Vol.2020-NL-246 No.11
Vol.2020-SLP-134 No.11
2020/12/2

言語モデルを用いたテキスト中の数字の予測

阪本 拓功[1,a)]　相澤 彰子[2,1,b)]

概要：数字を含む文章を深く理解するには，文中に現れる物体や現象の数量的な特徴に関する知識が必要である．しかし，従来の機械学習言語モデルは文章中の数字を他の単語同様に文字列トークンとして扱うため，数字の持つ大きさを学習に反映させることができず，そういった数量的常識の獲得が困難になっている．そこで，本研究では，文中のマスクされた数字を文脈から予測する数字穴埋めタスクを検証タスクとして用い，現行の機械学習言語モデルが数量的常識をどの程度獲得できているのか，実験を行い，獲得がまだ不十分であることを示した．また，従来の言語モデルの問題点の解決を目指し，数字の大きさをモデルの学習に利用する手法（1. 正解の数字と予測された数字の差の大きさに依存した損失関数を学習において使用する，2. 数字穴埋めタスクを回帰タスクとして解く）を提案し，従来手法と比較して提案手法が予測精度の向上に効果があることを確認した．

1. はじめに

数字や数量表現を含む自然言語文の意味を深く理解するには，文中に現れる物体や事象の数量的な特徴に関する知識を持っていることが必要である．例えば，「太郎くんは身長が 200cm ある．」という文章を読んだとき，人間ならば「太郎くんの身長が 200cm である」という情報からさらに一歩踏み込んだ，「太郎くんは背の高い人である」というところまでをこの文章から推論することができる．しかし，ヒトの身長が一般にどれくらいであるかの常識を持たないシステムではこのような推論は実現できない．そのため，数字を含む自然言語文の深い理解には，文章中に登場する物体や現象の長さや重さなど，実世界における数量的特徴に関する知識を持っていることが欠かせない．

近年，BERT，GPT-3 をはじめとする機械学習言語モデルが，多くの自然言語処理タスクにおいて人間と同等かそれ以上の高い性能を達成している [1] [2] [3] [4]．しかし同時に，数量的推論タスクや数字誤り検出/訂正タスクなど，数字の意味の深い理解や数量的常識が必要な自然言語タスクにおいては現行の機械学習モデルでもまだ性能が低いことも報告されており [5] [6]，こういったタスクにおける性能向上に関する試みが多く報告されている [7] [8] [9]．

[1] 東京大学
The University of Tokyo, 7-3-1, Hongo, Bunkyo, 113-8656 Tokyo, Japan
[2] 国立情報学研究所
National Institute of Informatics, 2-1-2, Hitotsubashi, Chiyoda, 101-8430 Tokyo, Japan
[a)] t_sakamoto@nii.ac.jp
[b)] aizawa@nii.ac.jp

機械学習言語モデルの数字の理解や数量的常識の獲得を困難にしている大きな問題点を二つ挙げる．一つ目は，通常の言語モデルは文章中の数字を他の単語同様に文字列トークンとして扱ってしまうことである [10]．数字を文字列トークンとして扱ってしまうため，数字の持つ大きさの理解が難しくなり，数字の文字列トークンと値の大きさの間のマッピングの獲得や，ひいては「ヒトの身長といったら大体〇〇 cm から〇〇 cm の間でおおよそこのような分布である」といった具体的な数量的常識の獲得が困難になる．二つ目に，BERT をはじめ多くの機械学習言語モデルで用いられているサブワードも数字や数量の理解を困難にしていると考えられる [11]．サブワードは，単語をより短いトークンに分割するもので，低頻度語であってもサブワードの組み合わせで表現できるという利点があるが，単語の場合と異なり数字から得られるサブワードは，もとの数字の意味との関係性が低いことが，数字の理解を困難にする原因となる．

上記の問題点を解決するため，本研究ではまず，機械学習言語モデルの獲得している数量的常識を評価，検証するための「数字穴埋めタスク」を定義する．．ここで数字穴埋めタスクとは文章中のマスクされた数字を文脈から予測するタスクであり，予測結果の正答率，予測値と正解の値の差の大きさで評価する．

次に，数字の文字列トークンの側面だけでなく大きさを持つ値としての側面を捉え，数字の大きさを学習に反映させる手法として以下の二つのアプローチを試みる．

(1) 事前訓練済み言語モデルを数字穴埋めタスク上でファインチューニングする際に，正解の数字と予測された

数字の差の大きさに依存した損失関数を使用する，
(2) 数字穴埋めタスクを回帰タスクとしてモデルの訓練，数字の予測を行う．

また，サブワードアプローチの弊害を解消するため本研究では数字についてはサブワード化せずに学習を進めた．

実験では，ベースラインモデルとして BERT の単語穴埋めモデルを使用し，パッセージの長さやドメイン，出現する数字の分布や性質の異なる四つのデータセット上で，結果の比較，データセット中の数字の性質と各手法の性能との関係の調査，提案手法の効果の確認を行った..

2. 関連研究

本研究で扱うような数字穴埋めタスクを扱う関連研究として [10]，[12] を紹介する．

文献 [10] では数字穴埋めタスクを用いて機械学習言語モデルの”numeracy”を検証し，”numeracy”を向上させる手法として文字レベルの RNN を用いる手法やマスクに入る数字の分布を混合ガウス分布として予測する手法及びそれらのアンサンブル手法を提案している．ここで，”numeracy”とは数字の意味を理解し，適切に使う能力のことである．文献中ではベースラインとして単純な LSTM モデルを使用しているが，本研究ではベースラインとして BERT を用いてモデルの数字を扱う能力の検証を行う．

文献 [12] は「鳥の足は 2 本」の”2”や「車のタイヤは 4 つ」の”4”など答えが一意に定まる常識的な数字穴埋めに焦点を当ててデータセットを作成し，それを用いて現行の機械学習モデルの数量的常識を検証している.. 本研究では数字穴埋めタスクには，(1) 答えが一意に定まる文字列トークンとしての側面に加えて，(2)「彼の身長は [MASK]cm」や「昨日観た映画の長さは [MASK] 分」といった，答えに幅のある量としての側面があることに注目し，機械学習言語モデルのより包括的な数量的常識について検討を行う．

3. 手法

3.1 数字穴埋めタスク

本研究では，機械学習言語モデルの持つ数量的常識の定量的な評価に，文中のマスクされた数字をその周囲の他の単語列から予測する数字穴埋めタスクを用いる．[10]，[12] 数字穴埋めタスクは以下のように定義される．

入力： 特殊記号 [MASK] でパッセージ中に出現する数字をちょうど一つマスクしたパッセージ
出力： 語彙中の数字トークンのランキング

単語穴埋めモデルは，数字のマスクをちょうど一つだけ含むような文章を入力として受け取って，マスクに入る数字を文脈から予測し，数字トークンのランキングの形で出力する．タスクの目的は正解の数字により近い数字をより上位に予測することである．本研究では，マスクされる数字は算用数字で表現された”1”, ”1,000”などに限り，”one”や”twenty”など英単語で表現された数字については扱わない．

3.2 評価指標

数字穴埋めタスクの評価指標としては，モデルの単語穴埋め精度を評価する評価指標 Hit@k accuracy と，正解の数字に近い数字が予測できているかを評価する評価指標である MdAE，MAPE，MdAPE の二種類の評価基準を用いる．

3.2.1 Hit@k accuracy

本研究の単語穴埋めモデルは最終的に softmax 関数を通してモデルの語彙上の確率分布を生成する．Hit@k accuracy は，生成した確率分布からつくられる語彙中の数字トークンのランキングを評価する評価指標であり，正解の数字トークンがランキングの上位 k トークン以内に存在する予測の割合を計算する [12]．実験では $k = 1, 3, 10$ の場合を計算し，評価に用いた．

3.2.2 MdAE，MAPE，MdAPE

Hit@k accuracy は単純に正解の数字が上位 k 個の予測に含まれるかを評価する評価指標であり，予測が正解の数値にどれだけ近いかについては考慮しない．しかし，数字穴埋めタスクの場合には一般に，不正解であったとしても正解の値に近い数字を穴埋めする予測の方が優れた予測であると考えられるため，Hit@k accuracy の他に正解の数字と予測された数字の差の大きさを評価する評価指標が必要である．

そこで本研究では文献 [10] に倣い，正解の数字と予測値の数直線上での近さを評価する指標として中央絶対誤差（Median Absolute Error，MdAE），平均絶対パーセント誤差（Mean Absolute Percentage Error，MAPE），中央絶対パーセント誤差（Median Absolute Percentage Error，MdAPE）を用いる．MdAE，MAPE，MdAPE は以下のように計算される．

$$\mathrm{MdAE} = \mathrm{median}\{|v_i - \hat{v_i}|\}$$

$$\mathrm{MAPE} = \frac{1}{N}\sum_{i=1}^{N}|\frac{v_i - \hat{v_i}}{v_i}|$$

$$\mathrm{MdAPE} = \mathrm{median}\{|\frac{v_i - \hat{v_i}}{v_i}|\}$$

ここで，v_i は正解の数字トークンの値，$\hat{v_i}$ は予測された数字トークンの値，N はマスクの総数である．

これらは回帰モデルの評価の際に一般的に用いられる指標である．数字穴埋めタスクの評価指標として使用する際の注意点については 7 節で考察する。

3.3 提案手法 1：数字の大きさを考慮した損失関数

従来の単語穴埋めモデルは出力として語彙（数字トークンのみ）上の確率分布を返し，それと正解の分布の間の交差エントロピー誤差を損失関数として学習を行う．通常の交差エントロピー誤差は正解でない語彙中の各トークンを対等に扱うが，数字穴埋めタスクの場合には数量的に正解の値に近い予測ほど誤差を小さく，遠い予測ほど誤差が大きくなるような損失関数を訓練に用いたい動機がある．それは一般に，正解が 10 であるマスクに 1 を穴埋めする予測と 9 を穴埋めする予測では後者の予測の方が優れた予測であるためである．そこで本論文では，単語穴埋めモデルを訓練する際の損失関数として，従来手法で一般的な交差エントロピー誤差（Cross Entropy Loss）に代わり，数字の大きさに依存した新たな損失関数 Loss_num ををを以下のように定義する．

$$\text{Loss\_num} = \sum_{i=1}^{N} \{\text{CrossEntropyLoss}_i \times (\log(\text{ans}_i) - \log(\text{pred}_i))^2\}$$

ここで N は全体のマスク数，ans_i は正解の数字トークンの値，pred_i はモデルによって 1 位で予測された数字トークンの値である．正解値と予測値の対数を取った差を計算し（先行研究 [11] の数字の処理に倣った）．通常の交差エントロピー誤差に掛け合わせたものを損失関数とする．事前訓練済みモデルを数字穴埋めタスクでファインチューニングする際にこの損失関数を使用することで，モデルが正解の値と数値的に近い数字トークンを予測する方向にファインチューニングされることが期待される．

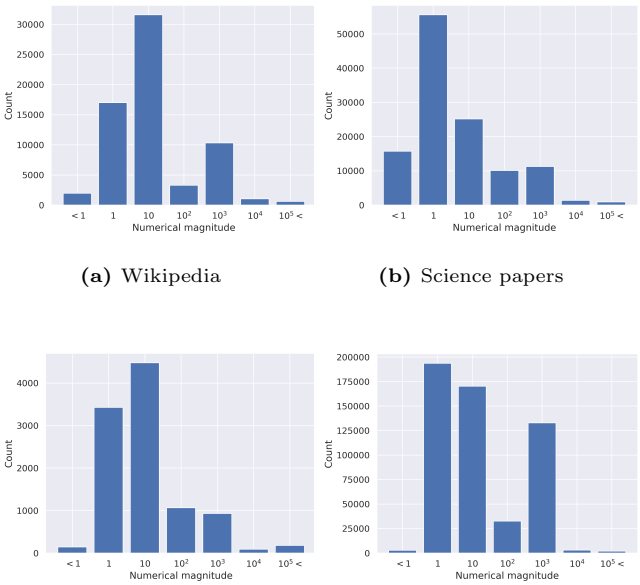
3.4 提案手法 2：数字穴埋めタスクを回帰タスクとして解くモデル

3.3 節の提案手法 1 はモデル自体は従来の単語穴埋めモデルを使用し，そのファインチューニングの際の損失関数として予測値や正解の数字の大きさを反映した損失関数を用いる手法であった．提案手法 2 では，マスクされた数字を一つ含む入力を受け取って，そこに含まれる数字を予測する数字穴埋めタスクを回帰タスクとして解く．

ここで提案する数字穴埋めモデルは，BERT の最終層に数値出力層を追加した構造になっており，出力層は BERT で処理された入力文章から 0 以上 MAXNUM 以下の数字を一つ出力する．ここで，MAXNUM は訓練データ中に出現した最大の数値である．ファインチューニング時の損失関数 Loss_MSE は回帰タスクでよく用いられる正解の値と予測値の間の平均二乗誤差（Mean Squared Error，MSE）を用いる．この際，外れ値の影響の軽減のため正解の数字と予測数値は共に対数をとって MSE の計算を行う．

$$\text{Loss\_MSE} = \sum_{i=1}^{N} (\log(\text{ans}_i) - \log(\text{pred}_i))^2$$

予測の際には出力を整数に丸め，それを 1 位の予測数字とする．それ以下の予測については一位の数字に近い整数を近い順に第二位の予測数字，第三位の予測数字... とする．



(a) Wikipedia **(b)** Science papers

(c) Financial tweets **(d)** Article titles

図 1 各データセットの訓練データ中の数字の分布

4. 実験

本節では実験に使用したデータセットやモデルなど，実験の設定について説明する．

4.1 データセット

本研究では，パッセージの長さやドメイン，データセットに出現する数字の分布が異なる以下の 4 つのデータセットを用いた．

- DROP（Wikipedia）[5]
- arXiv（science papers）[10]
- FinNum（financial tweets）[13]
- Numeracy-600K（article titles）[6]

ここで，データセットに含まれるパッセージや数字についての統計を**表 1** に示す．表 1 の”数字の種類”はデータセットに異なる数字が何種類出現するかをカウントしたもの，”同一パッセージ内での数字の重複”は同一パッセージに二回以上出現するような数字の個数である．また，各データセットに含まれる数字のオーダーごとの分布を**図 1** に示す．各図の横軸は左から 1 未満の数字，1 以上 10 未満の数字，...，10,000 以上 100,000 未満の数字，100,000 以上の数字の分布を示している．

4.1.1 Wikipedia

DROP は，回答に数量的推論スキルや数字の演算スキ

表 1 各データセットの訓練データに含まれるパッセージ及び数字についての統計

データセット	Wikipedia	Science papers	Financial tweets	Article titles
パッセージ数	4,329	14,821	3,992	420,000
平均パッセージ長（トークン数）	281.8	278.2	36.7	12.9
数字の総数	65,783	120,105	10,312	537,214
整数の割合	96.4%	80.0%	86.7%	98.5%
データセット中一回しか出現しない数字の割合	2.93%	3.69%	9.74%	0.36%
数字の種類	3,667	7,944	1,503	4,204
同一パッセージ内での数字の重複	25,269	57,372	1,354	22,555
平均値	1.57e6	5.55e16	2.02e15	2.98e7
中央値	24.0	5.0	20.0	17.0

ルを必要とするような機械読解タスクのデータセットである [5]．データセットは約 6,200 個のパッセージと約 96k 個の質問と回答ペアからなる．含まれるパッセージは Wikipedia から集められた文章で，年号や日付，長さや得点，人数など比較的幅広い種類の数字が含まれる．本研究では質問文と回答については使用せず，パッセージのみ使用した．本研究で使用している他のデータセットと比較すると，パッセージがおよそ 1 段落分ほどと長く，また，整数の割合が多いことが確認できる（表 1）．

4.1.2 Science papers

本研究では科学論文データセットとして文献 [10] で使用されている ARXIV の論文データセットを使用した．パッセージはそれぞれ一つ以上の数字を文章中に含み，パッセージが科学論文からの文章であるため，他のデータセットに比べ，小数の割合が高く，出現する数字の種類も比較的豊富であることが窺える（表 1）．

4.1.3 Financial tweets

これまで紹介した二つのデータセットはパッセージがある程度長く，数字穴埋めの際にマスクの周りの数字からヒントを得て予測を行うことができる．そこでパッセージ長が短く，同じパッセージ中に穴埋めのヒントとなりうる数字が少ない設定での数字穴埋めの精度の検証を行うため，数字を含むツイート文のデータセットと，同様に数字を含む記事タイトルのデータセットでも実験を行った．

FinNum は数字のカテゴリ分類タスクのデータセットである [13]．データセットに含まれる文章は金融に関するツイート文から集められており，株の買値や売値についての金額や日付などの数字が一個以上出現するようなツイート，約 5,700 ツイートからなる．本研究で使用している他のデータセットに比べ，ツイートであるため 1 パッセージの平均長が短めで同一パッセージ内での数字の重複が少ない．また，正確な金額を表記することが多いため小数の割合やデータセット中一回きりしか出現しないような数字の割合が多いなどの特徴がある（表 1）．数字の分布を見ると，他のデータセットに比べ大きな数字の割合が多く，これは金額などの大きな数字について言及するような機会が他のデータセットに比べて多いためと考えられる（図 1）．

4.1.4 Article titles

記事タイトルのデータセットとして本研究では Numeracy-600K の記事タイトルのパートを用いた [6]．データセットは The Examiner データセット[*1]に含まれる記事タイトルのうちタイトル中に数字を一つ以上含むようなもので構成されており，600k 個の記事タイトルを含む．データセットに含まれる数字の統計的分析としては，他のデータセットに比べ，パッセージ長が短く，整数の割合が非常に高い．また，データセットのサイズに比べ，出現する数字の種類は少なく，データセット中一回しか出現しないような数字の割合も非常に少ないといった他のデータセットには見られない傾向も見られる（表 1）．数字の分布を見ると，他のデータセットに比べて 1,000 以上 10,000 未満の数字の割合が非常に高いが，これは記事のタイトル中に年号が含まれることが多いことに起因すると考えられる．

4.2 実験設定

本研究では現行の機械学習モデルの数量的常識を検証，分析するため，ベースラインモデルとしてシンプルな BERT ベースの単語穴埋めモデルを用いた．具体的な構造は，BERT の最終層に softmax 層を追加した構造になっており，softmax 層は BERT で処理された入力文章から [MASK] に入る語の数字語彙上の確率分布を出力する．ここで，[MASK] に入る語は数字トークン一単語で，数字語彙には”10”や”2020”などの算用数字で表現された数字のみが含まれ，”ten”や”twenty four”などの英単語によって表現された数字は含まない．本実験では BERT-base-uncased を用いた．

本研究では，数字については BERT によるサブワード分割を実施せず，数字のワンワード化を行った．数字をワンワードで扱うことで，サブワード分割された数字から学習

[*1] https://www.kaggle.com/therohk/examine-the-examiner

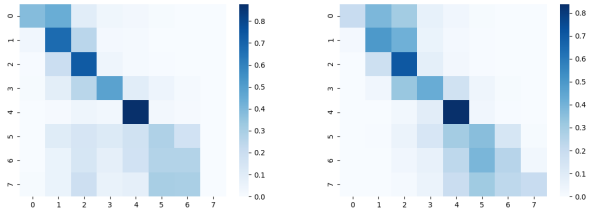
が困難であった数字の文字列とそれに対応する数字の大きさの間の明確なマッピングを学習しやすくなると考えられる．また，数字のワンワード化により，数字を予測する際にエンコーダデコーダモデル等を使用してサブワード列を予測する必要がなくなり，本研究で扱うようなナイーブな単語穴埋めモデルでも数字穴埋めタスクが扱いやすくなるというメリットもある．しかし，ワンワード化を行うことで低頻度数字についての学習が困難になるなどのデメリットも考えられ，それについては 7 節にて関連研究のトークナイズ手法と合わせて触れる．

5. 分析：提案手法 1

表 2 にナイーブな BERT モデル（ファインチューニングあり/なし）と提案手法 1 の結果を示す．ここで”BERT”はファインチューニングなしの事前訓練のみの BERT モデルの数字穴埋め結果，”Ft.BERT”は同じデータセット上の数字穴埋めタスクでファインチューニングされた BERT モデルの結果，”Ft.BERT w/ proposed loss”は提案手法 1 の損失関数を用いてファインチューニングされた BERT モデルの結果を表す．評価指標のうち左の三つは予測の上位 k トークン以内での正答率を測る評価指標なのでスコアが高いほど性能が良いような指標，右の三つの評価指標はエラーの数値的な大きさを測る指標であるのでスコアが低いほど性能が良いような指標である．以下，各データセットを Wikipedia（WP），Science papers（SP），Financial tweets（FT），Article titles（AT）と表現することとする．

まず，各データセットの”BERT”と”Ft.BERT”の項目を比較すると，どのデータセットにおいてもほぼ全ての評価指標でスコアが向上していることが確認できる．一部のデータセットでは MAPE のスコアが大きく低下しているが，これは MAPE が一部の大きなエラーの影響を受けやすい指標であることに起因すると考えられる．また，FT，AT でのファインチューニングによるスコアの伸びが WP，SP におけるスコアの伸びと比べて有意に大きいことが確認できる．これは，WP，SP が FT，AT と比べ平均パッセージ長が長く，ファインチューニングせずともマスクされた数字をマスク周りの数字や文脈からある程度予測することができるためと考えられる（表 1）．

次に，各データセットの”Ft.BERT”と”Ft.BERT w/ proposed loss”の項目を比較する．MdAE, MdAPE に注目すると SP と FT のデータセットにおいて，本提案手法の目的である予測の数値的な絶対誤差や絶対パーセント誤差の減少が達成されていることが確認できた．一方で WP や AT のデータセットにおいてはスコアが低下していた．これはデータセットに含まれる数字の性質の違いによるものだと考えられる．データの特性上 WP や AT には年号や日付，試合の得点などのような文字列トークンとしての側面が強い数字が多く，数字の大きさの理解が数字穴埋めの精度向上につながらなかった．一方で SP や FT には実験結果のスコアを表す数字であったり，詳細な金額など値や量としての側面の強い数字が WP や AT に比べ多く含まれ，数字の大きさの理解が数字穴埋めの精度向上につながったと考えられる．このような文字列トークンとしての数字が多いのか値や量としての数字が多いのかはデータセット中の整数の割合からも一部窺い知ることができる（表 1）．また，提案の損失関数は数字の大きさの理解を促し，予測の数値的な絶対誤差や絶対パーセント誤差を減少させることを目的とした損失関数であるが，一部データセットではわずかではあるものの Hit@k accuracy においても性能の向上が見られた．



(a) MWP model　　**(b)** REG model

図 2　正解値と予測値の桁数で分類した混同行列

6. 分析：提案手法 2

本節ではナイーブな単語穴埋めモデルと，数字穴埋めタスクを回帰タスクとして解くモデル（3.4 節の提案手法 2）の予測結果について比較，分析を行い，また，それらのモデルを用いたアンサンブル手法についても考察する．本節では前者のナイーブな単語穴埋めモデルを MWP モデル，後者の提案手法 2 のモデルを REG モデルと呼称する．

Wikipedia データセットにおける MWP モデル，REG モデルの結果を**表 3** に示す．なお，表 2 の Ft.BERT と性能値が異なるのは，以降のアンサンブルのために，データセットの半分で数字穴埋めを学習させているためである．表 3 の”Ensemble model”，”Best ensemble”の項については 6.1.4 節で触れる．

結果としては，REG モデルは MWP モデルに予測精度で大きく劣る結果となった．しかし，正解値と予測値のオーダーについての分析を行ったところ（**図 2**），REG モデルの方が MWP モデルよりも大きなエラーが少ないという結果が確認できた．図 2 は両モデルの予測値と正解値をそれぞれ桁数で分類した際の混合行列をヒートマップで表現したものである．(a)，(b) はともに縦軸が正解の数字の桁数，横軸がモデルによって予測された数字の桁数で分類されている．具体的には，混同行列において正解の値から桁が二つ以上，三つ以上，四つ以上異なるような大きなエラーの割合が MWP モデルではそれぞれ全体の 8.5%，3.4%，1.5%であったが，REG モデルではそれぞれ全体の

表 2 ベースラインモデル（ファインチューニングあり/なし）と提案手法 1 の結果

Dataset	Approach	Hit@1(%)	Hit@3(%)	Hit@10(%)	MdAE	MAPE(%)	MdAPE(%)
Wikipedia	BERT	23.8	32.1	45.0	7.0	3.28e3	42.9
	Ft.BERT	**28.5**	36.6	49.0	**5.4**	2.79e6	**25.0**
	Ft.BERT w/ proposed loss	**28.5**	**37.2**	**50.2**	6.0	**9.95e3**	28.6
Science papers	BERT	40.1	50.2	63.1	2.0	1.50e5	50.0
	Ft.BERT	45.5	55.5	67.7	**1.0**	**1.67e4**	33.3
	Ft.BERT w/ proposed loss	**48.4**	**57.6**	**69.2**	**1.0**	1.99e7	**25.5**
Financial tweets	BERT	19.9	27.8	43.2	10.0	1.71e4	85.1
	Ft.BERT	**40.5**	**49.1**	**60.0**	**3.0**	1.19e7	50.0
	Ft.BERT w/ proposed loss	40.0	48.2	59.4	**3.0**	**2.64e4**	**46.7**
Article titles	BERT	20.1	32.7	54.7	7.0	2.08e3	80.0
	Ft.BERT	**56.3**	**69.1**	**80.4**	**1.0**	**3.59e3**	**0.0994**
	Ft.BERT w/ proposed loss	55.7	68.5	80.0	**1.0**	4.57e3	0.0995

表 3 Wikipedia データセットにおける提案手法 2 とアンサンブル手法の結果

Approach	Hit@1(%)	Hit@3(%)	Hit@10(%)	MdAE	MAPE(%)	MdAPE(%)
MWP model	27.4	35.8	48.6	6.0	9.14e3	28.6
REG model	4.19	7.52	15.2	54.0	2.78e6	60.0
Ensemble model	26.9	35.1	48.0	6.0	2.15e5	28.8
Best ensemble	28.0	36.3	49.0	4.25	1.97e3	20.0

7.7%，1.8%，0.4%と大きく減少していた．これによって，REG モデルの全体の予測精度は低く，多くの数字については MWP モデルを用いた方がより良い予測が行えるが，数字の中には REG モデルを用いた方が MWP モデルを用いるより正確に予測が行えるような数字が一定数存在するということがわかった．そこで，以下では分析として MWP モデルと REG モデルのアンサンブル手法について考察し，実験を行った．

6.1 数値のタイプに合わせたモデルのアンサンブル

数字には文字列トークンとしての側面と大きさを持つ値としての側面がある．例えば，「クモの足は [MASK] 本」，「一週間は [MASK] 日」，「関ヶ原の戦いは [MASK] 年」といった文章の数字穴埋めを考えると，これらの数字は答えが一意に定まるような，定義や一対一対応の知識の意味合いが強い．そのため，これらの数字は，数字を文字列トークンとして捉えるモデルでも学習可能な文字列トークンの側面の強い数字であると考えられる．一方で，「彼は身長が高く，[MASK]cm もある．」や「昨日観た映画は [MASK] 分もあり，長かった．」といった文章の数字は，背景に何らかの分布を持つ量としての意味合いが強く，値としての側面の強い数字であると言える．後者のような数字を予測するためには，ヒトの身長の分布に関する知識や映画の長さの分布に関する知識を持っていることが必要で，そういった常識は数字を文字列トークンとしてしか捉えず，数字の大きさの理解が乏しいモデルでは困難である．

通常の単語穴埋めを行う従来の言語モデルは，数字も他の単語同様の文字列トークンとして扱うため，数字の文字列としての側面のみを捉えたモデルであると考えられる．一方，数字穴埋めタスクを回帰タスクとして解釈して予測する手法（3.4 節の提案手法 2）は，数字の値としての側面のみを捉えたモデルであると言える．

そこで，本節では，数字を文字列として捉えるモデルと値として捉えるモデルの二つのモデルを用いたアンサンブルを考える．具体的には，数字穴埋めタスクを以下の 2 ステップに分けて実行する．

(1) 文章中のマスクされた数字を，その周りの文脈から文字列の側面の強い数字か値の側面の強い数字か分類する．
(2) 文字列の側面の強い数字には従来の単語穴埋めモデルを用いて，値の側面の強い数字には提案手法 2 のモデルを用いて数字穴埋めを行う．

6.1.1 数字の分類モデル

数字の分類には，マスクされた数字をその文脈から分類する分類モデルを訓練し，それを用いた．モデルの構造はこれまでのモデル同様，BERT に最終層として，MWP モデルで予測した方が良い数字か REG モデルで予測した方が良い数字かを判定する出力層を追加した構造になっている．モデルは [MASK] を含むパッセージを受け取り，それを BERT で処理する．その後出力層にて BERT の処理結果から 0 以上 1 以下の数値を出力し，それにより MWP モデルで予測した方が良い数字か（出力が 0 に近かった場合）REG モデルで予測した方が良い数字か（出力が 1 に

表 4 数字の分類モデルの分類精度

Gold \ Pred	MWP model	REG model	sum
MWP model	62.8%	13.2%	76.0%
REG model	11.7%	12.2%	24.0%
sum	74.5%	25.5%	100%

近かった場合）判定する．

6.1.2 モデルの訓練

数字穴埋めモデル（MWP モデルと REG モデル）の訓練と 6.1.1 節の数字の分類モデルの訓練を別のデータセットで行うため，本実験ではデータセットを A と B 半分ずつ二つに分けてそれぞれを穴埋めモデルと分類モデルの訓練に用いた．

実験の流れを説明する．まず，分割した片方のデータセット A で数字穴埋めモデル（MWP モデル，REG モデル）の訓練を行い，それらで数字穴埋めができるようにする．次に，訓練した二つのモデルを用いてもう一方のデータセット B 中の数字を実際に予測し，二つのモデルの予測結果に基づいてデータセット B 中の各数字についてそれが MWP モデルで予測した方が良いものか REG モデルで予測した方がいいものか正解ラベルをアノテーションする．アノテーションは，MWP モデル，REG モデルの一位の予測値を実際の正解の数字と比べ，より近い数字を予測できていた方のモデルを正解ラベルとする．最後に，各数字について正解ラベルが付与されたデータセット B を用いて分類モデルの学習を行う．ここで分類モデルも判定の際には対象の数字はマスクされており，答えの数字から情報を得て判定はできないことを注意しておく．

6.1.3 分類モデルを用いた数字の穴埋め

数字の穴埋めを行う際は，6.1.2 節で訓練した分類モデルの分類結果を基に予測モデルを使い分ける．具体的には，分類モデルが MWP モデルで予測した方が良い数字と分類した数字には MWP モデルを，REG モデルで予測した方が良い数字と分類した数字には REG モデルを用いて数字の穴埋めを行う．これにより，数字のタイプごとに適切なモデルを使った予測を行うことができると考えられる．

6.1.4 結果

MWP モデルと REG モデルのアンサンブル手法の結果を示す．まず，マスクされた数字を含むパッセージを受け取ってその数字は MWP モデルで予測した方が良いものか REG モデルで予測した方が良いものかを判定する分類器の分類結果を**表 4** に示す．

まず，Wikipedia のテストセットに含まれる数字は 76%が MWP モデルで予測を行った方が正解に近かった数字，残りの 24%が REG モデルで予測を行った方が正解に近かった数字であった．前者の 76%の数字については，そのうちの 80%以上を正しく MWP モデルでの予測が向いている数字であると判定できていた．しかし，後者の 24%の数字については，ラベルの偏りの影響もあるかもしれないが MWP モデルで予測するべきか REG モデルで予測するべきかの判定精度はおよそ 50%で，あまり分類がうまくいっているとは言えない結果であった．全体としての予測精度は 75%ほどであった．

次に，分類器を用いた実際の予測精度を表 3 の”Ensemble model”の項に示す．いずれの評価指標においてもアンサンブル手法の性能は MWP モデルの性能以下という結果であった．この結果の原因としては，MWP モデルで予測するべき文字列トークンの側面が強い数字であるか，REG モデルで予測するべき値や量の側面が強い数字であるかの判定が曖昧であるということが挙げられる．これらの判定には絶対的な正解ラベルが存在せず，現状，訓練した予測モデルを使って訓練データに正解ラベルを付与している．しかし，この正解ラベルは付与する際に用いたモデルの性能に依存する．例えば，正解ラベルをつける際に性能の良い MWP モデルと性能の低い REG モデルを用いれば MWP モデルでの予測が向いているというラベルの個数が多くなり，逆に性能の低い MWP モデルと性能の良い REG モデルを用いれば REG モデルでの予測が向いているというラベルの個数が多くなる．このように明確な正解ラベルがないことが分類精度が上がらない原因であると考えられる．

ここで，分類器が 100%の精度で分類を行えたとした場合の予測精度を表 3 の”Best ensemble”の項に示す．このスコアは現状の MWP モデルと REG モデルを最も上手にアンサンブルした際に得られるスコアである．これより，アンサンブルをうまくやることでいずれの評価指標についてもまだ改善の余地があり，特にパーセント誤差については現状からまだ大きく減らすことができることがわかる．

7. 今後の課題

今後の課題としては，まず，数字のトークナイズ手法の検討が挙げられる．本研究では，数字の文字列と数字の持つ大きさのマッピングの獲得を容易にするため数字のワンワード化を行った．しかし，BERT は事前学習の際には数字もサブワード分割されており，ワンワード化された数字の意味の学習はファインチューニングでしか行えないこと，また，サブワードに分割される数字はそもそもデータセット中に頻繁には現れない低頻度数字であることが多いことなど，ワンワード化された数字の意味の学習にはまだ困難がある．本実験では，サブワード分割された数字を一つにまとめ直す方向のアプローチを行なったが，逆に，例えば一桁ごとに分割するなど，数字を決められたサブワードに更に分割する digit tokenize 等のアプローチも考えら

れる [8] [11]．そういった digit tokenize のような手法であれば，数字の文字列と数字の持つ大きさのマッピングの獲得と低頻度数字の意味の学習の双方に効果があることが期待できる．

また，数字穴埋めタスクの評価指標についても検討の余地がある．本研究では予測値と正解の値の差の大きさを評価する評価指標として MdAE，MAPE，MdAPE を用いたが，これらの指標にはそれぞれ注意しておかなくてはならないことがある．まず，これらのうち，各エラーの絶対誤差を用いる MdAE はデータのスケールに敏感な指標であり，日付や年号，金額など異なるスケールの数字が幅広く含まれているデータセットにおいては大きな数字に関する予測の精度に引っ張られやすい．また最終的に誤差の平均を計算する MAPE は外れ値のような極端なエラーに敏感で，これも 0 に近いような数字から何桁もの数字まで幅広く含まれるようなデータセットにおいてはそのような大きなエラーが起きやすく実際の性能のを評価の際にノイズとなってしまいやすい．そして最後に，パーセント誤差を扱う MAPE や MdAPE は，正解の数字よりも小さな数字を予測した予測よりも，正解の数字よりも大きい数字を予測した予測に対して大きなペナルティを課してしまうという問題がある．例えば，「今日は 10 月 31 日だ」の”31”に対して”1”を予測するような予測と「今日は 10 月 1 日だ」の”1”に対して”31”を予測するような予測はどちらも同程度の誤りであると感じるが，前者ははおよそ 100%のエラーとなるが，後者は 3000%のエラーとみなされてしまう．現状使用している評価指標のこのような問題点から数字穴埋めタスクのより適切な評価方法にはまだ検討の余地があると考えられる．

8. おわりに

本研究では，数字穴埋めタスクを検証タスクとして用いて，現行の機械学習言語モデルの持つ数量的常識を定量的に評価した．また，従来の言語モデルの，数字も他の単語同様に文字列トークンとして扱ってしまうという問題点を改善する手法として，学習に数字の大きさを反映させる手法を提案し，それらの効果を確認した．性質の異なる四つのデータセット上で実験を行い，データセット中の数字の性質とモデルの性能との関係の分析を行った．分析結果から，従来手法と提案手法のアンサンブル手法についても考察し，実験を行った．

参考文献

[1] Devlin, J., Chang, M.-W., Lee, K. and Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, *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)*, Minneapolis, Minnesota, Association for Computational Linguistics, pp. 4171–4186 (online), DOI: 10.18653/v1/N19-1423 (2019).

[2] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L. and Stoyanov, V.: RoBERTa: A Robustly Optimized BERT Pretraining Approach (2019).

[3] Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P. and Soricut, R.: ALBERT: A Lite BERT for Self-supervised Learning of Language Representations (2020).

[4] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I. and Amodei, D.: Language Models are Few-Shot Learners (2020).

[5] Dua, D., Wang, Y., Dasigi, P., Stanovsky, G., Singh, S. and Gardner, M.: DROP: A Reading Comprehension Benchmark Requiring Discrete Reasoning Over Paragraphs (2019).

[6] Chen, C.-C., Huang, H.-H., Takamura, H. and Chen, H.-H.: Numeracy-600K: Learning Numeracy for Detecting Exaggerated Information in Market Comments, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy, Association for Computational Linguistics, pp. 6307–6313 (online), DOI: 10.18653/v1/P19-1635 (2019).

[7] Hu, M., Peng, Y., Huang, Z. and Li, D.: A Multi-Type Multi-Span Network for Reading Comprehension that Requires Discrete Reasoning (2019).

[8] Geva, M., Gupta, A. and Berant, J.: Injecting Numerical Reasoning Skills into Language Models (2020).

[9] Spithourakis, G. P., Augenstein, I. and Riedel, S.: Numerically Grounded Language Models for Semantic Error Correction, *CoRR*, Vol. abs/1608.04147 (online), available from ⟨http://arxiv.org/abs/1608.04147⟩ (2016).

[10] Spithourakis, G. P. and Riedel, S.: Numeracy for Language Models: Evaluating and Improving their Ability to Predict Numbers, *CoRR*, Vol. abs/1805.08154 (online), available from ⟨http://arxiv.org/abs/1805.08154⟩ (2018).

[11] Wallace, E., Wang, Y., Li, S., Singh, S. and Gardner, M.: Do NLP Models Know Numbers? Probing Numeracy in Embeddings (2019).

[12] Lin, B. Y., Lee, S., Khanna, R. and Ren, X.: Birds have four legs?! NumerSense: Probing Numerical Commonsense Knowledge of Pre-trained Language Models (2020).

[13] Chen, C., Huang, H., Shiue, Y. and Chen, H.: Numeral Understanding in Financial Tweets for Fine-grained Crowd-based Forecasting, *CoRR*, Vol. abs/1809.05356 (online), available from ⟨http://arxiv.org/abs/1809.05356⟩ (2018).