歴史新聞データからのコーパス構築

田中 昂志† Chenhui Chu‡ 中島 悠太‡ 武村 紀子‡ 長原 一‡ 藤川 隆男*
† 大阪大学大学院情報科学研究科
‡ 大阪大学データビリティフロンティア機構
* 大阪大学大学院文学研究科
tanaka.koji@ist.osaka-u.ac.jp {chu, n-yuta, takemura, nagahara}@ids.osaka-u.ac.jp fuji@let.osaka-u.ac.jp

1 はじめに

大規模なテキストコーパスは自然言語処理に不可欠である．既存のコーパスは既に電子化されているテキストから作成されたものがほとんどである．例えば，構文解析のベンチマークである Penn Treebank[1] は電子化されている Wall Street Journal の新聞記事に対して品詞や構文情報を付与したものである．機械翻訳の評価型ワークショップ WMT で使われている対訳コーパス Europarl[2] は電子化されている欧州議会の多言語の議事録から対訳文をアライメントすることによって作成されたものである．

一方で，文学をはじめとする様々な分野では，研究対象となる資料の多くが紙などの物理媒体で保存されており，電子化されていない，もしくは物理媒体をスキャンしたのみで文字起こしなどによってテキスト化されていない．このような資料を電子化・テキスト化したうえで，特定のトピックの抽出などにより構造化することにより，自然言語処理技術を利用した種々の解析手法を適用可能となる．文学などの研究分野において，電子化・テキスト化・構造化などの処理は，元々の資料の価値を高めるものである．

本研究では歴史的新聞データベース Trove[*1]（オーストラリアの主要な日刊紙と地方新聞を網羅）を用いて，19 世紀から 20 世紀の約 120 年間にわたる特定のトピック public meeting に関して記述した記事のコーパスの作成方法を提案する．コーパス作成の手法は，始めに罫線を検出し，新聞の記事毎に画像をトリミングする．そして，トリミングした記事に対して Optical Character Recognition (OCR) を行い，特定のトピックの文字列を含む記事を抽出する．正解データを人手で作成し，評価を行った結果，特定の対象となる記事の始めの文と終わりの文の言語的特徴を用いた手法と比較して，過不足なく抽出できた記事の割合が 14.9% 向上した．提案手法は新聞データには特化しているが，年代や言語には非依存のため，Trove に限らず他の歴史的新聞のコーパス化にも対応できる．

[*1] https://trove.nla.gov.au

2 提案手法

提案手法の全体図を図 1 に示す．提案手法では，まず新聞画像内の罫線を検出し，トリミングすることで記事画像を抽出する．次に，抽出した全ての記事画像に対して OCR をすることで記事画像からテキストを抽出する．そして，検索文字列が含まれているか否かでフィルタリングを施すことで対象となる記事を抽出する．

2.1 トリミング

新聞画像内の罫線の検出及びトリミングには OpenCV を使用する．まず大津の 2 値化法 [3] を用いて新聞画像を 2 値化する．大津の 2 値化法とは，グレースケールの画像を白黒の 2 値画像に変換する手法であり，画素数のヒストグラムから分離度が最大となる閾値を求める．そして，輪郭追跡処理アルゴリズム [4] により 2 値化された画像の輪郭を抽出する．輪郭追跡処理アルゴリズムとは，2 値化された画像における境界部分を求める手法であり，輪郭となる画素を反時計回りに逐次検出していくことで輪郭を検出する．閾値以上の高さと閾値以下の幅を持つ領域をその新聞画像におけるカラムと判定し，閾値以上の幅と閾値以下の高さを持つ領域をその新聞画像における記事区切りと判定する．閾値は人手でチューニングを行い，トリミング操作で得られた画像を記事画像とする．

記事には図 1 の青線に示すような小カラムが存在する場合があるため，それに対応するため以下のような

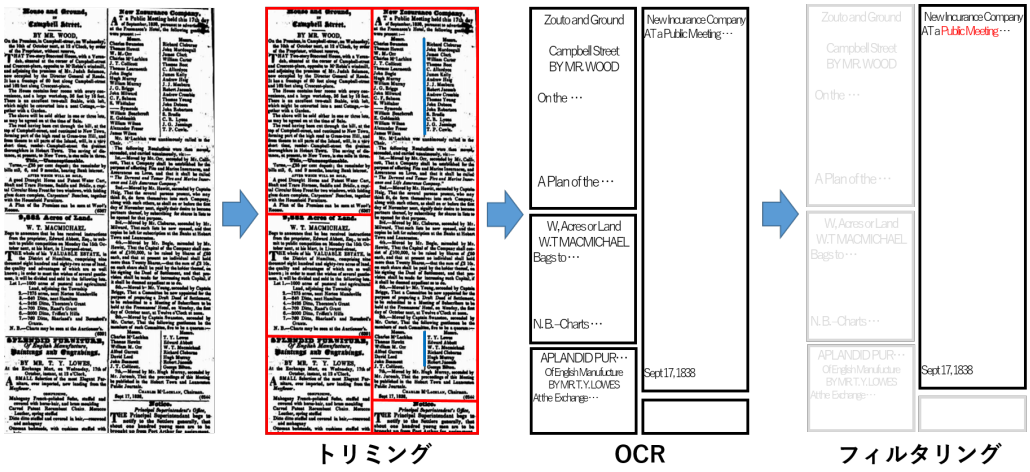

図 1: 提案手法の全体図

手法で縦分割するカラムを決定する．まず x 座標（水平方向）の値によりカラムをクラスタリングする．クラスタ内の最小の y 座標（垂直方向）と最大の y 座標が新聞画像全体の座標と比較して閾値以上の差であるクラスタは小カラムと判定し，分割に使用しないものとする．

2.2 OCR

OCR は一般的に文字の区切りの認識，大きさの正規化，特徴抽出，分類という手順で行われる．Google はオープンソースの OCR 手法として Tesseract[5] を公開しており，ニュース記事に対して，文字単位では 98.4%，単語単位では 97.4% の精度を達成している．

Tesseract の OCR を試したが，Google Drive の方が精度がよかったため本研究では記事画像からテキストを抽出するために，Google Drive[*2] の OCR 機能を使用する．

2.3 フィルタリング

記事画像の OCR 結果に検索文字列が存在するか否かでフィルタリングを行い，対象となる記事を抽出する．OCR による文字認識の誤りを許容するため，文字単位での類似度を条件とする．類似度の計算には Python の difflib モジュールの SequenceMatcher[*3]を使用する．SequenceMatcher は以下のように文字列間の類似度を計算する．

$$Similarity = \frac{2.0 * M}{T} \quad (1)$$

M は一致する文字数，T は比較する文字列の合計文字数を示す．

検索文字列の単語数に応じた単語 N-gram を記事内のテキストから得る．得られた N-gram と検索文字列との類似度を計算し，類似度の最大値が閾値以上の記事を対象記事とする．閾値は開発用データでの評価で最も F 値が高い閾値を用いる．

3 評価実験

3.1 使用データ

Trove からクロールした新聞画像データを用い，対象の記事は「public meeting」を含む記事とする．Trove から「public meeting」を含む新聞画像データを検索し，新聞 ID を得る．そして，Trove が提供している API を用いて，指定した新聞 ID の PDF データを取得する．OpenCV では PDF ファイルを扱えないので ImageMagick[*4]を用いて新聞の PDF データを PNG データに変換する．

本実験では，1838 年に刊行された「public meeting」を含む新聞 321 件を開発用 160 件，評価用 161 件にランダムに分割して使用する．対象記事の正解データは人手で抽出したものを用いる．図 2 に正解データの行数の分布を示す．

3.2 比較手法

ベースラインとして，画像による記事ごとのトリミングを行わず，Trove の OCR 結果のテキストの特徴量から記事の特定を行う手法を用いる．ベースラインの手法では，対象記事の初めの文と終わりの文の特徴をそれぞれ検出し，記事の特定を行う．ベースラインで用いる文の特徴は以下の通りである．

始めの文

「public meeting」を含む文から前 2 文を取得．

[*2] https://www.google.com/intl/ja_ALL/drive/
[*3] https://docs.python.jp/3/library/difflib.html
[*4] https://www.imagemagick.org/

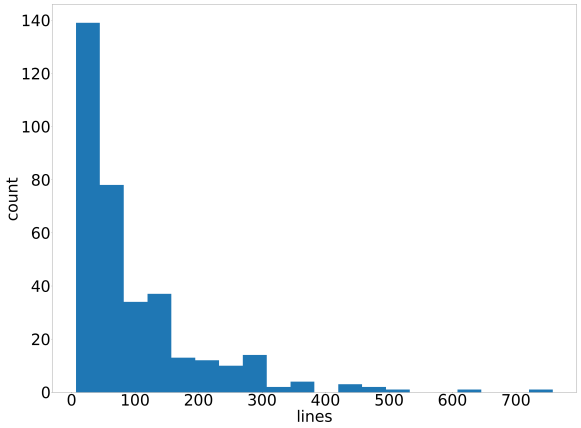

図 2: 正解データの行数の分布

終わりの文

Stanford parser[*5]を用いて固有表現抽出を行う．固有表現タグ（LOCATION, DATE, PERSON）を含む文かつ次の文が固有表現タグを含まない文を取得．

3.3 パラメータチューニング

2.1 節で言及したトリミングの際の罫線判定及び小カラム判定の閾値は，1 カ月の新聞データに対して実験的にチューニングを行う．また 2.3 節で言及したフィルタリングの際に開発用データで F 値が最も高くなる閾値を用いる．閾値を 0 から 1 まで 0.05 区切りで変化させ実験した結果，閾値 0.8 のときに F 値が最大となったため，フィルタリングの閾値は 0.8 とする．

3.4 評価手法

実験では，対象記事が抽出できているかを評価する記事レベルの評価と，記事の抽出の精度を評価する行レベルの評価を行う．それぞれの評価の方法は以下の通りである．

記事レベルの評価手法

抽出された記事内の「public meeting」を含む文と対象記事内の「public meeting」を含む文の類似度を計算し，類似度が閾値以上の場合抽出成功，閾値以下の場合抽出失敗とする．閾値は文として類似している実験的な値として 0.6 を用いる．ベースライン，提案手法それぞれに対して評価し，Precision, Recall, F1-score を計算する．

行レベルの評価手法

抽出された記事と対象記事の初めの行と終わりの行それぞれに対して調査を行う．抽出された記事が対象記事と比較して余剰・不足している行の割合を計算する．

[*5]https://nlp.stanford.edu/software/lex-parser.shtml

表 1: 記事レベルの評価

手法	Precision	Recall	F1-score
ベースライン	90.3	81.4	85.6
提案手法	74.9	77.6	76.2

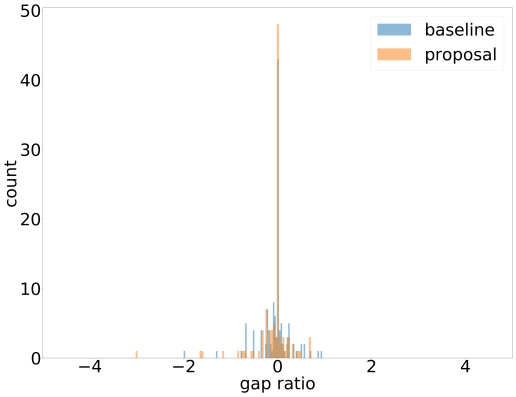

図 3: 行レベル評価（始めの行）

3.5 結果

記事レベルの評価

表 1 に記事レベルの評価結果を示す．提案手法よりベースラインの方が高い F 値を示した．ベースラインは記事の始めの行を取得するとき「public meeting」が含まれている文という特徴を使用しているため，記事レベルの評価の基準となる文を必ず含むように抽出していることが原因であると考えられる．ここで，ベースラインが抽出に失敗している記事が存在する理由として，1 つの新聞画像に複数の public meeting の記事が存在し，public meeting が OCR 誤りしている記事が含まれているためである．

行レベルの評価

図 3 に始めの行の評価結果，図 4 に終わりの行の評価結果を示す．横軸が余剰，不足している行の割合，縦軸が記事の数を表す．図 3, 4 より，記事の始め・終わりいずれの行においても，ベースラインと比較して提案手法の方が過不足なく抽出できている記事が多いことが分かる．また，始めの行と終わりの行共に過不足なく抽出できている記事の数は，ベースラインでは 5 件（3.1%），提案手法では 29 件（18.0%）であった．

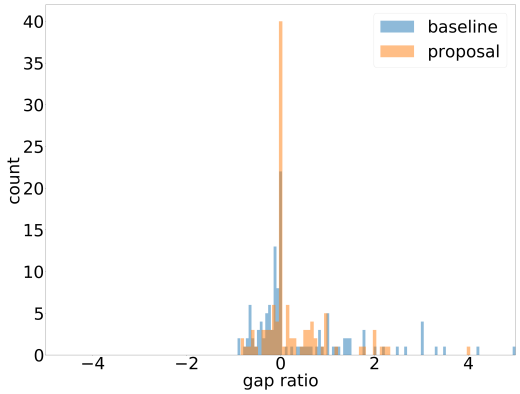

図 4: 行レベル評価（終わりの行）

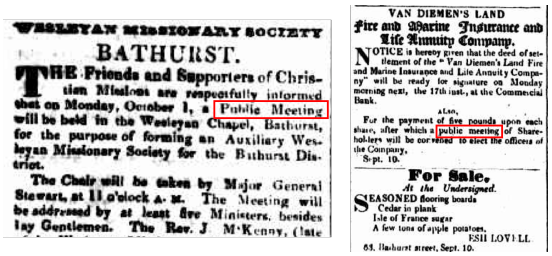

図 5: 記事の抽出結果例

よって，画像の特徴を用いて記事の区切りを検出する手法が，特定の記事を抽出する場合に有効であることが示せた．

3.6 考察

図 5 の左側は抽出に失敗した記事の例である．図 5 の左側の記事の赤線で囲まれている「public meeting」は，OCR を行った結果「Pohlle Meeling」と認識されていた．したがってフィルタリングの際に対象記事と判定されなかったため，抽出に失敗したと考えられる．これは，新聞画像の解像度が他の新聞画像と比較して低く，正しく OCR ができなかったことが原因であると考えられる．評価用データ中で，OCR 誤りによって抽出に失敗した例は 8.1% 存在した．

図 5 の右側は記事の抽出は成功しているが，余剰に抽出している例である．図 5 より，記事画像に対象記事が含まれているが，対象でない記事も同時に抽出されていることが分かる．これは，罫線が途中で途切れているため区切りの判定に失敗し，不適切なトリミングを行ってしまったことが原因であると考えられる．評価用データ中で，このように余剰に記事を抽出してしまう例は 26.7% 存在した．

抽出した記事が正解と比較して不足しているものもあった．原因として，記事の区切りではないが段落を区切っている小区切りが記事の区切りと判定された例が 3.1% 存在した．また，提案手法では，複数カラムに跨る記事を考慮していないため，複数カラムに跨る記事は別々にトリミングしてしまう．このような例は 1.9% 存在した．

本実験では取得できる新聞データの内，最も古い 1838 年の新聞を用いて実験を行ったため，印刷技術の向上などの要因を考慮すると，最も新しい 1954 年の新聞と比較して罫線の検出や OCR の精度が低くなると考えられる．よって，1838 年から 1954 年までのデータを用いて実験することを今後の課題とする．

4 まとめ

本研究では，歴史的新聞から特定のトピックの記事を抽出し，コーパスを作成することを目的とし，新聞の罫線によるトリミング，OCR 結果によるフィルタリングを用いた手法を提案した．Trove から取得した 1838 年の新聞データを用いて評価実験を行った結果，評価用データの 77.6% の記事から対象となる記事を抽出し，さらに 18.0% の記事が過不足なく抽出可能であることを示した．今後の課題として，抽出した記事を解析するために，OCR 誤り訂正，及び public meeting の開催日時や場所などの情報抽出に取り組む予定である．

参考文献

[1] Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*, 19(2):313–330, Jun. 1993.

[2] Philipp Koehn. Europarl: A Parallel Corpus for Statistical Machine Translation. In *Proc. of Machine Translation Summit*, pages 79–86, 2005.

[3] 展之 大津. 判別および最小 2 乗規準に基づく自動しきい値選定法. **電子通信学会論文誌** *D*, 63(4):349–356, Apr. 1980.

[4] Satoshi Suzuki and Keiichi Abe. Topological structural analysis of digitized binary images by border following. *Computer Vision, Graphics, and Image Processing*, 30(1):32–46, 1985.

[5] R. Smith. An overview of the tesseract ocr engine. In *Proc. of International Conference on Document Analysis and Recognition*, volume 2, pages 629–633, Sep. 2007.