# テキストセグメンテーションモデルのドメイン汎化性向上に向けたデータ拡張手法

久島務嗣 [1]　小林優佳 [1]　吉田尚水 [1]
永江尚義 [1]　岩田憲治 [1]
[1] 株式会社　東芝　研究開発センター
tsuyoshi2.kushima@toshiba.co.jp

## 概要

近年では，大規模コーパスと深層学習を用いた高性能なテキストセグメンテーションモデルが提案されている．しかし，既存手法は学習コーパスと異なるドメインのテキストでは顕著に性能が低下する．コーパスによって使われる名詞が大きく異なり，かつ，名詞に依存したモデルが学習されていることが，このドメイン汎化性低下の原因である可能性がある．そこで，本稿では名詞置換によるデータ拡張によって，ドメイン汎化性能の改善を試みる．実験の結果，ドメイン外の評価コーパスにおいて，名詞置換によって平均で F 値で 6.46，Pk スコアで 1.89 それぞれ改善し，名詞置換の効果を確認した．

## 1　はじめに

書籍や文書といった人によって作られたテキストは，章立てや段落などの構造を持つ．この構造情報は，高度なテキスト分析において有用である．しかし，自動化技術の発達によって，音声認識による音声書き起こしといった，構造を持たないテキストが大量に存在するようになった．構造を持たないテキストの分析は容易でないため，高性能なテキスト構造化技術が求められている．テキスト構造化技術の 1 つにテキストセグメンテーションがある．これは，構造を持たないテキストを意味的関連性がある複数の文のまとまり (セグメント) に分割し，構造化する技術である．近年では，大規模コーパスと深層学習を用いた手法が提案され [1, 2, 3, 4]，高性能な構造化が可能となってきている．

しかし，既存手法で高い性能が発揮できるテキストは学習データと似たテキストに限られる．Wikipedia や教科書といった，コーパスのテキストの種類である書式とトピックをドメインとすると，学習データとドメインが異なるテキストでは顕著に性能が低下する [3]．そのため，汎用的なモデルの実現には，ドメイン汎化性の向上が課題である．

このドメイン汎化性が低い原因として，モデルが学習データの名詞に過学習していることが考えられる．[3] で使われたコーパスを分析した結果，名詞は語彙の過半数を占め，学習データの語彙と評価コーパスの語彙との一致率が他の品詞に比べて低いことが分かった．したがって，学習データで高い性能を発揮するには，名詞に依存したモデルを学習する必要があり，さらに，名詞はコーパスによって大きく分布が異なるため，ドメイン汎化性が低下していると考えられる．よって，学習データのテキスト中の名詞を他の名詞に置換することで，特定の名詞に依存しない推定が可能になり，ドメイン汎化性能が向上すると期待される．

そこで，本稿では，名詞置換によって拡張したコーパスを使ってテキストセグメンテーションモデルを学習することで，モデルのドメイン汎化性を向上させる方法を提案する．また，置換対象とする品詞の比較実験によって，名詞置換は他の品詞に比べてモデルの汎化性向上に有効であることを示す．

まず，2 章で関連研究を挙げ，テキストセグメンテーションの課題を詳細に述べる．3 章で具体的なデータ拡張手法を説明し，4 章で実験の詳細，5 章で結果と考察をそれぞれ述べる．6 章では最後に本稿のまとめと今後の課題について述べる．

## 2　関連研究

深層学習を用いた既存研究は，Glavăs ら [1]，Lukasik ら [2]，Xing ら [3] の研究がある．これらの研究の学習・評価に用いられるコーパスを表 1 に示す．Section[5] と Wiki-727k[4] は，それぞれ学習データ・検証データ・評価データを含むコーパスである．

**表 1** コーパス統計量

| | 学習コーパス | | 評価コーパス | | | |
|---|---|---|---|---|---|---|
| | Section[5] | Wiki-727k[4] | Cities[6] | Elements[6] | Wiki-50[4] | Clinical[7] |
| トピック | 都市，疾患 | - | 都市 | 化学元素 | - | 医学 |
| 書式 | Wiki | Wiki | Wiki | Wiki | Wiki | 教科書 |
| 文書数 | 21,144 | 727,746 | 100 | 116 | 50 | 227 |
| 文書長 | 53.1 | 48.0 | 62.8 | 22.7 | 56.5 | 140.4 |
| セグメント長 | 8.0 | 8.2 | 5.1 | 3.2 | 8.2 | 34.8 |
| セグメント数 | 6.5 | 6.7 | 12.2 | 6.8 | 6.7 | 4.0 |

**表 2** 品詞別 語彙占有率

| 品詞 | Cities | Elements | Wiki50 | Clinical | ave |
|---|---|---|---|---|---|
| 名詞 | 0.69 | 0.60 | 0.60 | 0.53 | 0.60 |
| 動詞 | 0.09 | 0.14 | 0.17 | 0.18 | 0.14 |
| 形容詞 | 0.08 | 0.13 | 0.11 | 0.22 | 0.14 |
| 副詞 | 0.02 | 0.03 | 0.04 | 0.04 | 0.03 |
| 合計 | 0.88 | 0.90 | 0.92 | 0.97 | 0.91 |

**表 3** 品詞別 語彙一致率

| 品詞 | Cities | Elements | Wiki50 | Clinical | ave |
|---|---|---|---|---|---|
| 名詞 | 0.30 | 0.47 | 0.86 | 0.79 | 0.62 |
| 動詞 | 0.94 | 0.89 | 0.95 | 0.86 | 0.90 |
| 形容詞 | 0.66 | 0.67 | 0.92 | 0.76 | 0.77 |
| 副詞 | 0.90 | 0.90 | 0.93 | 0.83 | 0.89 |
| 全体 | 0.44 | 0.60 | 0.90 | 0.82 | 0.68 |

Glavăs らは，Wiki-727k を学習に用い，複数の評価コーパスで評価を行っている．モデルの汎化性評価には，学習データと異なるドメインのコーパスで評価する必要がある．しかし，Glavăs らは，Wiki 書式のコーパスと疑似的に生成されたコーパスでしか評価を行っておらず，モデルの汎化性の評価は十分でない．Lukasik らも同様に，十分なドメイン汎化性評価を行っていない．

一方で，Xing らは，Section を学習に用い，表 1 の評価コーパスで評価を行っている．学習データとドメインが異なる Elements・Wiki-50・Clinical によってモデルの汎化性を評価しているが，評価データと 3 つの評価コーパスの性能に乖離がある．

以上の様に，深層学習によってテキストセグメンテーションモデルを学習する既存研究では，モデルの汎化性評価が行われているものが少なく，行われていても，評価データの性能とドメインの異なるコーパスの性能に大きな乖離がある．様々なテキストに汎用的に利用できるテキストセグメンテーションモデルの開発には，モデルの汎化性を向上し，この乖離を埋めることが重要である．

## 3 提案手法

表 1 に示した評価コーパスの，品詞別の語彙占有率を表 2 に，Section の学習データと各評価コーパスとの品詞別単語一致率を表 3 に，それぞれ示す．表 2，表 3 には，語彙に占める割合の多い品詞上位 4 種のみを示している（詳細は付録参照のこと）．いずれのコーパスにおいても，名詞が語彙の過半数を占めており，学習データの語彙との一致率は名詞が最も低い．したがって，学習データで高い性能を発揮するには，名詞に基づいた学習が必要となり，さらに，他の品詞に比べて，名詞はコーパスによって分布が大きく異なるため，ドメイン汎化性の低下の原因になっている可能性がある．そこで，学習データ中の名詞を置換するデータ拡張によって，名詞に過学習せず，未知の名詞に頑健なモデルが学習され，モデルのドメイン汎化性が向上すると考える．

本稿で提案する名詞置換によるデータ拡張の例を表 4 に示す．表 4 の **original** には置換前のテキストを，**single** と **rand** は 2 つの置換方法の例を，色付き単語は抽出された名詞を，それぞれ表す．提案手法では，テキストから名詞を抽出し，抽出された名詞を学習データ内の他の名詞に置換する．

名詞の置換方法には，表 4 に示した **rand** と **single** の 2 つの置換方法を検討した．**single** は，置換先と置換元の名詞を 1 対 1 で対応付け，その対応に従って置換を行う方法である．一方，**rand** は 1 対 1 の対応付けをせず，名詞が出現する度にランダムに他の名詞に置換する方法である．したがって，**rand** は同じ名詞であっても異なる名詞に置換され得る．表 4 に示す例では，**original** には「Aleppo」という名詞が 2 回現れている．**single** では，「Aleppo」に「Gittinger」が対応付いているため，2 回とも「Gittinger」に置換されている．一方，**rand** では，同じ名詞であっても毎回異なる名詞に置換され得るため，1 回目は「Gittinger」に，2 回目は「Agustina」に置換されている．なお，「Gittinger」と「Agustina」は学習データから抽出された名詞である．

また，複数の単語から成る名詞も存在するが，複数語の名詞を置換の単位にすると，**single** が適用できないため，置換の単位は 1 つの単語とした．

表 4　名詞置換サンプル (色付き単語が名詞)

| |
|---|
| **original** |
| Aleppo has scarcely been touched by archaeologists since the modern city occupies its ancient site. |
| Aleppo appears in historical records as an important city much earlier than Damascus. |
| **single** |
| Gittinger has scarcely been touched by Espares since the modern hypertrophy occupies its ancient Wasserviertel. |
| Gittinger appears in historical inhabitants as an important hypertrophy much earlier than Ariqueanos. |
| **rand** |
| Gittinger has scarcely been touched by Espares since the modern hypertrophy occupies its ancient Wasserviertel. |
| Agustina appears in historical inhabitants as an important Biodiversität much earlier than Ariqueanos. |

# 4　実験

## 4.1　設定

実験に用いるモデルの概要図を図 1 に示す．図 1 中の $\mathbf{S_i}$ は文，$\mathbf{t_j}$ は文 $\mathbf{S_i}$ を構成するサブワード，$\mathbf{f_i}$ は文 $\mathbf{S_i}$ の特徴量，$\mathbf{p_i}$ は文 $\mathbf{S_i}$ がセグメント終端である確率，をそれぞれ表す．本稿では，文エンコード用 ( 前段 ) とセグメント終端推定用 ( 後段 ) に 2 つの事前学習済み BERT を用いる，Hier.BERT[2] を使用した．事前学習済み BERT は，Python のオープンソースライブラリである transformers[1] を使用した．

名詞置換は，エポック開始時に毎回実施した．また，名詞置換は 1/2 の確率で行われ，置換するか否かの判定はテキスト単位で行った．単語の抽出・品詞の特定には，Python のオープンソースライブラリである spaCy[2] を利用した．

名詞置換の有効性検証のため，動詞も対象として実験を行った．動詞は，語彙の占有率が名詞に次いで高く，名詞と同様に文の意味に大きくかかわるため，比較対象とした．よって，実験条件は置換方法 2 種（**single**・**rand**）と置換対象 2 種（名詞：**NOUN**・動詞：**VERB**）の組み合わせで，$\mathbf{NOUN_{single}}$・$\mathbf{NOUN_{rand}}$・$\mathbf{VERB_{single}}$・$\mathbf{VERB_{rand}}$ の 4 種となる．

名詞と同様に，動詞も置換の単位は 1 つの単語である．また，動詞は，活用を無視して置換すると，意味不明な文になったり，文法的に誤った文となってしまう．そのため，動詞の置換は，置換先の動詞を置換元の動詞に合わせて活用させて置換を行う．

ハイパパラメータの選定には Mosbach ら [8] を参考にして，学習率は [$2e^{-6}$, $5e^{-6}$, $1e^{-5}$, $2e^{-5}$] から，学習エポック数は [20, 30, 40, 50] から，それぞれ検証データでの F 値が最大になるものを選択した．また，Warmup も適用し，学習率を変化させるエポック数は，学習エポック数の [10%, 20%, 30%, 40%, 50%] から検証データでの F 値が最大になるものを選択した．なお，Warmup では学習率は線形に変化させた．

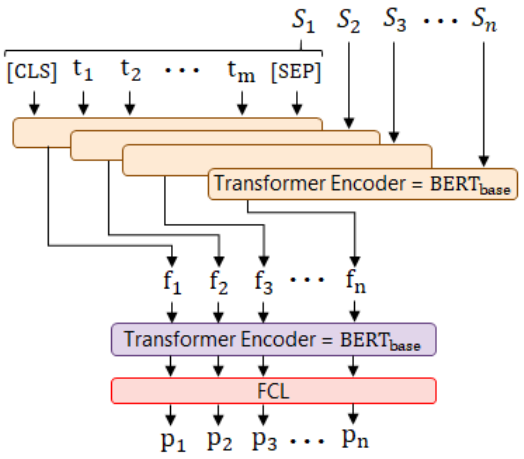


図 1　モデルの概要図

Lukasik ら [2] は，テキストの文数と文の文字数とを制限し，計算コストを下げた上で Hier.BERT の学習を行っている．本稿でも同様に，文字数を 64 文字，文数を 128 文にそれぞれ制限して学習を行った．文字数が 64 文字を超える場合には，文を分割し複数の文として前段の BERT に入力し，分割された文を後段の BERT に入力するときには，分割された文に同じ位置ラベルを与えて入力した．最後の全結合層に分割された文を入力するときには，分割された文の特徴量を平均した特徴量を入力した．文数が 128 文を超える場合には，テキストを分割し，複数の学習データとした．評価時は，窓幅 128 文の窓を 1 文ずつスライドさせ，各文の推定結果を平均することで，最終推定結果を取得した．

## 4.2　評価

学習には Section を使用し，評価コーパスには Section の評価データ・Cities・Elements・Wiki-50・Clinical の 5 つのコーパスを用いた．Section は Wiki 書式で都市と疾患のトピックを含むため，Elements・

1）https://github.com/huggingface/transformers
2）https://github.com/explosion/spaCy

表 5　評価結果 (F 値 /$P_k$スコア )

| | Section-train | Section-test | Cities | Elements | Wiki-50 | Clinical | OOD-Ave |
|---|---|---|---|---|---|---|---|
| Xing ら [3] | - / - | - / **9.7** | - / 16.1 | - / 39.4 | - / **26.8** | - / **30.5** | - / **32.2** |
| baseline | 95.25 / 3.29 | 73.32 / 17.81 | 66.70 / 21.67 | 49.87 / 37.92 | 36.94 / 36.82 | 27.14 / 48.83 | 37.98 / 41.19 |
| **VERB$_{single}$** | 91.14 / 6.09 | 67.96 / 20.04 | 63.25 / 24.16 | 48.25 / 38.21 | 34.70 / 37.56 | 24.36 / 50.93 | 35.77 / 42.23 |
| **VERB$_{rand}$** | 93.51 / 4.49 | 73.88 / 17.42 | 69.90 / 20.03 | 52.40 / 36.25 | 37.06 / 36.89 | 26.68 / 49.50 | 38.71 / 40.88 |
| **NOUN$_{single}$** | 87.55 / 8.19 | 75.89 / 16.39 | 75.39 / 16.09 | **61.55** / 30.34 | 42.29 / 34.05 | 29.58 / 51.37 | 44.47 / 38.59 |
| **NOUN$_{rand}$** | 87.74 / 8.05 | 75.84 / 16.36 | **75.52 / 15.72** | 61.38 / **29.80** | 41.79 / 34.07 | 30.07 / 50.17 | 44.41 / 38.01 |

Wiki-50・Clinical の 3 つのコーパスをドメイン外（OOD：Out-Of-Domain）として，ドメイン汎化性能の評価に用いた．Cities はドメイン内のコーパスとなるため評価の対象ではないが，Xing らとの比較のために評価に用いた．

評価指標には，F 値と $P_k$ スコア [9] を用いた．$P_k$ スコアは，窓幅 k 分の検出誤りを許す指標で，テキストセグメンテーションモデルの評価によく用いられる．$P_k$ スコアは，Python のオープンソースライブラリである segeval[3] を用いて計算した．

実験は，データ拡張を適用しない baseline と，**NOUN$_{single}$**・**NOUN$_{rand}$**・**VERB$_{single}$**・**VERB$_{rand}$** の 4 条件，合わせての 5 つの実験条件で行った．各実験条件に対して 5 つの seed で実験を行い，それぞれの結果の平均に対して評価を行った．

## 5　考察

実験の結果を表 5 に示す．表 5 中の OOD-Ave は OOD とした 3 つのコーパスの平均値を表す．各条件で最も高い数値を太字で示す．

baseline と 4 つの実験条件の OOD-Ave を比較すると，**VERB$_{single}$** 以外の条件で性能が改善した．置換対象とする品詞の違いによる改善幅の差を比較すると，**NOUN$_{single}$** と **NOUN$_{rand}$** を平均すると，F 値で 6.46，Pk スコアで 1.89 それぞれ改善したが，**VERB$_{single}$** と **VERB$_{rand}$** を平均すると，baseline に比べて性能が劣化した．よって，名詞を対象として置換するデータ拡張がドメイン汎化性の向上に有効であることが確認された．

名詞を置換対象とした条件の効果が高かった原因として，名詞への過学習が抑えられた可能性がある．**NOUN$_{rand}$** は単語が一貫性なく置換されるため，単語に依存しない推定を行う．したがって，**NOUN$_{rand}$** の学習データにおける F 値と Pk スコアを上回るためには，名詞に基づいた推定が必要となる．よって，名詞を置換対象とした条件以外では，**NOUN$_{rand}$** 相当の性能に達した後の学習では，名詞への過学習が起こっていると考えられる．**NOUN$_{single}$** と **NOUN$_{rand}$** の学習データにおける F 値と Pk スコアは，他の条件よりも劣化しているため，名詞を置換対象とした条件以外では名詞への過学習が起こっていたと考えられる．これが抑制されたため，名詞置換ではドメイン汎化性が向上したと考えられる．これは，名詞への過学習がドメイン汎化性低下の原因になっている仮説を支持する．

一般にテキストセグメンテーションにおいては，名詞の一貫性は重要な判断基準となり得る．そのため，**NOUN$_{rand}$** よりも，**NOUN$_{single}$** の方が改善幅が大きいことが期待される．しかし，置換方法による差が見られないという結果は，モデルが名詞の一貫性を考慮した推定をあまりしていないことを示唆する．よって，名詞の一貫性を考慮する機構等で，更にドメイン汎化性の改善ができる可能性がある．

また，個別のコーパスの結果を見ると，Cities と Elements は Xing らの結果を上回っているが，Clinical では，非常に低い結果となった．Clinical では，他のコーパスの傾向と異なり，recall よりも precision が低い結果となっており，セグメント終端の誤検出が多い（詳細は付録参照のこと）．これは，コーパスの書式によってセグメントの粒度が異なることに起因すると考えられる（表 1）．セグメント分割粒度を調整可能なモデルによって，Clinical においても適切なセグメンテーションができると考えられる．

## 6　おわりに

本稿では，ドメイン汎化性能の高いテキストセグメンテーションの実現に向け，名詞置換によるデータ拡張手法を提案した．実験の結果，名詞を置換するデータ拡張によって，平均して F 値で 6.46，Pk スコアで 1.89 それぞれ改善し，効果を確認した．評価コーパスの内，2 つのコーパスでは SOTA 手法を上回ったが，1 つのコーパスでは非常に低い結果となった．名詞の一貫性を考慮する手法や，セグメント粒度を調整可能な手法によって，更にドメイン汎化性能を向上することが今後の課題である．

3）　https://segeval.readthedocs.io/en/latest/

# 参考文献


[1] Goran Glavăs and Swapna Somasundaran. Two-level transformer and auxiliary coherence modeling for improved text segmentation. In **Proceedings of the 34th Association for the Advancement of Artificial Intelligence**, 2020.

[2] Michal Lukasik, Boris Dadachev, Kishore Papineni, and Gonçalo Simões. Text segmentation by cross segment attention. In **Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing**, pp. 4707—-4716, 2020.

[3] Linzi Xing, Brad Hackinenz, Giuseppe Carenini, and Francesco Trebbi. Improving context modeling in neural topic segmentation. In **Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing**, pp. 626––636, 2020.

[4] Omri Koshorek, Adir Cohen, Noam Mor, Michael Rotman, and Jonathan Berant. Text segmentation as a supervised learning task. In **Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics**, pp. 469—-473.

[5] Sebastian Arnold, Rudolf Schneider, Philippe Cudré-Mauroux, Felix A. Gers, and Alexander Löser. Sector: A neural model for coherent topic segmentation and classification. **Transactions of the Association for Computational Linguistics**, Vol. 7, pp. 169–184, 2019.

[6] Harr Chen, S.R.K. Branavan, Regina Barzilay, and David R. Karger. Global models of document structure using latent permutations. In **Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics**, pp. 371—379, 2009.

[7] Jacob Eisenstein and Regina Barzilay. Bayesian unsupervised topic segmentation. In **Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing**, pp. 334—-343, 2008.

[8] Marius Mosbach, Maksym Andriushchenko, and Dietrich Klakow. On the stability of fine-tuning bert: Misconceptions, explanations, and strong baselines. In **International Conference on Learning Representations**, 2021.

[9] Doug Beeferman, Adam Berger, and John Lafferty. Statistical models for text segmentation. **Machine Learning**, Vol. 34, No. 1, pp. 177—-210, 1999.

# A 参考情報

**表 A.1** 品詞別 語彙分析結果（語彙数/一致率）

| POS-TAG | Section-train | Section-val | Section-test | Cities | Elements | Wiki50 | Clinical |
|---|---|---|---|---|---|---|---|
| NOUN | 63,237 / - | 25,724 / 0.83 | 35,039 / 0.77 | 9,300 / 0.84 | 8,833 / 0.59 | 3,974 / 0.90 | 13,280 / 0.80 |
| PRON | 283 / - | 148 / 0.89 | 177 / 0.84 | 80 / 0.86 | 53 / 0.96 | 97 / 0.94 | 123 / 0.93 |
| PROPN | 199,211 / - | 52,212 / 0.60 | 84,877 / 0.54 | 22,366 / 0.07 | 2,971 / 0.13 | 2,934 / 0.81 | 2,452 / 0.74 |
| VERB | 15,960 / - | 8,229 / 0.90 | 10,383 / 0.86 | 3,976 / 0.94 | 2,676 / 0.89 | 1,972 / 0.95 | 5,499 / 0.86 |
| ADJ | 21,610 / - | 8,838 / 0.85 | 12,209 / 0.79 | 3,920 / 0.66 | 2,513 / 0.67 | 1,339 / 0.92 | 6,664 / 0.76 |
| ADV | 3,147 / - | 1,565 / 0.91 | 2,033 / 0.85 | 749 / 0.90 | 546 / 0.90 | 426 / 0.93 | 1,251 / 0.83 |
| ADP | 428 / - | 234 / 0.90 | 268 / 0.85 | 117 / 0.83 | 77 / 0.90 | 100 / 1.00 | 115 / 0.95 |
| DET | 116 / - | 74 / 0.89 | 82 / 0.83 | 35 / 0.91 | 25 / 0.96 | 40 / 1.00 | 50 / 0.98 |
| AUX | 461 / - | 158 / 0.61 | 278 / 0.44 | 112 / 0.45 | 33 / 0.91 | 87 / 0.58 | 75 / 0.83 |
| CCONJ | 87 / - | 44 / 0.89 | 57 / 0.77 | 23 / 0.74 | 18 / 0.83 | 18 / 1.00 | 34 / 0.79 |
| SCONJ | 136 / - | 90 / 0.93 | 105 / 0.91 | 55 / 0.93 | 38 / 1.00 | 53 / 1.00 | 77 / 0.97 |
| INTJ | 116 / - | 31 / 0.61 | 53 / 0.66 | 9 / 0.78 | 4 / 1.00 | 6 / 0.67 | 34 / 0.59 |
| NUM | 11,347 / - | 3,644 / 0.67 | 5,188 / 0.51 | 3,619 / 0.45 | 1,816 / 0.49 | 520 / 0.96 | 53 / 1.00 |
| PART | 19 / - | 8 / 0.88 | 7 / 1.00 | 6 / 0.50 | 6 / 0.50 | 8 / 0.75 | 5 / 1.00 |
| PUNCT | 2,395 / - | 286 / 0.14 | 752 / 0.10 | 960 / 0.05 | 22 / 0.18 | 24 / 0.04 | 8 / 0.00 |
| SYM | 107 / - | 35 / 0.34 | 41 / 0.46 | 8 / 0.38 | 2 / 0.50 | 2 / 1.00 | 2 / 1.00 |
| X | 6,449 / - | 1,262 / 0.34 | 2,316 / 0.29 | 839 / 0.23 | 171 / 0.23 | 98 / 0.38 | 245 / 0.42 |
| TOTAL | 292,825 / - | 93,133 / 0.71 | 139,230 / 0.63- | 40,811 / 0.44 | 17,784 / 0.60 | 10,894 / 0.90 | 26,551 / 0.82 |

**表 A.2** 実験結果 詳細

| | | accuracy | f1 | precision | recall | pk | | accuracy | f1 | precision | recall | pk |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **VERB$_{single}$** | train | 97.86 | 91.14 | 93.15 | 89.22 | 6.09 | **NOUN$_{single}$** | 97.04 | 87.55 | 90.85 | 84.47 | 8.19 |
| | val | 92.66 | 68.12 | 72.96 | 63.96 | 20.11 | | 94.79 | 76.60 | 83.39 | 70.85 | 16.27 |
| | test | 92.59 | 67.96 | 74.25 | 62.74 | 20.04 | | 94.59 | 75.89 | 84.18 | 69.10 | 16.39 |
| | cities | 85.08 | 63.25 | 61.44 | 65.35 | 24.16 | | 90.55 | 75.39 | 76.29 | 74.60 | 16.09 |
| | wiki50 | 88.93 | 34.70 | 62.78 | 24.09 | 37.56 | | 90.50 | 42.29 | 76.67 | 29.26 | 34.05 |
| | elements | 75.77 | 48.25 | 66.87 | 37.88 | 38.21 | | 80.65 | 61.55 | 76.63 | 51.71 | 30.34 |
| | clinical | 92.59 | 24.36 | 17.94 | 38.55 | 50.93 | | 93.21 | 29.58 | 21.09 | 49.88 | 51.37 |
| **VERB$_{rand}$** | train | 98.43 | 93.51 | 95.64 | 91.48 | 4.49 | **NOUN$_{rand}$** | 97.09 | 87.74 | 91.27 | 84.48 | 8.05 |
| | val | 94.30 | 74.46 | 80.83 | 69.03 | 17.34 | | 94.85 | 76.67 | 84.31 | 70.30 | 16.15 |
| | test | 94.13 | 73.88 | 81.74 | 67.41 | 17.42 | | 94.63 | 75.84 | 84.99 | 68.48 | 16.36 |
| | cities | 88.04 | 69.90 | 68.65 | 71.34 | 20.03 | | 90.78 | 75.52 | 77.96 | 73.32 | 15.72 |
| | wiki50 | 89.77 | 37.06 | 69.48 | 25.28 | 36.89 | | 90.56 | 41.79 | 79.03 | 28.43 | 34.07 |
| | elements | 77.47 | 52.40 | 71.90 | 41.31 | 36.25 | | 80.94 | 61.38 | 78.57 | 50.49 | 29.80 |
| | clinical | 93.50 | 26.68 | 19.70 | 41.34 | 49.50 | | 93.52 | 30.07 | 21.77 | 48.78 | 50.17 |