End-to-End 学習可能な記号処理層の検討と数量推論への応用における課題の分析

吉川 将司 [1, 2]　Benjamin Heinzerling[2]　乾 健太郎 [1, 2]
東北大学 [1]　理化学研究所 [2]
{yoshikawa,inui}@tohoku.ac.jp　benjamin.heinzerling@riken.jp

概要

記号処理の仕組みを DNN の微分可能な層として組み込み，end-to-end 学習可能にする方法を検討し，その応用として四則演算プログラム (電卓) 付き文章読解モデルを構築する．さらに，一般的な設定において提案モデルが電卓を活用するように学習しない問題に対し，人工データにより原因を調査をする．

1　はじめに

本稿では，離散記号処理関数 F を中間層として持つような DNN モデルを構築し，全体を誤差逆伝搬法によって end-to-end に学習する方法を検討する．また，その応用例として二項四則演算プログラム (電卓) 付き文章読解モデル (図 1) を構築し，数量推論を要する文章読解 (e.g., DROP [1]) に取り組む．[1)]

F は一般的なプログラミング言語における適当な数の引数を取る関数 (サブルーチン) である．課題として，F の微分則は不明な上，一般に DNN の中間の活性化を離散値に落とせば勾配が消失し，F より上流の層 (図 1 の N_1) を最終タスクの損失 $\ell(y)$ によって学習させることができない．そこで，本稿では 離散分布のサンプリングに対する微分可能な緩和を行う Gumbel Softmax trick (§ 2) を応用することで，DNN に組み込み end-to-end 学習可能にする F の代替を構築する方法を提示する．

電卓付き文章読解モデル (図 1) は，文章読解において標準的なスパン抽出 [3, 4] によるモデル (推論層) への入力を電卓の計算結果で拡張するものである．項抽出層が電卓への入力を予測し，電卓で四則演算を行い，推論層はその結果を活用して解答を行う．全体を微分可能にすることで，項抽出層を最終タスクの損失を最小化するように学習できる．数量推論は (文章理解の上に，) 広大な数の空間の構造

1)　本稿は，同著者による研究 [2] と同種の問題を扱うが，新たなアプローチとさらなる問題を調査したものである．

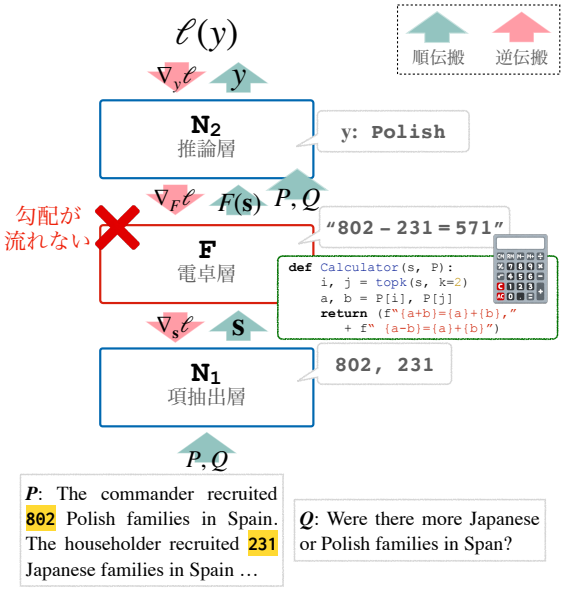

図 1　電卓付き文章読解モデル. 詳細は本文参照.

表 1　事前実験例．DROP dev データ中の比較を問う問題 (Orig) に対し，人手で数量を入れ替え問題を作り変えた (Other; 300 件)．提案モデルは電卓以外の根拠に依り頻繁に誤答してしまった．下の文例は説明のため簡潔であるが，実際は Wikipedia 文章である．

Orig (元の DROP の事例; F 値: **79.4**)
X has 2 pens. Y has 3 pens. Who has more pens? ⇒ Y
Other (数量と答えを入れ替え; F1: **20.2**)
X has 3 pens. Y has 2 pens. Who has more pens? ⇒ X

とその上の演算をモデルに理解させる必要があるが，その仕組みを外部知識層として括りだすことでデータ効率や推論の頑健性において強力なモデルが構築可能と考える．提案法は潜在変数モデルと関連し (§ 3)，理屈上，モデル全体としては電卓を活用しながら解答するように学習されると期待できる．

しかし，実際に構築したモデルは，期待される形で電卓を使わず，別の根拠に依る解法を選好するようになってしまった．この点が本稿第 2 の課題である．表 1 は予備実験の結果であるが，DROP [1] の

一部[2] でモデルを訓練した後，評価セット中の比較問題 (表中 `Orig`) に対し手を加え作成した別の事例 (`Other`) で性能を評価した．[3] 比較問題は，2 つの個体を数的な性質で比較する問題であるが，電卓で数の差を計算し，その符号に従って正しい個体を選択する，というのが最適な戦法に思われる．しかし，表が示すように，モデルは電卓を活用しないだけでなく，問題中の数を入れ替えてもその結果を反映せず，非本質的な根拠に依存しているようである．外部知識層を組み込む場合，モデル全体を end-to-end 学習可能にするだけではなく，モデルにその知識の有用性を認識させることが重要な課題である．[4]

本稿では，上述の問題に対し人工的な問題を用い調査を行う．人工データは，問題の形式に加え，DNN が敏感な分布的な偏りも制御できる．調査の結果，DROP で典型的な比較問題は電卓がなくとも DNN にとって簡単であり，問題を少し複雑にすれば提案モデルでも電卓を活用することがわかった．

2 DNN における離散記号変数

提案法は**離散潜在変数モデル**に関連するため，ここで簡潔にまとめる．[5] このモデルは x から y への離散潜在変数 $z \in \mathcal{Z}$ を介した生成過程を表し，パラメータ θ, ψ の確率分布で以下のように表される．

$$p_{\theta,\psi}(y|x) = \mathbb{E}_{p_\theta(z|x)}[p_\psi(y|z,x)]. \tag{1}$$

DNN ベースの自然言語処理への応用では検索による言語モデル [6, 7] が有名である．このモデルでは，入力テキスト x に対し，Wikipedia 記事 z を検索し (モデル $p_\theta(z|x)$)，その記事を活用しながら QA 等の目的タスクを解く (resp. $p_\psi(y|x,z)$).

多くの場合，潜在空間 $\mathcal{Z}$ は巨大であり，訓練では上の期待値は少数のサンプル $\{z_i\}_{i=1}^N$ ($N \ll |\mathcal{Z}|$) で近似される．しかし、これは θ による勾配の偏推定，ひいて学習の不安定性を招く可能性がある [8].

Gumbel-Softmax (GS) trick　潜在変数モデルの安定した勾配の近似計算のために GS trick [9] が提案されている．潜在空間 $\mathcal{Z}$ の要素を d ($= |\mathcal{Z}|$) 次元 one-hot ベクトルで表現する ($\mathcal{Z} = \{\mathbf{1}_i\}_{i=1}^d$)．さらに，$p_\theta(z|x)$ はスコア関数 $\boldsymbol{f}^\theta(x) \in \mathbb{R}^d$ と Softmax 関数で表現されると仮定する．

$$p_\theta(z = \mathbf{1}_i|x) = \mathtt{softmax}_{\tau,i}(\boldsymbol{f}^\theta(x)) = \frac{\exp(f_i^\theta(x)/\tau)}{\sum_j \exp(f_j^\theta(x)/\tau)}.$$

ここで $\tau > 0$ は温度パラメータである．*Gumbel trick* [10, 11] によれば式 (1) の期待値に対して，Gumbel 分布 p_ε を用いて以下が成り立つ．[6]

$$(1) = \mathbb{E}_{\boldsymbol{\varepsilon}\sim p_\varepsilon}[p_\psi(y|x, \mathtt{argmax}(\boldsymbol{f}^\theta(x) + \boldsymbol{\varepsilon}))].$$

$\mathtt{argmax}$ の微分はほとんど至るところ 0 であるため、GS trick では $\lim_{\tau\to 0} \mathtt{softmax}_\tau(\boldsymbol{s}) = \mathtt{argmax}(\boldsymbol{s})$ に着目し $\mathtt{softmax}_\tau$ で近似することを考える．さらに，上の期待値を 1 つの Gumbel ノイズ $\boldsymbol{\varepsilon}$ のみを用いて近似しても安定して学習できることが経験的に知られ，以下に対する誤差逆伝播法によって式 (1) のパラメータを end-to-end に最適化することができる．[7]

$$p_\psi(y|x, \mathtt{softmax}_\tau(\boldsymbol{f}^\theta(x) + \boldsymbol{\varepsilon})) \text{ where } \boldsymbol{\varepsilon} \sim p_\varepsilon.$$

また [9] では，順伝搬時には $\mathtt{argmax}$ を用い，逆伝搬時にその Jacobi 行列 $\frac{\partial \mathtt{argmax}}{\partial \boldsymbol{s}}(\boldsymbol{s})$ $(= 0)$ を $\mathtt{softmax}$ のもので代用する Straight-Through Gumbel-Softmax という手法も提案されている．本稿ではこの亜種の微分可能 $\mathtt{argmax}$ を $\mathtt{argmax}_\partial$ と書くことにする．

3 微分可能な記号処理層

本研究の鍵は，前節の技術を応用すれば多くの記号処理関数 F を簡単に深層モデルの微分可能な 1 層 F に変換できるということである．この方法は，驚くほどに単純である上，F の上流ネットワーク $\mathtt{N}_1$ が F への入力を学習し，下層ネットワーク $\mathtt{N}_2$ に対し F の出力が役立つようなものとなることを保証する．

命題 1　K 個の項空間 $\mathcal{A}_i$, K 引数記号処理関数 F を考える．F は $(\mathtt{a}_1, \ldots, \mathtt{a}_K) \in \mathcal{A} = \mathcal{A}_1 \times \cdots \times \mathcal{A}_K$ を記号処理の結果を表す可変長のトークン列 $\mathtt{r} = (\mathtt{r}_1, \ldots, \mathtt{r}_*) \in \mathcal{V}^*$ に変換する．このとき，対応する微分可能な F は，$\boldsymbol{s} \in \mathbb{R}^{|\mathcal{A}|}$ をすべての $\mathtt{a} \in \mathcal{A}$ に対するスコア $s_\mathtt{a}$ のベクトルとし、$R = [\mathbf{1}_{r_1}, \mathbf{1}_{r_2}, \ldots, \mathbf{1}_{r_*}]^\top \in \mathbb{R}^{*\times|\mathcal{V}|}$ を $\hat{\mathtt{a}} = \arg\max_{\mathtt{a}\in\mathcal{A}} s_\mathtt{a}$ に対する計算結果とした $F : \boldsymbol{s} \mapsto R$ として構築できる．

F から F の構築は簡単である．i 番目の項の組み合わせを $(\mathtt{a}_{i,1}, \ldots, \mathtt{a}_{i,K})$ とし，$F(\mathtt{a}_{i,1}, \ldots, \mathtt{a}_{i,K}) =$

2)　解答に 2 数の減算を要する比較，差，補部問題 1.4 万件 [5].
3)　提案モデルは数量を答える差問題 (*How many more pens does X have than Y?*) に対しては安定して高い性能を示す．
4)　関連して，検索による言語モデル [6] において，固有表現を重視した事前学習が，他タスクでの性能に大きく貢献するという非自明な結果も注目に値する．
5)　**記法**: $\mathbf{1}_k$ ($k \in \mathbb{N}$) で k 番目の要素が 1 の適当な次元の one-hot ベクトルを表す．$\boldsymbol{x} \in \mathbb{R}^n$ に対し，$\mathtt{argmax}(\boldsymbol{x})$ は最大の要素 x_k に対応する $\mathbf{1}_k$ を返す．$\mathtt{argmax}$ に関する議論では簡単のため x_k の大きさで同率一位になることはないと仮定する．
6)　$\boldsymbol{\varepsilon} \in \mathbb{R}^d$. u_i を $[0,1]$ 上の一様ノイズとし，$\varepsilon_i = -\log(-\log u_i)$.
7)　一般の離散分布 $\boldsymbol{\pi}$ に対する GS trick は，$\mathtt{softmax}(\log\boldsymbol{\pi} + \boldsymbol{\varepsilon})$ であるが，$\boldsymbol{\pi} = \mathtt{softmax}(\boldsymbol{f}^\theta)$ の場合 $\mathtt{softmax}(\boldsymbol{f}^\theta + \boldsymbol{\varepsilon})$ と等しい．

$(\mathsf{r}_{i,1}, \ldots, \mathsf{r}_{i,n_i})$，さらに計算結果の one-hot 表現を $E_i = [\mathbf{1}_{\mathsf{r}_{i,1}}, \ldots, \mathbf{1}_{\mathsf{r}_{i,n_i}}]^\top$ とすると，$\boldsymbol{s}$ に対して微分可能 argmax を適用して得た重み $\boldsymbol{z}$ による E_i の重み付き和として F は表現される:

$$\boldsymbol{z} = \mathsf{argmax}_\partial(\boldsymbol{s}), \quad F(\boldsymbol{s}) = \sum_{i=1}^{|\mathcal{A}|} z_i E_i.$$

ここで、可変行数の行列 E_i の和は適当に後部に 0 埋めを行うことで可能とする．

このようにして得た記号処理層 F は，微分可能 argmax，ひいては潜在変数モデルと関連し，理屈的には F を中間層とするモデルは F を活用するように学習する．この点については付録 A にまとめる．

4 電卓付き文章読解モデル

前節でのアイデアに基づき電卓付き文章読解モデル (図 1) を構築する．P, Q を文章と質問のトークン列とし，まとめて $x = (P, Q)$ とする．変数 y は Q に対する答えである．また $\mathfrak{X}(x)$ は電卓への入力候補の集合であり，x に出現するすべての数量のペアを要素とする．$\mathfrak{X}(x)$ の要素は $d = |\mathfrak{X}(x)|$ 次元の one-hot ベクトルで表すこともある．

項抽出層 まず x 中から電卓への入力 $\boldsymbol{z} \in \mathfrak{X}(x)$ を予測する．確率分布としては以下のように書ける．

$$p_\theta(\boldsymbol{z} = \mathbf{1}_i | x) \propto \exp f_i^\theta(x). \tag{2}$$

ここで $\boldsymbol{f}^\theta(x) \in \mathbb{R}^d$ はすべての $(a, b) \in \mathfrak{X}(x)$ に対するスコアであり，BERT [12] で x を走査して得た表現ベクトルで，数量 a, b を表すトークン列で先頭のもの (e.g., "1 ##2 ##3") を連結し 2 層 ReLU ネットワークで変換して得られる．

電卓層 式 (2) に従って抽出された数量に対し，電卓 F は二項四則演算の計算を行い結果を文字列として返す．本研究では演算として足し算と引き算を用い，すべての演算結果を列挙して返す．例えば $F(72, 23)$ = "95 = 72 + 23, 59 = 72 − 23, −59 = 23 − 72" などとする．この電卓を § 3 の手法に基づき記号処理層 F に変換しモデルに含める．

推論層 推論層 $p_\psi(y|x, F(\boldsymbol{s}))$ は文章読解において標準的なスパン抽出 [3, 4] によるものであるが，質問 Q，文章 P に加え電卓の計算結果文字列 $F(\boldsymbol{s})$ からも回答を抽出することができる点が異なる．

$$p_\psi(y|x, F(\boldsymbol{s})) = \sum_{(s,t) \in \mathcal{Y}(x,y)} q_\psi((s,t)|x, F(\boldsymbol{s})).$$

ここで，$\mathcal{Y}(x, y)$ は添字の組 (s, t) であり，トークン列 "[CLS] Q [SEP] P [SEP] $F(\boldsymbol{s})$ [SEP]" 上で s から t 番目のトークン列が y に一致するようなものの集合である．q_ψ は BERT で同じ文字列を走査して得たベクトル列の関数である．

表 2 実験データの例．斜体で示した部分を譲渡文と呼ぶ．どちらの問も根拠として，**9,816,091** (- ***6,224,450***) > **3,002,489** である (括弧内は譲渡文ありの場合).

P: John has 5,903,204 grapes, Mark has 8,907,756 grapes, Ryan has 7,646,494 grapes, **Anna** has **9,816,091** grapes, Kenneth has 8,091,683 grapes, Johnny has 1,078,950 grapes, **Carl** has **3,002,489** grapes, Alice has 5,953,680 grapes, Grace has 6,279,638 grapes, Kyle has 6,940,318 grapes, ***Anna gave 6,224,450 grapes to Grace.***
max Q: What is the maximum number of grapes owned either by Anna or Carl? **A**: 3,591,641
argmax Q: Who has more grapes: Anna or Carl? **A**: **Anna**

訓練 上述の 3 層を合成すると、提案モデルは $p_\psi(y|x, F(\boldsymbol{f}^\theta(x)))$ である．GS trick (§ 2) との関連から，このモデルの訓練には，電卓の使い方を探索するため Gumbel ノイズ $\boldsymbol{\varepsilon} \sim p_\varepsilon$ を加える．従って，学習では訓練データ上で以下の損失を最小化する．

$$\ell_{\mathsf{end}} = -\log p_\psi(y|x, F(\boldsymbol{f}^\theta(x) + \boldsymbol{\varepsilon})).$$

注意として，順伝搬時に動的に得られた電卓の計算結果トークン列 $F(\hat{\boldsymbol{z}})$ (with $\hat{\boldsymbol{z}} = \mathsf{argmax}(\boldsymbol{f}^\theta(x) + \boldsymbol{\varepsilon})$) 中にラベル y が含まれる場合，モデルはそのスパンも抽出するように更新される．

層ごとの教師信号 提案モデルの強みの 1 つに，層ごとの訓練が可能なことがある．具体的に，本研究では項抽出層に対して専用の教師信号 ℓ_{arg} を加えることを考える．訓練事例 (x, y) に対し y が数量であれば x 中の 2 数 (a, b) で $F(a, b)$ が y を含むようなものを全探索し抽出層の訓練に利用する．このようにして得たラベルと式 (2) の間の binary cross entropy を ℓ_{arg} とし，$\ell_{\mathsf{end}} + \ell_{\mathrm{arg}}$ で全体を訓練する．以下の実験では，人工的な設定ゆえに真の項抽出ラベルが手に入るが，現実的な問題設定を考慮して上の手続きで得られるもののみをラベルとして利用する．

5 実験

実験データ 実験には表 2 のような人工文章読解問題を用いる．人工データは，問題の構造そのものに加え，モデルが敏感なデータの偏りを制御できる．基本的に，ランダムに人が所有する物体 (grapes,

表 3　主な実験結果 (F1/項正答率)．各設定ごとに訓練・評価に用いるデータは異なる．列は含まれる問題のタイプ，行は各事例に譲渡文が含まれるかを表す．

F1/項正答率		問題タイプ		
		argmax	max	max&argmax
譲渡文	なし	98.1/0.6	99.5/1.8	98.8/9.1
	あり	92.2/2.3	99.0/100.0	98.9/100.0

表 4　汎化実験．6 桁以下の数字のみを含む 2 万件で訓練後，7 桁の argmax 問題のみの評価セットで F1/項正答率を評価．訓練/評価セットいずれも譲渡文を含む．

		BERT+スパン抽出	提案モデル
訓練データ	argmax	72.0/-	74.9/2.0
	max+argmax	-	78.5/99.8

etc.) を列挙し，それらの間の max/argmax 関係を問う．桁数に基づいて大小関係が推論できてしまうことを防ぐため，明示的に言及しない限り出現する数をすべて 7 桁で固定する．特殊な場合として，**譲渡文** (例最後の斜体文) を含める．このとき，質問は必ず譲渡イベントに参与した人について問うことにする．譲渡文なしでは，数量推論として**所有物の数量の比較**が必要であるが，譲渡文ありの場合は譲渡による**所有物の数量の変化**を含む 2 段階の計算をする必要がある．訓練/評価には 2 万/2 千件を用いる．問題の構造上，項抽出損失 ℓ_{arg} は譲渡文ありの max 問題の一部のみ[8] に与えられ，argmax 問題ではタスク損失 ℓ_{end} から抽出方法を学習する必要がある．

項抽出層，推論層では `bert-base-uncased` を用い，パラメータを共有させる．既存研究 [4] に従い，入力中の数量は桁ごとに分割する (例: `123` ↦ `1 ##2 ##3`)．評価指標として，DROP と同様の F1 スコアと項抽出の正答率 (2 項完答で 1 点) を報告する．項抽出は，譲渡文なしの場合は所有物の数量の比較に関するもの，譲渡文ありの場合は譲渡による数量の変化に関係する 2 項を正解とみなす．

5.1　実験結果

表 3 に問題のタイプ，譲渡文の有無を変えて訓練と評価を行った結果を示す．譲渡文なしの argmax 問題は，構造的に DROP の比較問題に対応するが，§ 1 での予備実験と同様に提案モデルは高い F1 を示す一方で，電卓は活用できていないことが観察できる．図 2 は，同じモデルで argmax 問題の事例を解かせたときに，入力と電卓計算結果のどのトークンに注目しているかを勾配ベースの手法 [13] で可視化

8)　表 2 の例において譲渡文が *Douglas gave 7,000,000 grapes to Grace.* であれば max Q の答えが変わるため存在しない．

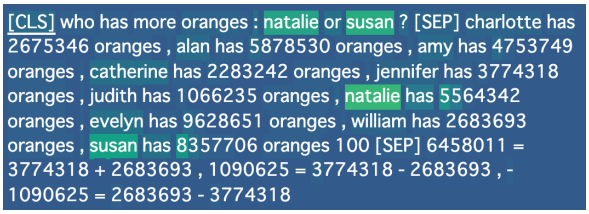

図 2　譲渡文なし，argmax 問題で訓練したモデルによる評価事例中のトークン重要度．$\hat{s}$ と $\hat{y}$ を項抽出層，推論層の予測とし，各トークン x_i の推論層入力の埋め込み表現 $\boldsymbol{e}_{x_i}$ について $\|\nabla_{\boldsymbol{e}_{x_i}}(-\log p_\psi(\hat{y}|x, F(\hat{s})))\|$ の大きさを示す．

したものであるが，モデルが敏感に反応しうるデータの偏りを極力廃した人工データでは，明らかに比較すべき人物と関連する数量 (の最初の数桁) に注目しており，この種の問題では電卓を利用せずともBERT 内部で解法を見つけることが可能であると示唆される．同様の観察から譲渡文なしの max 問題も電卓を使わずに解くことができるようである．

譲渡文ありの実験においては，max の設定で項正答率が 100%に達した．この事の要因の 1 つに，この設定では項抽出損失 ℓ_{arg} の効果があると推察される．また argmax のみ，max と argmax 両方を含む場合の 2 つを比べると後者の項正答率の高さから，max で得た電卓の使い方を argmax 問題に般化して活用するように学習できていることが観察される．これらから，現状の提案モデルが電卓を使うためには，(1) 2 段の推論を要する等**数量推論問題が簡単すぎないこと**，(2) max 問題等で**項抽出の仕方を少し教えること**の 2 点が有効であることが推測される．

最後に，提案モデルの未知の桁の数に対する頑健性を評価した (表 4)．表 2 と同じ譲渡文ありの事例で 6 桁以下の数のみ含む 2 万件[9] で訓練した後，7 桁の数のみ出現する 2 千件で評価したところ，電卓を活用できるモデルが最も高い頑健性 (F1) と動作の解釈性 (項正答率) を有していることがわかった．

6　終わりに

本稿では記号処理関数を DNN の微分可能な層として組み込む方法を検討し，その応用として電卓付き文章読解モデルを構築した．前者の鍵は Gumbel-Softmax trick を拡張することである．応用では，人工データによる実験でモデルの挙動の癖が明らかになったが，一般的な設定で電卓の有用性をモデルに認識させることは今後の課題である．

9)　[14] に倣い，事例に出現する数量 v を $d \sim U\{1, ..., 6\}, v \sim U[10^{d-1}, 10^d - 1]$ の 2 段階の一様サンプリングで得ることで事例に含まれる数が大きな値に偏ることを防いだ．

謝辞 本研究は JST CREST JPMJCR20D2 及び JSPS 科研費 20K23314, 21K17814 の助成を受けたものです.

参考文献

[1] Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In **Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)**, pp. 2368–2378, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[2] 吉川将司, 乾健太郎. 離散記号処理に対する近似的な微分構造の考察と数量推論を要する文章読解問題への応用. 言語処理学会年次大会発表論文集, 3 月 2021.

[3] Jambay Kinley and Raymond Lin. NABERT+: Improving numerical reasoning in reading comprehension. https://github.com/raylin1000/drop-bert, 2019.

[4] Mor Geva, Ankit Gupta, and Jonathan Berant. Injecting numerical reasoning skills into language models. In **Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics**, pp. 946–958, Online, July 2020. Association for Computational Linguistics.

[5] Tushar Khot, Daniel Khashabi, Kyle Richardson, Peter Clark, and Ashish Sabharwal. Text modular networks: Learning to decompose tasks in the language of existing models, June 2021.

[6] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. Retrieval augmented language model pre-training. In Hal Daumé III and Aarti Singh, editors, **Proceedings of the 37th International Conference on Machine Learning**, Vol. 119 of **Proceedings of Machine Learning Research**, pp. 3929–3938, Virtual, 13–18 Jul 2020. PMLR.

[7] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, **Advances in Neural Information Processing Systems**, Vol. 33, pp. 9459–9474. Curran Associates, Inc., 2020.

[8] Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. **Machine Learning**, Vol. 8, No. 3, pp. 229–256, 1992.

[9] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparametrization with gumbel-softmax. In **Proceedings International Conference on Learning Representations 2017**. OpenReviews.net, April 2017.

[10] Emil Julius Gumbel. **Statistical Theory of Extreme Values and Some Practical Applications. A Series of Lectures.** U.S. Government Printing Office, 1954.

[11] Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. In **International Conference on Learning Representations**, 2017.

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In **Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)**, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[13] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps, 2014.

[14] Rodrigo Nogueira, Zhiying Jiang, and Jimmy Lin. Investigating the limitations of transformers with simple arithmetic tasks, 2021.

A F の上流ネットワーク $\mathtt{N}_1$ の学習

本稿の目的は記号処理関数を DNN の 1 層とし，ネットワークとの合成関数 $x \xrightarrow{\mathtt{N}_1} \boldsymbol{s} \xrightarrow{F} F(\boldsymbol{s}) \xrightarrow{\mathtt{N}_2} y$ を end-to-end に学習することであった (図 1)．これに関する自然な疑問として，この予測 y に対する損失によりモデル全体を訓練したときに，$\mathtt{N}_1$ は F を介して $\mathtt{N}_2$ に有用な記号処理計算結果を渡すようになるだろうか，ということがある．これについては，GS trick との関係で肯定的に答えることができる．

命題 2　入力 x，予測 y，$\mathtt{N}_1(x) = \boldsymbol{s}$，$\boldsymbol{s}$ は F $(= F)$ に入力されるとする．さらに $\mathtt{N}_1$ は $p_\theta(z = \mathbf{1}_i|x) \propto f_i^\theta(x)$，$\mathtt{N}_2$ は $p_\psi(y|x, F(z))$ をモデル化する．このとき，Gumbel ノイズで摂動されたこれらの合成 $x \xrightarrow{\mathtt{N}_1} \boldsymbol{s} \xrightarrow{F} F(\boldsymbol{s}) \xrightarrow{\mathtt{N}_2} y$ は，

$$p_\psi(y|x, F(\boldsymbol{f}^\theta(x) + \boldsymbol{\varepsilon})) \text{ with } \boldsymbol{\varepsilon} \sim p_{\boldsymbol{\varepsilon}}, \tag{3}$$

であり，これは離散潜在変数モデル $p_{\theta,\psi}(y|x)$ を近似的にモデル化する．

計算結果 F' の条件付き分布 $p(F'|z)$ を仮定すれば $p_{\theta,\psi}(y|x)$ は以下のように展開できる．

$$p_{\theta,\psi}(y|x) = \sum_z \sum_{F'} p_\psi(y|x, F') p(F'|z) p_\theta(z|x). \tag{4}$$

F' は z から決定的に決まることに加えて，Gumbel trick を用いると，

$$\begin{aligned}(4) &= \sum_z p_\psi(y|x, F(z)) p_\theta(z|x) \\ &= \mathbb{E}_{p_\theta(z|x)}[p_\psi(y|x, F(z))] \\ &= \mathbb{E}_{\boldsymbol{\varepsilon} \sim p_\varepsilon}[p_\psi(y|x, F(\boldsymbol{f}^\theta(x) + \boldsymbol{\varepsilon}))].\end{aligned}$$

この式と式 (3) は GS trick のときと同様に近似的の関係にある．よって，式 (3) に関してパラメータを最適化すれば，$p_{\theta,\psi}(y|x)$ に関する尤度が最大化され，これは一方で $\mathtt{N}_1$ が我々の期待するような形で訓練されることを意味する．

本稿の文章読解モデルは，式 (4) に基づいて訓練することも可能であるが，今後の拡張として記号処理層を多段に適用することを念頭に置き，一度の順伝搬と逆伝搬で訓練可能な Gumbel Softmax 式のアプローチを検討する．