高速な契約書レビューのための計算量の削減

小林 尚輝 [1]　真鍋 陽俊 [2]　小田 悠介 [2,3]
[1] 東京工業大学　[2] LegalForce Research
[3] 東北大学 データ駆動科学・AI 教育研究センター
kobayasi@lr.pi.titech.ac.jp
{hitoshi.manabe, yusuke.oda}@legalforce.co.jp

概要

契約書には様々な規定や条項が記述されている．関心のある条項を自動で抽出することによって，これまで人手で実施されてきたレビューの業務コストが削減できる．それらの関心のある条項を契約書から抽出する情報抽出タスクとして契約書レビューが取り組まれている．従来研究では，抽出する内容を質問文とした BERT を用いた質問応答 (QA) モデルが広く使用されている．しかし，契約書のような長い文書を扱うための sliding window による小さな単位のテキスト（以下，セグメント）への分割と，質問文とセグメントを結合して入力事例とする BERT の構造から，従来の QA モデルでは BERT による処理回数が膨大となり非効率である．提案手法は，軽量な分類器により各事例が抽出すべきテキストを含むかを事前に識別することで，性能をほとんど劣化させることなく，BERT により処理される事例の数を 44%削減した．

1　はじめに

契約書は取引の内容を証するために作成され，互いの権利と義務を明確にする重要な役割を持つ．契約締結リスクとなりうる規定や条項を契約書に記載しないために，専門家の手による契約書レビューは非常に重要な手続きとなる．しかし，数十ページにわたる契約書を専門家の手を借りてレビューするには高いコストを要する．そこで，自然言語処理技術を用いた契約書レビューの自動化によるコストの削減が期待されている．

このような背景から，契約書を対象とした研究が幅広く行われており，契約書レビューは契約書を入力として契約書に含まれる関心のある条項を抽出する情報抽出タスクとして研究されている．また，関心のある条項は契約書の種類によって異なるため，

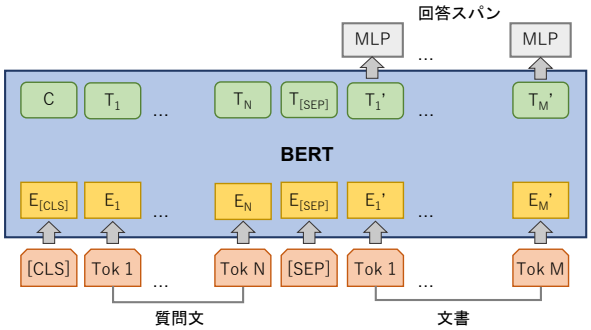

図 1　BERT を用いた質問応答モデル

様々なデータセットとそれに応じた抽出項目が提案されている．取り分け，CUAD [1] は構成する契約書数が多く，アノテーションに多くの時間と費用を費やしている．CUAD は契約書に対して，41 種類の抽出項目（例えば，当事者名を表す Parties や契約締結日を表す Agreement Date など）と各抽出項目に該当する契約書中のテキストスパンが付与されている．その他にも，秘密保持契約のみを対象とした ContractNLI [2] がある．ContractNLI は契約書に対して 17 通りの仮説が与えられ，それぞれの仮説が契約書中でどのように言及されたかを 示す 3 つのラベル（含意する/矛盾する/言及されない）と，そのラベルの根拠となる契約書中の文が付与されている．

従来研究 [1, 2] では BERT [3] を用いた質問応答 (QA) モデルにより契約書に対する情報抽出を行っている．BERT は大規模なテキストから事前学習された言語モデルで，図 1 に示されるように [SEP] トークンで結合された質問文と文書のペアを入力事例（以下，事例）とし，回答となるテキストスパンの始点/終点を推定するように finetune することで質問応答 (QA) タスクで高い性能を達成している．ここで，契約書を入力文書とし，抽出したい内容に応じた質問文と合わせて入力することで，BERT による契約書を対象とした情報抽出が可能となる．

このとき，BERT が扱える最大長を超える契約書

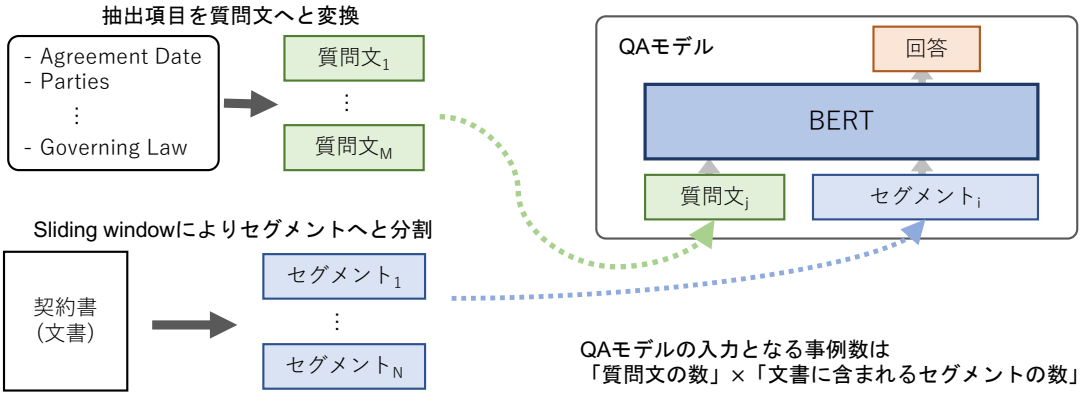

図 2　QA モデルを用いた契約書レビューの手続き

は sliding window を用いてより小さいセグメントへと分割されて処理される．BERT を用いた QA モデルによる契約書レビューは，数十ページに及ぶ契約書全体から質問文に該当するテキストスパンを抽出する．そのため，sliding window により分割された全セグメントに質問の数だけ BERT を適用する必要があり大幅に時間がかかる．しかし，抽出すべきテキストスパンは各質問に対して一つであるため，多くの事例は回答スパンを含まないという傾向がある．実際に，CUAD における 99.987%の事例は回答スパンを内包しない．

そこで，本稿は QA モデルによる契約書レビューを高速に行うために事例の削減を行う．具体的には，各事例に対し，高速な分類器を用いて事例のセグメントが回答を内包するかどうかの二値分類を行い，分類器により回答を内包すると判別された事例のみを QA モデルの入力とすることで BERT により処理される事例を削減する．

また，BERT の高速化手法として，重みパラメータの削減によるモデルサイズの圧縮 [4] が提案されている．これらの手法と提案法はモデルサイズと処理するデータ量の異なるボトルネックへのアプローチであり，速度改善において競合しない．

実験では，CUAD をデータセットとして用いて，テキストスパンの抽出性能をほとんど損なうことなく 44%の事例を削減した．

2　質問応答モデル

本節では BERT を用いた質問応答 (QA) モデルの説明を行う．図 1 に示されるように QA モデルは，トークン列により表現された質問文と文書を [SEP] トークンで結合して BERT の入力とし，文書側の各トークンに適用された MLP により回答スパンの先頭/末尾のスコアを推定する．このとき BERT の扱える最大長を超える文書は sliding window を用いてセグメントに分解され，各セグメントごとに質問文と結合されて入力される．

	事例数	回答を含む事例数 (割合)
CUAD	2529.8	34.1　(0.01)
SQuAD	341.8	147.4　(0.43)

表 1　CUAD と SQuAD による 1 文書あたりの事例数および回答を含む事例数とその割合の比較

3　計算量

本節では 1 契約書あたりの BERT の処理回数から契約書レビューにおける QA モデルの計算量を考える．図 2 に示されるように，契約書レビューにおける抽出項目に対応した M 通りの質問文と契約書に含まれる N 個のセグメントが与えられ，それらの組み合わせが BERT に入力される．したがって，質問文とセグメントのペアである事例の数が BERT による処理回数となり，事例の数は $M \times N$ 通りとなる．

CUAD から開発データとして抽出した 80 文書を対象として，1 文書あたりの事例数，および，回答を含む事例数との割合を算出[1)]した結果を表 1 に示す．比較のために SQuAD [5] の開発データも同様に算出した．

表より，CUAD の事例数は SQuAD に比べて約 7.4 倍多い．つまり 1 文書あたりの BERT による処理回数が約 7.4 倍となる．また，CUAD は事例数が多い一方で，SQuAD に比べ回答スパンを内包する事例数およびその割合が極めて低く，大半の事例が回答を含まないことがわかる．したがって，全ての事例

1)　質問文は 64 トークン，sliding window はセグメントの大きさと stride 幅をそれぞれ 448，および 256 トークンとした．

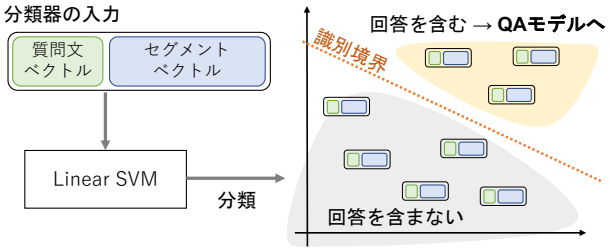

図 3　LinearSVM を用いた分類器

に一様に BERT を適用する従来の QA モデルによる契約書レビューは，その計算量の殆どを回答を含まない事例に費やすことになり非効率的である．

4　計算量を減らすための分類器

回答を含むセグメントを持つ事例はわずかであるため，事前に高速な分類器によって回答を含まないと判別できるならば，QA モデルに入力される膨大な事例数を減らすことができ，計算量の削減につながる．したがって，本節では各事例についてそのセグメントが質問に対する回答を内包するかを分類するための分類器を提案する．提案する分類器の概要を図 3 に示す．

分類器には計算速度の観点から BERT と比較して高速に動作する LinearSVM を使用する．QA モデルと一貫した入力形式を扱えるように，分類器の入力は QA モデルと同様にセグメントと質問文を用いる．セグメントは内包される uni-gram と bi-gram の TF-IDF ベクトル，質問文は Bag-of-Words ベクトルによりベクトル化して分類器の入力とする．

5　実験設定

5.1　データセット

分類器の学習，評価に用いるデータセットは CUAD[2] を使用した．CUAD は，学習データと評価データが配布されており，学習データを分割して開発データを用意した．このとき，文書サイズの偏りを軽減するために学習データを文書の長さによってソートしたのち 10 個のビンへと分割し，各ビンにおいてランダムに 80%を新しい学習データ，残りの 20%を開発データとして分割した．

5.2　評価尺度

本実験では QA モデルにより処理される事例を削減して計算量を減らすことが目的であるため，事例

2）https://github.com/TheAtticusProject/cuad

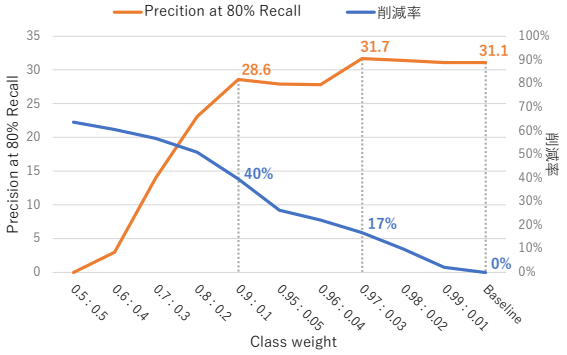

図 4　開発データを用いた class weight の変化における性能と削減率の比較

の削減率と QA モデルの性能の二つにより評価を行う．

削減率は全ての事例のうち，分類器によって回答を含まないと分類された事例の割合とする．

QA モデルの性能は予測したスパンと正解のスパンをもとに算出される．ここでは先行研究 [1] に従い，Precision at 80% Recall (P@R80) により QA モデルの性能を評価する．

5.3　ハイパーパラメータ

本実験の LinearSVM には scikit-learn [6] を利用し，クラスごとの正則化係数 C を変化させる class weight の値を開発データにより選択した．class weight は図 4 の横軸に示される 0.5:0.5 から 0.99:0.01 の 10 通りから探索した．class weight を変化させたときの開発データによる QA モデルの性能と削減率を図 4 示す．例えば class weight が 0.7:0.3 であれば，回答を含む，回答を含まないの各クラスに対応する正則化係数 C にそれぞれ 0.7 と 0.3 が掛け合わされる．したがって，グラフの右側に行くにつれて回答を含むと分類されやすくなる． グラフの右端は分類器を用いない（全ての事例に回答を含むと分類した場合と等しい）ベースラインである．

開発データの 結果から，性能が高い 0.97:0.03，および性能が高く削減率の良い 0.9:0.1 の 2 つの class weight を評価データでベースラインと比較することとした．

6　実験結果

表 2 に評価データによる実験結果を示す．性能と削減率がトレードオフである傾向は開発データと同様であり，class weight により削減率を制御できていることがわかる．特に，class weight=0.9:0.1 において性能が 1.8 ポイント減少するが，44%の事例を削

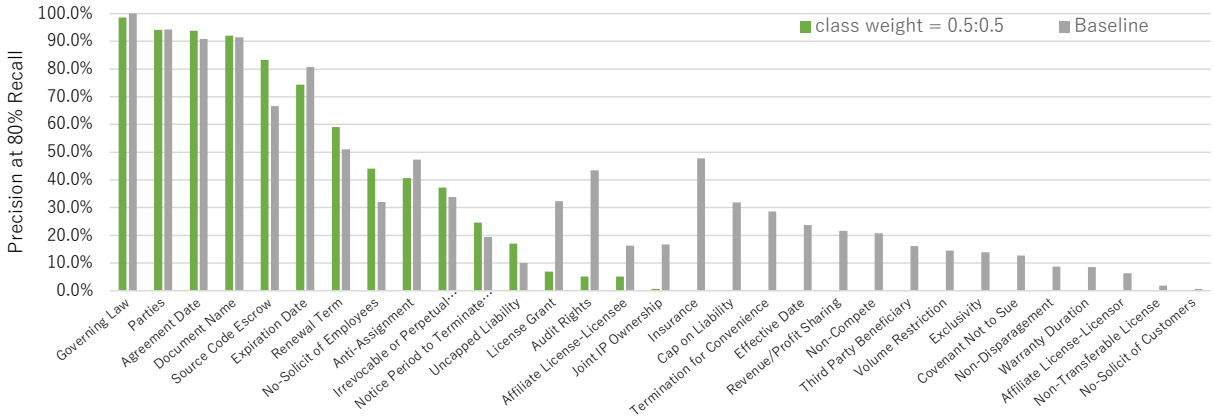

図 5　評価データによる class weight = 0.5 : 0.5 を対象としたラベルごとの性能の比較

	class weight	P@R80	削減率
ベースライン	-	33.4	-
+ 分類器	0.97 : 0.03	34.1	21.9%
	0.9 : 0.1	31.6	44.4%

表 2　評価データによる性能および削減率の比較

減できることを示した．また，class weight=0.97:0.03 において性能がわずかに向上する．これは事前に回答を含まない事例を削減することにより，QA モデルが候補として予測するスパンの質が向上したためと考えられる．

図 5 に評価データを用いたラベル毎の性能を示す．事例の削減による各ラベルの性能低下の影響を調べるために，削減率の高い class weight=0.5:0.5 の分類器をベースラインと比較した．CUAD は 41 種類のラベルがあるが，ベースラインにおける P@R80 の値が 0 ポイントのラベルはグラフから除いた．ラベルは class weight=0.5:0.5 の性能によってソートされており，ラベルによって性能が低下するラベルとそうでないラベルがあることがわかる．性能の劣化しなかった Governing Law や Agreement Date などのラベルは特定の表現（Governing Law であれば"governed by"，Agreement Date であれば月を表す英単語など）を高い確率でセグメントに含んでおり，N-gram を素性とした分類器で十分に分類可能であったと考えられる．一方で，性能の劣化したラベルは N-gram で捉えられない，例えば，言い換えなどの表現が含まれると考えられる．この結果を踏まえて，今後の改善として意味の近い単語を類似するベクトルによって表現できる単語分散表現 [7, 8] を素性として用いることが考えられる．

7　まとめ

本稿では，BERT を用いた QA モデルによる契約書レビューの高速化のために，BERT の入力事例を高速な分類器を用いて事前に分類することで QA モデルの計算量を削減する手法を提案した．CUAD を用いた実験では，分類器のパラメータである class weight によりトレードオフの関係にある削減率と性能を選択できることを示し，評価データにおいては 1.8 ポイントのわずかな性能の低下で 44%の事例が削減できることを示した．さらなる今後の改善として，分類器として用いた LinearSVM への入力となるセグメントや質問文のベクトルを単語分散表現に置き換えることで言い換えなどに頑健な分類が可能になると考えられる．

参考文献

[1] Dan Hendrycks, Collin Burns, Anya Chen, and Spencer Ball. Cuad: An expert-annotated nlp dataset for legal contract review. *NeurIPS*, 2021.

[2] Yuta Koreeda and Christopher Manning. ContractNLI: A dataset for document-level natural language inference for contracts. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pp. 1907–1919, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[3] 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.

[4] J. S. McCarley, Rishav Chakravarti, and Avirup Sil. Structured pruning of a bert-based question answering model. 2019.

[5] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 2383–2392, Austin, Texas, November 2016. Association for Computational Linguistics.

[6] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, Vol. 12, pp. 2825–2830, 2011.

[7] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, Vol. 26. Curran Associates, Inc., 2013.

[8] Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics.