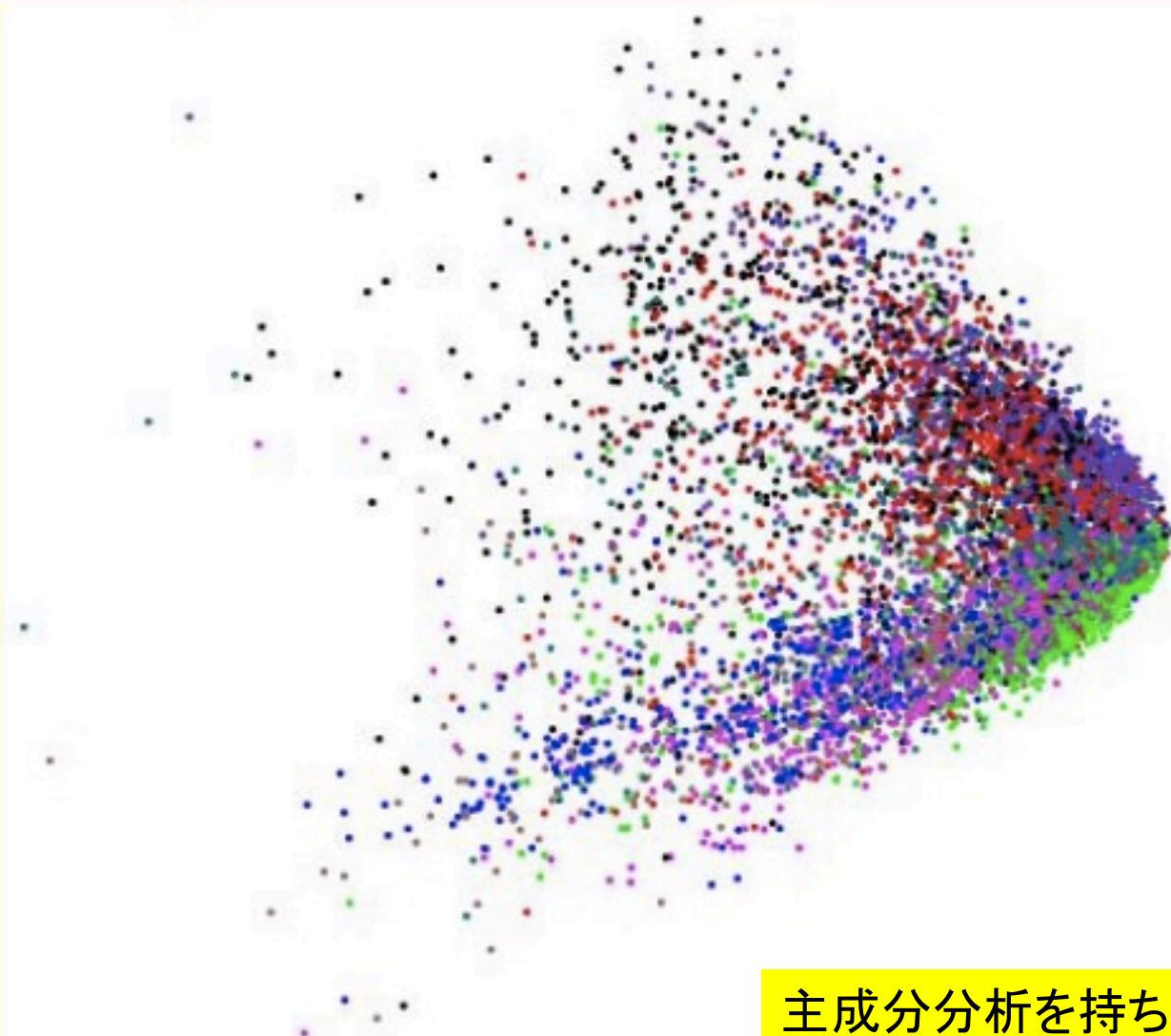
書籍の分類へのAutoencoderの利用

- 先の例は、「画像圧縮技術」の一種として理解してもいいのだが、この論文で、Hintonは、もっと面白い例を紹介している。書籍の分類に、このAutoencoderを使おうというものである。
- よく使われる単語を2000個ほど選ぶ。ある本にこれらの単語が何個含まれているかをカウントする。そうすると、ある本に2000次元の整数からなるベクトルを対応づけることができる。このベクトルをAutoencoderの入力に与えて、Autoencoderがこのベクトルを出力に再現できるように訓練をする。Autoencoderの中央のボトルネックの部分で、10次元のベクトルにすると、ある本に10個の数字を対応づけることができる。
- Hintonは、40万冊のビジネス書を対象に、この方法で得られた10個の数字が、書籍の分類に有効かどうかを実験した。結果を、二次元に可視化したものが、次の図だ。見事に、分類に成功している。

First compress all documents to 2 numbers.
Then use different colors for different document categories



First compress all documents to 2 numbers using a type of PCA
Then use different colors for different
document categories



主成分分析を持ちいた分類

Semantic hashing (意味的ハッシング)

- 重要なことは、「画像」と「書籍」では、対象のデータの性質はまるで異なるのだが、Autoencoderは、そのいずれに対しても、高次元のデータを低次元のデータに変換しているということである。別の言葉で言えば、それは、対象の高次元のデータから、低次元のデータを、元の情報のエッセンスとして取り出しているのである。
- Hintonは、こうしたAutoencoderの働きを、**Semantic hashing (意味的ハッシング)**と呼んでいる。SHA-1のようなハッシングでは、ハッシュ化されたデータから元のデータを復元することは不可能なのだが、Semantic hashingされたデータは、データの次元は低いものの、元の情報の中核部分を保持している。

Encoder-Decoderの機械翻訳への応用

HintonのAutoencoderは、そのままの形では、機械翻訳に利用されることはなかったと思う。ただ、2014年に、Ilya Sutskever らは、RNN(ここでは多層のLSTM)の持つシーケンスをシーケンスに変換する能力が、機械翻訳に応用できるという論文 **"Sequence to Sequence Learning with Neural Networks"** <https://goo.gl/U3KtxJ> を発表する。

2014年 Sutskever et al. “**Sequence to Sequence Learning with Neural Networks**” <https://goo.gl/U3KtxJ>

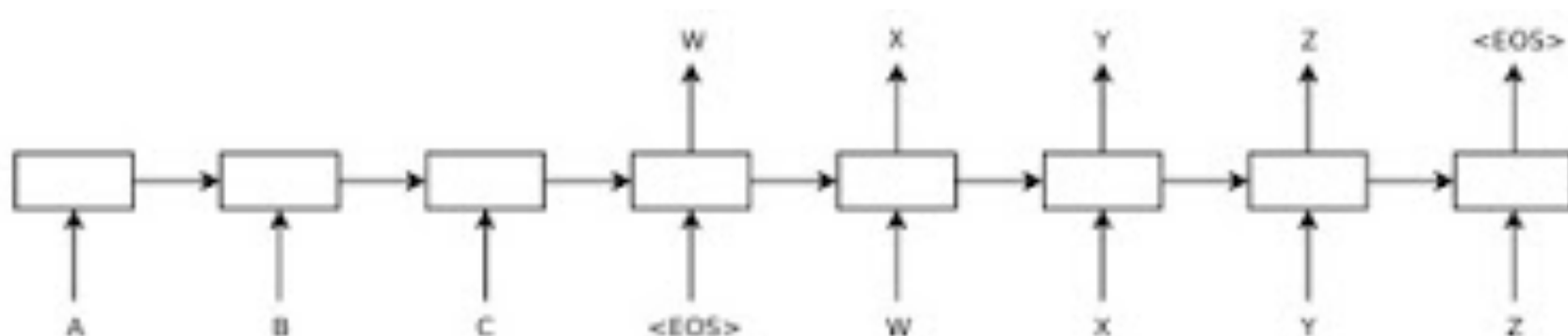
「ディープニューラルネットワーク(DNN)は、難しい学習課題でも優れたパフォーマンスを達成する強力なモデルである。DNNは、ラベル付けられた訓練用のデータが利用可能な時には、いつもうまく機能するというものの、シーケンスをシーケンスにマップすることに、DNNを利用することはできない。

我々は、この論文で、シーケンスを学習するエンド・トゥ・エンドの汎用のアプローチを提示する。そこでは、シーケンスの構造に最小限の前提しか課していない。

我々の方法では、入力のシーケンスを固定次元のベクトルにマップするのに、多層のLong Short-Term Memory(LSTM)を利用する。その後、別の深いLSTMが、このベクトルから目的のシーケンスをデコードする。」

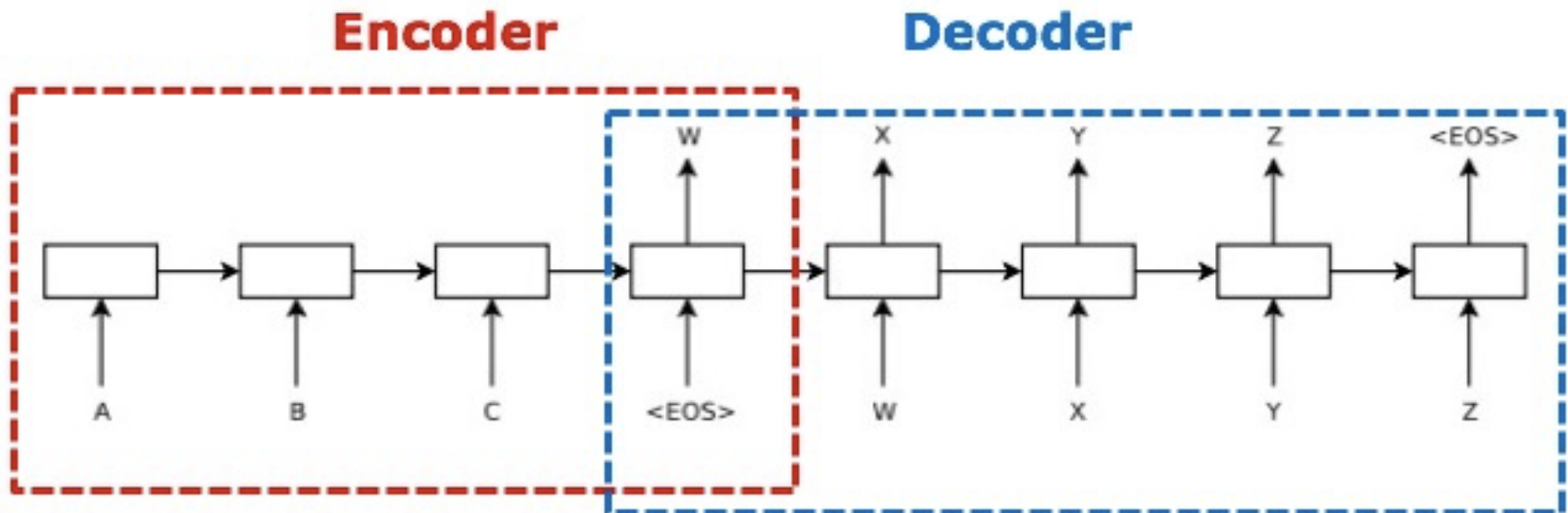
Sequence to Sequence

□ この論文の次の図を見て欲しい。

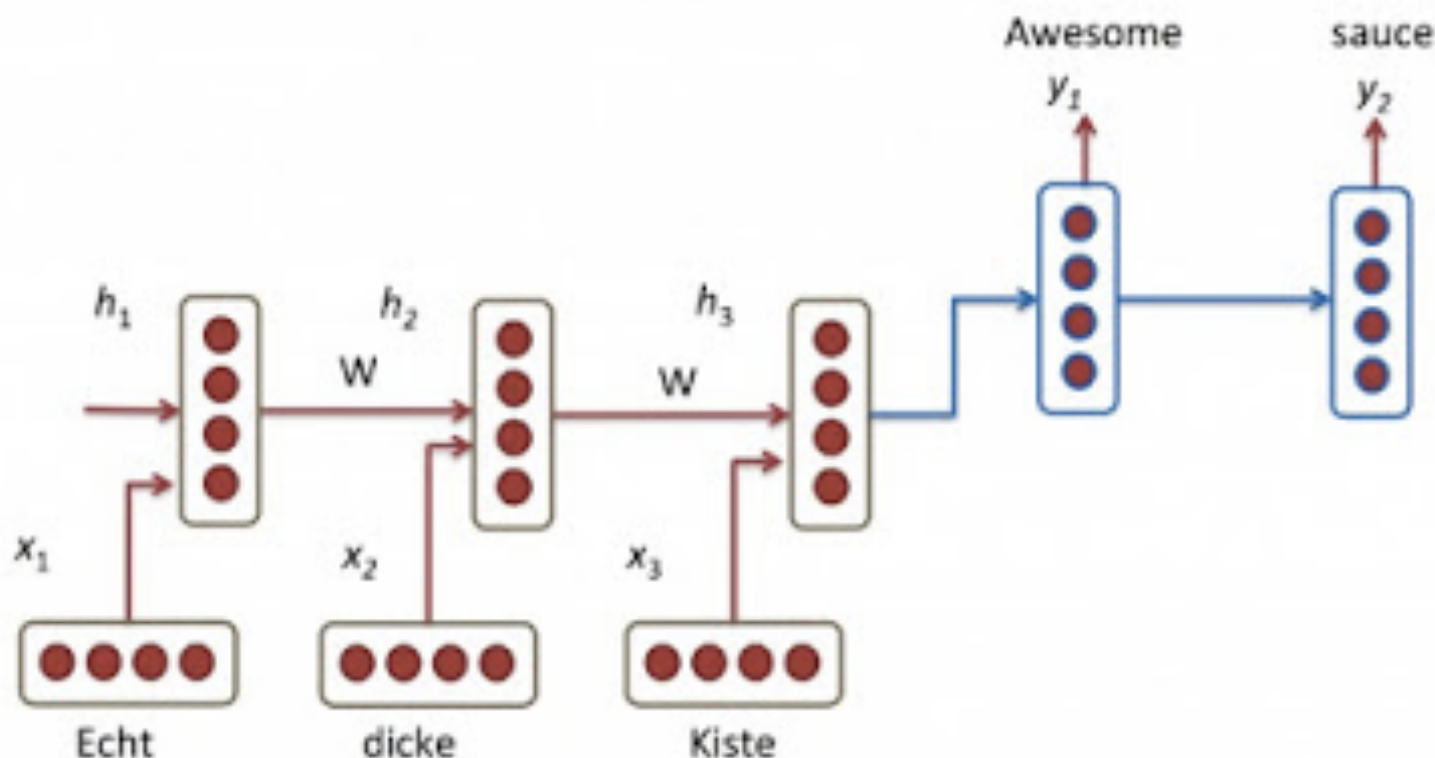


□ この図は、このシステムが、ABCというシーケンスが与えられた時、xyzというシーケンスを返すことを表している。<EOS>は、End of Sequence でシーケンスの終わりを表す特別な記号である。(これが、シーケンスの構造に課せられた「最小限の前提」である。)

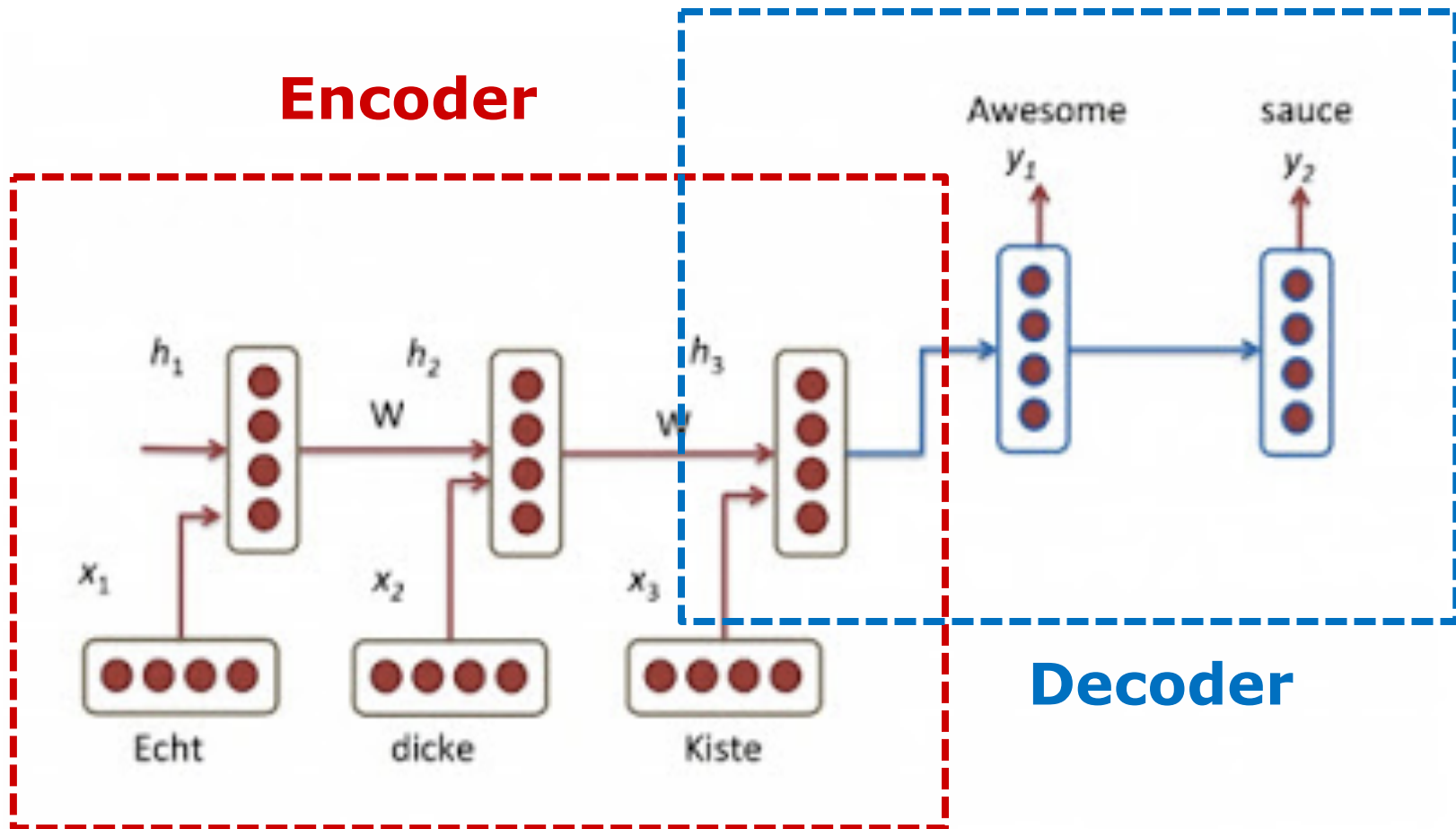
- これが、先に見たEncoder-Decoderのパターンであることは、次のようにしてわかる。
- 先行するLSTM群は、入力シーケンスABCを受け取って、それを固定長のベクトル w に変換している。後行のLSTM群は、そのベクトル w を受け取って、それから出力シーケンスxyzを生成する。すなわち、先行のLSTM群をEncoder、後行のLSTM群をDecoderと考えることができる。中間に生成され、両者で共有される w は、先のHintonのボトルネック部だと思えばいい。



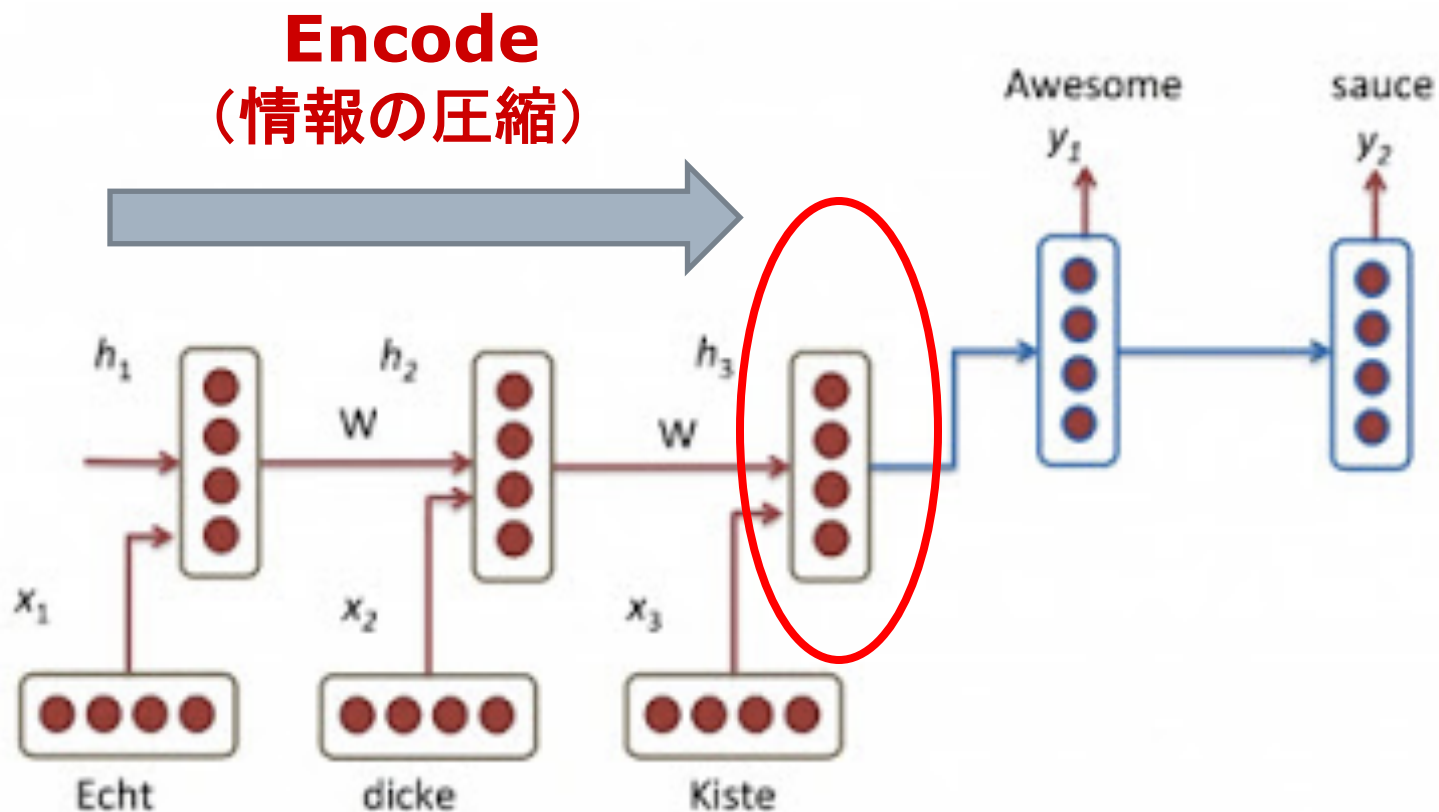
- 次の図(<https://goo.gl/JGckBP> から)は、こうしたメカニズムで、RNNが、独文の "Echt dicke Kiste" を英文の "Awesome sauce" に翻訳する様子を表している。(ここでは、文章の終わりを表す<EOS>は、省略されている)



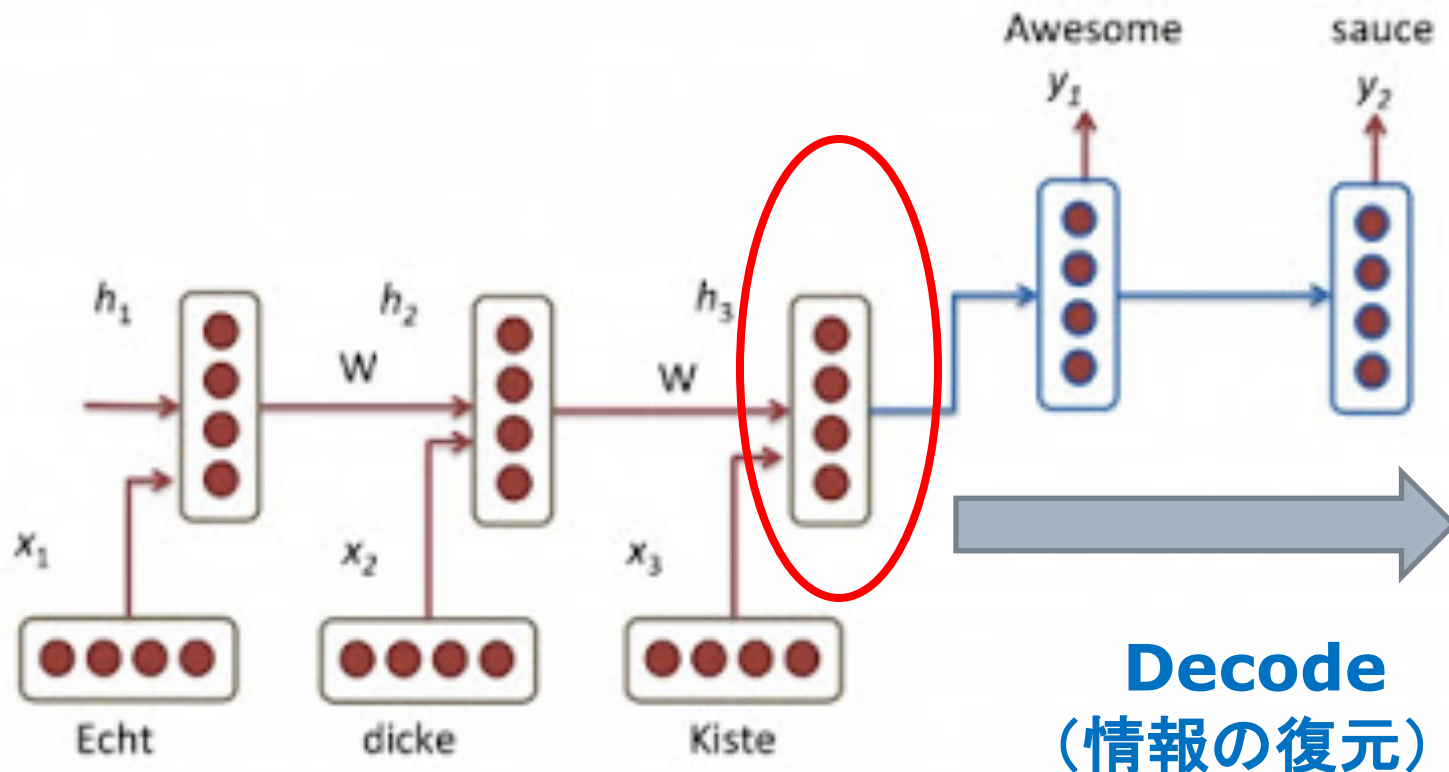
- 次の図(<https://goo.gl/JGckBP> から)は、こうしたメカニズムで、RNNが、独文の "Echt dicke Kiste" を英文の "Awesome sauce" に翻訳する様子を表している。(ここでは、文章の終わりを表す<EOS>は、省略されている)



- ここでは、Encoder部が、文章の最後にとるRNNの内部状態 h_3 が、そのままDecoder部に渡されることが示されている。入力シーケンスの情報のエッセンスが、この内部状態 h_3 に凝縮されていると考えればいい。



- AutoencoderのDecoder部が、圧縮された情報から元の情報を復元しようとするように、ここでは、その情報から、「同じ意味」を持つ、別の言語の文章を復元しようとする。



- Ilya Sutskever らは、このアーキテクチャーで、英語をフランス語に翻訳するシステムを作成し、BLEUのスコアで、34.81という高得点をたたき出した。
- この時のシステムは、5段重ねのLSTMで構成され、それぞれが 8,000次元の状態からなる384M個のパラメーターを持つものだった。

Attention Mechanism

新しいGoogle翻訳のシステムは、全く新しいアイデアに基づいてスクラッチから作り上げられたものではない。その基本的なアイデアは、2016年の5月にarXivに投稿された次の論文に多くを負っている。それは、先の、Ilya Sutskever らのシステムを発展させたものだ。

Bahdanau, D., Cho, K., and Bengio, Y. “**Neural machine translation by jointly learning to align and translate**” <https://goo.gl/HZxbNH>

Bahdanau, D., Cho, K., and Bengio, Y. **“Neural machine translation by jointly learning to align and translate”** <https://goo.gl/HZxbNH>

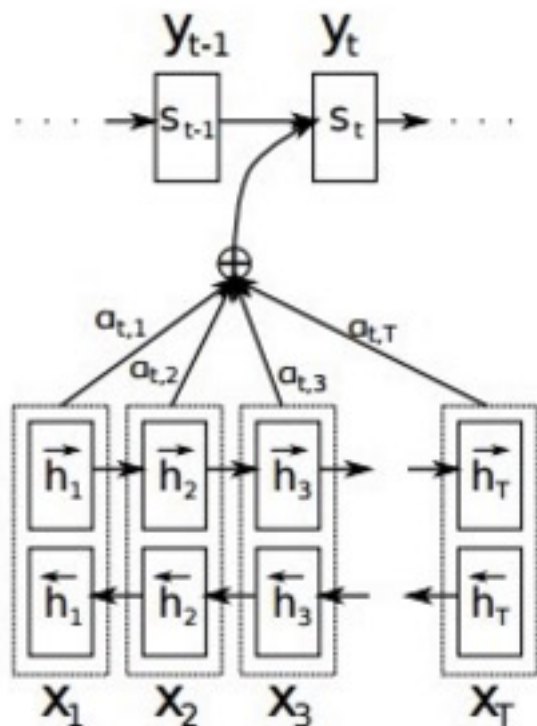
「近年、ニューラル機械翻訳として提案されたモデルは、多くの場合、Encoder-Decoderのファミリーに属している。そこでは、ソースの文が固定長ベクトルにエンコードされ、そこからデコーダが 翻訳文を生成する。この論文では、**固定長ベクトルの使用が、この基本的なEncoder/Decoderアーキテクチャの性能を改善する上でのボトルネックになっている**と推論し、モデルに自動的に、ターゲット・ワードを予測するのに重要なソース・文の一部について、(ソフト)検索を可能とすることによって、これを拡張することを提案する。その際、これらの部分を明示的にハードセグメントとして形成する必要はない。」

固定長ベクトルがボトルネック

- 先に見た、Ilya Sutskever らの翻訳システムでは、翻訳されるべき文は、Encoderで、一旦、ある決まった大きさの次元（例えば8000次元）を持つベクトルに変換される。このベクトルからDecoderが翻訳文を生成する。入力された文が、長いものであっても短いものであっても、途中で生成され以降の翻訳プロセスすべての出発点となるこのベクトルの大きさは同じまま。このシステムでは、長くても短くても入力された文全体が、一つの固定長のベクトルに変換されるのだ。
 - 確かに、そこは翻訳の精度を上げる上でのボトルネックになりうる。事実、Ilya Sutskever らのシステムでは、文の長さが長くなるにつれて、翻訳の精度が低下されるのが観察されるという。
-

この論文の基本的アイデア

- 文全体に一つの固定長のベクトルを割り当てるのではなく、翻訳時に、ソース・文の一部を改めて見直して、その部分から提供される情報を翻訳に生かそうということだ。



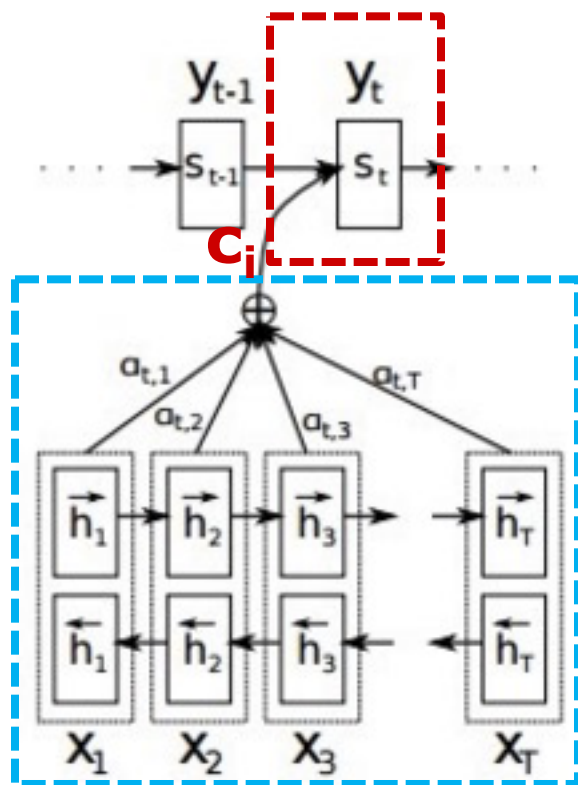
$a_{3,2}$ が大きい場合、これは、Decoderがターゲット文の第3の単語を生成しながら、ソース文の第2の状態に多くの注意を払うことを意味する。

「ここで、 y はデコーダによって生成された翻訳された単語であり、 x は原文の単語である。上記の図は双方向のリカレント・ネットワークを使用しているが、それは重要ではない。逆方向は無視していい。

重要な部分は、各デコーダの出力するワード y_t が、Encoderの最後の状態だけでなく、すべての入力状態の重みづけられた結合に依存することである。

a は、出力ごとに、それぞれの入力状態をどの程度考慮されるべきかを定義する重みである。したがって、 $a_{3,2}$ が大きい場合、これは、Decoderがターゲット文の第3の単語を生成しながら、ソース文の第2の状態に多くの注意を払うことを意味する。 a は、通常、1に合計されるように正規化される(それらは、入力状態に対する確率分布である)。」

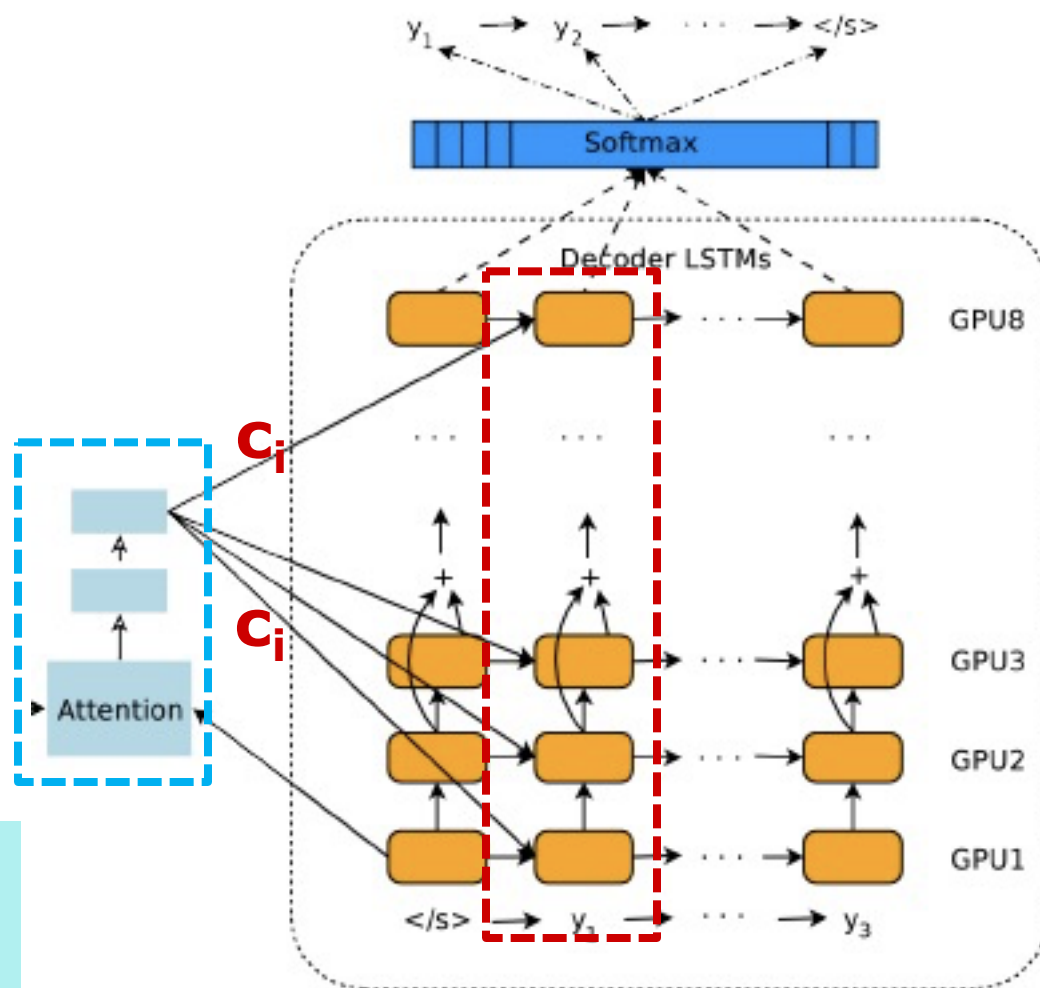
Bahdanau et al.



Decoderの内部状態 s_i は、
先行するノードの内部状態 s_{i-1} と
先行するノードの出力 y_{i-1} と、
Context c_i で決まる。

$$s_i = f(s_{i-1}, y_{i-1}, c_i)$$

GNMT



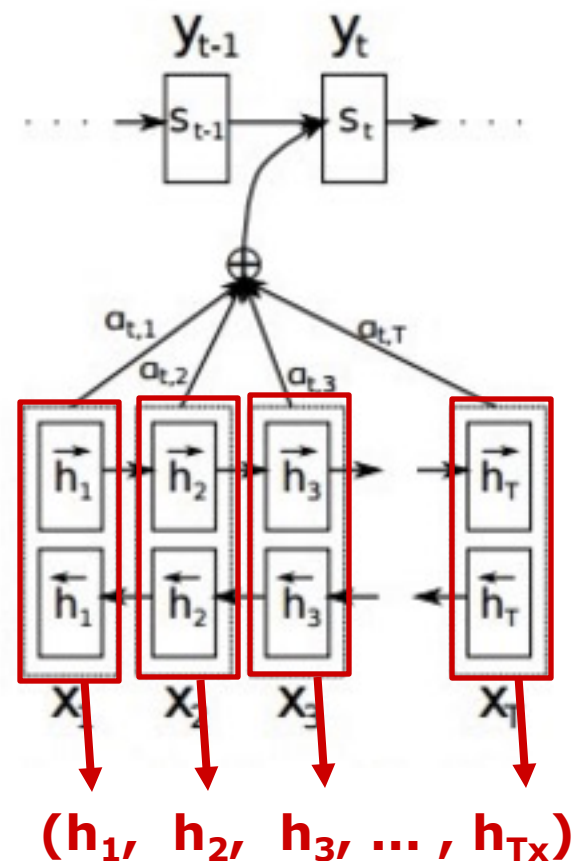
Annotation

このシステムでは、Encoderが、入力のシーケンスをAnnotationのシーケンス ($h_1, h_2, \dots, h_{Tx}$) に変える。

先の図には、Annotation h_i の名前は、直接には書き込まれていなかった。

上下に並んだ 逆向きの矢印を持つ h_i を囲む四角が書かれているのが、この四角が Annotation h_i である。ここでは、それを赤い四角で囲んだ。

この論文では、右向きの隠れ層の状態と左向きの隠れ層の状態の「連結」として h_i が実装されている。GNMTでも同様である。



Context

Docorderが、入力の $(x_1, x_2, \dots, x_{T_x})$ から、 t 番目の語 y_t を生成しようとする時に、このAnnotationのシーケンス $(h_1, h_2, \dots, h_{T_x})$ からの情報は、次の式で、 c_i に束ねられてDecoderに流れ込んでくる。

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

この c_i を、**Context**と呼んでいる。

Decoderの内部状態

Decoderの内部状態 s_i は、先行するノードの内部状態 s_{i-1} と先行するノードの出力 y_{i-1} と、このContext c_i で決まる。

$$s_i = f(s_{i-1}, y_{i-1}, c_i)$$

「直観的には、これは、DecoderにAttentionのメカニズムを実装する。Decoderは、ソースの文のいくつかの部分を、注意を払うべき文章だと決定する。Decoderに、Attentionのメカニズムを持たせることで、Encoderは、ソース文内のすべての情報を固定長のベクトルにエンコードする負担から解放される。この新しいアプローチにより、情報は、Annotationのシーケンス上に広がって拡散することができ、その情報は、Decoderによって選択的に取り出すことができる。」

Alignment Model

Context c_i を定義している、それぞれのAnnotation h_j にかかる重み α_{ij} は、次の式で定義される。

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

先の図の $\alpha_{t,1}, \alpha_{t,2}, \dots, \alpha_{t,T}$ を全部足し合わせると、1になる。
SoftMaxと同じだ。

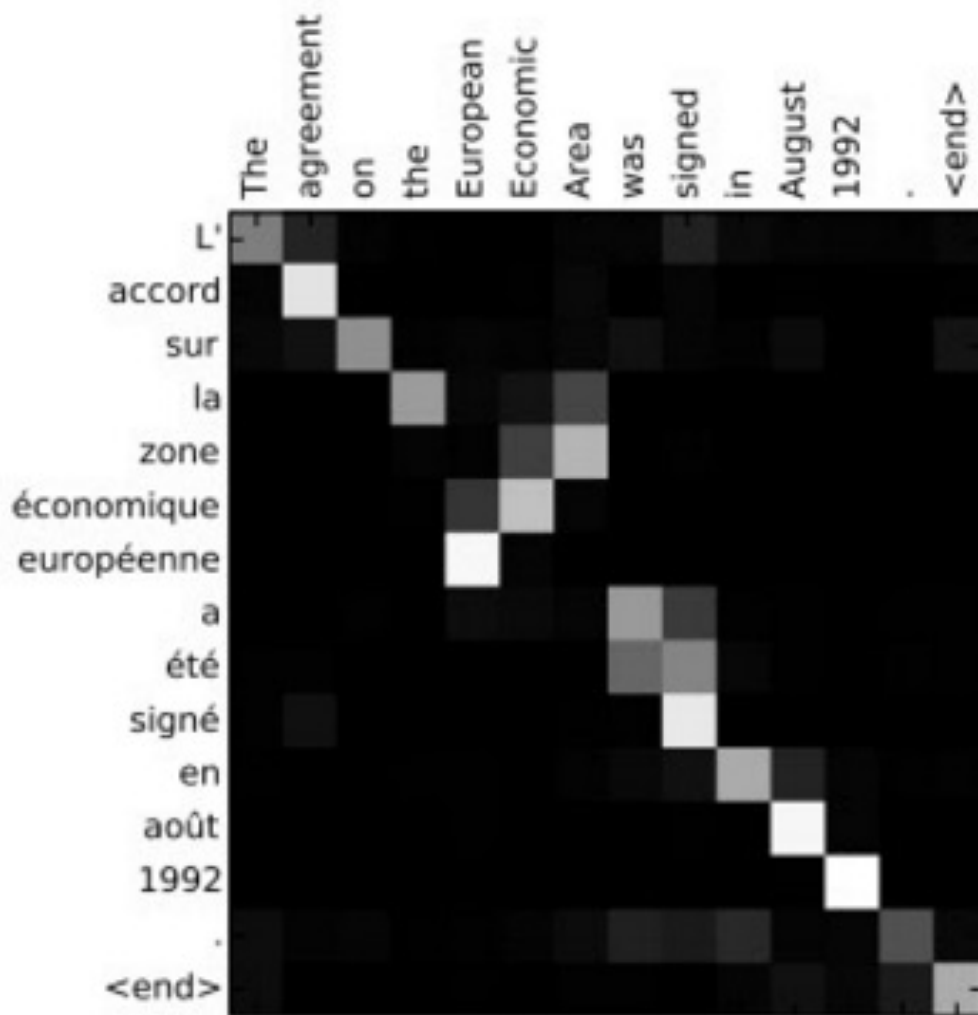
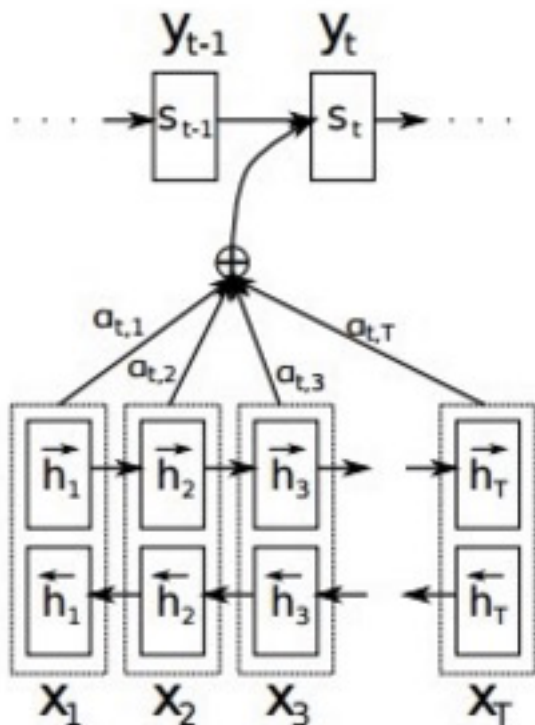
α_{ij} の計算の元になる、 e_{ij} は、次のように計算される。

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

この指標 e_{ij} (**Alignment Model**と呼ばれる)は、入力の i 番目付近と、出力の j 番目付近が、うまくマッチしているかを示すものだ。
 e_{ij} は、Decoderの $i-1$ 番目の状態 s_{i-1} (y_i を出力する直前の状態である)と、Encoderの j 番目のAnnotation h_j で決まる。

次の図は、英語からフランス語の翻訳で、英語の入力のj番目の語と、フランス語の出力のi番目の語について、 a_{ij} の値をグレースケールで示したものである(白は1、黒は0)。

Annotationによる Word Alignment



- 二つの言語の語順が、同じであれば、対角線上に、1が集まることになるのだが。
- この図は、次の英語とフランス語の翻訳が、下線部分で、語順が逆になることを反映している。

The agreement on the European Economic Area was signed in August 1992 .

L' accord sur la zone économique européenne a été signé en août 1992 .

- もっとも、Word Alignmentについては、90年代の「統計的機械翻訳モデル」においても、熱心に研究されていたので、こうしたアプローチは、新しいものではない。

Wordpiece

正直にいうと、僕は最初、論文のこの部分が、何をしているのかよくわからなかった。アプローチが僕の常識の「想定外」だった。その意味では、GNMTらしさが明確に出ているところだと思う。

と言って、言語へのアプローチとして、逸脱があるわけではないのだ。原理的に言えば、言語は、書かれた文字の集まりではなく、まずは、発話された音の連なり(順序は重要)としてあるということと、コトバとその意味の結合は、「恣意的」なものだということにたちかえればいいのだと思う。

この部分は、“Zero-Shot”論文とも、繋がっている。

文をどのように分割するか？

- 機械翻訳の対象は、もちろん、ソース側でもターゲット側でも文なのだが、その文をどのように翻訳機械に入力として与えるのかには、いくつかの選択肢がある。
- 一つは、文を「文字」単位で分割して、1文字ずつ入力することだ。もちろん、文末には、文末であることを示す記号を置く。
- もう一つの方法は、文を「語」に分割して、一語ずつ入力することだ。ここでも、文末の記号は必要である。
- 1文字ずつ読んで、正しい翻訳ができるのか？ それができるのである。次の英語 -> フランス語翻訳の結果は、文字単位の方がワード単位よりいい翻訳結果が得られることを示している。（なぜか？） ただし、計算時間は、5倍から10倍かかる。

Character Mode

Table 4: Single model results on WMT En→Fr (newstest2014)

Model	BLEU	CPU decoding time per sentence (s)
Word	37.90	0.2226
Character	38.01	1.0530
WPM-8K	38.27	0.1919
WPM-16K	37.60	0.1874
WPM-32K	38.95	0.2118
Mixed Word/Character	38.39	0.2774
PBMT [15]	37.0	
LSTM (6 layers) [31]	31.5	
LSTM (6 layers + PosUnk) [31]	33.1	
Deep-Att [45]	37.7	
Deep-Att + PosUnk [45]	39.2	

同じ文字セットを使っている
のなら、一文字単位での
「翻訳」は、驚くほど精度は
高い！

ただ、効率は
悪い

wordへの分割の問題

- ただ、自明と思える文のwordへの分割にも問題があるのだ。
(特に、日本語の場合) 次の二つの文を考える。

The stratosphere extends from about 10km to about 50km in altitude.

成層圏は、高度10kmから50kmの範囲にあります。

- 英語の場合、文のワードへの分割は、ワード間にスペースが存在することで、ほぼ自明である。ところが、日本語の場合にはそうではない。
- 文は語から、「文法的」に構成されているのだが、文を語に分割するには、「文法」の知識に基づいたParsingが必要なのだ。
(例えば、Mecabを使う)

辞書にない語をどう扱うか？

- ニューラルマシン翻訳モデルでは、名前、数字、日付といった、基本的には辞書にないオープンな語彙の翻訳であっても、固定した語のボキャブラリで動作することがよくある。
- 語彙外(OOV : Out Of Vocaburary)単語の翻訳に対処するアプローチには、大きく2つのカテゴリーがある。
- 1つのアプローチは、珍しい単語を、ソースからターゲットへの単純にコピーすることである。(珍しい単語の大部分は、名前や番号で、それらに対しては、コピーが正しい翻訳になる。)あるいは、アテンション・モデルに基づいて、外部のアライメントモデルを使う、または、より複雑な特殊な目的のポインティングネットワークを使う方法もある。
- もう一つのアプローチの広いカテゴリーは、サブ・ワード(sub-word)を単位として使うことである。サブ・ワードには、文字(character), 語(word)と文字の混在、あるいは、もっと賢いサブ・ワードがある。

ワードピース・モデル

我々の最も成功したアプローチは、第2のカテゴリ（サブワードユニット）に分類される。Google音声認識システムの日本語/韓国語の分割問題を解決するために最初に開発されたワードピースモデル（WPM）[35]の実装を採用した。

このアプローチは、完全にデータ駆動型で、可能性のある文字列に対して確定的な分割を生成することが保証されている。これは、ニューラルマシン翻訳で、辞書にない珍しい言葉を扱うために[38]で使用された方法に似ている。

任意の単語を処理するために、我々はまず、与えられた訓練されたワードピース・モデルでワード達をワードピースに分割する。特別な語境界記号が、モデルの訓練の前に追加される。元のワード列を曖昧さなくワードピース列から復元することができる。デコード時に、モデルはまずワードピース列を生成し、ワードピース列は対応するワード列に変換される。

ワード列とそれに対応するワードピース列の例

□ ワード列:

Jet makers feud over seat width with big orders at stake

□ ワードピース列:

_J et _makers _fe ud _over _seat _width _with
_big _orders _at _stake

- 上記の例では、ワード「ジェット」は2つのワードピース「_J」と「et」に分割され、ワード「feud」は2つのワードピース「_fe」と「ud」に分割されている。他のワードは単一のワードピースとして残っている。"_"はワードの先頭を示す特殊文字である。
-

ワードピース・モデル

- ワードピースモデルは、データドリブン・アプローチを使って生成され、変化するワードの定義の元でも、トレーニングデータの言語モデルの可能性を最大にする。
 - 訓練コーパスと、望むトークンの数 D が与えられた時、最適化の問題は、コーパスが選択されたワードピースモデルに従って分割されたとき、結果として得られるコーパスがワードピースの数が最小であるように、 D 個のワードピースを選択することである。
 - この最適化問題に対する我々の総当たりのアルゴリズムは、[38]と同様であり、[35]でより詳細に説明される。
-

2012年 [35] Schuster, M., and Nakajima, K. "Japanese and Korean voice search" <https://goo.gl/DUxQuv>

1. まず、単位語の目録を、ベーシックUnicode(日本語の漢字・ひらがな・カタカナ、韓国語のハングル、そしてすべてのASCIIコードを含む)で作る。最終的には、日本語でトータル22000語、韓国語11000の目録になる。
2. この目録を使った訓練データ上に、言語モデルを構築する。
3. 現在の語の目録から二つの単位語を取り出し、一つに結合して新しい単位語を作り、単位語の目録を一つ増やす。可能な組み合わせの中から、言語モデルに追加された時、訓練データ上で、類似度を一番増大させるものを、新しい単位語として選ぶ。
4. 事前に定義された単位語の限界に到達するか、類似度の増大が、ある閾値以下に下がるまで、2. の操作を繰り返す。

- [35]で使用された元の実装と比較して、ワードの始めにのみ特殊記号を使用し、ワードの両端では使用しない。また、基本的な文字の数をデータに応じて管理可能な数に減らし(西洋言語では約500、アジア言語ではより多く)、残った文字を特別な未知の文字にマップして、非常にまれな文字で、ワードピースの語彙を汚染しないようにする。
- トータルで8kないし32kの間のワードピースの語彙を使用すると、我々が試したすべての言語対の間で良好な精度(BLEUスコア)と高速なデコード速度の両方が達成されることがわかった。

Champollionが、"*Je tiens l'affaire!* »と叫んで倒れたのは、象形文字が、表音文字と表意文字の混在であることに気づいたからなのだが、日本語は、もともとそうである。その上、ひらがな、漢字、カタカナ、ASCIIが混在している。(顔文字も)

[35]で、取り上げられている日本語の「読み」の難しさの例が面白い。"AKB48", "W-INDS", "シャ乱Q", "価格.com",

-
- 前述のように、翻訳では、ソースからターゲットにまれな固有名あるいは番号を直接コピーすることがよくある。このような直接のコピーを容易にするため、**ソース言語とターゲット言語の両方に共通のワードピースモデルを使用する**。このアプローチを使用すると、ソースとターゲットの文の同じ文字列がまったく同じ方法で分割されることが保証され、システムがこれらのトークンを簡単にコピーできるようになる。
 - **ワードピースは、キャラクターの柔軟性とワードの効率との間のバランスを達成する**。我々のモデルは、ワードピースを使用したときに、BLEUスコア全体が改善されていることがわかる。おそらく、我々のモデルが文字のみに頼らずに本質的に無限のボキャブラリを効率的に扱うからである。文字のみの場合は、~~入力列と出力列の平均長さをずっと長くするので、より多くの計算が必要になる。~~

SentencePiece

最近、Googleが発表した、WordPieceモデルに基づく、文分割のライブラリー。文を直接、WordPieceに分割する。僕には、WordPieceを理解するのに、とても役に立った。

<https://github.com/google/sentencepiece>

SentencePieceとは何か？

- SentencePieceは、主に、ニューラルネットワークベースのテキスト生成用(たとえば、ニューラルネットワーク機械翻訳のような)に設計された、教師なしのテキストTokenizerとDeTokenizerである。
 - SentencePieceはサブワード単位(ワードピースとして知られている[Wu et al.] [Schuster et al.])と、バイトペア符号化(BPE)[Sennrich et al.]の再実装である。
 - SentencePieceは、トークン化ツールを事前にトークン化された文から訓練する以前のサブワード・アプローチとは異なり、元の生の文からtokenizerとdetokenizerを直接訓練する。
-

SentencePiece

ユニークなトークンの数が、事前に決められる

- ニューラルマシン翻訳モデルは、通常、固定された数の語彙で動作する。
 - 無限の語彙数を仮定するほとんどの教師なしの単語分割アルゴリズムとは異なり、SentencePieceは、最終的な語彙サイズが固定されるようにセグメンテーションモデルを訓練する（例えば、8k、16k、または32k）。
 - 空白は、基本的な記号として使用される。
-

モデルの生成・訓練

```
% spm_train --input=data/botchan.txt --model_prefix=m --vocab_size=1000
unigram_model_trainer.cc(494) LOG(INFO) Starts training with :
input: "../data/botchan.txt"
... <snip>
unigram_model_trainer.cc(529) LOG(INFO) EM sub_iter=1 size=1100 obj=10.4973 r
trainer_interface.cc(272) LOG(INFO) Saving model: m.model
trainer_interface.cc(281) LOG(INFO) Saving vocabs: m.vocab
```

語彙数の指定

生成された
モデルと語彙

モデルを利用した encodeとdecode

```
% echo "I saw a girl with a telescope." | spm_encode --model=m.model
_I _saw _a _girl _with _a _te le s c o pe .

% echo "I saw a girl with a telescope." | spm_encode --model=m.model --output_format=id
9 459 11 939 44 11 4 142 82 8 28 21 132 6

% echo "9 459 11 939 44 11 4 142 82 8 28 21 132 6" | spm_decode --model=m.model --input_format=id
I saw a girl with a telescope.
```

おそらく、こうした数字がGNMTの
入力に与えられているのだと思う。

Google多言語ニューラル機械翻訳

GNMTの多言語拡張についての論文である。

ゼロ・ショット翻訳が可能であることを示し、また、インターリンガの存在を示唆するなど、非常に刺激的である。

言語に対するいくつかの「仮説」と、ディープラーニングでの結果をある種の「実験」として、結びつけようとするスタイルは新しいものである。GNMTは、CERNの加速器のように、人類の共通実験プラットフォームとして皆に認知されているわけではない(ラザフォードの原子核の発見のレベルかも)、方法論的には無意識的なものかもしれないのだが。それは、BLEUスコアだけでなく、言語そのものの謎に向き合うことになるはずだ。

Googleの多言語ニューラル機械翻訳

"Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation"

Melvin Johnson et al.

<https://goo.gl/islUXa>

2016年



2016年 Melvin Johnson et al. **"Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation"**

<https://goo.gl/islUXa>

「我々は、単一のニューラル機械翻訳(NMT)モデルを使用して、複数の言語どうしを翻訳する、シンプルで洗練されたソリューションを提案する。」

「共有ワードピースのボキャブラリを使用することで、多言語NMTはパラメータを増やさずに、単一のモデルを利用することができる。これは、従来の多言語NMTの提案よりも大幅に簡単なものだ。」

「我々のモデルは、訓練中に明示的には見られなかった言語ペアの間の暗黙的な橋渡しを実行することも学ぶことができた。それは、翻訳の学習とゼロ・ショット翻訳がニューラル翻訳で可能であることを示している。」

「我々のモデルには、普遍的なインターリンガ表現が存在することを示唆する分析を示し、複数の言語を混在させた時に起きる、興味深い例を示す。」

Google 多言語ニューラル機械翻訳の概要

多言語の翻訳を、言語ペアの数だけのシステムによって行うのではなく、一つのシステムで行うことにより、さらに多くの言語へのスケール・アップが容易になる。

システムが単純になり、コーパスの少ない言語にもメリットが生まれ、ゼロ・ショット翻訳も可能になる。

多言語翻訳を単一モデルで

- ニューラル機械翻訳(NMT)は、多くの大規模な環境で急速に採用されてきた機械翻訳に対するアプローチである。ただ、このようなシステムのほとんどは、単一の言語のペアのために構築されていた。
 - 基本的なNMTアーキテクチャを大幅に変更することなく、単一のモデルを使用して複数言語のペアを処理するための十分に単純で効率的な方法は、それまでなかった。
 - この論文では、単一のモデルを使用して多言語間で翻訳を行うための簡単な方法を紹介している。この方法では、ターゲット言語を示す人工的なトークンを、入力シーケンスに追加するだけで、従来のNMTモデルアーキテクチャに、変更を加える必要はない。
-

特徴1: シンプルさ

- モデルのアーキテクチャは変更されていないため、新しいデータを単純に追加するだけで、より多くの言語へのスケーリングは簡単に行われる。
 - ターゲット言語が変更された場合には、それを示す新しいトークンが使用されるだけである。これにより、複数の言語を扱う際に必要なモデルの数を減らすことができるため、製品版の展開も簡単なものになる。
 - Googleでは、ソースとターゲットとして合計100以上の言語をサポートしているため、もし、それぞれの翻訳のモデルが、1つの言語ペアしかサポートしていないとすれば、すべての言語ペア間の最良の翻訳を得るためには、理論的には、 100^2 ものモデルが必要になる。明らかに、これは製品版の環境では問題となる。
-

特徴2: リソースの少ない言語の翻訳の改善

- 多言語NMTモデルでは、すべてのパラメータが、モデル化されているすべての言語ペアによって暗黙のうちに共有される。これにより、モデルは、訓練・学習を通じて言語の境界を越えて一般化される。
- 利用可能なデータがほとんどない言語のペアと豊富なデータを持つ言語のペアが単一のモデルに混在すると、リソースの少ない言語ペアの翻訳品質が大幅に向上する。

フランス語 -> 日本語 で翻訳してみたのだが、
フランス語 -> 英語 -> 日本語 のブリッジの方が
精度が高いように思われる。

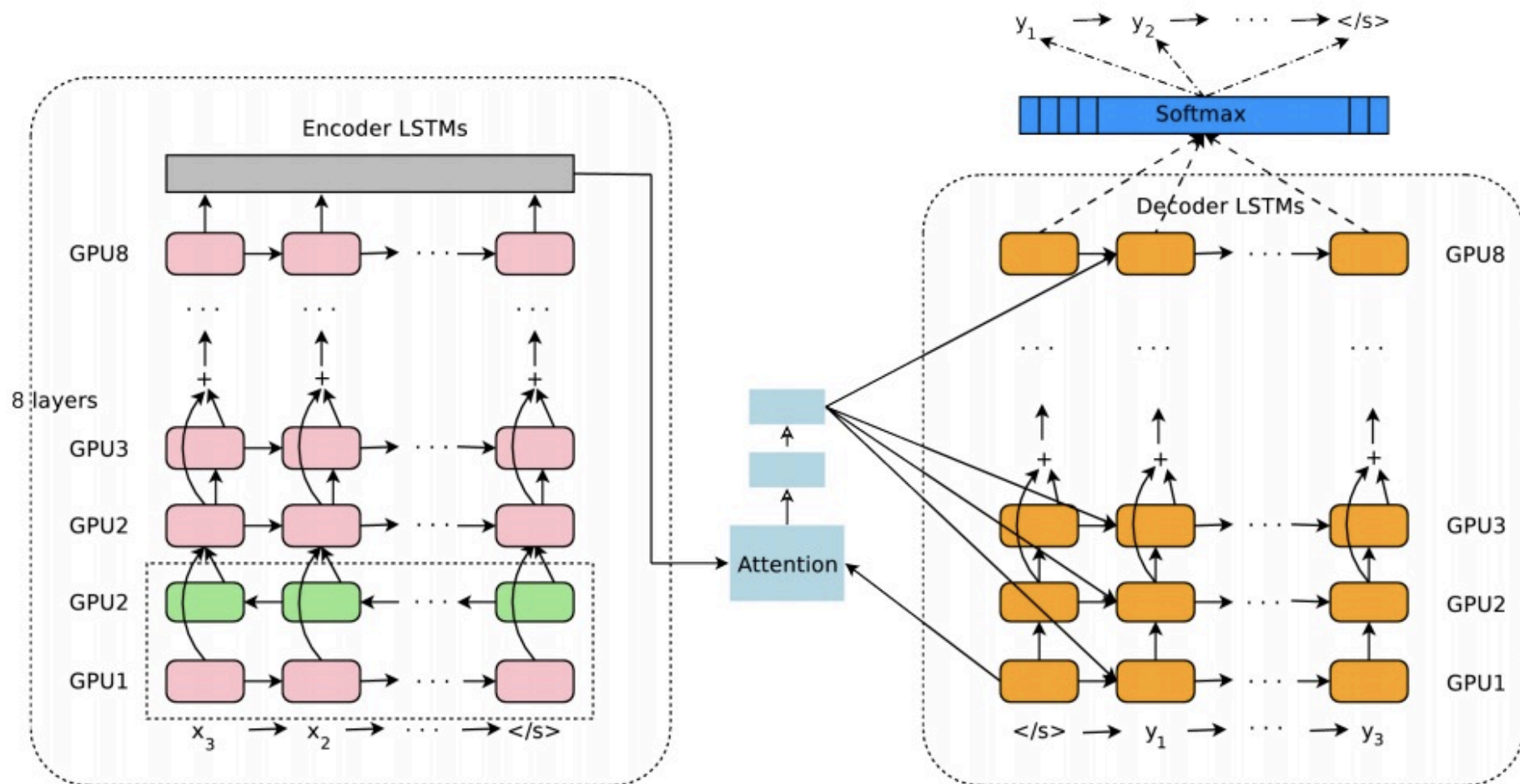
特徴3: ゼロ・ショット翻訳

- 単一のモデルで複数の言語ペアをモデリングすることの驚くべきメリットは、モデルが、これまで見たことのない言語ペア間の翻訳(ゼロ・ショット翻訳)を暗黙のうちに学習することである。
 - たとえば、ポルトガル語から英語、英語からスペイン語のサンプルで訓練された多言語NMTモデルでは、その言語ペアの学習データは存在しないにもかかわらず、ポルトガル語からスペイン語への合理的な翻訳が生成される。
 - 問題の言語ペアの追加データをほとんど使わずに、ゼロ・ショット言語ペアの品質を簡単に改善できる。
-

多言語翻訳のシステム・アーキテクチャー

多言語モデルのアーキテクチャは、Googleのニューラル・マシン・トランスレーション(GNMT)システムと同じものである。異なるのは、ターゲット言語を示すトークンを先頭に付け加えるところだけである。

Google 多言語ニューラル機械翻訳のアーキテクチャ



多言語対応でも、GNMTの
アーキテクチャーをそのまま使う

入力データの変更

ターゲット言語を示すトークンの追加

- 単一システム内で多言語データを利用できるようにするため、入力データの簡単な変更を提案している。入力文の冒頭に人工的なトークンを導入して、モデルが翻訳する対象言語を示す。

Hello, how are you? -> ¿Hola como estás?



<2es> Hello, how are you? -> ¿Hola como estás?

ターゲット(翻訳先)が、スペイン語であることを示す。

ソース言語の指定は不要

- ❑ ソース言語を指定していないことに注意。モデルはソース言語を自動的に学習する。(システムは、すべてUnicodeを使う)
- ❑ ソース言語を指定しないと、スペルは同じだが異なるソース言語の、意味が異なる単語の翻訳が曖昧になる可能性があるという欠点があるのだが、コード切り替えを使って簡単に入力进行处理できるという利点がある。ほとんどすべての場合で、文脈が正しい翻訳を生成するのに十分な言語の証拠を提供することがわかる。

余談だが、Unicodeでは、象形文字も楔形文字も、線形文字Bも表現できるらしい。使ったことないけど。

複数言語を、同時にモデル化する

- トークンを入力データに追加した後、複数の言語ペアからなるすべての多言語データを一度にモデル化する。
 - 我々は、トレーニングに使用されるすべてのソースとターゲットのデータを共有するワードピース・モデルを使用する。通常は32,000ワードピース。
 - 実装は、TensorFlow 。
 - このアプローチは、われわれが知りうる選択肢の中で、最もシンプルで最もエレガントなものである。
-

実験結果

まず、多言語モデルは、単一言語ペアのモデルより、ハンディがあることを確認しよう。にもかかわらず、多対1のモデルは、単一モデルより翻訳の精度が向上する。

12種類の言語からなる多言語モデルは、278Mのパラメータを持つのだが、合計3.33Bのパラメータを持つ12のモデルを実行するのと、ほぼ同じ性能が達成できる。

単一言語ペアモデルと多言語モデルの比較

- ❑ 多言語および単一言語のペアモデルは、すべて、同じ環境（1024ノードおよび8つのLSTMレイヤーと32kの共有ワードピース・モデル・ボキャブラリを使用）で訓練され、同じパラメータ数を持つ。
 - ❑ これは多言語モデルには、不公平なものである。なぜなら、 N が多言語モデルで結合された言語対の数であれば、単一言語ペアモデルに比べて、言語対ごとに利用可能なパラメータの数は、 N 倍だけ減少する。
 - ❑ 多言語モデルはまた、組み合わせたボキャブラリーを処理しなければならない。
 - ❑ 実験を簡単にするため、すべてのモデルでパラメータの数を一定に保つことを選択した。
-

訓練に利用されたコーパス

- ❑ WMT'14の英語(En) <-> フランス語(Fr)データセットには、3,600万の文のペアが含まれている。
 - ❑ WMT'14の英語(En) <-> ドイツ語(De)データセットには、500万の文のペアが含まれている。
 - ❑ Googleは、内部に、英語 <-> 日本語(Ja)、英語 <-> 韓国語(Ko)、英語 <-> スペイン語(Es)、英語 <-> ポルトガル語(Pt) 等々の多くのデータセットを持っているが、その規模は、先のWMTのデータセットより、2～3桁大きいという。
 - ❑ Googleニューラル機械翻訳では、GPU100個を使って、フルトレーニングには最大1,000万ステップ、収束までには3週間かかることがあるという。
-

多言語モデルを三つの異なる構成で訓練する

- 多言語モデルでは、一つまたは複数のソース/ターゲット言語を持つことができるので、次の三つの構成で、モデルを訓練した。
 - 多くのソース言語を1つのターゲット言語(多対1)に、
 - 1つのソース言語から多くのターゲット言語(1対多)に、および
 - 多くのソース言語を多くのターゲット言語に(多対多)
-

多対1の場合: 実験環境

- 最初の実験は、WMTデータセット上で行った。ここでは、ドイツ語 -> 英語とフランス語 -> 英語を組み合わせせて多言語モデルを学習した。ベースラインは、2つの単一言語ペアモデルで、ドイツ語 -> 英語とフランス語 -> 英語は独立して訓練されている。これらの実験は、言語ペアごとのデータ量が等しくなるようにオーバーサンプリングで1回実行し、もう一回はオーバーサンプリングなしで実行した。
- 2番目の実験は、製品版データに基づいている。ここでは、日本語 -> 英語、韓国語 -> 英語とオーバーサンプリングを組み合わせている。
- 最後の3番目の実験は、製品版データの上で、スペイン語 -> 英語、ポルトガル語 -> 英語の両方を組み合わせた。オーバーサンプリングを使用した。

多対1の場合: 実験結果

- すべての実験について、多言語モデルは、言語ペアごとに利用可能なパラメータの数に関しての上記の欠点にもかかわらず、ベースライン単一システムより優れていた。
- このメリットを説明する可能性のある仮説の1つは、モデルがより多くの英語データをターゲット側に持ち、ソース言語が同じ言語ファミリーに属していること。それで、モデルは有用な一般化を学んでいると思われる。
- WMT実験では、French -> Englishの最大利得は、+1.27 BLEUである。両方のWMTテストセットの結果は、私たちが知る限り、単一のモデルの他の公開された、どの最先端の結果よりも優れている。製品版の実験では、多言語モデルがベースラインの単一システムよりも+0.8 BLEUほど優れていることがわかる。

多対1の実験結果

Model	Single	Multi	Diff
WMT German→English (oversampling)	30.43	30.59	+0.16
WMT French→English (oversampling)	35.50	35.73	+0.23
WMT German→English (no oversampling)	30.43	30.54	+0.11
WMT French→English (no oversampling)	35.50	36.77	+0.27
Prod Japanese→English	23.41	23.87	+0.46
Prod Korean→English	25.42	25.47	+0.05
Prod Spanish→English	38.00	38.73	+0.73
Prod Portuguese→English	44.40	45.19	+0.79

日本語、韓国語からの英訳は、他の言語の場合に比べて、かなり、低い。

多対1の場合: 実験結果

- オーバーサンプリングは、より大きい言語ペア (En -> Fr) の低品質を犠牲にして、小さな言語ペア (En -> De) に役立っている。オーバーサンプリングのないモデルは、期待されるように、小さなモデルと比較して、より大きな言語でより良い結果を達成する。
- これらの両極端の間に、両方の言語に利益をもたらす設定があり、その設定を見つけることが将来の仕事の範囲であることを期待している。この効果は、小規模なデータセット (WMT) で顕著であり、製品版のデータセットではそれほど顕著ではないこともわかる。

1 対多の実験結果

Model	Single	Multi	Diff
WMT English→German (oversampling)	24.67	24.97	+0.30
WMT English→French (oversampling)	38.95	36.84	-2.11
WMT English→German (no oversampling)	24.67	22.61	-2.06
WMT English→French (no oversampling)	38.95	38.16	-0.79
Prod English→Japanese	23.66	23.73	+0.07
Prod English→Korean	19.75	19.58	-0.17
Prod English→Spanish	34.50	35.40	+0.90
Prod English→Portuguese	38.40	38.63	+0.23

多対多の場合：実験結果

- 単一言語モデルと同じモデルサイズと語彙サイズを持つ多言語製品版モデルは、ベースラインに非常に近いことがわかる。場合によっては、ベースラインモデルよりも優れている場合もあるのだが、他のモデルでは悪化している。
- この効果は、このモデルでは、利用可能なすべてのトレーニングデータを利用せず、各言語のペアがベースラインで見られるデータの4分の1しか見ていないためだと考えられる。
- WMTデータセットでは、小さな言語ペアのオーバーサンプリングの影響をみつと、先と同様の傾向に気付く。オーバサンプリングは、より小さい言語ペアを、大規模なものを犠牲にして助けるのだが、オーバーサンプリングしないことが、逆の効果を持つようには見えない。

1対多の実験結果

	Model	Single	Multi	Diff
WMT English→German (oversampling)		24.67	24.49	-0.18
WMT English→French (oversampling)		38.95	36.23	-2.72
WMT German→English (oversampling)		30.43	29.84	-0.59
WMT French→English (oversampling)		35.50	34.89	-0.61
WMT English→German (no oversampling)		24.67	22.61	-2.06
WMT English→French (no oversampling)		38.95	38.16	-0.79
WMT German→English (no oversampling)		30.43	29.22	-1.21
WMT French→English (no oversampling)		35.50	35.93	+0.43
Prod English→Japanese		23.66	23.12	-0.54
Prod English→Korean		19.75	19.73	-0.02
Prod Japanese→English		23.41	22.86	-0.55
Prod Korean→English		25.42	24.76	-0.66
Prod English→Spanish		34.50	34.69	+0.19
Prod English→Portuguese		38.40	37.25	-1.15
Prod Portuguese→English		44.40	44.02	-0.38
Prod Spanish→English		38.00	37.65	-0.35

大規模実験

- ここでは、12の製品版の言語のペアを単一言語のペアモデルと同じ数のパラメータを使用して単一の多言語モデルに結合した結果を示す。上記のように、入力にはターゲット言語トークンを前置する必要がある。データのバランスを取るために、より小さな言語のペアからサンプルをオーバーサンプルする。
- 結果は次の表に要約されている。多言語モデルは最高の単一モデルにかなり近く、場合によっては同等の品質を達成することさえある。
- 278Mのパラメータを持つ単一のモデルが、合計3.33Bのパラメータを持つ12のモデルを実行することができることは注目に値する。

大規模多言語モデル

- ❑ 多言語モデルでは、トレーニングの時間とコンピューティングリソースの12分の1で収束する必要がある。もう1つの重要な点は、単一のモデルより少し長く訓練しただけなので、個々の言語のペアは、その単一の言語のペアモデルと比較して、データの12分の1以下のデータしか見ていないということ。
- ❑ ここでも、多言語モデルにとっては、比較はやや不公平なものなのだが、利用可能なすべてのデータで訓練されたより大きなモデルが、ベースラインと同等またはそれ以上の品質を達成すると期待できる。
- ❑ 要約すると、多言語NMTを使用することで、品質の低下がほとんど、あるいは、まったくないように言語をグループ化でき、トレーニング効率も向上し、モデル数も少なくなり、製品化が容易になるという利点がある。

大規模な多対多の実験結果

Model	Single	Multi	Diff
Prod English→Japanese	23.66	21.10	-2.56
Prod English→Korean	19.75	18.41	-1.34
Prod Japanese→English	23.41	21.62	-1.79
Prod Korean→English	25.42	22.87	-2.55
Prod English→Spanish	34.50	34.25	-0.25
Prod English→Portuguese	38.40	37.35	-1.05
Prod Portuguese→English	44.40	42.53	-1.87
Prod Spanish→English	38.00	36.04	-1.96
Prod English→German	26.43	23.15	-3.28
Prod English→French	35.37	34.00	-1.37
Prod German→English	31.77	31.17	-0.60
Prod French→English	36.47	34.40	-2.07

ゼロ・ショット翻訳

この多言語システムは、明示的な訓練データがない言語ペアの間でゼロ・ショット翻訳を実行できる。

著者らは、これを、「我々の知る限り、これは真の多言語ゼロ・ショット翻訳の最初のデモである」と主張している。

ゼロ・ショット翻訳

- このアプローチの興味深い利点は、明示的な訓練データがない言語ペアの間にゼロ・ショット翻訳を実行できることである。
 - これを実証するために、2つの異なる言語ペア、ポルトガル語 -> 英語と英語 -> スペイン語(モデル1)と、英語 <-> ポルトガル語、英語 <-> スペイン語(モデル2)の4つの異なる言語ペアのデータで訓練されたモデルの、2つの多言語モデルを使う。
 - これらのモデルの両方が、スペイン語 -> ポルトガル語で、合理的に良質なポルトガル語を生成できることを示す。
 - 著者らは、これを、「私たちの知る限り、これは真の多言語ゼロ・ショット翻訳の最初のデモです。」と主張している。
-

ゼロショット翻訳の実験結果

- 最も興味深いのは、モデル1とモデル2の両方が合理的な品質でゼロショット変換を実行できるということである((d)と(e)を参照)。
- モデル2は、BLEUポイントで3点近くモデル1より優れていることに注目すべきである。言い換えれば、ソース側でスペイン語を追加し、ターゲット側でポルトガル語を追加すると、ポルトガル語 -> スペイン語のゼロ・ショット翻訳に役立つということ。
- これは、共有アーキテクチャにより、モデルがこれらすべての言語間のインターリンガを学習できるためだと考えている。
- ゼロ・ショット翻訳が問題なく機能するという楽しい事実の他に、第3言語による明示的なブリッジ処理が不要なため、デコードのスピードが半分になるという利点もある。

ゼロ・ショット翻訳の実験結果

Portuguese→Spanish BLEU scores using various models.

	Model	BLEU
(a)	PBMT bridged	28.99
(b)	NMT bridged	30.91
(c)	NMT Pt→Es	31.50
(d)	Model 1 (Pt→En, En→Es)	21.62
(e)	Model 2 (En↔{Es, Pt})	24.75
(f)	Model 2 + incremental training	31.77

少量のデータによる段階的な訓練の重要性

- 最後に、モデル2は、少量のポルトガル語 -> スペイン語の平行データ(表5(c)よりも1桁小さい)で段階的に訓練した時、最良の品質を得た。
 - これは、我々の多言語モデルが、暗黙的なブリッジングによるゼロ・ショット変換を実行できることを示している。それは、少量のデータで段階的に訓練されると、強力なNMTベースラインよりも優れている。
 - この結果は、特に重要である。英語以外のリソースの少ない言語ペアにとって、ソース言語とターゲット言語の両方が英語でない言語ペアの平行・データを取得する方がはるかに難しいからである。
-

ビジュアルな分析 インターリンガの存在

言語にかかわらず、ネットワークは、同じ意味を持つ文章が同じような方法で表現される何らかの共有表現を学習しているのか？ この質問に対しては、そうだという。これも、画期的。モデルは、訓練された言語ペアを扱うのと同じ方法で、ゼロ・ショット翻訳を実行しているのか？ これについては、まだよくわからないという。

実験結果が提起する問題

- モデルを複数の言語にまたがってトレーニングすることで、個々の言語レベルでのパフォーマンスが向上し、ゼロ・ショット翻訳が有効になることがわかるということが、この論文の結論なのだが、この結果は、これらのタスクがモデル内でどのように処理されるかについて、次のような問題を提起する。
 1. 言語にかかわらず、ネットワークは、同じ意味を持つ文章が同じような方法で表現される何らかの共有表現を学習しているのか？
 2. モデルは、訓練された言語ペアを扱うのと同じ方法で、ゼロ・ショット翻訳を実行しているのか？
-

ビジュアル化による分析

- ネットワークによって使用される表現を研究する1つの方法は、翻訳中のネットワークの活性化の状態を見ることである。
- 研究の出発点は、アテンション・ベクトルのセット、すなわち、エンコーダとデコーダのネットワークを接続するレイヤ内の活性化の状況を見ることだ。
- 単一の文の翻訳は、一連のアテンション・ベクトルを生じさせる。この文脈において、共有表現に関する元々の問題は、異なる文のベクトルのシーケンスがどのように関連しているかを調べることで研究することができる。例えば、次のように。
- ソースまたはターゲット言語に応じて、文章がクラスタリングされているのか？
- あるいは、言語にかかわらず、同様の意味を持つ文章がクラスタリングされるのか？

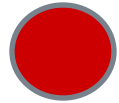
インターリンガの証拠

- いくつかの訓練されたネットワークは、実際に共有表現の強いビジュアルな証拠を示す。
- たとえば、以下の図2は、英語 <-> JapaneseとEnglish <-> Koreanでトレーニングされた多対多モデルから作成されたものである。モデルの実際の動作を視覚化するために、意味論的に同一の言語間フレーズの74個のトリプルからなる小さなコーパスから始めた。
- つまり、それぞれのトリプルには、同じ基本的な意味を持つ英語、日本語、韓国語のフレーズが含まれている。これらのトリプルをコンパイルするために、私たちは日本語と韓国語の翻訳と対になった、英語の文章のための正しい(Ground Truth)データベースを検索した。

英語



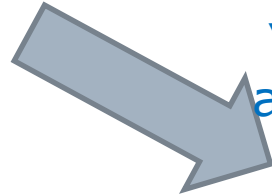
三つの言語で
同じ意味を持つ文を
一つのトリプルに
まとめる。



日本語

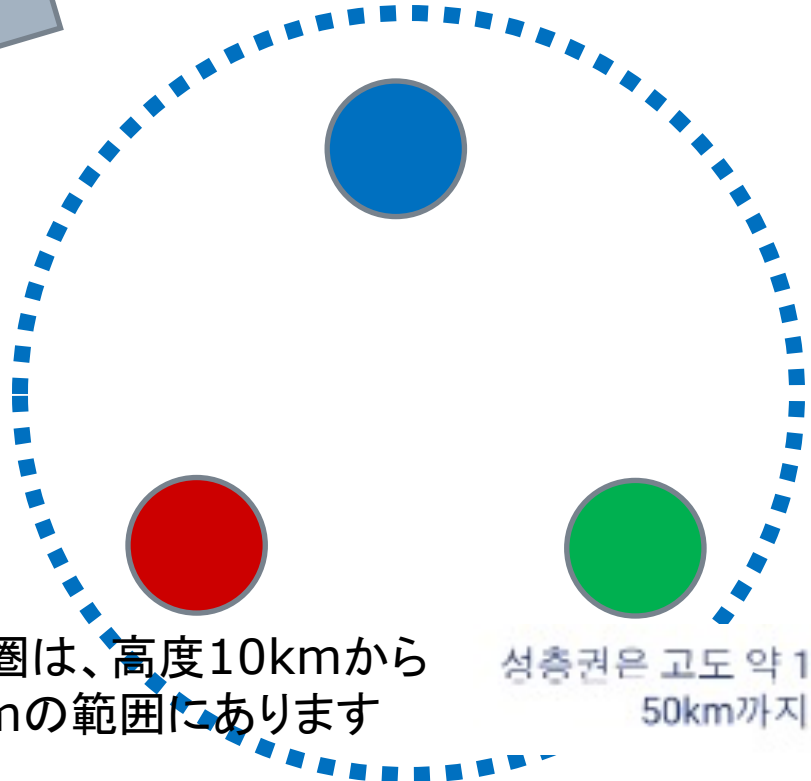


韓国語



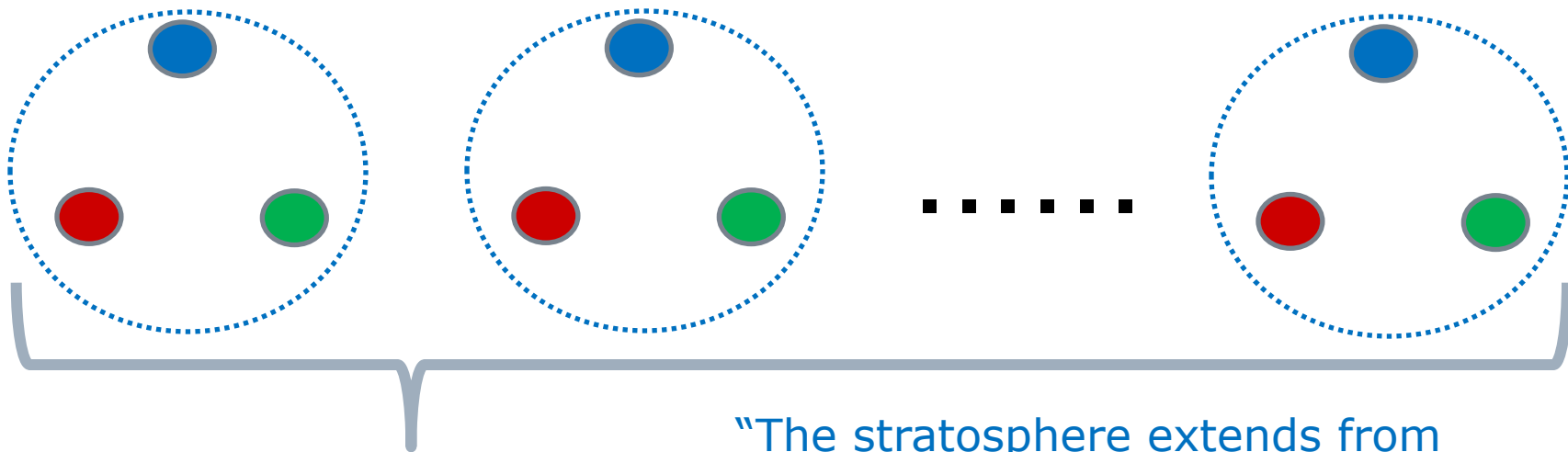
例えば

"The stratosphere extends from
about 10km to about 50km in altitude."



成層圏は、高度10kmから
50kmの範囲にあります

성층권은 고도 약 10km부터 약
50km까지 확장됩니다.

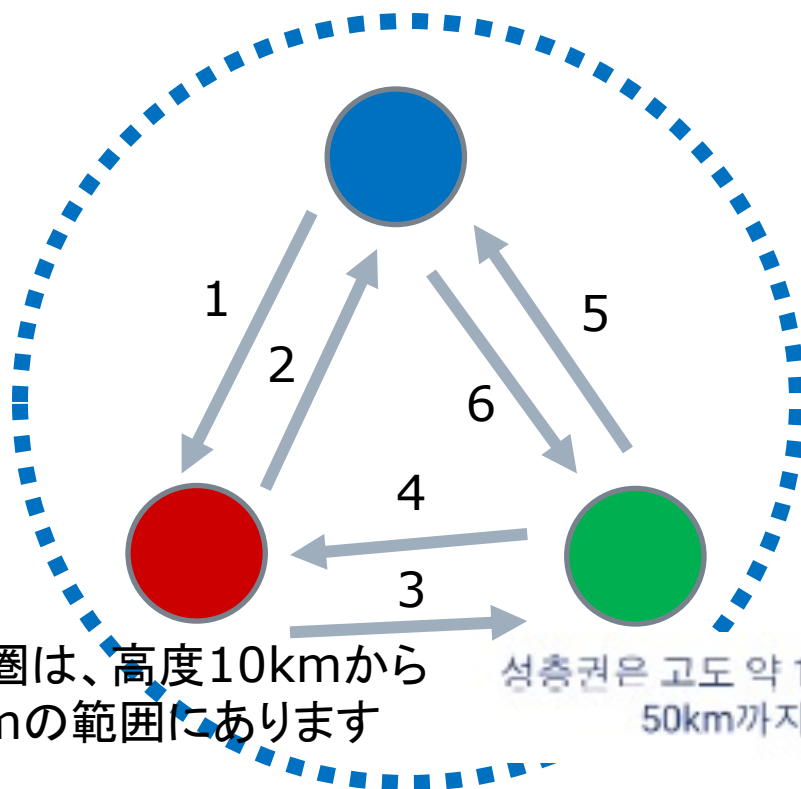


74個のトリプルからなる
コーパスを準備する。

各トリプルの各センテンスを
他の2つの言語に翻訳する。
一つのトリプルについて、
6つの翻訳が生まれる。

このコーパスでは、
 $74 \times 6 = 444$ の
翻訳が生まれる。

“The stratosphere extends from
about 10km to about 50km in altitude.”



成層圏は、高度10kmから
50kmの範囲にあります

성층권은 고도 약 10km부터 약
50km까지 확장됩니다.



この 444の翻訳を
行うのに、システムは
9,978ステップ要した。

この時、発せられた
9,978個のアテンション
ベクトルを、t-SNEを用
いて二次元に投射した
のが左の図。

同じトリプル内の文の
翻訳で発せられるアテン
ション・ベクトルは、同じ
色で表している。

同じ色が、近くに集まっ
てクラスターを形成して
いるのが見て取れる。
(?)

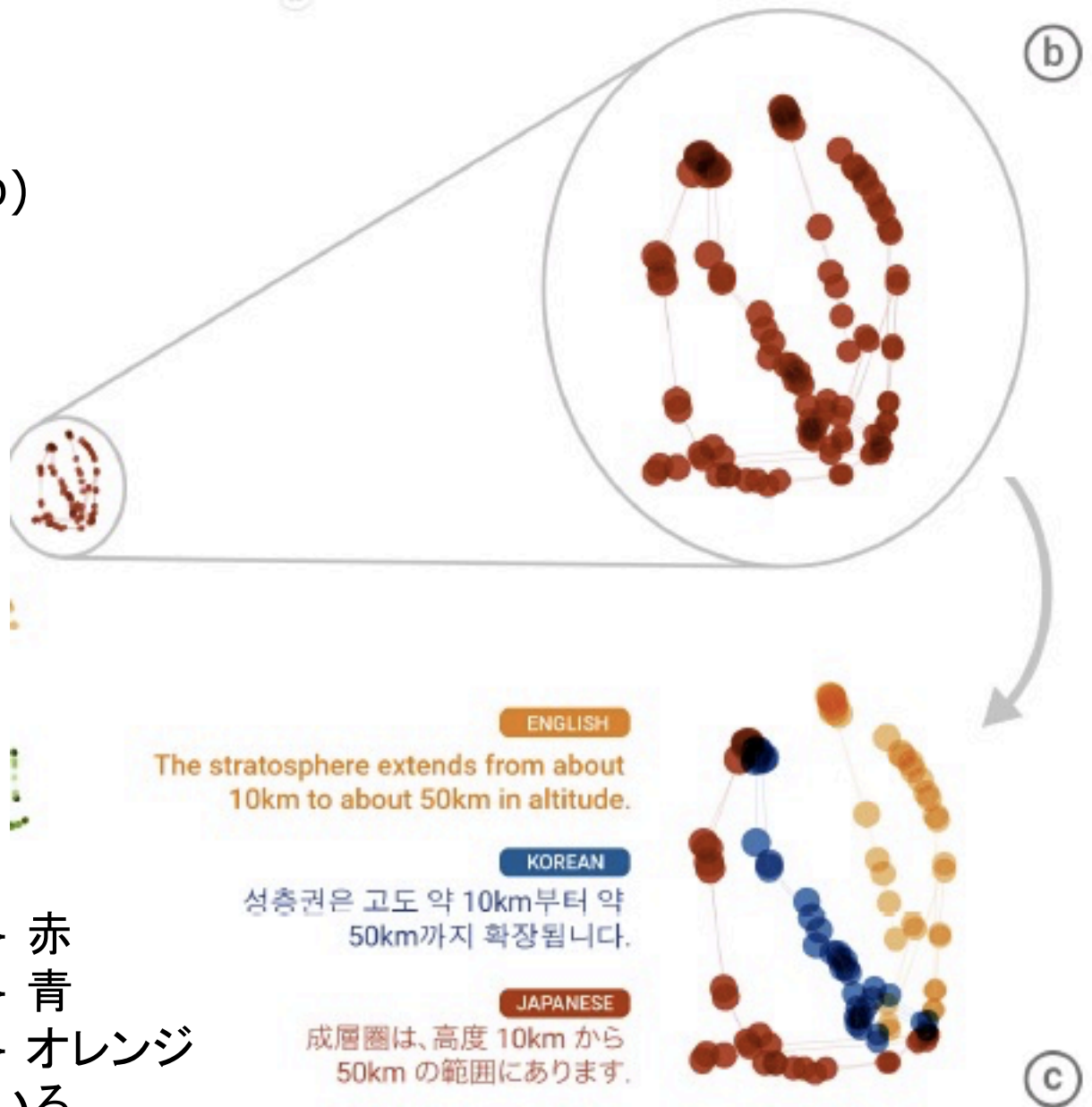
言語にかかわらず、同様の意味を持つ文章が
クラスタリングされている。

この同じ色のクラスター(b)
は、元の同じ意味を持つ
トリプルから生まれたもの
である。

同一の文から発せられる
アテンション・ベクトルは、
線で結ばれている。

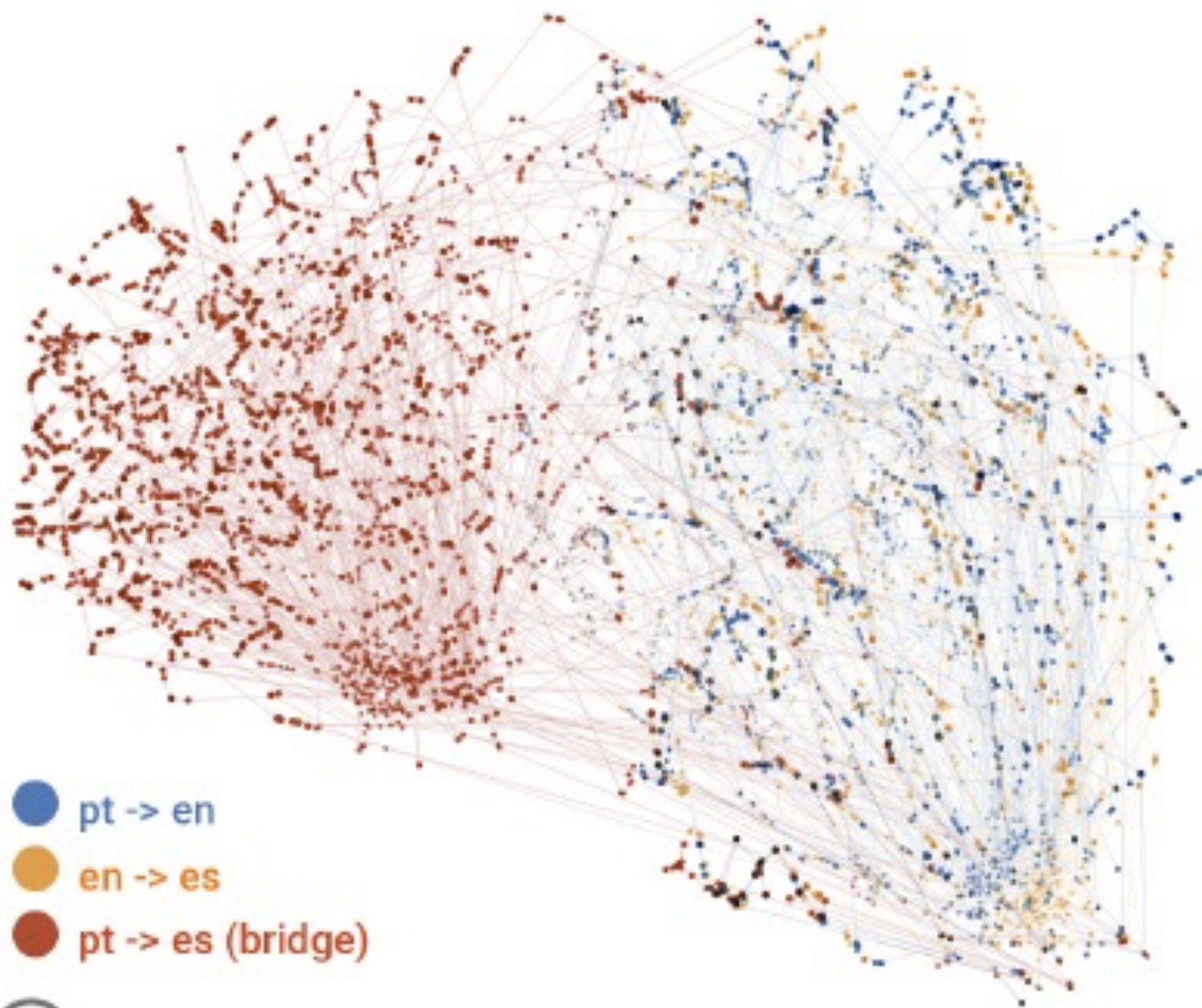
(c)は、翻訳ソースが同じ
言語を同じ色で塗り分け
たもの。

日本語->{英語,韓国語} 赤
韓国語->{英語,日本語} 青
英語->{日本語,韓国語} オレンジ
は、クラスターを形成している。



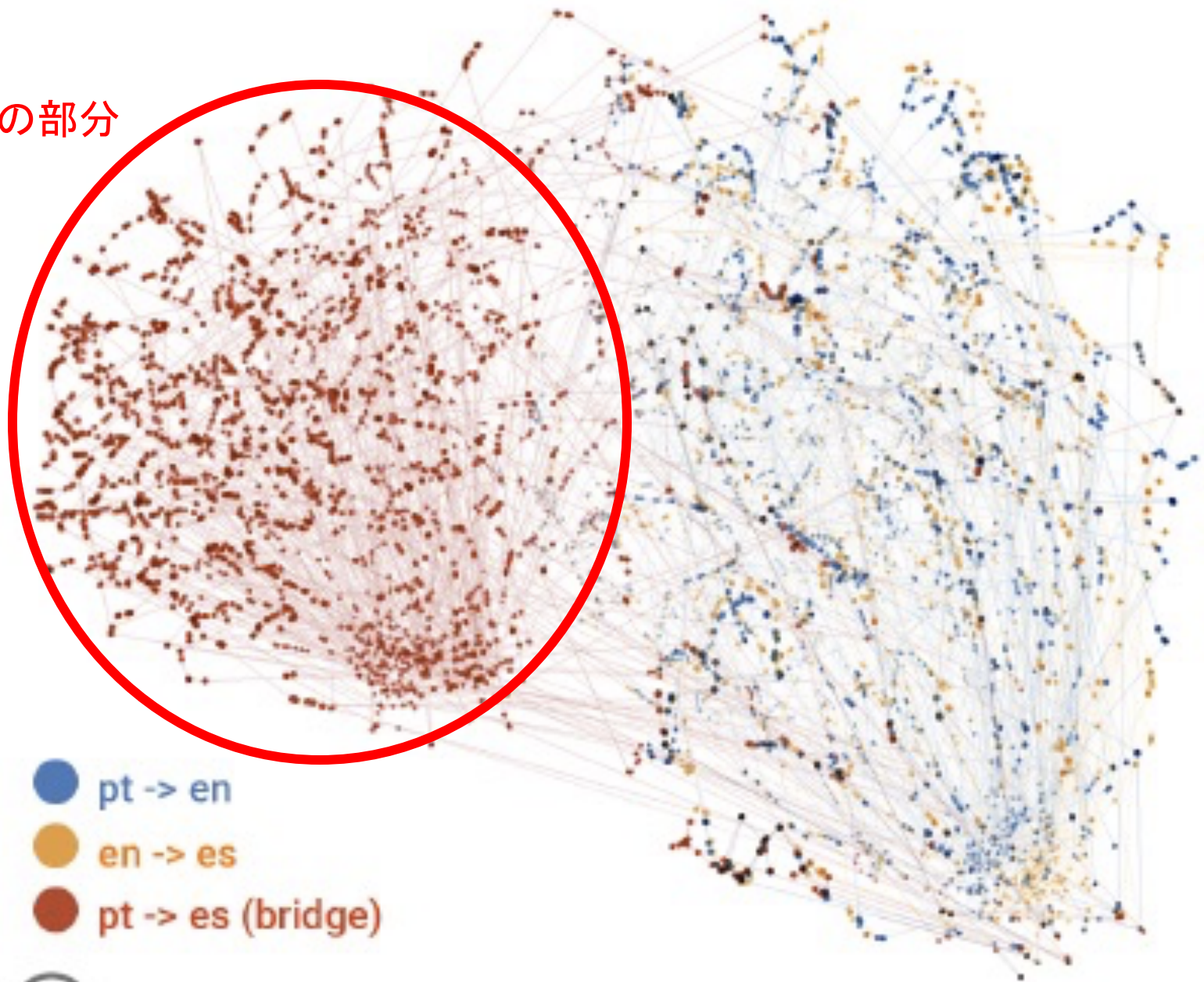
部分的に分離された表現

- すべてのモデルが、そのようなクリーンな意味クラスタリングを示すわけではない。ある一つの言語ペアからだけの多くのアテンション・ベクトルを含む、大きなクラスターを観察した。
- たとえば、次の図は、ポルトガル語 -> 英語(青)と英語 -> スペイン語(黄色)で訓練され、ポルトガル語からスペイン語(赤)のゼロショット変換を実行したモデルからの注意ベクトルのt-SNEプロジェクションを示しています。
- 左の大きな赤い領域には、主にゼロショットのポルトガル語 -> スペイン語の翻訳が含まれている。言い換えれば、かなりの数の文では、ゼロ・ショット翻訳は、2つの訓練された翻訳方向とは異なる埋め込みを持っている。



a

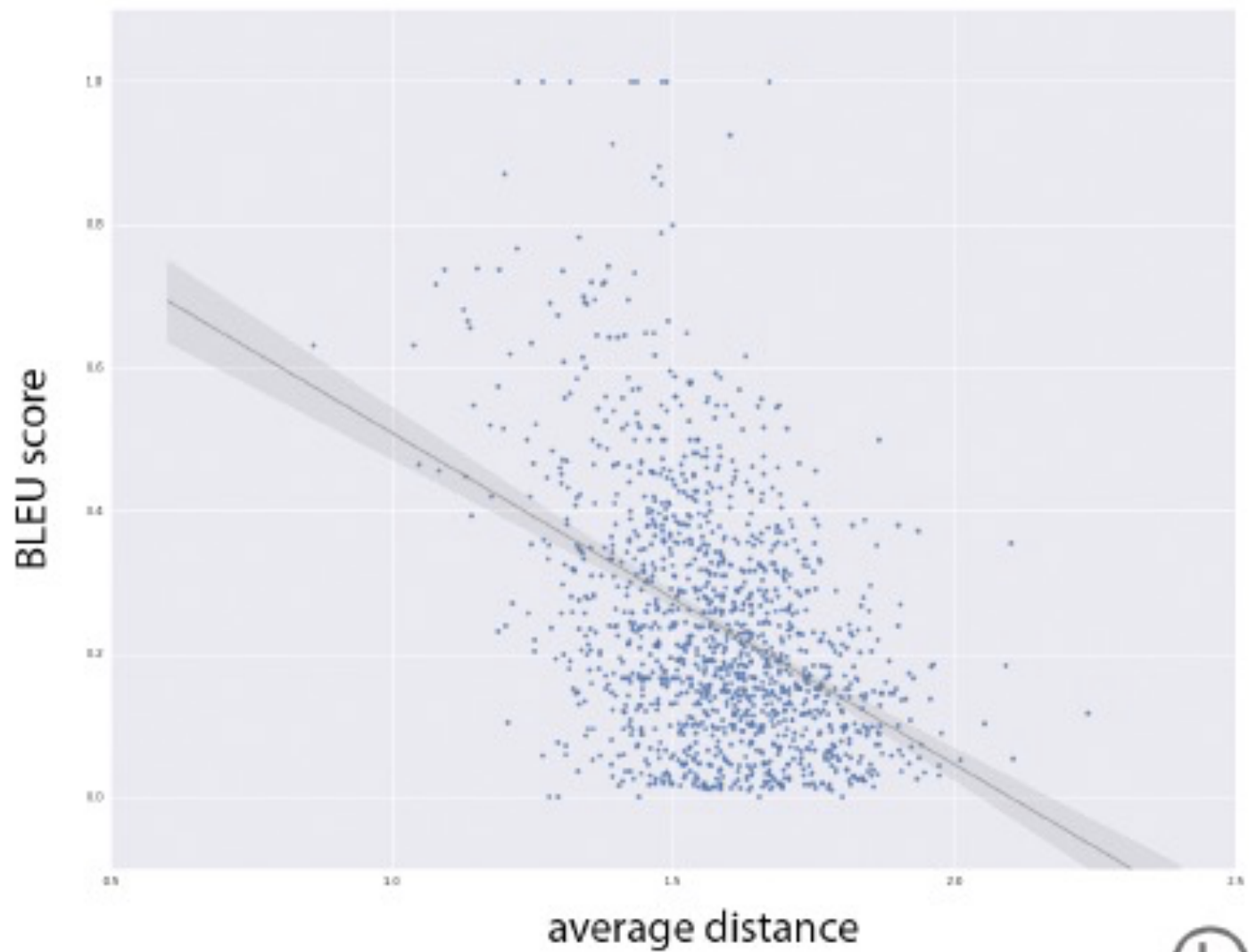
この部分



a

-
- 「分離された」ゼロ・ショット変換の大きなクラスターが、何らかの重要性を持っているかどうかを尋ねるのは自然なことだ。
 - 確定的な回答にはさらなる調査が必要なのだが、この場合、分離された領域のゼロ・ショット翻訳ではBLEUスコアが低くなる傾向がある。(次の図) この相違度スコアの値はピアソン相関係数が-0.42のゼロショット変換の品質と相関があり、中程度の相関を示している。
 - 将来の研究のための興味深い領域は、訓練された言語ペアを介した翻訳の埋め込みと比較することによって、デコーダ内のゼロショット変換の品質を予測するために、埋め込みの地図とモデル性能との間のより信頼できる対応を見つけることである。
-

ゼロ・ショット翻訳のBLEUスコアと
ゼロ・ショット翻訳とノンブリッジ翻訳
との平均点距離の散布図



混合言語

多言語モデルで、文章の途中で言語を切り替えて混在させるとどうなるか？ 訓練データ中には、そうしたチャンポンの例はないにもかかわらず。正しい翻訳が得られる。

出力をチャンポンにできるか？ それは難しかった。

言語の混合

- 追加のソーストークンを使用してランダムなソース言語から選択した単一のターゲット言語に翻訳する仕組みを持っているので、ソースまたはターゲット側で言語が混在しているときに何が起こるかを考え、ました。次の2つの実験を行った。
 - 1. 多言語モデルは、文章の途中での多言語入力(コード切り替え)をうまく処理できるか？
 - 2. 多言語モデルがトリガーされ、1つではなく2つのターゲット言語トークンが重み付けされて1つに加わる(これらのトークンの重み付けされた埋め込みをマージするのと同様)場合、どうなるか？
-

ソース言語でのコード転換

- ここでは、{日本語、韓国語} -> 英語 データで訓練された多言語モデルの例を示す。このモデルを使用して、ソースに日本語と韓国語を混在させると、このモデルではコード切り替えが処理できる。
- そのようなコード切り替えサンプルはトレーニングデータには存在しないにもかかわらず、多くの場合正しい英語の翻訳が生成される。
- モデルには、個々の文字/ワードピースが語彙ボキャブラリー内に存在するため、異なるタイプの記述を効果的に処理できるのだ。

- Japanese: 私は東京大学の学生です。 → I am a student at Tokyo University.
- Korean: 나는 도쿄 대학의 학생입니다. → I am a student at Tokyo University.
- Mixed Japanese/Korean: 私は東京大学학생입니다. → I am a student of Tokyo University.

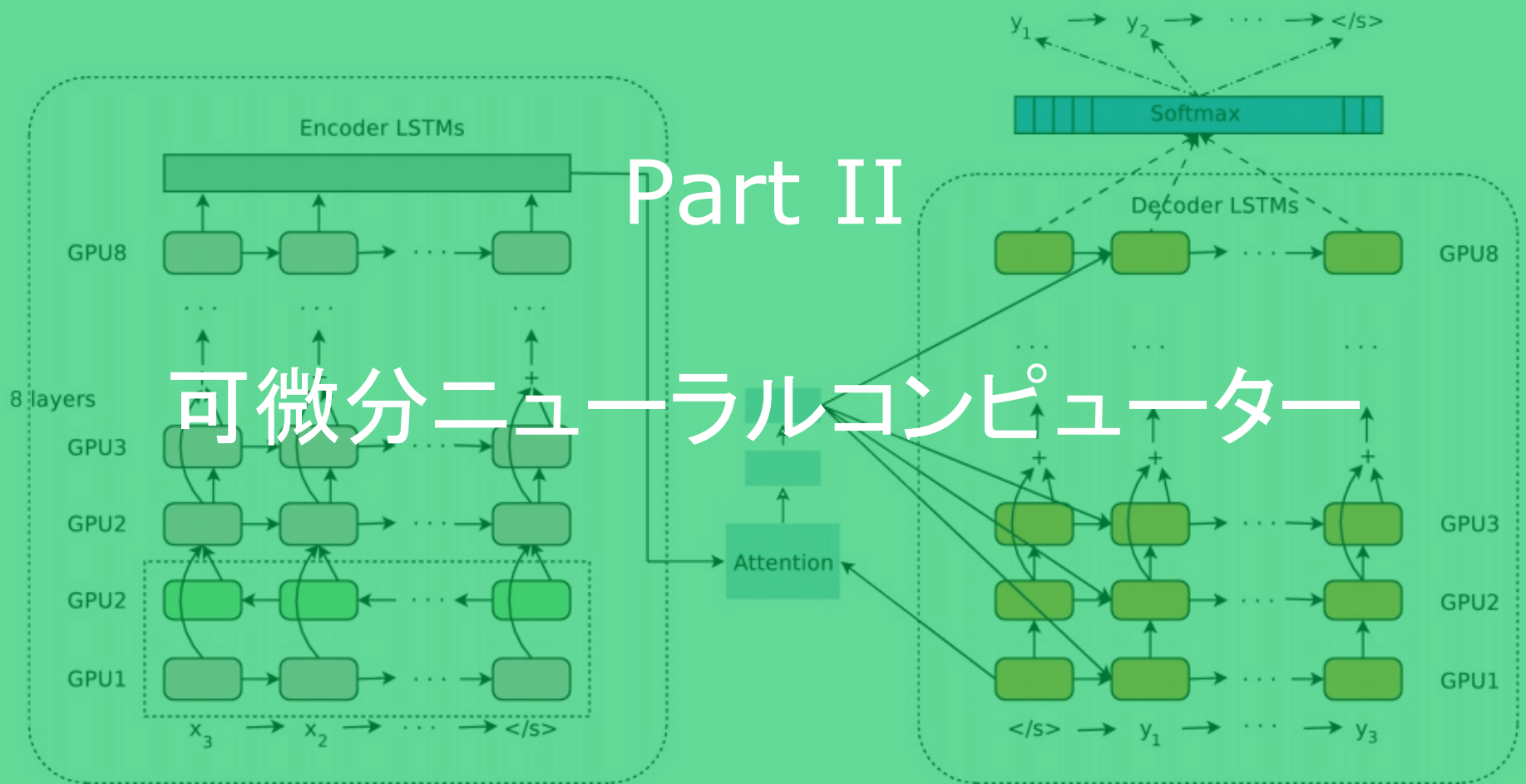
重みづけられたターゲット言語の選択

- エンコーダLSTMの最下層に “<2ja>” の埋め込みベクトルを与えるのではなく、線形結合 $(1-w) \langle 2ja \rangle + w \langle 2ko \rangle$ を与える。
- 明らかに、 $w = 0$ の場合、モデルは日本語を生成する必要があり、 $w = 1$ の場合、それは韓国語を生成するはずだが、 w がその間の値をとる時、何が起こるのかということ。
- 結果はそれほど驚くことではない。たいていの場合、出力は $w = 0.5$ 付近で言語を切り替える。場合によっては、 w の中間の値について、モデルは文の途中で、言語を切り替える。
- この動作の可能な説明は、デコーダLSTMによって暗黙的に学習されたターゲット言語モデルが、特にこれらの言語が異なった記述を使用する場合、異なる言語の単語を混在させることを非常に困難にする可能性があるということ。

Russian/Belarusian:	I wonder what they'll do next!	
$w_{be} = 0.00$	Интересно, что они сделают дальше!	
$w_{be} = 0.20$	Интересно, что они сделают дальше!	
$w_{be} = 0.30$	Цікаво, что они будут делать дальше!	
$w_{be} = 0.44$	Цікаво, що вони будуть робити далі!	ウクライナ語が出てくる！
$w_{be} = 0.46$	Цікаво, що вони будуть робити далі!	
$w_{be} = 0.48$	Цікаво, што яны зробіць далей!	
$w_{be} = 0.50$	Цікава, што яны будуць рабіць далей!	
$w_{be} = 1.00$	Цікава, што яны будуць рабіць далей!	
Japanese/Korean:	I must be getting somewhere near the centre of the earth.	
$w_{ko} = 0.00$	私は地球の中心の近くにとこかに行っているに違いない。	
$w_{ko} = 0.40$	私は地球の中心近くのとこかに着いているに違いない。	
$w_{ko} = 0.56$	私は地球の中心の近くのとこかになっているに違いない。	
$w_{ko} = 0.58$	私は지구의中心의가까이에어딘가에도착하고있어야한다。	
$w_{ko} = 0.60$	나는지구의센터의가까이에어딘가에도착하고있어야한다。	
$w_{ko} = 0.70$	나는지구의중심근처어딘가에도착해야합니다。	
$w_{ko} = 0.90$	나는어딘가지구의중심근처에도착해야합니다。	
$w_{ko} = 1.00$	나는어딘가지구의중심근처에도착해야합니다。	
Spanish/Portuguese:	Here the other guinea-pig cheered, and was suppressed.	
$w_{pt} = 0.00$	Aquí el otro conejillo de indias animó, y fue suprimido.	
$w_{pt} = 0.30$	Aquí el otro conejillo de indias animó, y fue suprimido.	
$w_{pt} = 0.40$	Aquí, o outro porquinho-da-índia alegrou, e foi suprimido.	
$w_{pt} = 0.42$	Aqui o outro porquinho-da-índia alegrou, e foi suprimido.	
$w_{pt} = 0.70$	Aqui o outro porquinho-da-índia alegrou, e foi suprimido.	
$w_{pt} = 0.80$	Aqui a outra cobaia animou, e foi suprimida.	
$w_{pt} = 1.00$	Aqui a outra cobaia animou, e foi suprimida.	

Part II

可微分ニューラルコンピューター



可微分ニューラルコンピューター

Googleニューラル機械翻訳も素晴らしいのだが、ニューラルネットワークの世界では、さらに大きな変化が進行中である。その最前線と言っているのが、去年の10月に、GoogleのDeepMindチームが、Nature誌に発表した次の論文である。

Alex Graves et al. **"Hybrid computing using a neural network with dynamic external memory"** <https://goo.gl/r4g5xt>

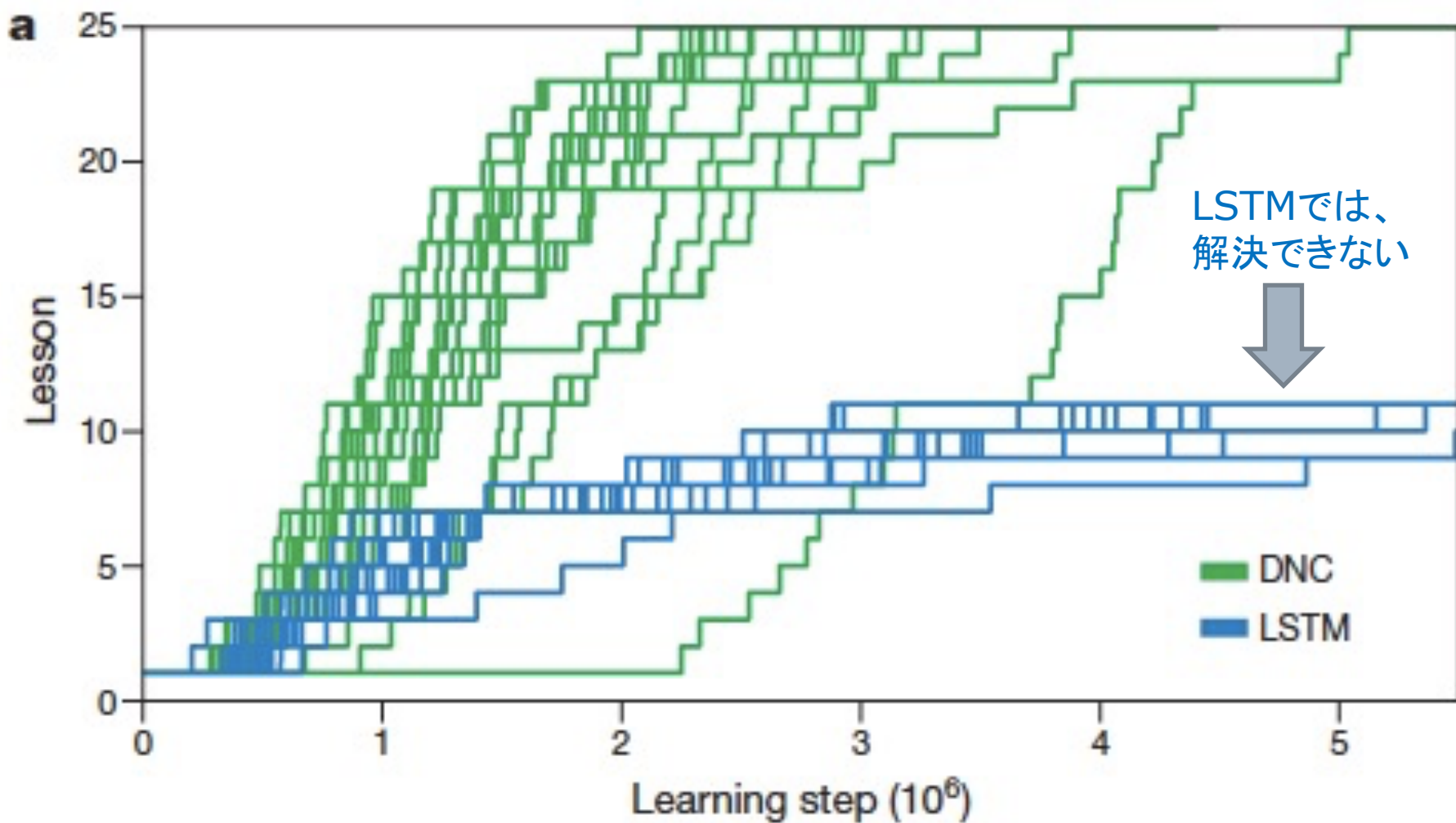
人工のニューラルネットワークは、感覚の処理、シーケンスの学習、強化学習には、極めて熟練しているのだが、変数やデータ構造を表現したり、長い時間のスケールでデータを格納する能力には限界を持っている。そうした限界は、ニューラルネットワークが、外部メモリを持たないことによるものだ。

この論文で、我々は、「可微分ニューラル・コンピュータ(DNC: Differentiable Neural Computer)」と呼ばれる機械学習モデルを提案する。このモデルは、従来のコンピュータのランダムアクセスメモリに類似した、外部メモリ行列との間で読み書きが可能なニューラルネットワークで構成される。それは、従来のコンピュータと同じ様に、複雑なデータ構造を表現し操作するためにメモリを使用できるが、ニューラルネットワークのように、データからそれを行う方法を学ぶことができる。

DNCは、複雑な問題の解決能力を持つ

- DNCを教師あり学習で訓練すると、それは、自然言語で書かれた論証や推論の問題を真似するように作られた総合的な質問に正しく答えることができることを示す。我々は、DNCが、ある特定の点の間の最短経路を見つけ出したり、無作為に生成されたグラフで欠けているリンクを推定したりするタスクを学習することが出来ることを示す。さらに、DNCは、これらのタスクを輸送ネットワークや家系図などの特定のグラフに一般化することもできる。
- DNCを強化学習で訓練すると、それは、目標の変化を記号列で指定するブロックを移動させるパズルを解くことが出来た。
- これらをまとめると、我々の結果は、DNCが、外部読み書きメモリなしのニューラルネットワークでは到達できない、複雑で構造化されたタスクを解決する能力を持っていることを示している。

DNCは、LSTMでは解決できない問題を
解くことができる



メモリー利用の利点

- 現代のコンピュータは、計算とメモリーを分離する。計算は、プロセッサによって実行される。プロセッサは、計算の対象を出し入れするのに、アドレス可能なメモリーを利用することができる。
 - これにより、2つの重要な利点を得られる。新しい情報を書き込むための拡張可能なストレージの利用と、メモリーの内容を変数として扱う機能である。変数はアルゴリズムの汎用性にとって非常に重要である。つまり、同じ手続きを、一つのデータだけでなく他のデータで実行するのに、アルゴリズムは、それが読み出すデータのアドレスを変更するだけでいいからである。
-

ニューラル・ネットワークの弱点

- コンピュータとは対照的に、人工ニューラルネットワークの計算リソースとメモリーのリソースは、ニューラルネットワークの重みとニューロンの活動の中に、混じり合っている。
 - これは、大きな負債である。タスクの増大によってメモリが必要とされても、これらのネットワークは、新しいストレージを動的に割り当てることもできず、タスクの変数によって実現される、値とは独立に動作するアルゴリズムも簡単には学習できないからである。
-

-
- 最近のブレイクスルーは、ニューラルネットワークは感覚の処理、シーケンスの学習と強化学習には、極めて熟練していることを示しているのだが、認知科学者や神経科学者は、ニューラルネットワークには、変数やデータ構造を表現する能力と外部からの干渉なしに長い期間にわたってデータを格納する能力では限界があると論じてきた。
 - 我々は、ニューラルネットワークに外部メモリへの読み書きのアクセスを提供することで、ニューラルコンピューティングとコンピュータによる処理の利点を組み合わせることを目指している。
 - メモリーへのアクセスは、記憶間の干渉を最小限に抑える、かつ、長期間の保存を可能にする為に、狭いところに集中して行われる。システム全体は微分可能であり、それ故、勾配降下法でエンドツーエンドで訓練することができ、ネットワークは目標指向の方法でメモリを操作し、編成する方法を学習できる。

DNCシステム概論

-
- DNCは外部メモリ行列に結合されたニューラルネットワークである。(ネットワークの動作は、メモリが容量まで満たされていない限りメモリのサイズに依存しないため、メモリは「外部」と見なされる)。このメモリ行列をDNCのRAMと考えるならば、「コントローラ」と呼ばれるネットワークは、微分可能なCPUである。その動作は、勾配降下法で学習される。
 - DNCアーキテクチャは、メモリを選択的に読み書きすることができ、メモリの内容を繰り返して修正することが可能である点で、近年のニューラル・メモリー・フレームワークとは異なっている。DNCの以前の形式であるニューラル・チューリング・マシンは、同様の構造を持っていたが、メモリアクセスの方法がより制限されていた(詳細は「Method」の章を参照されたい)。
-

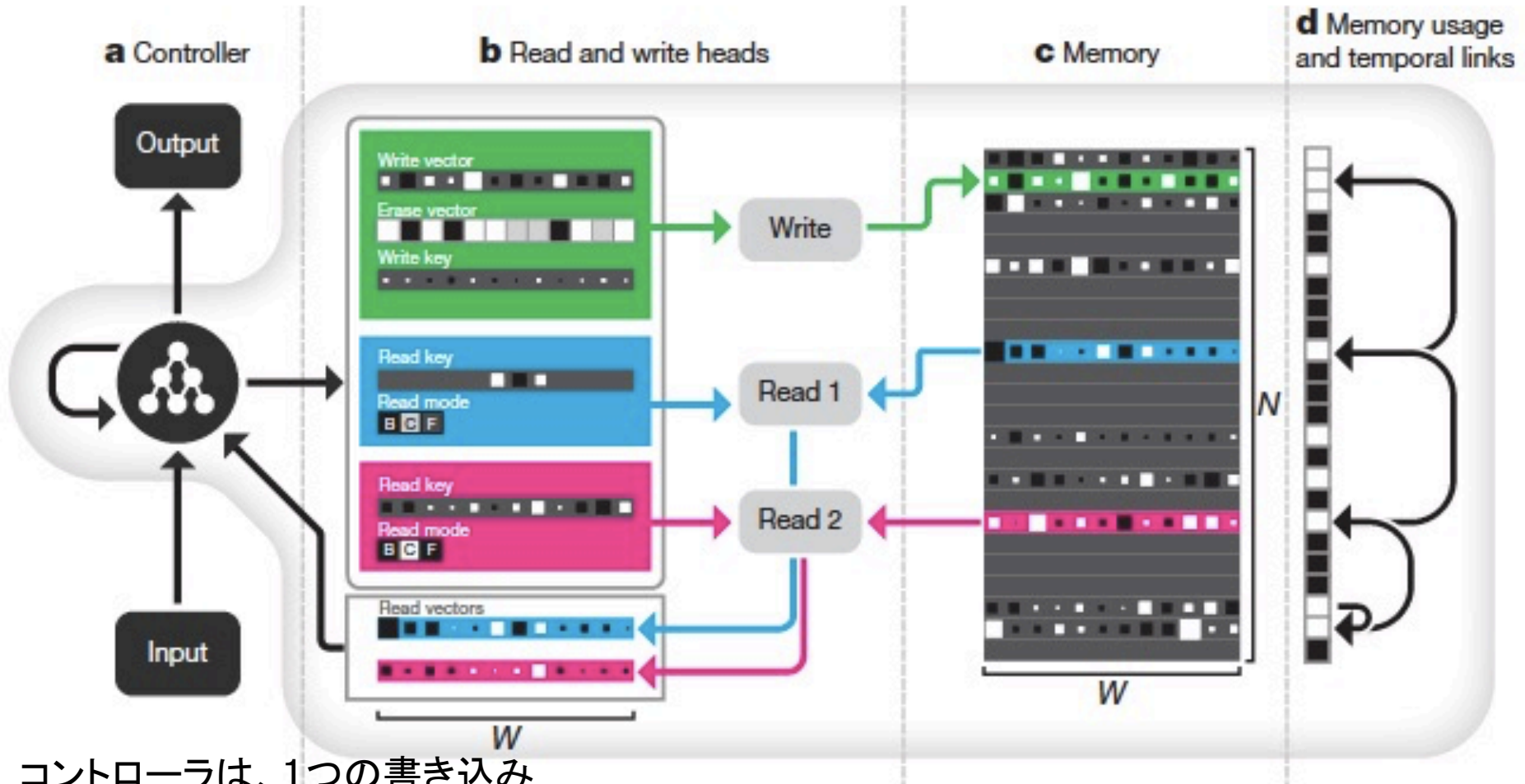
-
- 従来のコンピュータが、メモリ内容にアクセスするためにユニークなアドレスを使用するのに対し、DNCは微分可能なAttentionメカニズムを使用して、 $N \times W$ メモリ行列である M 内の N 行、「ロケーション」上に分布を定義する。この分布が、各ロケーションが読取りまたは書込み操作に、関わる程度を表現している。我々はそれを「重み付け」と呼んでいる。
 - メモリ M 上の読み出しで、重み付け w^r によって返される読み出しベクトル r は、メモリ・ロケーションにわたって重み付けされた和である。 $r = \sum_{i=1}^N M[i, \cdot] w^r[i]$ である。ここで、「 $\cdot$ 」は全ての $j = 1, \dots, W$ を表している。



-
- 同様に、書き出しの操作は、書き出しの重み付け w を使う。まず、消去ベクトル e を使って消去し、その後で、書き出しのベクトル v を追加する。 $M[i,j] \mapsto M[i,j](1-w[i]e[j]) + w[i]v[j]$
 - 重みづけを決定して適用する機能の単位は、読み書きヘッドと呼ばれる。ヘッドの動作は、以下に要約する。形式的な記述は、「Methodの」章を参照されたい。
-

Controller

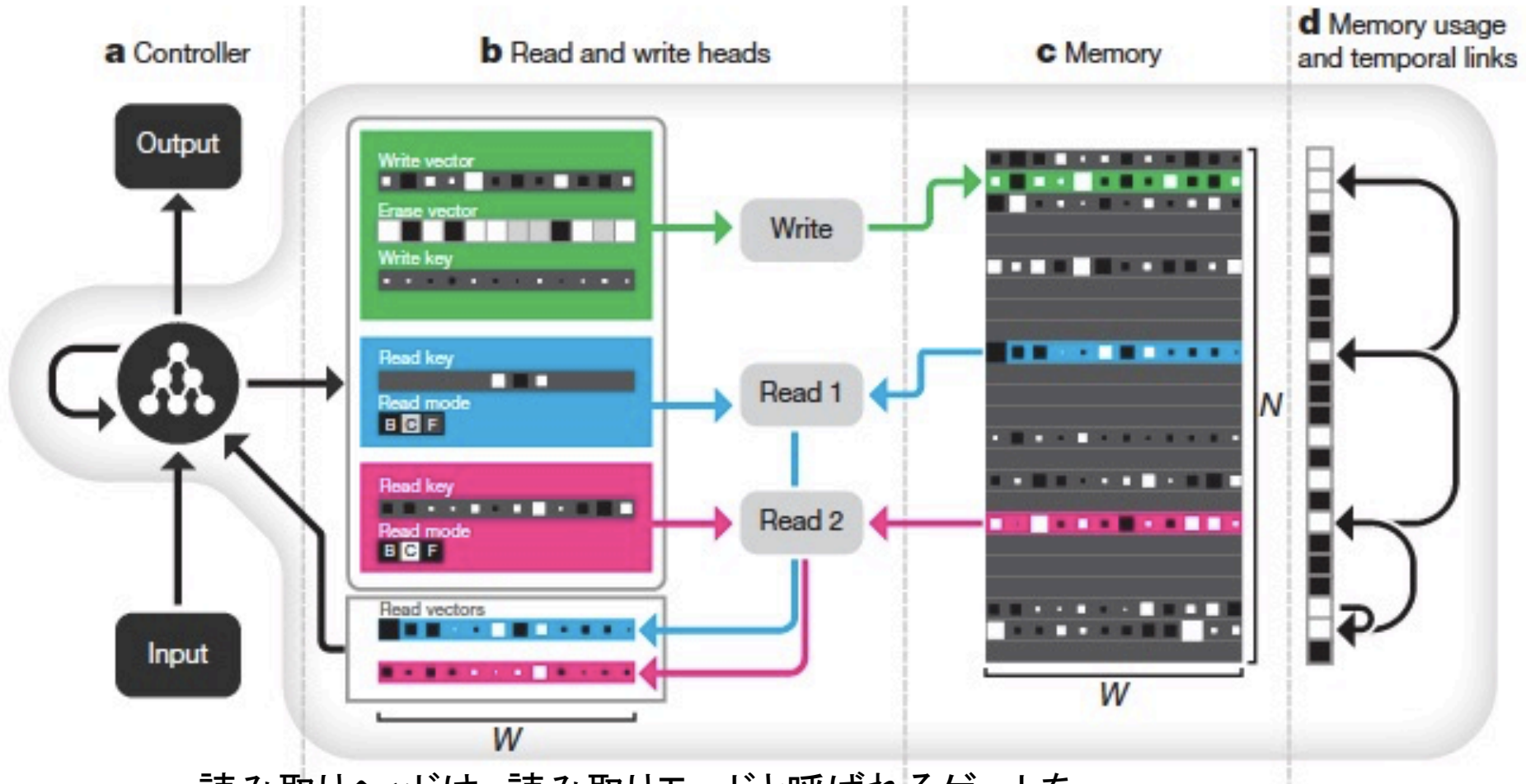
リカレント・コントローラ・ネットワークは、
外部のデータソースから入力を受け取り、
出力を生成する。



コントローラは、1つの書き込み
ヘッド(緑)と複数の読み取りヘッド
(この場合は2つの青とピンク)を
パラメータ化するベクトルも出力する。

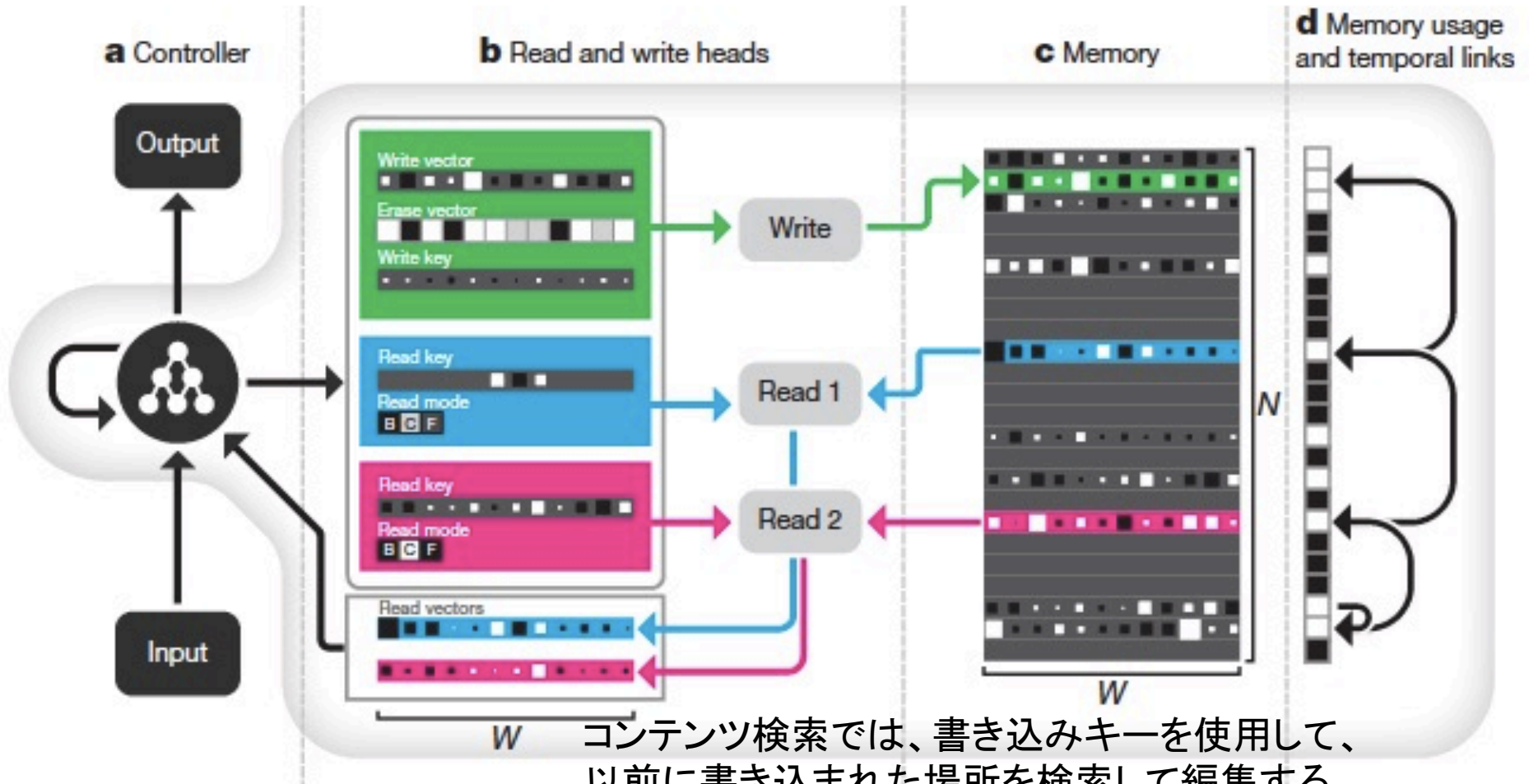
Head

書き込みヘッドは、 $N \times W$ メモリ行列を編集するために使用される書き込みベクトルおよび消去ベクトルを定義し、その要素の大きさおよび符号は、それぞれ四角の領域および色の濃さによって示されている。



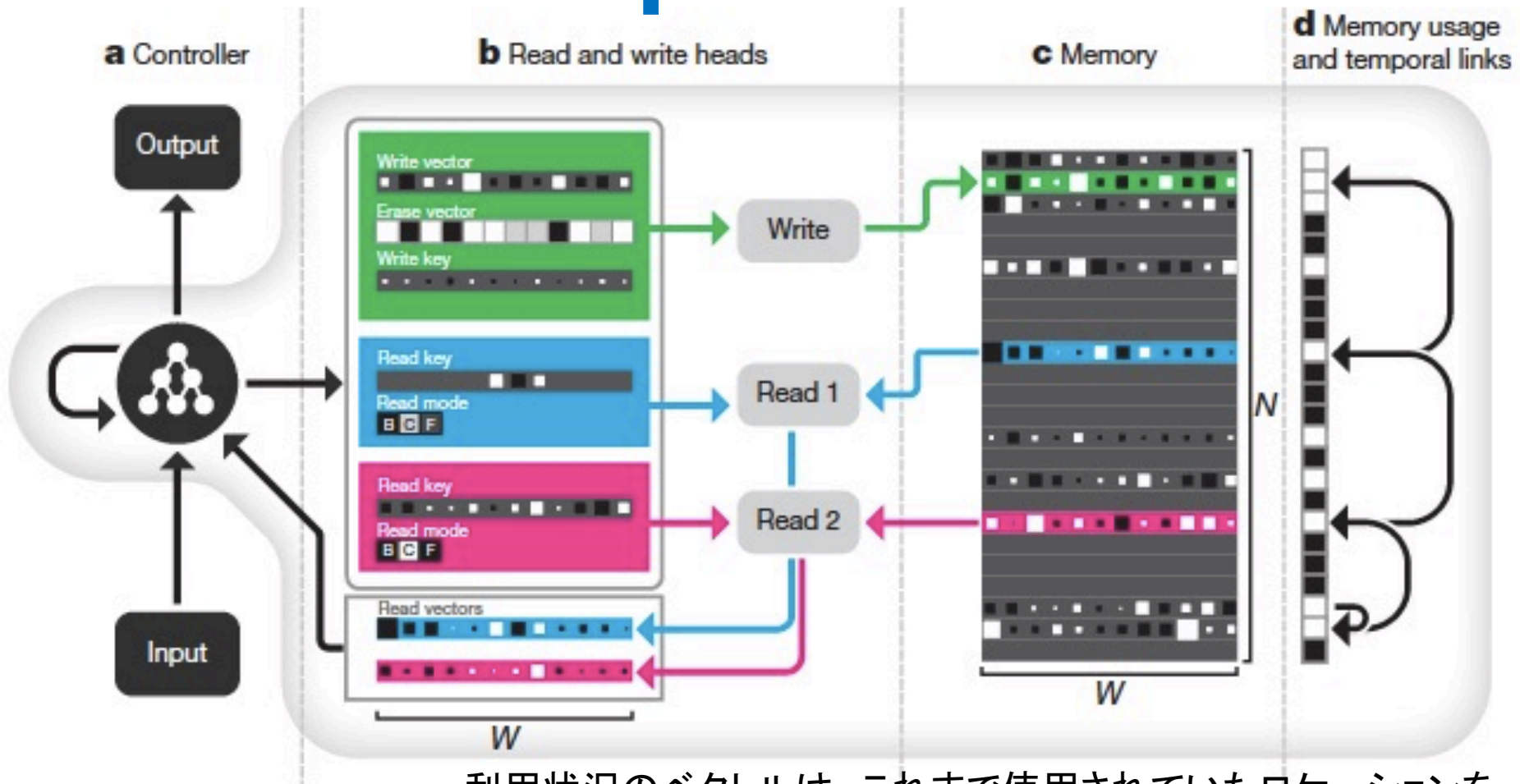
読み取りヘッドは、読み取りモードと呼ばれるゲートを使用して、読み取りキー('C')を使用してコンテンツ検索と、書かれた順に前方('F')または後方('B')への読み出しとを切り替える。

Content Lookup



コンテンツ検索では、書き込みキーを使用して、以前に書き込まれた場所を検索して編集する。書き込みキーは、メモリ行列内の行 すなわちロケーションにわたって書き込み動作を選択的に集中させる重み付けの定義に、影響を与えることができる。

Memory Usage and Temporal Links



利用状況のベクトルは、これまで使用されていたロケーションを記録し、時間的リンク・行列は、ロケーションに書き込まれた順序を記録する。ここでは、ロケーションの順序は、矢印で表されている。

ヘッドとメモリーの間の相互関係

ヘッドは3つの異なる形の微分可能なAttentionを用いる。

コンテンツ検索

- 第一は、コンテンツ検索である。そこでは、コントローラによって出力されたキー・ベクトルが、類似性の尺度(コサイン類似度を使う)に従って、メモリ内の各ロケーションの内容と比較される。この類似性のスコアが、連想記憶のために読取りヘッドによって利用される重み付け、あるいは、メモリ内の既存のベクトルを修正するために書込みヘッドによって使用される重み付けを決定する。
- 重要なことは、メモリ・ロケーションのコンテンツと部分的にしか一致しないキーも、そのロケーションに強く注目するために使用できるということである。これによって、パターンを完成させるような形式も可能になる。なぜなら、メモリのロケーションを読み取ることによって回復される値には、キーには表現されていない追加の情報が含まれているからである。

-
- 一般に、キー値での値の検索は、外部メモリ内の連想データ構造をナビゲートするための豊富なメカニズムを提供する。これは、あるアドレスのコンテンツが他のアドレスへの参照を効果的にエンコードできるからである。



時間的リンク行列

- 第二のAttentionメカニズムは、 $N \times N$ の時間的リンク行列 L である。それは、連続して書き込まれたロケーション間の遷移を記録する。 $L[i, j]$ は、 i が j の後に書き込まれた次の位置であれば1に近く、そうでない場合には0に近い値を持つ。
- 任意の重みづけ w に対して、操作 Lw は、フォーカスを、 w で強調された後に書き込まれた位置に、前方に滑らかにシフトさせる。 $L^T w$ は、フォーカスを後方にシフトする。これにより、DNCは、連続した書き込みが、近接した時間の範囲内に発生しなかった場合でも、シーケンスを書き込んだ順序で復元するネイティブな機能を提供する。

メモリー割り当て

- 第三のAttentionの形式は、書き込みのためのメモリーを割り当てる。各ロケーションの「利用状況」は0と1の間の数として表され、使用されていない場所を選ぶ重み付けが、書込みヘッドに送られる。
- 利用状況は、ロケーションへの書き込みごとに自動的に増加するだけでなく、読み取りごと減らすことができる。これにより、コントローラは不要になったメモリーを再割り当てできる。
- 割り当てメカニズムはメモリーのサイズと内容とは無関係である。つまり、あるサイズのメモリーを使用してあるタスクを解決するためにDNCを訓練し、後で再訓練することなく、大容量のメモリーにアップグレードすることができる。原理的には、これは、任意のロケーションの最小使用可能量が特定の閾値を超えるたびに、自動的にロケーションの数を増加させて、無制限の外部メモリーを使用することを可能にする。

海馬の機能との類似

- Attentionメカニズムのデザインは、主に計算上の考慮によって動機づけられている。コンテンツ検索は、連想データ構造の形成を可能にする。時間的リンクは、入力シーケンスの逐次検索を可能にする。メモリーの割り当ては、書き込みヘッドに未使用のロケーションを提供する。
 - しかし、興味深いのは、DNCの記憶メカニズムと哺乳動物の海馬の機能の能力との間には、類似点があることである。DNCの記憶の修正は速く一次的であり得るのだが、それは、海馬CA3およびCA1シナプスの長期的な増強に似ている。
-

-
- 海馬歯状回は、ニューロンの形成をサポートすることが知られている領域であるが、そこでは記憶の表現をさらに疎なものにして、それにより記憶容量を拡大しているのではと提案されている。我々のモデルでの、利用状況に基づくメモリ割り当てと疎な重み付けは、それと類似の機能を提供しているのかもしれない。
 - 人間の「自由想起」の実験では、最初に提示されたのと同じ順序の時、想起の確率が高まることが示されるのだが、それは、時間的なコンテキスト・モデルによって説明される海馬依存の現象である。それは、我々のモデルでの時間的なリンクの形成とある類似性を有する。（「method」の章を参照されたい。）
-

DNCは、どんな課題に
取り組んでいるか？

Synthetic question answering experiments

John is in the playground.
Bob is in the office.
Where is John? **A:playground**

John is in the playground.
Bob is in the office.
John picked up the football.
Bob went to the kitchen.
Where is the football? **A:playground**
Where was Bob before the kitchen? **A:office**

John picked up the apple.
John went to the office.
John went to the kitchen.
John dropped the apple.
Where was the apple before the kitchen? **A:office**

Towards AI-Complete Question Answering: A Set of Prerequisite Toy Tasks

- One long-term goal of machine learning research is to produce methods that are applicable to reasoning and natural language, in particular building an intelligent dialogue agent. To measure progress towards that goal, we argue for the usefulness of a set of proxy tasks that evaluate reading comprehension via question answering. Our tasks measure understanding in several ways: whether a system is able to answer questions via chaining facts, simple induction, deduction and many more.

-
- The tasks are designed to be prerequisites for any system that aims to be capable of conversing with a human. We believe many existing learning systems can currently not solve them, and hence our aim is to classify these tasks into skill sets, so that researchers can identify (and then rectify) the failings of their systems. We also extend and improve the recently introduced Memory Networks model, and show it is able to solve some, but not all, of the tasks.

Sample

1 Mary moved to the bathroom.

2 John went to the hallway.

3 Where is Mary? bathroom 1

4 Daniel went back to the hallway.

5 Sandra moved to the garden.

6 Where is Daniel? hallway 4

7 John moved to the office.

8 Sandra journeyed to the bathroom.

9 Where is Daniel? hallway 4

10 Mary moved to the hallway.

11 Daniel travelled to the office.

12 Where is Daniel? office 11

13 John went back to the garden.

14 John moved to the bedroom.

15 Where is Sandra? bathroom 8

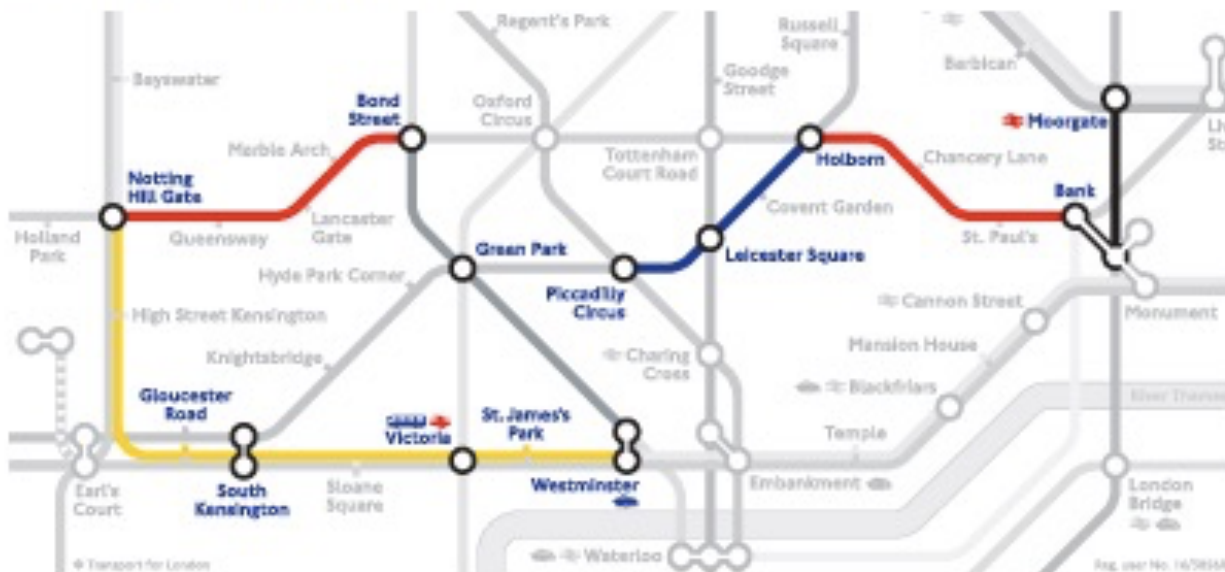
We found that a single DNC, jointly trained on all 20 question types with 10,000 instances each, was able to achieve a mean test error rate of 3.8% with task failure (defined as $> 5\%$ error) on 2 types of questions, compared to 7.5% mean error and 6 failed tasks for the best previous jointly trained result.

We also found that DNCs performed much better than both long shortterm memory (LSTM; at present the benchmark neural network for most sequence processing tasks) and the neural Turing machine.

Graph experiments

Such sentences could easily be rendered in graphical form: for example “John is in the playground” can be diagrammed as two named nodes, ‘Playground’ and ‘John’, connected by a named edge ‘Contains’. In this sense, the propositional knowledge in many of the bAbI tasks is equivalent to a set of constraints on an underlying graph structure.

b London Underground



Underground input:

(OxfordCircus, TottenhamCtRd, Central)

(TottenhamCtRd, OxfordCircus, Central)

(BakerSt, Marylebone, Circle)

(BakerSt, Marylebone, Bakerloo)

(BakerSt, OxfordCircus, Bakerloo)

⋮

(LeicesterSq, CharingCross, Northern)

(TottenhamCtRd, LeicesterSq, Northern)

(OxfordCircus, PiccadillyCircus, Bakerloo)

(OxfordCircus, NottingHillGate, Central)

(OxfordCircus, Euston, Victoria)

84 edges in total

b London Underground



Traversal

Traversal question:

(BondSt, _, Central),

(_, _, Circle), (_, _, Circle),

(_, _, Circle), (_, _, Circle),

(_, _, Jubilee), (_, _, Jubilee),

Answer:

(BondSt, NottingHillGate, Central)

(NottingHillGate, GloucesterRd, Circle)

⋮

(Westminster, GreenPark, Jubilee)

(GreenPark, BondSt, Jubilee)

Shortest-path

Shortest-path question:

(Moorgate, PiccadillyCircus, _)

Answer:

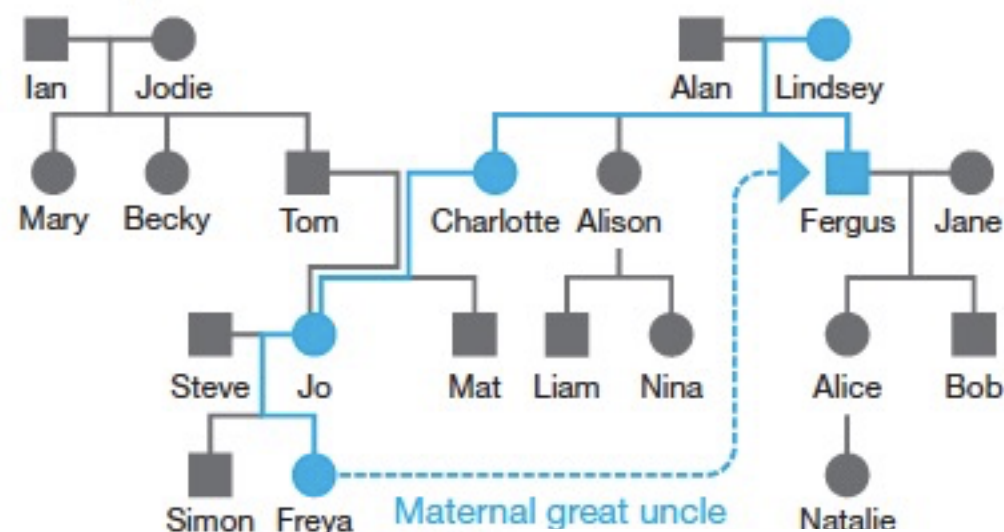
(Moorgate, Bank, Northern)

(Bank, Holborn, Central)

(Holborn, LeicesterSq, Piccadilly)

(LeicesterSq, PiccadillyCircus, Piccadilly)

c Family tree



Family tree input:

(Charlotte, Alan, Father)

(Simon, Steve, Father)

(Steve, Simon, Son1)

(Nina, Alison, Mother)

(Lindsey, Fergus, Son1)

⋮

(Bob, Jane, Mother)

(Natalie, Alice, Mother)

(Mary, Ian, Father)

(Jane, Alice, Daughter1)

(Mat, Charlotte, Mother)

54 edges in total

Inference question:

(Freya, _, MaternalGreatUncle)

Answer:

(Freya, Fergus, MaternalGreatUncle)

As a benchmark we again compared DNCs with LSTM. In this case, the best LSTM network we found in an extensive hyper-parameter search failed to complete the first level of its training curriculum of even the easiest task (traversal), reaching an average of only 37% accuracy after almost two million training examples; DNCs reached an average of 98.8% accuracy on the final lesson of the same curriculum after around one million training examples.

Block puzzle experiments

Our environment, which we term Mini-SHRDLU, contains a set of numbered blocks on a grid board. An agent, given a view of the board as input, can move the top block from a column and deposit it on top of a stack in another column. At every episode, we generated a start board configuration and several possible goals. Each goal, identified by a single-letter label, was composed of several individual constraints on adjacent block pairs that were transmitted one constraint per time-step

(goal 'T' is "block 6 below 2; block 4 left of 1; ...")
After all of the goals were presented, a single goal label was chosen at random, and the agent was cued to satisfy that goal.

