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

吉川 将司 [1,2]
東北大学 [1]
yoshikawa@tohoku.ac.jp

乾 健太郎 [1,2]
理化学研究所 [2]
inui@ecei.tohoku.ac.jp

1 はじめに

本稿では，**離散記号処理モジュール** F を中間層として持つような深層学習モデルを構築し，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**電卓モジュール付き文章読解モデル**（図 1）を実装し，実験を行う．

F は一般に Python インタープリタのような**プログラム**を考える．しかし，当然ながら F の微分則は不明であるし，また一般的に DNN の中間の活性化を離散値に落とせば勾配が 0 になってしまい，F より上流のモジュールを，最終タスクの損失に関する勾配によって学習させることができない．そこで，本稿では argmax 活性化に対する微分可能な緩和を行う Gumbel Softmax trick (§ 3.1) を一般化し，F に対する**擬似的な Jacobi 行列**を構築する方法を提案する．

記号推論を要する問題として，**数量推論を要する文章読解** (DROP [1]) に取り組み，深層学習モデルに，F として簡単な二項四則演算プログラム（**電卓**と呼ぶ）を取り込むことを考える．数量推論は深層学習による自然言語処理では難題であり，近年の巨大な言語モデルでも限られたパターンしか対応できないことが報告されている [2]．本研究は，現状の言語モデルの延長ではこの問題は解決できないと考え，外部モジュールを組み込む方策を検討する．

2 問題設定

具体的なモデルの構造について本旨に関係する範囲で整理し，詳細は付録 A にまとめる.[1] 文章と質問を P, Q とし，対応する答え（正解ラベル）を $y^\star$,

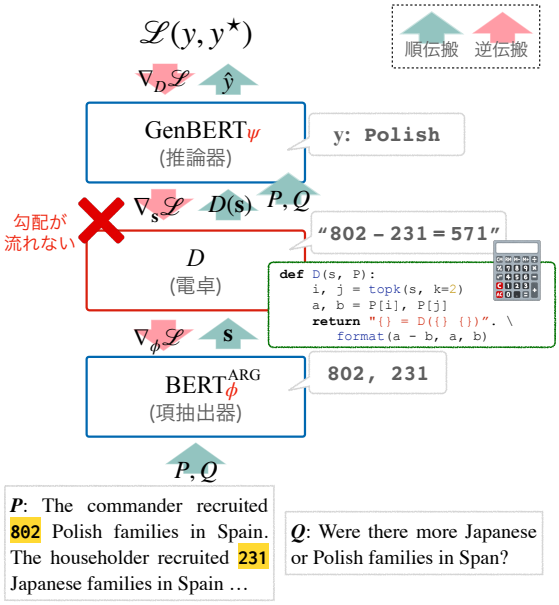

図 1 電卓付き文章読解モデル．詳細は本文参照．

モデルの予測を y，損失を $\mathscr{L}(y, y^\star)$ とする．これら変数の具体的な構造には立ち入らず，$\mathscr{L}$ は y で微分できると仮定する（e.g., 確率分布間の KL 誤差）．

モデル ベースの文章読解モデルとして GenBERT [3] を用い，それに対して電卓機能を追加する．パラメータ ψ の関数として $\mathrm{GenBERT}_\psi$ は BERT [4] に基づき，“[CLS] Q [SEP] P” のサブワード列を入力とし，入力テキスト中のスパンを抽出するか，文字列を直接生成することによって問題を解く．電卓による拡張では，別の BERT ベースのモデル $\mathrm{BERT}^{\mathrm{ARG}}_\phi$（項抽出器）に P 中から問題を解くために有用そうな数量項を 2 つ抽出させ（図 1 の “802”, “231”），電卓 D により計算を行った結果の文字列 Z （ここでは引き算として “571 = D(802, 231)”）を連結した “[CLS] Q [SEP] P [SEP] Z” を使って $\mathrm{GenBERT}_\psi$ に推論させる．$\mathrm{BERT}^{\mathrm{ARG}}_\phi$ の出力は $\boldsymbol{s} \in \mathbb{R}^K$ （K は P 中の数量項の組の数）であり，要素 s_i は数量項の組に対するスコアである．電卓は s_i の大きさに従って数量項の組をサンプリングし Z を構築する．ここで，Z は BERT

1) **記法**：$\boldsymbol{e}_k$ ($k \in \mathbb{N}$) で k 番目が 1 の適当な次元の one-hot ベクトルとする．語彙 $\mathcal{V}$ は全順序集合とし $\mathtt{w} \in \mathcal{V}$ に対し $\boldsymbol{e}_\mathtt{w}$ とも書く．また $\mathbf{1}$ はすべての要素が 1 のベクトルである．$n-1$ 次元単体は $\triangle^n = \{\boldsymbol{\mu} \in \mathbb{R}^n | \forall \mu_i \geq 0, \sum_{i=1}^n \mu_i = 1\}$ とし，タイプライター書体の $\mathtt{argmax}(\boldsymbol{x})$ は $\boldsymbol{x} \in \mathbb{R}^n$ 中の最大要素 x_i について $\boldsymbol{e}_i$ を返す関数とする．これは一般的な演算子を使って $\mathtt{argmax}(\boldsymbol{x}) = \arg\max_{\boldsymbol{\mu} \in \triangle^n} \boldsymbol{x}^\top \boldsymbol{\mu}$ と書ける．argmax に関する議論では x_i の大きさが同率一位になることはないと仮定する．

ベースの $\mathsf{GenBERT}_\psi$ に直接入力されるため文字列である必要はなく，BERT の入力層のサブワード語彙 $\mathcal{V}$ に対応した one-hot ベクトルを並べた行列

$$[\underbrace{\boldsymbol{e}_{5}\ \boldsymbol{e}_{\#\#7}\ \boldsymbol{e}_{\#\#1}}_{F(\boldsymbol{s});\text{計算結果 }(L\text{ トークン})}\ \overbrace{\boldsymbol{e}_{=}\ \boldsymbol{e}_{\mathsf{D}}\ \boldsymbol{e}_{(}}^{\text{常に含まれる数学記号}}\ \underbrace{\boldsymbol{e}_{8}\cdots\boldsymbol{e}_{\#\#1}}_{A(\boldsymbol{s});\text{抽出結果}}\ \boldsymbol{e}_{)}]^\top = Z = D(\boldsymbol{s}) \quad (1)$$

であればよく，以降 Z をそのような行列とする．注意として，演算記号 “-” を使わず抽象的に “D” とまとる．これにより $[F_+(\boldsymbol{s})\ F_-(\boldsymbol{s})\ F_\times(\boldsymbol{s})\ \boldsymbol{e}_=, \ldots]$ のように式 1 の “=” の左に F_* を並べることで簡単に演算を増やすことができる．実際に用いる演算は付録 A.2 にまとめる．予測は以下のように整理される．

$$y = \mathsf{GenBERT}_\psi(P, Q, D(\mathsf{BERT}^{\mathsf{ARG}}_\phi(P, Q) + \boldsymbol{g}))$$

ここで $\boldsymbol{g}$ は Gumbel ノイズである (§ 3.1)．BERT 等の言語モデルに Z のようなヒントを与えた場合，推論性能が大きく向上することが報告されており [5, 6]，本モデルが期待通りに動作すれば，文章読解問題を高い精度で解くことが期待できる．

問題　項抽出器 $\mathsf{BERT}^{\mathsf{ARG}}_\phi$ に問題を解くために有効な数量項を抽出させるために，そのパラメータ ϕ を文章読解の損失 $\mathcal{L}(y, y^\star)$ を最小にするように推定したい．勾配ベースの学習法を用いるため，$\mathcal{L}$ の ϕ による勾配 $\nabla_\phi \mathcal{L}(y, y^\star)$ を計算したいが，それは

$$\nabla_\phi \mathcal{L}(y, y^\star) = \frac{\partial \mathsf{BERT}^{\mathsf{ARG}}_\phi}{\partial \phi} \frac{\partial D}{\partial \boldsymbol{s}} \frac{\partial \mathsf{GenBERT}_\psi}{\partial Z} \nabla_y \mathcal{L}(y, y^\star)$$

であり，式 1 から $\partial F/\partial \boldsymbol{s}$ と $\partial A/\partial \boldsymbol{s}$ を計算する必要がある．後者については Gumbel-Softmax trick の簡単な応用で対処できるため詳細は付録 A.2 に載せるにとどめるが，前者の $\partial F/\partial \boldsymbol{s} \in \mathbb{R}^{K \times L \times |\mathcal{V}|}$ をいかに近似/推定すればよいか，が本研究が扱う問題である．本研究では，Sofmax を含む関数クラスの Jacobi 行列を分析し (§ 4)，その結果に基づいて $\partial F/\partial \boldsymbol{s}$ の近似を人工的に設計する手法を提案する（§ 5）．[2)]

3　関連研究

3.1　Gumbel-Softmax trick

図 1 のような電卓つき文章読解モデルを作る場合，項抽出結果 Z に対する $D(Z)$ が確率 1 で決まることに着目し，潜在変数モデル（P, Q への依存は略）

$$P(y) = \mathbb{E}_{P_\phi(Z)}[P_\psi(y|D(Z))]$$

を検討することは尤もであり，近年検索ベースの言語モデル [6] 等が成功を収めている．しかし，この方法では，モデル内に記号処理モジュールを増やすごとに潜在変数の組み合わせが爆発的に増えるため，確率の計算が困難になってしまう．特に，勾配ベースの手法で学習する際，一般的に限られた数の $Z_i \sim P_\phi(Z)$ により勾配を近似的に推定することになるが，REINFORCE アルゴリズム [7] のような不偏推定量の計算法が存在しながらも，分散が大きく学習が不安定なことが知られている．D が恒等写像で，Z がカテゴリカル分布に従う場合のこの問題に対する安定した代替案である Gumbel-Softmax trick [8] では，サンプリング操作をノイズと argmax 関数評価に切り分け (*reparameterization trick*; $P(y) = \mathbb{E}_{\epsilon \sim P_\epsilon}[P_\psi(y|\mathsf{argmax}(f_\phi(\epsilon)))]$)，温度パラメータ $\tau > 0$ を小さくした $\mathsf{softmax}(\boldsymbol{x}/\tau) = [e^{x_i/\tau}/(\sum_j e^{x_j/\tau})]_i$ で argmax を近似する（**微分可能な緩和**）ことで end-to-end 学習を行う．さらに，順伝搬時は argmax を計算し，逆伝搬時に softmax の Jacobi 行列 $\nabla\mathsf{softmax}$ を使う Straight-Through (ST) Gumbel-Softmax という手法も提案されている．本研究は D として恒等写像から記号処理モジュールに一般化し，その場合における近似 Jacobi 行列を考えるものである．

softmax が argmax の近似であることは，特に前者が正則化 argmax [9, 10] の一種であることからわかる．正則化 argmax はある正則化関数 Ω に関して，

$$\mathsf{argmax}_\Omega(\boldsymbol{x}) = \arg\max_{\boldsymbol{\mu} \in \triangle^n} \boldsymbol{x}^\top \boldsymbol{\mu} - \Omega(\boldsymbol{\mu})$$

の形をした関数のクラスであり，softmax は Ω に負の Shannon エントロピー ($-H_S(\boldsymbol{\mu}) = \sum_i \mu_i \log \mu_i$) を選んだ場合に相当する．温度パラメータ τ を下げることと argmax の関係は以下のように正則化の効果を落とすことに相当する．

$$\begin{aligned}\mathsf{argmax}_\Omega(\boldsymbol{x}/\tau) &= \arg\max_{\boldsymbol{\mu} \in \triangle^n} (\boldsymbol{x}/\tau)^\top \boldsymbol{\mu} - \Omega(\boldsymbol{\mu}) \\ &= \arg\max_{\boldsymbol{\mu} \in \triangle^n} \boldsymbol{x}^\top \boldsymbol{\mu} - \tau\Omega(\boldsymbol{\mu}) \xrightarrow{\tau \to 0} \mathsf{argmax}(\boldsymbol{x})\end{aligned} \quad (2)$$

上の関係は softmax に限らない一般の argmax_Ω に成り立ち，微分可能緩和として sparsemax [9] などの

2)　Z_i を式 1 のように i 番目の数量項の組に対する計算結果から作る行列とすれば，F は $F'(\boldsymbol{s}) = \mathsf{argmax}(\boldsymbol{s})^\top [Z_1, \ldots, Z_k]^\top$ と書け，この argmax で ST Gumbel-Softmax を使うという簡単な方法でも電卓の近似微分が実現できることに本稿執筆時に気づいた．しかし，これをナイーブに実装する場合 Z_i を並べた巨大な行列を作る必要があり，空間計算量的に問題である．提案法は（逆伝搬時はもう少し工夫が必要であるが）それを迂回できる可能性がある．また，F' の定式化の場合，各 Z_i に対する重み付けの比を学習することに相当するが，提案法の Jacobi 行列は（各時刻に $\mathcal{V}$ の要素を予測する）$\boldsymbol{s} \mapsto Z$ の系列生成器の $\boldsymbol{s}$ に関する微分を想起させる形になっている．F と F' の近似 Jacobi 行列の特性や，学習問題にどのような含意を持つのかは興味深い問題であり，今後詳細に調べたい．

他のインスタンスを使う方針もあり得る．§ 4 では，一般性のある正則化 argmax に対し分析を行う．

3.2 記号処理モジュールの微分

本研究と最も関連のある既存研究は Estimate and Replace 法 [11] である．この手法を電卓付き文章読解モデルに応用する場合，まず電卓の記号処理を模倣するネットワークを人工データ上で事前訓練し，文章読解モデルを訓練する場合には，この模倣ネットワークで電卓を置き換えて勾配を推定 (estimate) し，テスト時には真の電卓に戻して (replace) 使う．この方法では原理的に最終タスクによる勾配を上流まで流すことができるが，その性質については不明である．特に，論文 [11] による評価実験が単純な人工タスクに限られていることは，汎用性に関し限界を示唆する．推定される勾配の質は，模倣ネットワークの模倣精度に大きく依存すると予想され，精度を上げるためにネットワークを大きくすれば，モデル全体の学習に要する計算コストが犠牲になる．

本研究の動機は，離散記号モジュールの微分がどうあるべきかを数理的に分析し，上述の問題を克服する勾配推定法を確立することである．その糸口として，以下正則化 argmax の Jacobi 行列を分析する．

4 $\mathtt{argmax}_\Omega$ の Jacobi 行列の分析

一般の正則化 argmax $\boldsymbol{\ell}$ がもつ以下の基本的な性質（証明は [10] 参照）を前提としてその Jacobi 行列 $\partial\boldsymbol{\ell}/\partial\boldsymbol{x}$ の性質を調べる．以下 $\boldsymbol{z}=\boldsymbol{\ell}(\boldsymbol{x})$ とする．

- （**単調性**） $\forall\epsilon>0,\ \ell_i(\boldsymbol{x}+\epsilon\boldsymbol{e}_i)\geq\ell_i(\boldsymbol{x})$
- （**正規性**） $\boldsymbol{\ell}(\boldsymbol{x})\in\triangle^n$ i.e., $\ell_i(\boldsymbol{x})\geq 0,\ \sum_{i=1}^n\ell_i(\boldsymbol{x})=1$
- （**定数に対する不変性**） $\forall c\in\mathbb{R},\ \boldsymbol{\ell}(\boldsymbol{x}+c\mathbf{1})=\boldsymbol{\ell}(\boldsymbol{x})$

性質 1：符号　まず注目する点として，これらの Jacobi 行列は，常に対角成分が非負，非対角成分が非正の正方行列である．単調性から対角成分 $\partial\ell_i/\partial x_i$ は非負であり，正規性から 1 を z_i 間で分配しないといけないため $\partial\ell_i/\partial x_j\ (i\neq j)$ は非正となる．具体的に $\mathtt{softmax}$ の Jacobi 行列はそのようになっている：

$$\frac{\partial\mathtt{softmax}_i}{\partial\boldsymbol{x}_j}=\begin{cases}z_i(1-z_i) & i=j\\ -z_iz_j & i\neq j\end{cases}\tag{3}$$

そもそも，Jacobi 行列の要素 $\partial\ell_i/\partial x_j$ の意味をナイーブに解釈すれば，$\mathtt{argmax}_\Omega$ の役割からして「**z_i を大きくするには x_i を大きくし $(\partial\ell_i/\partial x_i\geq 0)$，$x_j\ (j\neq i)$ を小さくしなければいけない $(\partial\ell_i/\partial x_j\leq 0)$**」ということが表現されているはずである．

性質 2：値の大きさ　一方で Jacobi 行列のそれら要素の大きさに関してはどのように解釈できるだろうか？解釈には様々な可能性がありえるが，1 つとして **出力 z が peaky になるほど Jacobi 行列が零行列に近づくことで勾配の流れを妨げる弁のような機能を持つ**と考える．[3] 厳密でないが，z を peaky にすることは，入力 $\boldsymbol{x}$ に対して温度 τ を小さくすることで模倣できるが，式 2 よりこれは至るところ勾配 0 の argmax に近づくことになり，同様に $\partial\boldsymbol{\ell}/\partial\boldsymbol{x}$ も零行列に近づくと考えられる．このことは，具体的な Jacobi 行列（式 3）からも見て取れる．

性質 3：入力への間接的依存性　最後に，興味深いことに $\mathtt{softmax}$ の Jacobi 行列が $\boldsymbol{z}$ を介して記述でき，$\boldsymbol{x}$ への依存が間接的である．一般論としては，この事実は正則化 argmax の定数に対する不変性に関係する．$\mathtt{argmax}_\Omega$ は明らかに非単射であるが，その Jacobi 行列は $\boldsymbol{z}$ で決まり情報量の多い $\boldsymbol{x}$ を必要としない．具体的に，任意の $\boldsymbol{x}_1,\boldsymbol{x}_2\in\mathbb{R}^n$ に対して，$\boldsymbol{\ell}(\boldsymbol{x}_1)=\boldsymbol{\ell}(\boldsymbol{x}_2)$ ならば $(\partial\boldsymbol{\ell}/\partial\boldsymbol{x})(\boldsymbol{x}_1)=(\partial\boldsymbol{\ell}/\partial\boldsymbol{x})(\boldsymbol{x}_2)$ である．仮に Jacobi 行列間の不等号が成り立てば，$\boldsymbol{x}_1,\boldsymbol{x}_2$ 周辺の適当な変化 ϵ に対する $\boldsymbol{\ell}$ の挙動が異なることになるが，$\boldsymbol{\ell}(\boldsymbol{x}_1)=\boldsymbol{\ell}(\boldsymbol{x}_1+\epsilon\mathbf{1})\neq\boldsymbol{\ell}(\boldsymbol{x}_2+\epsilon\mathbf{1})=\boldsymbol{\ell}(\boldsymbol{x}_2)$ となり仮定に反するからである．

5 F の擬似的な微分

前節で $\mathtt{argmax}$ の微分可能緩和の Jacobi 行列を分析したが，仮に F に対する微分可能緩和 $\tilde{F}$ が存在すれば，およそ § 4 冒頭の仮定を満たし，その Jacobi 行列 $\partial\tilde{F}/\partial\boldsymbol{s}$ にもこの分析と同様の性質が成り立つと考えられる．しかし $\tilde{F}$ の具体的な関数形は closed な形ではわからない．そこで ST Gumbel-Softmax (§ 3.1) と同様に順伝搬時には one-hot ベクトルを並べた $Z=F(\boldsymbol{s})$ を使い，逆伝搬時には，上の分析に基づき**人工的に作った近似 Jacobi 行列** $J\ (\approx\partial\tilde{F}/\partial\boldsymbol{s})$ を用いれば勾配 $\nabla_\phi\mathcal{L}(y,y^\star)$ を推定できる．具体的に，大きな方針は**性質 1** より J の要素の符号は F の望ましい入出力関係に基づき設定し，その値の大きさを**性質 2, 3** に基づいて設計することを考える．

まず簡単に，数量項の組の数を K とし，入出力は必ず一桁で桁の繰り上がりは起きないとした**一桁足し算** $F^1:\mathbb{R}^K\to\{\boldsymbol{e}_n\}_{n\in\mathcal{V}'}\ (\mathcal{V}'=\{1<2<\ldots<9\})$ の場合を考えると，その近似 Jacobi 行列 $J\in\mathbb{R}^{K\times 9}$ は，

3) これを機能として捉える動機の 1 つに，著者による STE [12, 13] と ST Gumbel-Softmax を比較した予備実験がある．弁機能のない STE では argmax 前のスコアの分布の最大値が学習につれ大きく上振れし，ある種の過適合が見られた．

性質 1 より $i = 1, \ldots, K$ ごとのある $d_i, d'_i \geq 0$ を使って下の例のように足し算結果によって決めることができる．($K = 3$, s_i に対応する数の組を右に示す)．

$$J = \begin{bmatrix} \overbrace{d_1}^{0+1=1} & \ldots & \overbrace{-d'_1}^{2+3=5} & \ldots & \overbrace{-d'_1}^{4+5=9} \\ -d'_2 & \ldots & d_2 & \ldots & -d'_2 \\ -d'_3 & \ldots & -d'_3 & \ldots & d_3 \end{bmatrix} \begin{matrix} \\ \\ \\ \end{matrix} \overbrace{\begin{matrix} (0,\ 1) \\ (2,\ 3) \\ (4,\ 5) \end{matrix}}^{\text{入力}}$$

d_i, d'_i の値は**性質 2, 3** を考慮して $\boldsymbol{s}$ を適当な $\mathtt{argmax}_\Omega$ で正規化した値を用い，弁機能も備えさせる．具体的に本稿では Softmax を使い，式 3 を基に $\boldsymbol{z} = \mathtt{softmax}(\boldsymbol{s})$ として $d_i = d'_i = z_i(1 - z_i)$ とする．[4)]

次に，入出力の桁数に関して一般の演算 $Z = F(\boldsymbol{s}) \in \mathbb{R}^{L \times |\mathcal{V}|}$ の近似一階微分 $J \in \mathbb{R}^{K \times L \times |\mathcal{V}|}$ ($J_{ijk} \approx \partial F_{jk} / \partial s_i$) を考える（図 2）．このとき，$J_i$ の要素の符号は $\boldsymbol{s}$ のなかで s_i が最も大きいときに F の期待される出力で決めればよく，つまり $Z_i = F(\mathbf{e}_i) \in \mathbb{R}^{L_i \times |\mathcal{V}|}$ に依る．さらに J の要素の大きさも一桁電卓の場合と同様に調整して $J_i = d_i Z_i - d'_i(1 - Z_i)$ としたい．しかし桁の繰り上がり等で必ずしも $L_i = L$ ではないため，$L_i < L$ の場合後ろを 0 で埋め，$L_i > L$ の場合 Z_i の先頭 L 行だけを使うことで行列の大きさを調整した $\tilde{Z}_i \in \mathbb{R}^{L \times |\mathcal{V}|}$ で $J_i = d_i \tilde{Z}_i - d'_i(1 - \tilde{Z}_i)$ とする．

6 実験

設定 数量推論を必要とする文章読解問題を集めた DROP データセット [1] で実験を行う．開発セットにおける電卓あり/なしの GenBERT [3] の性能を比べる．基本的な実験設定は GenBERT 論文のものに従い，GenBERT 論文で構築された人工データを用いた事前学習のあと，DROP の学習セットで追加学習する．注意すべき点として，電卓ありの場合は別の BERT ベースの $\mathtt{BERT}^{\mathrm{ARG}}_{\phi}$ もモデルに含まれるためパラメータ量の点でフェアな比較ではない．

結果 開発セットでの結果を表 1 に載せる．残念ながら，現段階では電卓の有無によって性能に差は見られず，性能面における電卓の有用性は示されなかった．そこで，項抽出器により抽出された数量が問題を解くために有益なものとなっているかを評価した．項抽出に対する正解データは存在しないため Amazon Mechanical Turk を用いる．具体的に，電卓付きモデルが正答した開発セットの問題からランダムに 100 件抽出し，質問，文章，答え，抽出された数量項をワーカーに見せ，質問から答えに至る過程で数量項を使うかを尋ねた．1 問に対し 3 人のクラウドに評価してもらう．結果，3 人中 k 人が数量項が有益であると判断した問題数は，$k = 1, 2, 3$ に対しそれぞれ 100 問中 40, 14, 8 であった．こちらに関してもあまり良い結果とはいえず課題が残った．

現段階の直感的な考察として，推論器 $\mathtt{GenBERT}_{\psi}$ が電卓から与えられる計算結果が問題を解くために有益であることに気づけない一方で，そのことにより，他方では項抽出器に質の良い教師信号が流れないという状況が起きているのではないかと考えられる．今後さらなる調査を行う予定である．

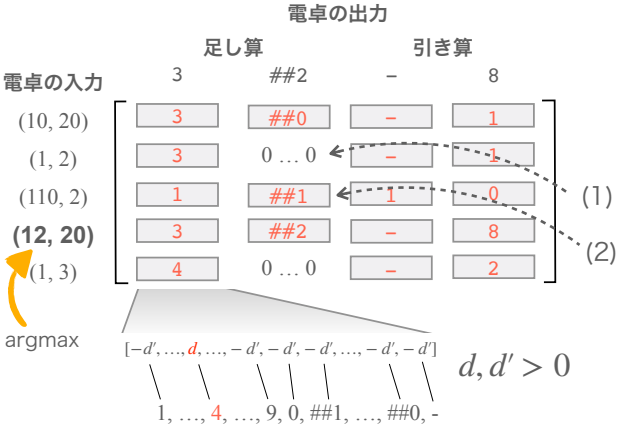

図 2 提案法による Jacobi 行列の計算．(1) 桁が少ない場合 0 で埋め，(2) 桁がはみ出る場合は下を切る．

表 1 DROP 開発セットにおける結果．

手法	EM	F1
GenBERT	68.8	72.3
+ 電卓 (Softmax Jacobian)	68.2	72.4

7 終わりに

本稿では記号処理モジュールの微分を検討し，その応用例として電卓付き文章読解モデルを構築した．鍵となるアイデアは，記号処理モジュールの近似微分の概形はその望ましい入出力関係で決まり，細部は argmax の近似である softmax 等を参考に決めれば良い，ということであった．現状の結果は課題が多く残る．しかし，このような仕組みが完成した暁には数量推論に限らず，理論言語学的な推論の仕組みや外部知識辞書をモデルに統合するなどの応用が可能となり，その利益は大きいと思われ，今後もこの方針の追求に邁進するつもりである．

謝辞 本研究は JSPS 科研費 20K23314 の助成を受けたものです．

4) Sparsemax に基づく方法を実験結果とともに付録 B に載せる．$\mathtt{argmax}_\Omega$ と異なり F の入出力はベクトルと行列で形が非対称であり，その点を考慮しながら設計しなければならない．

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]Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. 2020.

[3]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.

[4]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.

[5]Alon Talmor, Oyvind Tafjord, Peter Clark, Yoav Goldberg, and Jonathan Berant. Leap-of-thought: Teaching pre-trained models to systematically reason over implicit knowledge. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

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

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

[9]Andre Martins and Ramon Astudillo. From softmax to sparsemax: A sparse model of attention and multi-label classification. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, Vol. 48 of *Proceedings of Machine Learning Research*, pp. 1614–1623, New York, New York, USA, 20–22 Jun 2016. PMLR.

[10]Mathieu Blondel, André F.T. Martins, and Vlad Niculae. Learning with fenchel-young losses. *Journal of Machine Learning Research*, Vol. 21, No. 35, pp. 1–69, 2020.

[11]A. Jacovi, G. Hadash, E. Kermany, B. Carmeli, O. Lavi, G. Kour, and J. Berant. Neural network gradient-based learning of black-box function interfaces. In *International Conference on Learning Representations (ICLR)*, 2019.

[12]Geoffrey Hinton. Neural networks for machine learning, coursera, lecture 15d - semantic hashing : 3:05 - 3:35. https://www.cs.toronto.edu/~hinton/coursera/lecture15/lec15d.mp4, 2012.

[13]Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation, 2013.

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

A　提案法のモデルの構造について

ここでは文章 P のサブワード列を $x_1, \ldots, x_N$ とする．GenBERT 論文 [3] と同様に，サブワードへの単語分割時に，数として判定されたトークンは “1234” $\mapsto$ “1, ##2, ##3, ##4” というように必ず 1 文字単位で区切ることにする．N_i として，x_i から始まるこのような数サブワード列の長さとする（x_i が数でなければ適当に 0 とする）．さらに m_i は x_i が数サブワード列の先頭トークンであれば 1，それ以外 $-\infty$ であるマスク定数とする．

A.1　数量項抽出モデル

BERT によって文字列 “[CLS] Q [SEP] P” に対応するサブワード列を走査して得たベクトル表現の列を $\boldsymbol{h}_{\texttt{[CLS]}}, H^Q, \boldsymbol{h}_{\texttt{[SEP]}}, H^P$ とする．ここで $H^P = [\boldsymbol{h}_1^P, \ldots, \boldsymbol{h}_N^P]^\top$ とすると，項抽出スコア $\boldsymbol{s}$ は $\boldsymbol{h}_i^P$ を一度線形変換してマスクを掛けた

$$s'_i = m_i \times (\boldsymbol{w}^{\arg\top} \boldsymbol{h}_i^P + b^{\arg}),$$
$$\boldsymbol{s} = (s'_i + s'_j)_{\substack{i=1,\cdots,N \\ j=i+1\cdots N}},$$

であり，式 1 の $A(\boldsymbol{s})$ は，

$$(\hat{x}_i, \hat{x}_j) = \arg \max_{\substack{(x_i,x_j); \\ \substack{i=1,\cdots,N \\ j=i+1\cdots N}}} s'_i + s'_j \tag{4}$$

$$[\mathbf{1}_{\hat{x}_i} \mathbf{1}_{\hat{x}_{i+1}} \ldots \mathbf{1}_{\hat{x}_{N_i}} \mathbf{1}_{\hat{x}_j} \mathbf{1}_{\hat{x}_{j+1}} \ldots \mathbf{1}_{\hat{x}_{N_j}}] = A(\boldsymbol{s})$$

である．各数量項サブワード列の先頭トークンに付与されたスコア s'_i を，その項を抽出するかどうかの判断に用い，もしその数量項が抽出されれば後ろの連続した N_i トークンを項抽出結果として取り出す．本研究ではこの $A(\boldsymbol{s})$ を微分可能にするために，ST Gumbel-Softmax と同様に誤差逆伝搬時には式 4 の操作に対する Jacobi 行列として Softmax 関数のものを用いる．

A.2　電卓モジュール

電卓の機能　本稿で用いた電卓の二項演算の種類は，既存研究 [14] を参考に選んだ．2 つの数量項を x, y として，以下の 9 つの演算である．

- 和：$x + y$, $(x + y) \times 100$, $(x + y) \div 100$
- 差の絶対値：$|x - y|$, $|x - y| \times 100$, $|x - y| \div 100$
- 積：$x \times y$, $(x \times y) \times 100$, $(x \times y) \div 100$

和，差，積に対して 100 倍や 100 で割る場合も考え，これらの計算の結果が整数値ではない場合は，整数にキャストしたものを計算結果とする．これらの演算を F_i $(i = 1, \ldots, 9)$ として，以下を推論モデルに渡す．

$$[F_1(\boldsymbol{s})F_2(\boldsymbol{s}) \ldots F_9(\boldsymbol{s})\ \boldsymbol{e}_=\boldsymbol{e}_\texttt{D}\boldsymbol{e}_(\ A(\boldsymbol{s})\ \boldsymbol{e}_)]^\top = D(\boldsymbol{s})$$

P に数量項が 2 つ以上含まれる場合，スコアの大きさに依らず常に大きい方から 2 つ抽出し，数量項の数が 1 つ以下の場合は項抽出をせず，電卓を用いなかったことを表すダミー表現として $\boldsymbol{e}_{\texttt{[SEP]}}^\top = D(\boldsymbol{s})$ とする．

A.3　推論モデル (GenBERT)

推論部分 $\texttt{GenBERT}_\psi$ は基本的に論文 [3] のものと同じである．このモデルは，文章内のスパン抽出，質問内のスパン抽出，文字列の生成の 3 つのアクションを持ち，数量項抽出時と同様に BERT に P, Q（提案法の拡張ではさらに $D(Z)$ も含む）を入力し，どのアクションを行うかを $\boldsymbol{h}_{\texttt{[CLS]}}$ の関数として決定した後，対応するアクションを H^Q, H^P の関数として行うことで答えを予測する．

損失 $\mathscr{L}$ も論文 [3] に従い，各アクションによる予測確率を周辺化したものの対数の負を損失とする．詳細については論文を参照頂きたい．

B　Sparsemax に基づく Jacobi 行列

Sparsemax [9] に基づく F の Jacobi 行列の設計の仕方を紹介する．ここで $\mathbb{1}_{[(\text{条件})]}$ は (条件) が満たされるとき 1，それ以外 0 とする．$\texttt{sparsemax}_i(\boldsymbol{s}) = \max(0, s_i - \theta(\boldsymbol{s}))$（$\theta$ はある閾値関数）の Jacobi 行列は以下である：

$$\frac{\partial \texttt{sparsemax}_i}{\partial s_j} = \begin{cases} (1 - 1/N)\mathbb{1}_{[z_i>0]} & i = j \\ -1/N\mathbb{1}_{[z_i,z_j>0]} & i \neq j \end{cases}$$

ここで，N は $z_i > 0$ な添字 i の数である．

Sparsemax に基づく方法では基本的な方針は節 5 の Softmax の場合と同じであるが，Jacobi 行列の要素の大きさを上式の Sparsemax 活性化関数に基づいて設計する．具体的に，$\boldsymbol{z} = \texttt{sparsemax}(\boldsymbol{s})$、$N$ を $z_i > 0$ な添字 i の数として，$d_i, d'_i \geq 0$ を以下のように設定する．

$$\begin{cases} d_i = (1 - 1/N)\mathbb{1}_{[z_i>0]} \\ d'_i = (1/N)\mathbb{1}_{[z_i>0]} \end{cases}$$

この Jacobi 行列を使った場合の DROP データセット上での結果は EM/F1 = 67.3/71.6 であり，Softmax による Jacobi 行列を使った場合とあまり差は見られなかった．