Renga Block: Q-grams Blocking を用いた 高速名寄せ・高速テキストクラスタリングの実装

山城 颯太
LINEヤフー株式会社
soyamash@lycorp.co.jp

概要

名寄せ（Entity Resolution）のサブタスクとして，大量の表記集合のうち確実に同一エンティティを指さない表記ペアを比較対象から除外する Entity Blocking が広く研究されている [1]．本研究は Q-grams Blocking に着目し，Zipf 則 [2] と分布仮説 [3] を考慮したシンプルなアルゴリズムを構築することで，高速かつ高精度な名寄せが実現できることを確認した．また，同アルゴリズムを文書集合に対して適用することで，高速かつ高精度なテキストクラスタリングが実現できることを確認した．

1 はじめに

名寄せ（Entity Resolution）は現実世界の同一エンティティを指し示す複数の表記同士をまとめ上げるタスクであり，データベース内のエントリの重複や表記ゆれの吸収に利用される．しかし，対象表記が N 件存在し，網羅的に全表記ペアを比較した場合，比較回数は $\mathcal{O}(N^2)$ となり，大規模データにスケールしない．その解決策として，比較する表記ペア数を削減する Blocking 手法が様々に提案されている [1]．その一つに Q-grams Blocking が存在する．Q-gram とは文字 N-gram のことを指し，例えば「LINEヤフー」に含まれる $q = 2$ の Q-gram は [‘LI’, ‘IN’, ‘NE’, ‘Eヤ’, ‘ヤフ’, ‘フー’] となる．Q-grams Blocking は同一 Q-gram を一つ以上共有する表記ペアのみを比較対象とすることで，比較回数を削減する．しかし，ある Q-gram k の出現数が n_k だとすると，k を共有する表記ペアの数は $\mathcal{O}(n_k^2)$ となる．Q-gram の出現数はべき乗則に従う [2] ため，高頻度 Q-gram を考慮した場合の比較回数は依然として多いままである．一方で，考慮の対象とする Q-gram が高頻度であるほど，その Q-gram の一致が判明したことによって得られる情報量は減少し，表記ペア同士が同一エンティティを指し示す確率は減少すると考えられる．そこで本研究では，低頻度 Q-gram のみを考慮の対象として Q-grams Blocking を行う，ボトムアップ型の階層的クラスタリングアルゴリズムを提案する．提案手法は，クラスタ結合時に素性平均ベクトル同士を比較することで分布仮説 [3] を自然に考慮する．

提案アルゴリズムを適用しうる別のタスクとして，テキストクラスタリング，あるいはトピックモデリングが存在する．これらは文書集合内で同じトピックについて書かれた文書同士をまとめ上げるタスクであり，ユーザ生成コンテンツの分類や自動タグ付与に利用される．特にトピックモデリングはクラスタリング結果に加えて，各クラスタの中心となるトピックを抽出し出力する．我々の提案アルゴリズムは Q-gram を素性とし，エンティティ同士をまとめ上げるが，同アルゴリズムにおいて，文中の形態素等を素性とすれば，文同士を効率的にまとめ上げることができる．本研究におけるテキストクラスタリングでは，名詞・名詞句のみを素性とする[1)]．

提案手法の極めて実用的な特徴として，以下の三つが挙げられる．(1) 教師ラベルや事前学習済みモデルを必要とせず，ドメインに依存しない．(2) 実行のたびに出力が変わらない決定的なアルゴリズムである．(3) 必要となる二つのパラメータのうち一つ（coverage threshold）をタスクごとに固定することで，ユーザは一つのパラメータ（similarity threshold）のみを変更することでクラスタの粒度を調整できる．

2 提案手法

2.1 アルゴリズム詳細

提案手法を Algorithm 1 に示す．このアルゴリズムは，エンティティリスト $\mathcal{S}$，similarity threshold θ，

1) 素性として採用する品詞は，言語やドメインに応じて変更することができる．

Algorithm 1 提案クラスタリング手法

Input $\mathcal{S}$, θ $(0 < \theta < 1)$, λ $(0 < \lambda < 1)$
Output $\mathcal{U}$

1: $\mathcal{F} \leftarrow$A list of features ordered by increasing frequency in $\mathcal{S}$.
2: $\mathcal{U} \leftarrow$A disjoint set union of clusters, where initially each element of $\mathcal{S}$ forms its own cluster.
3: **for** $k \leftarrow 1$ to $|\mathcal{F}| * \lambda$ **do**
4: $\quad \mathcal{S}_k \leftarrow$ A sublist of elements in $\mathcal{S}$ that have $\mathcal{F}[k]$.
5: $\quad$ **if** $|\mathcal{S}_k| = 1$ **then**
6: $\quad\quad$ **continue**
7: $\quad$ **end if**
8: $\quad$ **for** $i \leftarrow 1$ to $|\mathcal{S}_k| - 1$ **do**
9: $\quad\quad$ **for** $j \leftarrow i + 1$ to $|\mathcal{S}_k|$ **do**
10: $\quad\quad\quad f_i \leftarrow$The feature vector of $\mathcal{S}_k[i]$.
11: $\quad\quad\quad f_j \leftarrow$The feature vector of $\mathcal{S}_k[j]$.
12: $\quad\quad\quad s_f \leftarrow$The similarity score between f_i and f_j.
13: $\quad\quad\quad$ **if** $s_f < \theta$ **then**
14: $\quad\quad\quad\quad$ **continue**
15: $\quad\quad\quad$ **end if**
16: $\quad\quad\quad c_i \leftarrow$The centroid vector of $\mathcal{S}_k[i]$'s cluster.
17: $\quad\quad\quad c_j \leftarrow$The centroid vector of $\mathcal{S}_k[j]$'s cluster.
18: $\quad\quad\quad s_c \leftarrow$The similarity score between c_i and c_j.
19: $\quad\quad\quad$ **if** $s_c < \theta$ **then**
20: $\quad\quad\quad\quad$ **continue**
21: $\quad\quad\quad$ **end if**
22: $\quad\quad\quad$ MergeClusters($\mathcal{U}, i, j$)
23: $\quad\quad$ **end for**
24: $\quad$ **end for**
25: **end for**
26: **return** $\mathcal{U}$

coverage threshold λ を入力として受け取り，出力としてクラスタリング結果を保持する素集合森 $\mathcal{U}$ を返す．素集合森はそれぞれの集合を木構造で表現しており，各ノードは親ノードへの参照を持つ．

まず，エンティティリスト $\mathcal{S}$ に出現するすべての Q-gram を取得し，その出現頻度を数え上げ，出現頻度の昇順にソートする (line 1)．素集合森 $\mathcal{U}$ のサイズは $|\mathcal{S}|$ であり，初期値として，各エンティティに対応する各ノードの親ノード参照は自身を指す．つまり，初期化時点で各エンティティは自身のみを要素として持つ要素数 1 のクラスタに所属する (line 2)．ソート済み素性リストに対して線形探索を行い，2 回以上出現した各素性 $\mathcal{F}[k]$ を持つ $\mathcal{S}$ 中のすべてのエンティティをリスト $\mathcal{S}_k$ として取得する．素性出現数はべき乗則に従うので，半数近くの素性は出現数 1 としてここで除外される．また，coverage threshold λ に基づいて，素性リスト $\mathcal{F}$ のうち $|\mathcal{F}| * \lambda$ 個まで到達したら探索を打ち切る (line 3-7)．各 $\mathcal{S}_k$ 中のすべてのエンティティペアを取得し，それぞれのエンティティから Q-gram 素性ベクトル[2] を取得する (line 8-11)．クラスタ同士を結合するか否かを決定する判定処理は二段階に分けられる．まず第一段階として，上記素性ベクトルペア同士の cosine 類似度が similarity threshold θ 以上であることを確認する (line 12-15)．第二段階として，上記エンティティペアが所属するそれぞれのクラスタの平均ベクトル[3] を取得し，平均ベクトル同士の cosine 類似度が similarity threshold θ 以上であることを確認する (line 16-21)．上記二段階の判定処理を両方通過した場合のみ，上記エンティティペアが所属するそれぞれのクラスタ同士を結合する (line 22)．

上記提案手法をテキストクラスタリングに適用する際は，入力 $\mathcal{S}$ を文リストに置き換え，使用する素性を文中に出現する名詞・名詞句に置き換えることで同様に処理することができる．

2.2 正当化

連続値埋め込みとの比較: 離散的な素性を利用する提案手法への懸念として，ニューラルネットを用いた連続値埋め込みと比較して，(1) 同表記かつ異なる語義を持つ素性を区別できない，(2) 異表記かつ同義の素性を異なる素性として区別してしまう，といった二つの欠点が予測される．ニューラルベースの埋め込みは分布仮説 [3] を理論的根拠とした周辺文脈からの事前学習によって各サブワード（または語）の埋め込み表現を獲得している．一方，提案手法はクラスタ同士の結合可否の判定材料として，周辺文脈[4] を含めた素性ベクトル同士を cosine 類似度で比較している．従って，結合候補のクラスタ同士に，同表記の異なる語義を持つ素性が含まれる場合でも，周辺文脈が大きく異なれば結合しない．逆に，異表記だが同義の素性が含まれる場合でも，周辺文脈が類似しているならば結合する．この仕組みは連続値埋め込み同様に，(1)，(2) の課題に対処することを意図している．この極めてシンプルな処理が実用上は十分に機能することを 3 節の実験で示す．

頻度に基づく考慮素性の打ち切り: 提案アルゴリズムでは低頻度素性を優先してクラスタ結合を行っ

2) 各素性の出現頻度に基づく離散ベクトル．
3) Euclidean distance に基づく centroid の代わりに Spherical k-means で使用される Cosine similarity に基づく centroid を試してみたがわずかに精度が下がったので，比較的実装が簡潔になる前者を採用している．
4) ここで周辺文脈と呼んでいるのは，第二段階の判定で使用する，すでに同クラスタと判定された他サンプルの素性を包含した平均ベクトルを指している．

ていき，coverage threshold λ に達した段階で線形探索を打ち切っている．この工夫により，大幅な比較回数の削減を達成している一方，最終的な精度を犠牲にしている懸念がある．実際には，$\lambda = 1.0$ として全探索した場合よりも，探索を途中で打ち切った場合のほうが，同程度か，もしくは高い精度になることを 3 節で示す（詳細な考察は Appendix A に記す）．

2.3 実装

提案アルゴリズムは順方向インデックスと転置インデックスを用いることで高速化できる．特にテキストクラスタリングにおいては，転置インデックスから TF-IDF 値を算出することで，各クラスタのトピックを容易に抽出することができる[5]．

3 実験

我々は 2 節の手法を C++で実装し，Renga Block と名付け，社内向けに提供している[6]．以下では各種タスクにおける Renga Block の性能を比較する．

3.1 名寄せタスクの設定

名寄せタスクの実験データとして日本語 Wikipedia[7]記事本文中のハイパーリンクを抜き出し，5 回以上出現したリンク元表記・リンク先記事タイトルペアを利用した．リンク先記事をランダムに 1 万件選び，その記事をリンクしているリンク元表記 13,891 件を入力データとする（平均長 10.0 文字）．同一記事を指しているリンク元表記のペアについて，モデルがそれらを同一クラスタとして出力していれば正解とみなす．各表記ペアを一単位とする F1-score を算出し，モデル同士を比較する．Renga Block の素性は表記に含まれる Q-gram（$q = 2$）とする．パラメータは $\lambda = 0.95$ で固定し，経験的に $\theta = 0.5$ を採用した．また，全探索した場合との比較のために $\lambda = 1.0$ で固定した Renga Block の結果を併記する．

3.2 名寄せタスクのベースライン

既存研究の多くはモデルを学習させるための教師データ，紐付ける対象の知識ベース，あるいは表記以外のメタデータを必要とする [4]．もしくは事前学習済みモデルのような計算コストの大きいモデルを用いて表記ペアごとに判定を行う [5]．これらは本研究が取り組むドメイン非依存の高速な名寄せタスクには適さない．そこで本研究のベースラインとして，日本語汎用テキスト埋め込みモデル Ruri[8] [6] の出力[9]に対して密度ベースのクラスタリング手法である HDBSCAN [7] を適用する[10]．HDBSCAN のハイパーパラメータには，基本的に BERTopic [8] で使用されている値を流用している[11]．公平な比較のために GPU は使用していない．

3.3 名寄せタスクの結果

結果を表 1 に記す．Renga Block はベースラインと比較して Precision が高く，F1-score では 4.7pt の精度向上を達成している．実行時間は約 9,000 分の 1 である[12]．一方ベースラインの Ruri + HDBSCAN は Recall の値が優れているが，Precision が大きく下回っている．本実験条件において，名寄せのような表層の特徴が重要となるタスクに対して，連続値埋め込みが十分な性能を発揮できないことを示している．Ruri は Wikipedia を事前学習しているので，比較的有利な実験設定であることに注意されたい．Appendix B に出力例とともに定性評価を記す．また，Renga Block を全探索した場合は F1-score が微減しており，実行時間も約 4 倍に増えている．λ による探索の打ち切りが実行速度を改善するのみならず，精度を悪化させず，むしろ改善することを示している．

データ規模に対して Renga Block の探索打ち切りがどの程度効果的かを確認するために，入力表記リストを 140,428 件（10 万記事）に増やした場合の結果を表 2 に記す．実行時間はそれぞれ，探索を打ち切った（$\lambda = 0.95$）場合は約 8.9 倍の増加にとどまっているのに対し，全探索を行った場合は約 43.6 倍に増加している．このことから入力データを増加させ

5） ただし本研究では，トピック抽出性能については比較実験の対象外としている．
6） 以下で紹介されている Azuki を利用して API 提供している． https://techblog.yahoo.co.jp/entry/2020091730016720/
7） Wikipedia データは森羅プロジェクトが再配布しているバージョンを使用した．http://shinra-project.info/
8） https://huggingface.co/cl-nagoya/ruri-base
9） クエリ prefix を付与した平均プーリング
10） 前実験において，高速な Repeated Bisection 実装に Q-gram 素性を与えて提案手法と比較したが，精度が大きく劣り，速度的にも下回ったのでベースラインとして採用していない．https://github.com/fujimizu/bayon
11） 正解最小クラスタの要素数に合わせて *min_cluster_size* のみ 2 にしてある．また，BERTopic 同様にクラスタリング直前に次元削減（UMAP）を適用したところ，実行速度は大幅に改善するものの精度が著しく下がったので外してある．https://maartengr.github.io/BERTopic/index.html
12） ベースライン実行時間の約 99%はクラスタリング．

表 1　名寄せタスク実験結果

	Pre.	Rec.	F1	time [ms]
Ruri + HDBSCAN	35.5	**54.4**	43.0	619,296
Renga Block (全探索)	49.1	40.0	44.1	275
Renga Block ($\lambda = 0.95$)	**60.3**	39.5	**47.7**	**68**

表 2　名寄せタスク実験結果（入力データ約 10 倍）

	Pre.	Rec.	F1	time [ms]
Renga Block (全探索)	**46.3**	6.6	11.6	11,979
Renga Block ($\lambda = 0.95$)	26.2	**21.8**	**23.8**	**606**

た場合でも，Renga Block の探索の打ち切りによって，実行時間の増加幅を抑えられることがわかる．

3.4　テキストクラスタリングの設定

テキストクラスタリングタスクの実験データとして Wikinews[13] のトピック文から構築された MewsC-16 [9] の日本語サブセットを使用した．記事件数は 1,984 件，ラベル種類数は 12 件，平均長は 94.1 文字である．同一ラベルが付けられている記事ペアについて，モデルがそれらを同一クラスタとして出力していれば正解とみなす．各記事ペアを一単位とする F1-score を算出し，モデル同士を比較する．Renga Block の素性は文中に含まれる名詞・名詞句・未知語とする．パラメータは $\lambda = 0.95$ で固定し，経験的に $\theta = 0.25$ を採用した．[14] また，全探索した場合との比較のために $\lambda = 1.0$ で固定した Renga Block の結果を併記する．ベースラインは Ruri-Base の出力[15] を UMAP [10] で次元削減，HDBSCAN でクラスタリングした．ハイパーパラメータには BERTopic で使用されている値を流用した．UMAP は実行ごとに出力が変わるため，10 回試行した平均スコアを記している．公平な比較のために GPU は使用していない．また，特殊ドメインに対する汎用性を確認するために医療機器の説明文データである JMDN[16] に対する精度を併記する．件数 1,115 文，ラベル数 96 件，平均長は 110.6 文字である．

13）https://www.wikinews.org/

14）F1-score が最も高かった $\theta = 0.1$（26.0%）を用いた場合，本来別々のラベルが付けられるべき複数クラスタの記事を大きな一つのクラスタにまとめてしまっていた．実用上は巨大クラスタ内の記事のみを再帰的にクラスタリングすることで対処可能であるが，ここでは $\theta = 0.25$ として十分にクラスタ分割された結果について考察する．

15）文章 prefix を付与した平均プーリング

16）https://huggingface.co/datasets/oshizo/JMDNClustering-ja https://www.std.pmda.go.jp/scripts/stdDB/conf/stdDB_confjmdn.cgi

表 3　テキストクラスタリング実験結果（MewsC-16）

	Pre.	Rec.	F1	time [ms]
Ruri + UMAP + HDBSCAN	57.5	19.0	**27.2**	87,068
Renga Block (全探索)	17.8	**52.0**	26.5	564
Renga Block ($\lambda = 0.95$)	**92.6**	13.8	24.0	**209**

表 4　テキストクラスタリング実験結果（JMDN）

	Pre.	Rec.	F1	time [ms]
Ruri + UMAP + HDBSCAN	**67.9**	13.2	21.9	80,564
Renga Block ($\lambda = 0.95$)	24.4	**37.8**	**29.6**	**114**

3.5　テキストクラスタリングの結果

MewsC-16 に対する結果を表 3 に記す．Renga Block は Precision が高く，代わりに Recall がベースラインに対して少々下回っており，F1-score としてはやや劣る．しかしベースラインモデルの Ruri + UMAP + HDBSCAN は実行ごとに出力結果が安定せず，F1-score の標準偏差は 4.46 であった．一方 Renga Block は決定的なアルゴリズムであり，実行のたびに出力が変化することはない．実行時間は約 400 分の 1 である[17]．Appendix B に出力例とともに定性評価を記す．また，Renga Block を全探索した場合は F1-score が微増しており，3.3 節の傾向と異なる．これは Appendix A で考察している通り，探索の打ち切りによって比較漏れが起きる確率は対象サンプルの素性数に依存するため，短文が多い MewsC-16 データでは探索の打ち切りが精度に悪影響を与えたと考えられる．

JMDN に対する結果を表 4 に記す．ベースラインモデルは事前学習データとドメイン上の乖離がある場合に十分な精度が出ない傾向にあるが，Renga Block はある程度精度を維持していることがわかる．

4　まとめ

我々は Entity Blocking に着想を得たドメイン非依存の決定的ボトムアップクラスタリングアルゴリズムを提案し，これを名寄せタスクとテキストクラスタリングタスクへと適用する Renga Block を実装した．本実験条件下の連続値埋め込みによるベースライン手法と比較して，名寄せの場合はより高精度かつ数千倍の速度，テキストクラスタリングの場合は同程度，もしくはより高精度かつ数百倍の速度で動作することを確認した．Renga Block は現在社内向け API として提供されており，応用先として，大規模データストリームへの適用などが考えられる．

17）ベースライン実行時間の約 97%は次元圧縮．

参考文献

[1] George Papadakis, Dimitrios Skoutas, Emmanouil Thanos, and Themis Palpanas. Blocking and Filtering Techniques for Entity Resolution: A Survey. **ACM Comput. Surv.**, Vol. 53, No. 2, March 2020.

[2] George K. Zipf. **Human Behaviour and the Principle of Least Effort**. Addison-Wesley, 1949.

[3] Zellig Harris. Distributional structure. **Word**, Vol. 10, No. 2-3, pp. 146–162, 1954.

[4] Vassilis Christophides, Vasilis Efthymiou, Themis Palpanas, George Papadakis, and Kostas Stefanidis. An Overview of End-to-End Entity Resolution for Big Data. **ACM Comput. Surv.**, Vol. 53, No. 6, December 2020.

[5] Meihao Fan, Xiaoyue Han, Ju Fan, Chengliang Chai, Nan Tang, Guoliang Li, and Xiaoyong Du. Cost-Effective In-Context Learning for Entity Resolution: A Design Space Exploration. In **2024 IEEE 40th International Conference on Data Engineering (ICDE)**, pp. 3696–3709, 2024.

[6] Hayato Tsukagoshi and Ryohei Sasano. Ruri: Japanese General Text Embeddings. **arXiv:2409.07737**, 2024.

[7] Ricardo J. G. B. Campello, Davoud Moulavi, and Joerg Sander. Density-Based Clustering Based on Hierarchical Density Estimates. In Jian Pei, Vincent S. Tseng, Longbing Cao, Hiroshi Motoda, and Guandong Xu, editors, **Advances in Knowledge Discovery and Data Mining**, pp. 160–172, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[8] Maarten Grootendorst. BERTopic: Neural topic modeling with a class-based TF-IDF procedure. **arXiv:2203.05794**, 2022.

[9] Sosuke Nishikawa, Ryokan Ri, Ikuya Yamada, Yoshimasa Tsuruoka, and Isao Echizen. EASE: Entity-Aware Contrastive Learning of Sentence Embedding. In **Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies**, pp. 3870–3885, Seattle, United States, July 2022. Association for Computational Linguistics.

[10] Leland McInnes, John Healy, and James Melville. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. **arXiv:1802.03426**, 2020.

表 5　テキストクラスタリングの各モデルの出力例

共通文	秋田放送によると、2004 年に秋田県大館市の小坂製錬小坂線（小坂鉄道）で起きた脱線事故は、国土交通省航空・鉄道事故調査委員会の調べで、「枕木の腐食が原因」との結果が出た。: **事故** 朝日新聞によると、東日本旅客鉄道（JR 東日本）は 17 日、羽越線脱線事故で不通になっていた羽越本線の鶴岡-余目-酒田間の運転を 19 日の始発列車から再開すると発表した。: **事故**
Ruri	大阪ステーションシティ。: **経済** 鉄道博物館の概観（資料、GFDL）日本・埼玉県さいたま市にある鉄道博物館が 10 月 14 日 (UTC+9) に開館した。: **文化** 東京都交通局と東京都地下鉄建設は 13 日、2008 年 3 月の開業を予定している新交通「日暮里・舎人線（仮称）」の路線名および駅名を正式決定し、これを公表した。: **経済**
Renga Block	『さくら』・『みずほ』に使用の N700 系 7000 番台『つばめ』用 800 系朝日新聞・日本経済新聞によると、JR 九州と JR 西日本は、九州新幹線鹿児島ルートの全線開業日が 2011 年 3 月 12 日（UTC+9）に決まったと、9 月 15 日に正式発表した。: **経済** 朝日新聞・毎日新聞によると、JR 西日本は、大和路線（関西線）を走行する電車の運転席に設けられている列車無線の補助スピーカーの配線が切断されている事例が、6 月（UTC+9）以降で 4 件あったと発表した。: **事件** 産経新聞によると、JR 西日本の大阪環状線などの電車約 20 両について、緊急時に周辺の電車に対し停止信号を送る「防護無線」の予備電源のヒューズが、何者かによって抜き取られていたことが、7 月 15 日（UTC+9）に判明した。: **事件**

表 6　名寄せタスクの各モデルの出力例

Ruri	Renga Block
文科省, 文部官僚, 文部省, 文部科学大臣	文部科学, 文部科学大臣, 文部科学大臣杯, 文部科学大臣表彰, 文部科学大臣賞/文部科学大臣奨励賞
アイルランド中央銀行, アイルランド貴族院, 中国銀行（香港）, 中銀香港, 哈爾濱鉄路局, 長春軌道客車股份有限公司, 香港鉄路有限公司	アイルランド中央銀行, アイルランド貴族院, アイルランド庶民院, 統一アイルランド党

A　探索打ち切りの影響

単純化した計算により，探索の打ち切りによって同義エンティティ同士が比較されない確率をおおまかに見積もる．エンティティ e が m 個の Q-gram を持ち，各 Q-gram はその種類によらず一律同じ確率 p_t で同義の別表記 Q-gram にそれぞれ独立に置換され，言い換えエンティティである e' が生成されていると考える．ある Q-gram が coverage threshold λ に対して探索対象となる確率を p_c とおく．探索対象の Q-gram を低頻度 Q-gram と呼び，探索対象外の Q-gram を高頻度 Q-gram と呼ぶことにする．e と e' のペアが提案アルゴリズム上で比較されないのは以下二通りの場合のみである．(i) e に含まれる m 個の素性がすべて探索対象外の高頻度 Q-gram である．(ii) e に含まれる m 個の素性のうち k 個が探索対象内の低頻度 Q-gram であるが，k 個の低頻度 Q-gram がすべて e' 中では別の Q-gram に置換されている．上記 (i), (ii) の場合分けを考慮して，e の出現確率と，e の言い換えエンティティであるにもかかわらず探索対象とならない e' の出現確率の積である $P_{miss}(e, e')$ は以下のように表せる．

$$P_{miss}(e, e') = \sum_{0 \le k \le m} {}_mC_k p_c^k p_t^k (1 - p_c)^{m-k} \tag{1}$$

$$= (p_c p_t - p_c + 1)^m \tag{2}$$

式 (2) から，m の値が十分大きい場合は $P_{miss}(e, e')$ は無視できるほど小さくなり，m の値が小さい場合は p_c が大きいほど，また p_t が小さいほど $P_{miss}(e, e')$ はある程度まで小さくなることが予想できる．

B　定性評価

表 6 に名寄せタスクの出力例を記す．Renga Block の出力は低頻度 Q-gram の一致に基づいてクラスタが形成されていることは明らかであり，解釈性に優れる．一方，Ruri + HDBSCAN は表層より深い意味を捉えてまとめ上げる傾向が見られる．その一方，「中央銀行」と「中銀」のように文脈によっては関係ない略称を結び付けるような傾向も見られ，低頻度サブワード埋め込みが出力に悪影響を与えている可能性がある．また，中国の各種鉄道企業を一つのクラスタにまとめてしまっており，事前学習の知識が厳密なクラスタ分割を阻害している傾向が見られる．

表 5 にテキストクラスタリングの出力例を記す．各モデルの出力したクラスタに共通する文を「共通文」行とし，ベースラインモデルである Ruri + UMAP + HDBSCAN が出力したクラスタのみに含まれる誤り文を「Ruri」行，Renga Block が出力したクラスタのみに含まれる誤り文を「Renga Block」行として記している．各文の末尾にコロン区切りで正解ラベルを記している．ベースラインモデルの出力は実行ごとに異なるので，平均 F1-score に最も近いスコアとなった出力を用いている．共通文の例から，このクラスタには鉄道関連の事故に関する記事が集まっていると判断できる．しかしベースラインモデルは事故に限らない鉄道関連の記事を集め Precision を悪化させており，事前学習によって得た知識を上手くクラスタリングに活用できていないことがわかる．また，本来 Outlier として取り除くべき極端に短い文をクラスタに含めてしまっている．一方，Renga Block は主に鉄道関連の**事件**に関する記事を**事故**の記事と区別できずに誤っている．しかしその誤り例は人目で確認しても分類が難しいものであり，「発表」，「UTC+9」のような各ニュース記事カテゴリ独特の手がかり語に上手く着目してクラスタリングを試みていることがわかる．