自然言語学習データの自動生成と技術文書からの知識抽出に関する基礎的研究

Fundamental Study on Automated Learning Data Generation and Knowledge Acquisition from Technical Documents

○恒川　裕史[*1]
Hiroshi TSUNEKAWA[*1]

*1 株式会社竹中工務店技術研究所　上席研究員
Research and Development Institute, TAKENAKA Corporation

Summary: A method to generate training data automatically has been proposed. Deep learning has been applied in various fields of industry. In the construction business, the experience and research results have been accumulated as a huge amount of textual information, and we believe that it will be important to extract knowledge from such technical documents when applying AI. Deep learning needs huge amount of data, so to generate training data is attempted. To generate the training data, sentences are extracted from technical documents, templates are applied to them, and term substitution is performed using the masked language model. In terms substitution, negative examples are generated by changing technical terms and positive examples are generated by changing non-technical terms. It is shown that training data can be safely generated by avoiding the main part of the sentence by parsing. An original architectural language model is developed, incorporating a tokenizer that encompasses architectural terms and an architectural corpus. Furthermore, a transformer-based discriminator is trained using the generated training data, and its accuracy is evaluated using separate verification data. Comparing the results with published language models, the accuracy of the created architectural language model is found to be the highest at 0.78.

キーワード: 自然言語処理; ディープラーニング; 機械学習; 自己教師あり学習; 言語モデル
Keywords: Natural language processing; deep learning; machine learning; self supervised learning; language model.

1. はじめに

産業の様々な分野で AI 技術の活用が進んでいる。当初は数値データの機械学習や画像認識がそのメインであったが、深層学習、特に Transformer[4]を応用した技術の進展により、テキスト情報もその対象となりつつある。建築業務ではこれまでの経験や研究の成果が膨大なテキスト情報として蓄積されており、AI 適用に当たっては、こうした技術文書から知識を抽出することが重要になると考え、継続的な研究を行ってきた。本研究は、文献[1)2)3)]に加筆修正したものである。

2. 検討の方針と課題設定

数値データでは、業務で作成・収集したデータをそのまま学習データにできる場合が多いが、テキスト情報の場合は、新たに作成する以外にない。深層学習による自然言語処理技術は、従来に比べ必要な学習データ数が少なくなってきてはいるものの、やはり数千件以上のデータが必要である。こうした学習データは、近年ではアノテータと呼ばれる人々により、クラウドソーシングなどを利用して作成されることが多くなっている。しかし、質の良い多数のアノテータを使うには相当なコストがかかる。そこで、本研究では、技術文書からの学習データ自動生成を試みた。画像認識の分野では既に自己教師あり学習[5)]としてこうした試みが行われている。現在脚光を浴びている GPT3[6)]を始めとした Transformer による仕組みは、教師なし学習による言語モデル作成と課題に即した教師あり学習を組合わせたものであり、後半には学習データが必要である。また、生成 AI では生成された内容が真実か否かを判定することが必要で、そこにも学習データが必要である。本研究の最終目標をユーザの自由な質問に対して回答することと設定し、本論文では、最初の試みとして、文の正誤判定、いわゆる○×問題の学習データを生成し学習・推論することを目的とした。画像ラベル判別の負例は無関係な画像で良いが、○×問題ではそうはいかず、むしろ難しい課題である。学習データ生成には、テンプレート置換に加えて言語モデルによる用語置換を用いた。言語モデルは本来は類似した文の生成を目的としているが、それを用いて判別の難しい○×問題を生成するところに本研究の特徴がある。生成した学習データを用いた従来技術と深層学習との性能比較で

その後の判定手法を確定した。自由な質問に対応するための検討での専門用語であるか否かで正負例を生成し分ける点も本研究の特徴と言える。社内外技術文書に基づき技術支援する場面を想定しているが、内容を示す必要性から対象文書を公共建築工事標準仕様書とした。

3. 手法

3.1. 学習データ自動生成の方針と検討ステップ

○×問題の学習データでは、○と×に対応する正例と負例との 2 種類のデータを作る必要がある。技術文書から取り出した文を正例とし、対応する負例を生成すれば最低限の学習データを生成できる。これを BASE 法とした。その際、質問文は、「単一の文で意味の通る質問文とする」ことを方針とした。また、最終目標に向けて、次に正例、負例にバリエーションを持たせて学習データを増やし、これを EXT 法とした。

3.2. BASE 法

方針に従い、図、表、式などを参照する項目は除外した。また、仕様書は章、節、項などと階層構造になっており、末端の文のみでは何を対象に記述しているのかわからない場合がある。この場合は、文章の構造を解析し、適宜、表現を補った。こうして作成した文を正例とした。負例は、4 種類のテンプレートによる置換と用語の置換で正例に対して 1:1 の割合で作成した。

一つ目のテンプレートは、監督職員への対応である。監督職員の承諾を受ける、報告する、検査を受ける。このパターンには 7 種類の対応があり、これを入れ替える。二つ目は、法律である。仕様書には「職業能力開発促進法」など、いくつかの法律が引用されている。自動抽出された法律は 16 あり、これらの法律を入れ替える。三つ目は、人である。「技能資格者」、「施工管理技術者」を代表とする 2 種類の用語があり、それぞれ 4 パターンの人を表す用語の中で入れ替える。四つ目は、「次の部分」として項目が列挙されている箇所で、列挙部分を本文に挿入したものを正例とし、動詞を否定（元々否定の場合は肯定）形にして負例とする。

用語の置換には、Transformer の一種である BERT[7]を利用した。前述した言語モデル作成（事前学習）で、BERT は隠された単語を推定する Masked LM の学習を行うが、これは、文としての自然な単語の並び方を学習するもので、似たような用語に置き換えたい今回のニーズに合致している。置換は、以下の手順で行った。

1. 専門用語自動抽出システム termex[8]を用い、処理結果から不要な語を除いた上で専門用語を抽出する
2. 文書構造に基づいて補った部分を除き、Cabocha[9]で構文解析し、主語、述語、目的語を抽出する
3. 主語、述語、目的語およびその他の部分で専門用語の出現頻度を基準に、頻度が等しい場合は優先度を使い、式(1)でスコアを計算し、置換か所を決定する
4. BERT の事前学習モデルで置換か所を推定し、置換候補の中から置換前の単語を除外して BERT モデルの出力値により専門用語を優先的に選択する

$$score = Nw + a \quad (1)$$

ただし、Nw: 専門用語の頻度、a: 優先度である。

3.3. EXT 法

最終目標に向け、EXT 法では質問文の自由度を上げることを目指した。BASE 法では、簡単のため、全正例と異なる文を生成できればそれを負例として採用していたが、正例と異なっていても意味的に元の技術文書に適合するものは本来の目的に照らして負例とは言い難い。意味解析が困難な現状の自然言語処理技術では、これらの課題を即座に解決するのは困難である。しかし、ユーザの自然な質問に答えるシステムの構築に向けた試みとして、意味的に正例、負例に適合しつつ、文の表現の自由度を上げることを試みた。

BASE 法では、文書から切り出した正例 1 件に対して 1 件の負例のみを生成していたが、表現の自由度を上げるという方針に従い、テンプレート置換で生成できる可能な限りの負例を生成した。また、BASE 法の用語置換では、構文解析をして主語、述語、目的語を重点的に置き換えたが、その後の分析で、こうした文の主要部分を替えると、思わぬ副次的な効果で正例と考えざるを得ない文が生成されてしまうことが判明した。その例については、実施事例で説明する。このため、あえて文の主要部分を避けることで、このような副次的効果を抑えることとした。更に、用語置換を行う際、BASE 法では専門用語を 1 つのマスクで置き換えて置換を行っていた。しかし、すべての専門用語が言語モデルのトークンに含まれる訳ではない。利用した言語モデルのトークン化の関係で、例えば「設計図書」は「設計」と「図書」という複数のトークンに分割されるため、マスクとトークンとの間には不整合が起きていた。そのため EXT 法では、BERT のトークン化に準じた数のマスクで置換を行うこととした。また、文表現の自由度を上げるという方針も鑑み、当該専門用語が例えば 3 つのトークンで表されるとすると、3,2,1 個の 3 種類の連続したマスクに置き換えて置換を行った。この時、複数のマスクで置換されるトークンを一度に推定するのではなく、文の前方から順に推定した。また、同じ意味を持つ用語に置き換えたものも負例とは言い難いため、同義語辞書[10,11]を用いて置換用語ペアをチェックし、同義語には置換しないこととした。

負例の数を増やしたため、バランスの取れた学習をす

るためには同じ数の正例が必要である。そのため、BERTの置換による正例の生成を試みた。負例では異なる文意とするために専門用語を置き換えたが、正例ではその反対に、意味が異ならないように専門用語を除いた単語を置き換える。それでも、置き換えた単語が元の単語と反対の意味になるような単語であると、文の意味が異なってしまう。そのため、置換は対義語辞書[12]を用いて置換用語ペアをチェックし、対義語には置換しないようにした。また、試行錯誤により対義語辞書を補強した。

3.4. 学習法

最初にBASE法を用いて従来手法と深層学習との比較を行う。従来手法には、勾配ブースティングに基づくXGBoost[13]を用いた。設定した課題は文単位で正誤を答える二値判定である。このため、ブースターは決定木、入力はBag of Words、目的変数は2クラス分類とした。深層学習は、用語置換でも用いたBERTとした。BERTによる学習は、前述の通り事前学習と教師あり学習（微調整）で行われる。二値判定は、BERTの例題であるポジティブかネガティブかを判断する感情分析と同じである。したがって、微調整の学習には、例題同様にTransformer層の後に全結合層を介してSoftmaxで2クラス分類出力するネットワークを用いた。

事前学習結果はいわゆる言語モデルであり、BERTの場合はMasked言語モデルであるが、既にいくつかのモデルが公開されている。微調整に比べて事前学習には多くの計算機資源と計算時間が必要であるため、こうした公開モデルを使うことができれば利便性が高い。また、BERT以降様々な手法が研究されており、精度が更に向上している。このため、公開されている言語モデルの中から、BERTおよびBERT以降の新手法による言語モデルの適用性を検討する。しかしながら、こうした公開モデルには建築特有の単語が登録されておらず、精度に影響することが予想される。このため、次節で説明するように、新たに建築用の言語モデルも作成した。

学習にはTransformers[14]を用いた。

3.5. 建築用言語モデル

言語モデルでは文をトークンというデータ列に変換する。言語モデルの作成にはトークンへの建築用語の追加と、建築用学習データが必要であるので、順に説明する。

まず、一般的な建築用語としてJCCS[15]から一般用語を除いた7,489語と、公共建築工事標準仕様書から専門用語自動抽出システム[8]で抽出し、JCCSとの重複や、「○○等」のような無意味な用語を除いた3,535語を用いて、基本辞書に追加する辞書を作成した。

建築用の学習データとしては、以下のものを利用した。なお、著作権法30条の4により、情報解析の場合は著作物の利用が許されている[16]。

a. 建築学会全国大会予稿集10年分
b. 建築関連法令128件
c. 公共建築工事標準仕様書（建築、電気、機械）

aとcはPDFから、bはe-Gov法令検索からダウンロードしたXMLからテキストを抽出した。aでは、所属や名前と思われる個所は除外した。cでは、古い法令で促音が大文字の「つ」で表記されたものがあったため、BERTの学習データを作成してNERにより促音の「つ」を同定し、「っ」に置き換えた。抽出した文は、括弧内を除く句点で分割した上で、句点で終わるものおよびaについてはタイトルを文と考え、unicode正規化(NFKC)した上で、学習データとした。表1にそれぞれのデータ数を示す。

表1　言語モデル用学習データ

	a	b	c
行数	1,894,515	24,960	17,308
データ量	281MB	4.42MB	1.97MB

トークン化器の学習は文献17)に倣い日本語Wikipediaから抽出したテキストからランダムに選んだ100万行に、上記の建築用データを加えて行った。その際、韓国語やアラビア語などの文は除外した。また、数値は多様なバリエーションの割には効果が低いことを考慮し、数値のトークンができることを防ぐために、トークン化器学習の際は連続する数値をすべて1文字の0に置き換え、学習後に0〜9をトークン（単独およびサブワード）として影響のなさそうなトークンと入れ替えた。

言語モデルの学習は、2種類の方法で行った。Arch fine tuneは、事前学習済み言語モデルを微調整する方法、Archは上記で作成したトークン化器を使って一から学習する方法である。前者は短期間で作成できるが、建築用語をトークンとすることができない。反対に後者は建築用語をトークンとする純然たる建築用言語モデルを作成できるが、学習には計算機材と時間が必要である。

3.6. 検証用データ

学習データ生成から推論までの手法の妥当性は別途用意する検証用データの精度で確認する。自由な質問に対して回答できることを目指すEXT法の能力検証には本来なら自由質問を用いるべきであるが、まだそのようなレベルではなく、また難度の設定も困難であるため、BERTによる置換の延長線上で作成した。前述の学習用データの生成では負例の専門用語、正例の非専門用語は、ともに1つの文で1か所のみ置換を行った。これに対して検証用データは、複数個所の置換を行った。更に、日

本語には語順を変えられるという BERT を生んだ英米語にはない特徴があるため、意味が変わらない範囲で置換後の文の語順に変化を付けたものを使用した。

前述した通り、学習データは、近年ではアノテータにより作成されることが多い。しかし、アノテータには癖あるいは思想によるバイアスがあり、またアノテーションした量に応じて報酬が払われることから効率を求める結果、ある種の傾向が生じてしまうため、学習手法は課題の本質ではなく、癖をヒントに学習してしまうことが知られている[18]。本研究においても、一定の機械的な方法により学習データを生成するため、同じような危険は避けられない。こうした傾向を確認するため、少々意地悪な検証データも設定した。具体的には、専門用語置換に加えて非専門用語の置換を施した負例を生成した。学習段階ではそれぞれ負例は専門用語置換、正例は非専門用語置換しか学習しておらず、それらが混合した時にどのような判定をするかは興味をそそるところである。

表 2　専門用語例

term	importance	frequency
コンクリート	10934728	301
試験	3438708	182
工法	2512224	244
工事	1931160	114

表 3　生成した負例の数（BASE）

	template				BERT
	1	2	3	4	
counts	226	11	38	13	2,287

表 4　BERT 置換による負例数（BASE）

	subject	predicate	object	Other
counts	389	152	1,669	77

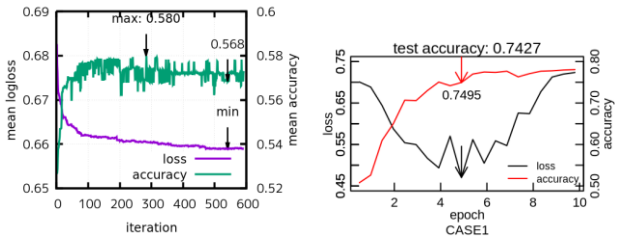
4. 試行結果

4.1. BASE 法による従来法との比較

専門用語抽出では、6,069 語の専門用語が抽出された。抽出された用語の例を重要度順に表 2 に示す。

仕様書から抽出し、章、節名などを補った質問文は、2,588 件得られた。負例は 4 つのテンプレートで計 288 例、BERT による用語置換で 2,287 例作成し、計 2,575 組の学習データを生成した。その内訳を表 3 に示す。テンプレートの中では、1 が多い結果となった。優先度 a は、質問文としての自然さを考慮して主語: 0.3, 述語: 0.4, 目的語: 0.5, その他: 0.1 と設定した。その内訳を表 4 に示す。置換候補を 20 位まで抽出したが、その中に置換前の用語が存在した質問文が 72%あり、BERT の優秀さを確認した。事前学習モデルには cl-tohoku/bert-large-japanese[17]を用いた。これは 24 層、隠れ層 1,024、16 ヘッドのモデルである。質問文と置換結果例を下に示す。章、節名などを補った部分を下線で、置換か所を[]で示す。

正例：工事現場管理の…の保安責任者について、保安責任者は、関係法令に基づき、適切な[保安業務]を行う。
負例：工事現場管理の…の保安責任者について、保安責任者は、関係法令に基づき、適切な[管理]を行う。



(a)XGBoost　　(b)BERT

図 1　評価結果（BASE）

表 5　BERT の Accuracy（BASE）

1	2	3	4	5	average
0.7427	0.7524	0.7301	0.7864	0.7689	0.7561

表 6　BERT のタイプ別 Accuracy（BASE）

true	false				
	1	2	3	4	BERT
0.81	0.71	0.27	0.71	0.62	0.70

全学習データを5分割したCross Validation(CV)により、非訓練データのlossが最小になるようにハイパーパラメータチューニングをした。XGBoostでは学習率ほかを合計2,880回グリッドサーチした。評価結果を図1(a)に示す。XGBoostではiterationが決定木の数に対応するためCVではiterationごとの平均lossを最小化した。BERTではOptuna[19]を用いて学習率ほかを50回探索した。この際、CVのケースごとの最小lossを指標とし、チューニングに使うデータと最終的な精度を計算するデータ(test)は別にした。事前学習モデルは質問文作成時に用いたものと同じものとした。CV各ケースの評価結果を表5に、一例を図1(b)に示す。XGBoostに比べてBERTの精度が高く、Transformerの有効性を確認した。表6に、質問文タイプごとの精度を示す。正例に比べて負例のaccuracyが低く、難しい負例を生成できていることがわかる。負例2が極端に低いのは、学習データの少なさのためと思われる。

4.2. EXT 法による学習データ生成

BASE 法では構文解析を用いて SVO 部分で置換を行った。下に示した①が技術文書から抽出した質問文。②は主語である「受注者」を「作業」に置換したものである。確かに仕様書にこのような文は無いため負例としてまったく不適切という訳ではないが、文としては正しいことを主張しており、支援システムの負例としては相応しくない。③は SVO を避けて「標準仕様書」を「設計要綱」に置換しており、安全に負例が生成できている。SVO を除外したため、生成できた質問文は 2,485 件に減少した。

表 7　生成した正例負例の数（EXT）

type	負例					正例
	1	2	3	4	BERT	
counts	1,171	93	170	13	18,313	17,275

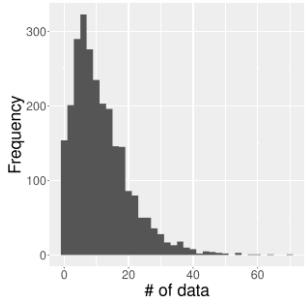
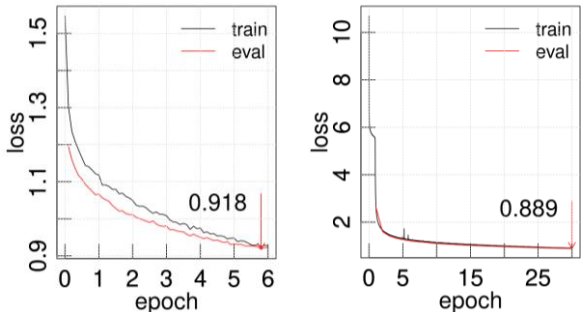


図 2　質問文毎の学習データ数（EXT）



(a) Arch fine tune　　(b)Arch

図 3　建築用言語モデルの学習状況

① 受注者は、設計図書（別冊の図面、標準仕様書、特記仕様書、現場説明書及び現場説明に対する質問回答書をいう。以下同じ。）に従い、責任をもって履行する。
② 作業は、設計図書…責任をもって履行する。
③ 受注者は、設計図書（別冊の図面、設計要綱、…

質問文ごとの正例と負例の数を同じにするため少ない方に合わせた結果、全体で 30,739 対の学習データが生成された。質問文ごとのデータ数の分布を図 2 に示す。図から、生成できた学習データの数には質問文によりかなりのばらつきがあることがわかる。あまり大きなデータ数のばらつきは学習に悪い影響を与えるため、最大 10 件までを最終的な学習データとし、19,760 対のデータを生成した。内訳を表 7 に示す。

4.3. 建築用言語モデルの作成

Arch fine tune は、cl-tohoku/bert-large-japanese をベースに、前述の建築用の学習データで微調整を行った。5.8 エポックで eval loss が最小となったため、この時点の結果を言語モデルとした。Arch は日本語 Wikipedia を含めたすべてのデータを建築用語を加えたトークン化器を用いてトークン化し、ネットワークの構造パラメータは文献 17 に従い、事前学習を行った。いずれも、全体の 1 割を言語モデル用検証データとした。図 3 に学習状況を示す。

4.4. 検証用データの作成

検証用データは、専門用語を 2 か所または 4 か所(T2, T4)、非専門用語を 2 か所または 4 か所(N2, N4)置換したものと、専門用語と非専門用語を 1 か所ずつ(T1N1)、2 か所ずつ(T2N2)置換したものを作成した。生成した 1,275 件から手作業で正しくないものを除外して 755 件とし、正例と負例の数を揃えて 736 件のデータを設定した。次に、語順変更は、文を句読点で分割した後、GiNZA[20]で構文解析を行い、同じ文節にかかる複数文節の順序を入れ替えた。上記のデータの中で語順変更できた文の中から、元の質問文ごとに 1 件をランダム抽出した 349 件から手作業で誤りを除外して 179 件のデータとし、正例と負例の数を揃えて 170 件を選択した。最終的に、906 件のデータを検証用とした。

表 8　検証ケース

case	data	language model	network
$Bert_{LargeOrg}$	base	bert-large-japanese[17]	$Bert_{Large}$
$Bert_{Large}$	ext	bert-large-japanese[17]	
$Arch_{Finetune}$		Arch fine tune	
$Arch_{(10/20/30)}$		Arch	
$Bert_{Base}$		bert-base-japanese-v2[17]	$Bert_{Base}$
ElectraIzumi		electra-base-japanese-discriminator[22]	$Electra_{Base}$
ElectraUD		transformers-ud-japanese-electra-base-discriminator[23]	

4.5. EXT 法学習データによる言語モデルの比較

検証用データを使い、各言語モデルの精度を検証した。試行ケースを表 8 に示す。ネットワークには、$Bert_{Large}$ に加えてパラメータ数の少ない $Bert_{Base}$ と、Bert と同じサイズでも精度が高いと言われる Electra[21]を試した。学習データを 元の質問文単位で 5 分割し、学習:開発:テスト=3:1:1 に割り当てて CV で乱数初期値や学習率などで 50 回のパラメータチューニングを行い、図 4 に示すように開発セットの平均ロスが最小となるパラメータの時の重みを使って検証精度を比較した。各ケースの検証精度の平均とアンサンブル（多数決）および、微調整時のテスト精度(mean test)を図 5 に示す。$Arch_{Finetune}$ は平均検証精度では $Bert_{Large}$ にわずかに及ばないが、アンサンブルでは若干改善された。Arch は 10 エポックでは平均検証精度で前述の 2 ケースに及ばないが、20, 30 エポックと学習が進むと精度が改善され、特にアンサンブルでは $Bert_{Large}$ に比べて 4%以上改善した。Electra は Bert より精度が高いとされているが、いずれも $Bert_{Base}$ より検証精度が低かった。表 9 にタイプ別 accuracy を示す。置換トークン数が増えるほど accuracy は高く問題としては易しくなることがわかる。また、$Arch_{30}$ は $Bert_{Large}$ より T2 で 10%以上と N2 での 2 倍改善しており、専門用語での精度が向上したこと、8%改善している $Arch_{fine\ tune}$ との比較から

コーパスの影響が大きいことがわかる。N2,N4 の $Arch_{30}$ と $Arch_{fine\ tune}$ の比較で、両者の差が大きいことから、トークン化の影響はむしろ非専門用語置換に影響していることが分かる。専門用語と非専門用語を混合したタイプはいずれも 50%程度と精度が悪く、特に T1N1 では非専門用語に惑わされて accuracy が 50%を割っている。語順を変更していないもの(nRep)と変更したもの(Rep)では、むしろ変更したものの方の精度が高い結果となり、$Arch_{30}$ ではその傾向が強まっている。当該言語モデルにとり、変更した語順の方が自然だったということになる。

表 9　タイプ別 Accuracy（EXT）

case	N2	N4	T2	T4	T1N1	T2N2	nRep	Rep
$Bert_{Large}$	0.72	0.81	0.70	0.89	0.43	0.56	0.74	0.75
$Arch_{Finetune}$	0.70	0.79	0.78	0.90	0.40	0.59	0.74	0.76
$Arch_{30}$	0.77	0.85	0.81	0.90	0.42	0.57	0.77	0.84

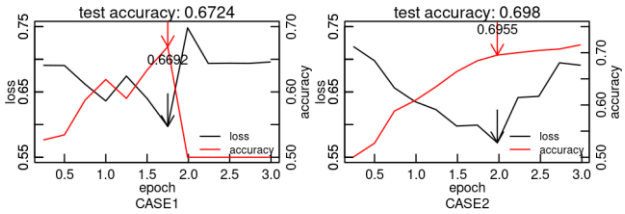


図 4　学習結果例（$Arch_{30}$）

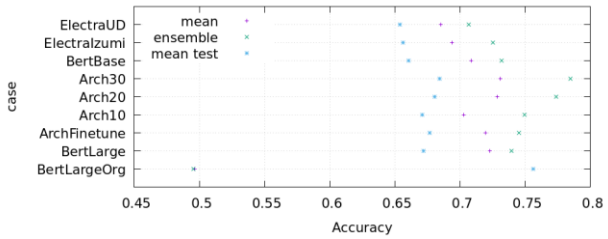


図 5　test / 検証 accuracy の比較

5. おわりに

技術文書から学習データを自動的に生成する方法としてテンプレートによる方法と言語モデルによる方法を提案し、限定的ではあるものの未知のデータに対して7～8割の精度があることを確認した。また、建築用言語モデルを作成し、その有効性を確認した。言語モデルの学習に使うコーパスの質の良さが注目されている[24]が、その点では今回使用した建築用コーパスは質が良いと思われる。しかし、実際にはコーパスで微調整した $Arch_{Finetune}$ は既存言語モデルからそれほど改善されておらず、トークンへの建築用語の追加が重要であることを確認した。また、$Bert_{Large}$ のケースで使用した言語モデルは100エポックの学習をしているのに対して Arch は30エポックでその精度を超えている。言語モデルは学習が進むにしたがって精度が向上することが知られており、更なる改善が期待できる。更に、学習に用いたものとは性格の異なるデータを検証に使い、自由な質問への回答に向けては、より多様な学習データが必要であることを確認した。

[参考文献]

1) 恒川裕史：技術文書からの知識抽出と自然言語学習データの自動生成に関する基礎的検討，日本建築学会大会梗概集, 2021
2) 恒川裕史：技術文書からの知識抽出と自然言語学習データの自動生成に関する基礎的検討その 2 : 自由質問に向けた試み，日本建築学会大会梗概集, 2022
3) 恒川裕史：技術文書からの知識抽出と自然言語学習データの自動生成に関する基礎的検討その 3 : 建築用言語モデル，日本建築学会大会梗概集, 2023
4) Ashish Vaswani et al.:Attention Is All You Need, Advances in Neural Information Processing Systems(NIPS2017), 30, 2017
5) Ashish Jaiswal et al.: A Survey on Contrastive Self-Supervised Learning, arXiv:2011.00362, 2020
6) Tom B. Brown et al.: Language Models are Few-Shot Learners, arXiv:2005.14165, 2020
7) Jacob Devlin et al.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, Human Language Tech., Vol.1, 2019
8) 湯本紘彰ほか:出現頻度と連接頻度に基づく専門用語抽出, 第 145 回自然言語処理研究会，情報処理学会, 2015
9) 工藤 拓，松本 裕治: チャンキングの段階適用による係り受け解析，情報処理学会論文誌, 2002
10) http://compling.hss.ntu.edu.sg/wnja/, (accessed 2022-03-18)
11) 伝康晴ほか：「コーパス日本語学のための言語資源：形態素解析用電子化辞書の開発とその応用」,日本語科学, Vol.22, pp.101-123, 2007
12) 荻原亜彩美ほか:『分類語彙表』に対する反対語情報付与, 言語処理学会第 25 回年次大会発表論文集, 2019
13) Tianqi Chen et al.:XGBoost: A Scalable Tree Boosting System, Proc. of Int. Conf. on KDDM, pp.785-794,2016
14) Thomas Wolf et al.:Transformers: State-of-the-Art Natural Language Processing, EMNLP: System Demonstrations, 2020
15) 社会基盤情報標準化委員会: 建設情報標準分類体系（JCCS）Ver.2.0 の公開について, JACIC, https://www.jacic.or.jp/hyojun/jccs-ver2r.html, (accessed 2020-9-18)
16) 上野達弘:情報解析と著作権,人工知能,36 巻 6 号,2021
17) https://github.com/cl-tohoku/bert-japanese, (accessed 2023-03-02)
18) Mor Geva et al.: AreWe Modeling the Task or the Annotator? An Investigation of Annotator Bias in Natural Language Understanding Datasets. EMNLP-IJCNLP, 2019
19) Takuya Akiba et al.: Optuna: A Next-generation Hyperparameter Optimization Framework. In KDD.2019
20) 松田寛：GiNZA - Universal Dependencies による実用的日本語解析，自然言語処理, Vol.27, No.3, pp.695-701, 2020
21) Kevin Clark et al.:ELECTRA: Pre-Training Text Encoders as Discriminators Rather Than Generators, ICLR 2020
22) Masahiro Suzuki et al.:Construction and Validation of a Pre-Trained Language Model Using Financial Documents, Proc. of SIG-FIN27, 2021
23) https://huggingface.co/megagonlabs/transformers-ud-japanese-electra-base-discriminator(accessed 2023-03-02)
24) Suriya Gunasekar et al.: Textbooks Are All You Need, arXiv:2306.11644, 2023