第 26 回年次大会予稿

LOD データセット間のリンクにおけるクラウドソーシング適用の試み
An attempt to apply crowdsourcing on link between LOD datasets

新井 叡樹 †, 阪口 哲男 ‡*
Eiju ARAI, Tetsuo SAKAGUCHI

† 筑波大学 情報学群 知識情報・図書館学類
College of Knowledge and Library Sciences, School of Informatics, University of Tsukuba
‡ 筑波大学 図書館情報メディア系
Faculty of Library, Information and Media Science, University of Tsukuba
〒 305-8550 茨城県つくば市春日 1-2
E-Mail: saka@slis.tsukuba.ac.jp
* 連絡先著者 Corresponding Author

近年 Linked Open Data (LOD) としてデータの公開が促進されつつあり，様々な主題領域の LOD データセットが多くの機関等から提供されている．それらの活用のためにはデータセット間のリンクが重要であり，同一あるいは関連のある事物について網羅的なリンクをいかに軽便に行うかが課題である．そこで，本研究では近年活用されつつあるマイクロタスク型クラウドソーシングをリンク付けに適用することを提案し，その適用手法の開発とサンプルデータを用いた実験結果について報告する．

In recent years, publishing Linked Open Data (LOD) are promoted, and LOD datasets of various subject areas are provided by many organizations. Links between datasets are important for utilizing them, and how to carry out comprehensive links on the same or related things becomes a problem. Therefore, in this research, we propose applying microtask type crowdsourcing which is being used recently to linking, and develop a method and report the experimental result using sample data.

キーワード: Linked Open Data, Crowdsourcing, Microtasks

1 はじめに

近年，個人や行政機関，企業等の組織から様々なデータが Linked Open Data (LOD) として提供される事例が増えている．また，それらのデータの利活用が増えているほか，利活用を促進するためにワークショップやコンテスト等も開催されている．各組織等が提供している LOD データセットの利活用が促進されるには，そこに含まれるデータそのものが有用であることに加えて他のデータセットと相互にリンクされていることが重要である．その一方で，異なるデータセット間を網羅的にリンクするのは膨大な組合せについてリンクの可否を判断する必要があり，そのコストが課題となる．

本研究では，異なるデータセット間のリンクを支援するために，マイクロタスク型のクラウドソーシングの適用を提案する．その適用方法としては様々な方式が考えられるが，ここでは 2 つのデータセットからリンクすべき組合せの候補を抽出し，その候補についてリンクすべきか否かを人に判断してもらうためのタスクを生成することにした．

2 LOD におけるリンクと課題

Linked Data は，「構造化されたデータを Web 上で相互にリンクづけして，それらを公開できる一連のしくみを提供する実践的方法」[1] と述べられているように，リンク付けされていることが大前提となっている．そして，オープンデータにその Linked Data を適用したものが LOD である．LOD では RDF データモデルが用いられることが多いため，本研究も RDF に基づいて行った．RDF では，Subject (主語)，Predicate (述語)，Object (目的語)

の Triple (三つ組/トリプル) の単位でデータを記述する．

LOD では様々な事物を URI (Uniform Resource Identifier) で識別し，その事物間の関係や事物とそれを説明するリテラル[1] の属性を述語で表す．述語もその概念を明確に識別するため，URI が用いられる．例えば，A という事物と B という事物が同じであるという関係は述語「owl:sameAs」[2] で表される．このように LOD では述語によって様々な事物が互いにリンクされる．

各組織等がデータセットを作成し，公開する際はそのデータセット内の事物間をリンクするのはもちろんであるが，既存の他データセットともリンク付けすることが一般的である．その際，対象となる他データセットを網羅的に捜し出すだけのコストをかけるのが難しかったりすると，著名なデータセットとのみリンクすることになる．また，公開後に関連性の高いデータセットを継続的に探索し，リンク付けするようなこともコストがかかってしまう．そのためか，The Linking Open Data Cloud Diagram[2] などを見ればわかるように，多くのデータセットを結び付けるハブのようなデータセットがいくつか存在し，他のデータセットはそのハブにリンク付けされることが多い．

このような状況では，2 つのデータセットを用いて有用なアプリケーションを構築しようとしても，直接リンク付けされておらず，ハブを経由した間接的なリンクを利用することが考えられる．間接的であっても意味的には十分な場合もあるかも知れないが，途中に別の事物が介在することによって意味的にずれが含まれる可能性も考えられる．そのため，可能ならば異なるデータセット間は直接リンク付けされている方が望ましい．

しかしながら，前述のようにハブ以外にリンク付けするべきデータセットを網羅的に捜し出したり，継続的にリンク付けを更新していくようなコストをかけるのもなかなか難しいのが実際であると思われる．

そこで，そのリンク付けするべき候補を機械的に捜し出し，リンク付けを支援するための研究がいくつかなされて来ている．マッチングのためのヒューリステイクスを宣言して行うものとして Silk[3]，また機械学習を用いた手法として RiMOM[4] や山根らの手法 [5] 等がある．これらの手法ではそれぞれ優れた精度を実現しているが，あらゆる分野のデータセットを対象とするようなものにまでは至っておらず，また少ないとはいえ結果に誤りが含まれてしまう．

本研究では完全な機械化を行うのではなく，最後にクラウドソーシングによって人に判断してもらうことで，意味的に正確なリンクを行えるようにすることを目指す．

3 クラウドソーシングの適用

3.1 手法の概要

本研究では，機械的にリンク候補を抽出した後，リンク判定作業をマイクロタスク型クラウドソーシングで実行するためのタスク生成を行う．リンク付けを行う対象となる 2 つのデータセットの探索については済んでいるものとし，またその中にある主語として記述されているリソースをリンク対象として候補の抽出を行う．リンク付けする述語としては同一の事物を表す owl:sameAs のみとして，下記のような流れで処理を進める．

1. リンクを付与するデータセットを 2 つ読み込ませる．
2. リンク候補として抽出するときのリソース間の類似度の閾値を設定する．
3. それぞれのデータセットが持つリソースの類似度を総当たりで算出する．
4. 類似度が設定した閾値を超えたら，そのリソース同士をリンク候補とする．
5. リンク候補として抽出されたリソースに関する情報を表形式に変換する．
6. ワーカに 2 つの表を提示し，それらを比較するリンク判定タスクを行ってもらう．

3.2 リンク候補の抽出

類似度計算については，リソースの URI は用いず，リソースのプロパティ中のリテラルである文字

[1] 文字列や数値等

[2] 本来は http://www.w3.org/2002/07/owl#sameAs という URI で表されるが長くなるので，http://www.w3.org/2002/07/owl# を「owl:」というプリフィックスで表している．本稿では以下同様．

列を対象とし，岡崎らの手法 [6] を参考に次のように求める．文字列 x と y のそれぞれの N-gram 集合を X, Y とし，その集合間類似度 $String\_sim(x, y)$ は，次の式で表す．

$$String\_sim(x, y) = \frac{|X \cap Y|}{|X \cup Y|} \tag{1}$$

N について，$min(|x|, |y|)$ をリテラル x, y の文字数の最小値とし，次のように定める．

$$min(|x|, |y|) < 3 \Rightarrow N = min(|x|, |y|) \tag{2}$$

$$min(|x|, |y|) \geq 3 \Rightarrow N = 3 \tag{3}$$

次にリテラル l がリソース r を説明するものとし，先に記述されているほど l は r を特徴付けていると考え，リテラルの重み LW を次のように定義する．

$$LW(l_{ij}) = (N_{L_i} + 1) - order(l_{ij}) \tag{4}$$

ここで，L_i は r_i のリテラル集合であり，$L_i = \{l_{ij} | j \in [1, N_{L_i}], r_i$のリテラル $\}$(N_{L_i}は L_iの要素数). $order(l_{ij})$ は r_i における l_{ij} の出現順位である．

次に類似度を計算する 2 つのリソースを r_1, r_2 とする．r_1, r_2 についてのリテラルの集合を L_1, L_2 とする．

1. r_1, r_2 の名前を表すプロパティを指定する．
2. L_1, L_2 の要素の全ての組合せの $String\_sim$ を算出する．
3. 求めた $String\_sim$ が設定した閾値を超えたら，その 2 つのリテラルをペアとする．ただし，名前を表すリテラルの $String\_sim$ が閾値を超えた場合は手順 4, 5 を飛ばし，r_1, r_2 をリンク候補とする．また，抽出されたペアを構成するリテラルはそのペアにのみ属するとし，異なるペアに共通して同じリテラルが含まれる場合は，類似度が高い方を優先する．
4. L_1, L_2 の全ての要素の LW を計算する．
5. 抽出されたペア同士の重みの積和を r_1 と r_2 の類似度 $Sim(r_1, r_2)$ とし，これが設定した閾値を超えたら r_1, r_2 をリンク候補とする．

例えば，抽出されたペアが $\{(l_{11}, l_{21}), (l_{13}, l_{25}), (l_{14}, l_{26})\}$ だった場合，$Sim(r_1, r_2)$ は，

$$Sim(r_1, r_2) = \begin{array}{l} LW(l_{11}) \cdot LW(l_{21}) + \\ LW(l_{13}) \cdot LW(l_{25}) + \\ LW(l_{14}) \cdot LW(l_{26}) \end{array} \tag{5}$$

となる．また，リテラルの重みをあるリソースが持つリテラルの数によらず類似度を 0 以上 1 以下に，次のように正規化する．

$$\frac{LW(l_{ij})}{\sqrt{\sum_{k=1}^{N_{L_i}} k^2}} \tag{6}$$

類似度の閾値については実験の項で述べる．

3.3 リンク判定用タスクの生成

前節で計算した類似度に基づいてリンク候補となるリソースの組を抽出するが，その URI を見ただけではワーカはその組をリンクするべきか否かを判断することが難しい．そこで，本研究ではリソースの URI だけを見せるのではなく，リソースのプロパティを表形式で見せることにした．そのプロパティを見ることで人が判定しやすくなると考えた．

また，リソースの URI やプロパティとなっている述語や目的語にも URI が含まれるが，それらも全て見せると長い割には意味がわかりにくいと考え，URI を短縮して表示する．URI から Prefix を抽出し，抽出した Prefix を短い文字列で置き換えることで URI を短縮形に変換し，表中に提示する．

あるデータセットに出現する URI 集合を U，U の Prefix の集合を $P_1(U), P_2(U), P_3(U)$ とし，$U = \{u_i | i \in [1, N_U], あるデータセットに出現する URI\}$, $P_1(U) = \{p_{1j} | j \in [1, N_{P_1U}], 1$ 回の切り分けでできた U のPrefix候補 $\}, P_2(U) = \{p_{2k} | k \in [1, N_{P_2U}], 2$ 回の切り分けでできた U のPrefix候補 $\}$, $P_3(U) = \{p_{3l} | l \in [1, N_{P_3U}], 3$ 回の切り分けでできた U の Prefix候補 $\}$ とする．本研究では，1 種類の Prefix 候補に対し，2 種類以上の元の URI が先頭からマッチするとき，その Prefix 候補を Prefix として抽出する．

1. u_i を末尾から “/” または “#” で切り分けていき，Prefix 候補 p_{1j}, p_{2k}, p_{3l} を生成する．
2. p_{1j} と先頭からマッチする u_i が出現する回数を数え，$Match(p_{1j}, U)$ とする．

3. $Match(p_{1j}, U) \geq 2$ ならば，p_{1j} を Prefix とする．Prefix として抽出された p_{1j} と先頭からマッチする u_i を U から削除する．

4. P_1 の全ての要素について 2〜3 を繰り返す．

5. P_2，P_3 について，2〜4 と同様の処理を行う．

6. Prefix の末尾が “#” の場合，末尾の “#” から次の “/” までの文字列で Prefix を置き換え，Prefix の末尾が “/” の場合，末尾の “/” から次の “/” までの文字列で Prefix を置き換える．

7. Prefix として “http://” や “https://” 等の不適切なものが抽出されている場合は，それを削除する．

例えば，“http://www3.city.sabae.fukui.jp/rdf/refuge#1” という URI から Prefix を抽出し，短縮形に変換すると Prefix は “http://www3.city.sabae.fukui.jp/rdf/refuge#”，短縮形は “refuge:1” となる．その Prefix を求める過程例を図 1 に示す．

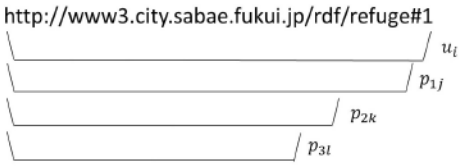


図 1: URI の Prefix を求める過程例

なお，データセット中で prefix 宣言がされている場合はそれを用いる．また，owl，rdf，rdfs 等一般的に用いられている Prefix はそれを用いる．

このようにして，短縮形を求めた上で，リンク候補の 2 つのリソースの表を並べて表示することになるが，それぞれの表の中のプロパティの並び順が比較に影響を与える可能性がある．そこで，本研究では次の 3 つの順で実験を行い，比較することにした．

1. ペアをなすリテラルの類似度が高い順 (図 2)

2. リテラルが文字であるものを先に，数字を後にした上でペアをなすリテラルの類似度が高い順 (図 3)

3. データセット中で記述されている順 (図 4)

表1

ラボーゼかわだ	
rdf1s131i2:5	
rdf1s131i:name	ラボーゼかわだ
rdf1s131i:tel	0778-65-0012
wgs84_pos:lat	35.947934
wgs84_pos:long	136.300829
rdf1s131i:address	鯖江市上河内町第19号37番地2
rdfs:label	5
rdf1s131i:zipcode	916-1235

表2

ラボーゼかわだ	
refuge:337	
rdfs:label	ラボーゼかわだ
refuge:tel	+81-778-65-0012
geo:lat	35.948078
geo:long	136.300819
refuge:address	上河内町19−37−2
sabae:ordinalNumber	337
rdfs:label2	La Pause Kawada
rdf1s307i:numberofpopulation	190
rdf:type	refuge:Refuge
sabae:locate	area:10

図 2: 類似度順の表示例

表1

ラボーゼかわだ	
rdf1s131i2:5	
rdf1s131i:name	ラボーゼかわだ
rdf1s131i:tel	0778-65-0012
rdf1s131i:address	鯖江市上河内町第19号37番地2
rdf1s131i:zipcode	916-1235
wgs84_pos:lat	35.947934
wgs84_pos:long	136.300829
rdfs:label	5

表2

ラボーゼかわだ	
refuge:337	
rdfs:label	ラボーゼかわだ
refuge:tel	+81-778-65-0012
refuge:address	上河内町19−37−2
rdfs:label2	La Pause Kawada
geo:lat	35.948078
geo:long	136.300819
sabae:ordinalNumber	337
rdf1s307i:numberofpopulation	190
rdf:type	refuge:Refuge
sabae:locate	area:10

図 3: 文字を数字より優先した類似度順の表示例

4 実験と結果

前節までで述べたリンク候補の抽出と表形式での提示方式について評価するために，サンプルのタスクを実行してもらった上でアンケートを取った．実験に用いたデータセットは「鯖江市公共施設」[7] と「鯖江市避難施設」[8] である．両者には座標を表すプロパティとして “http://www.w3.org/2003/01/geo/wgs84_pos#” の “lat” と “long” が用いられており，これらは文字列としての類似度計算を行わず，位置のずれが約 50m 以内になるように数値比較を行った．それ以外の数値については類似度計算を行っていない．また，文字列の類似度計算については，この鯖江市のデータの他に数件のデータセットについて再現率が 1 となり適合率が最大となった，閾値 0.5 を用いた．

実験方法

実験は筑波大学の学生 12 人を対象に行った．実際の LOD データセットから生成したタスクを 1 人当たり 8 件行った．前半 4 件は URI の短縮形を用

表1

ラボーゼかわだ	
rdf1s131i2:5	
rdfs:label	5
rdf1s131i:name	ラボーゼかわだ
rdf1s131i:tel	0778-65-0012
rdf1s131i:zipcode	916-1235
rdf1s131i:address	鯖江市上河内町第19号37番地2
wgs84_pos:lat	35.947934
wgs84_pos:long	136.300829

表2

ラボーゼかわだ	
refuge:337	
rdfs:label	ラボーゼかわだ
geo:lat	35.948078
geo:long	136.300819
sabae:ordinalNumber	337
refuge:address	上河内町19−37−2
refuge:tel	+81-778-65-0012
rdfs:label2	La Pause Kawada
rdf1s307i:numberofpopulation	190
rdf:type	refuge:Refuge
sabae:locate	area:10

図 4: データセット中の記述順の表示例

いたタスク (TypeA) で，後半 4 件は短縮しない元の URI を用いたタスク (TypeB) である．また，被験者を 3 グループに分け，それぞれ 3 つの並び順のタスクを割り当てた．図 5 はタスク画面例である．


表1は「鯖江市の公共施設」、表2は「鯖江市の避難施設」に関するのデータの一部です

TypeA

表1

衛陽会館	
rdf1s131i:2:3	
rdfs:label	3
rdf1s131i:name	衛陽会館
rdf1s131i:tel	0778-52-5789
rdf1s131i:zipcode	916-0027
rdf1s131i:address	鯖江市桜町2丁目7番1号
wgs84_pos:lat	35.947885
wgs84_pos:long	136.179382

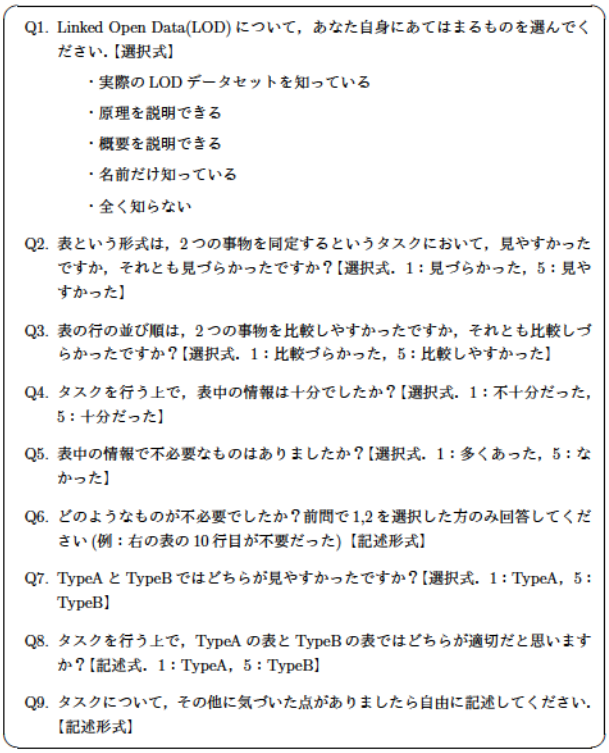
表2

衛陽会館	
refuge:9	
rdfs:label	衛陽会館
geo:lat	35.947928
geo:long	136.17939
sabae:ordinalNumber	9
refuge:address	桜町2丁目7−1
refuge:tel	+81-778-52-5789
rdfs:label2	Kyoyo hall
rdf1s307i:numberofpopulation	168
rdf:type	refuge:Refuge
sabae:locate	area:1

タスク1-1　「表1が示す場所」と「表2が示す場所」は同じ場所ですか？ *

- ○ 同じ
- ○ 異なる
- ○ わからない

図 5: タスク画面例


Q1. Linked Open Data(LOD) について，あなた自身にあてはまるものを選んでください.【選択式】

- ・実際の LOD データセットを知っている
- ・原理を説明できる
- ・概要を説明できる
- ・名前だけ知っている
- ・全く知らない

Q2. 表という形式は，2 つの事物を同定するというタスクにおいて，見やすかったですか，それとも見づらかったですか？【選択式．1：見づらかった，5：見やすかった】

Q3. 表の行の並び順は，2 つの事物を比較しやすかったですか，それとも比較しづらかったですか？【選択式．1：比較づらかった，5：比較しやすかった】

Q4. タスクを行う上で，表中の情報は十分でしたか？【選択式．1：不十分だった，5：十分だった】

Q5. 表中の情報で不必要なものはありましたか？【選択式．1：多くあった，5：なかった】

Q6. どのようなものが不必要でしたか？前問で 1,2 を選択した方のみ回答してください (例：右の表の 10 行目が不要だった)【記述形式】

Q7. TypeA と TypeB ではどちらが見やすかったですか？【選択式．1：TypeA，5：TypeB】

Q8. タスクを行う上で，TypeA の表と TypeB の表ではどちらが適切だと思いますか？【記述式．1：TypeA，5：TypeB】

Q9. タスクについて，その他に気づいた点がありましたら自由に記述してください.【記述形式】

図 6: アンケート設問一覧

実験結果

8 件のタスクについての正答率は表 1 のようになった．TypeA の正答率は 96%で，TypeB は 94%である．タスク実行後に行ったアンケートの項目を図 6 に示す．また，自由記述を除いた回答を表 2 から表 10 に示す．

結果より URI については TypeA すなわち短縮形を用いる方が見やすく，適切であるという意見が多く，概ね短縮そのものは意図通りの効果があったと考えられる．ただし，見やすさに比べて適切さで低い評価が増えており，短縮による情報損失が課題であると考えられる．また行の並び順については類似度順が比較的良い回答が多く，また正答率も類似度順の方がやや良い傾向が得られ，特にデータセット内での出現順は比較しづらいという意見もあった．表中の情報が十分であったという意見が多かった一方で，不必要な情報も多かったという結果が得られたが，表中に含める情報の取捨選択を機械的に実行するのは難しく今後の課題であると思われる．なお，LOD に関する知識と正答率の間にはほぼ相関はなく，不特定多数のワーカに実行してもらうクラウドソーシングを適用すること自体の有効性はあるものと思われる．

5　おわりに

本研究では，異なるデータセット間のリンク付けにマイクロタスク型クラウドソーシングを適用するための手法として，リンクする候補の抽出とタスクにおけるリンク候補の提示方式について提案し，実験の結果ある程度良好な結果が得られた．特に RDF データモデルのようなグラフ構造データを比較する際にワーカに表形式で示しても有効であることを示すことができた．しかしながら，実験者数がまだ少ない点と，実験に用いたデータセットが特定の 2 つのみに留まっているため，本提案の有効性が一般的であるかどうかまではまだ結論づけることはできない．また，owl:sameAs のみではなく，様々な関係に基づくリンク付けを行うための候補抽出方式などまだまだ多くの課題が残されている．

参考文献

[1] トム・ヒース; クリスチャン・バイツァー （武田英明監訳）：「Linked Data: Web をグローバルなデータ空間にする仕組み」．近代科学社，149p., 2013.

[2] “The Linking Open Data cloud diagram”. http://lod-cloud.net/ （2017 年 11 月 23 日参照）.

[3] Volz, Julius; Bizer, Christian; Gaedke, Martin: “Discovering and Maintaining Links on the Web of Data”, *The Semantic Web - ISWC 2009*, pp. 650–665, 2009.

[4] Juanzi, Li; Jie, Tang; Yi, Li: “RiMOM: A Dynamic Multistrategy Ontology Alignment Framework”, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 21, No. 8, pp. 1218–1232, 2009.

[5] 山根昇平; 鵜飼孝典:「オープンデータの Linked Data への変換–リンク付与とボキャブラリ統一の自動化–」, 人工知能学会研究会資料. 人工知能学会セマンティック Web とオントロジー (SWO) 研究会.

[6] 岡崎直観; 辻井潤一:「集合間類似度に関する簡潔かつ高速な類似文字列検索アルゴリズム」, 自然言語処理, Vol. 18, No. 2, pp. 90–117, 2011.

[7] 鯖江市役所:「鯖江市公共施設」. http://linkdata.org/work/rdf1s131i, (2017 年 11 月 23 日参照).

[8] 鯖江市役所:「鯖江市避難施設」. http://linkdata.org/work/rdf1s307i, (2017 年 10 月 19 日参照).

表 1: タスク正答率

タスク	正答率
タスク 1 (TypeA)	100%
タスク 2 (TypeA)	92%
タスク 3 (TypeA)	92%
タスク 4 (TypeA)	100%
タスク 5 (TypeB)	100%
タスク 6 (TypeB)	100%
タスク 7 (TypeB)	83%
タスク 8 (TypeB)	92%

表 2: Q1: LOD について

実際の LOD データセットを知っている	1 名
原理を説明できる	1 名
概要を説明できる	3 名
名前だけ知っている	6 名
全く知らない	1 名

表 3: Q2: 表形式は見やすかったか

1	2	3	4	5
1 名	2 名	2 名	6 名	1 名

表 4: Q3a: 行の並び順は比較しやすかったか (類似度順)

1	2	3	4	5
0 名	0 名	3 名	0 名	1 名

表 5: Q3b: 行の並び順は比較しやすかったか (文字優先の類似度順)

1	2	3	4	5
0 名	2 名	1 名	1 名	0 名

表 6: Q3c: 行の並び順は比較しやすかったか (データセット内の出現順)

1	2	3	4	5
1 名	3 名	0 名	0 名	0 名

表 7: Q4: 表中の情報は十分だったか

1	2	3	4	5
0 名	0 名	3 名	6 名	3 名

表 8: Q5: 表中の情報に不要なものはあったか

1	2	3	4	5
2 名	6 名	2 名	1 名	1 名

表 9: Q7: TypeA と TypeB はどちらが見やすいか

1	2	3	4	5
8 名	3 名	1 名	0 名	0 名

表 10: Q8: TypeA と TypeB が適切か

1	2	3	4	5
5 名	3 名	2 名	1 名	1 名