Vol.2018-CE-146 No.6
2018/10/20

Linked Dataを用いた俯瞰的な多肢選択式問題自動生成手法の提案

奥原 史佳[1,a)]　清 雄一[1,b)]　田原 康之[1,c)]　大須賀 昭彦[1,d)]

概要：近年，教科全体や科目内全体で俯瞰的な学習が求められている．特に教科・科目間の関連を図った横断的な学習が求められていること，多肢選択式問題が大量かつ広範囲の出題に向いている形式であることから，俯瞰的な多肢選択式問題が有用であると考えられる．“俯瞰的な問題” とは，広い関連知識を含み全体像を捉えさせるような内容を指す．一方で適切な問題を人手で生成・収集することはコストがかかる．本研究では，俯瞰度を考慮した多肢選択式問題の自動生成手法を提案する．本手法では，多肢選択式問題を構成する問題と選択肢に対してそれぞれ要件を設定し，知識ベースとして Linked Data を利用したアプローチを考案する．Linked Data とは，構造化されたデータ同士をリンクさせることができるグラフデータである．多肢選択式問題のうち，単純な単数選択形式と複数の解が存在することが保証される複数選択形式を生成するアルゴリズムを考案して生成した．生成問題に対する評価は，被験者として教員免許状所持者と学生に対して出題した上で，要件に対応する複数の評価項目について満たす問題かどうかを確かめた．

Generation of Multiple Choice Questions Including Panoramic Information Using Linked Data

Fumika Okuhara[1,a)]　Sei Yuichi[1,b)]　Yasuyuki Tahara[1,c)]　Akihiko Ohsuga[1,d)]

1. はじめに

学校ではカリキュラムが設定されており，この内容を不足なく学習するべきである．カリキュラム・マネジメントの重要性として，“教科横断的な視点から教育活動の改善を行っていくこと” が重要な鍵であると述べられている [1] ことから，教科全体で俯瞰的な学習活動が求められていると考えられる．

多肢選択式問題は，資格試験や検定試験などで広く用いられている形式の 1 つである．この出題形式は，大規模な受験に対する迅速な採点が容易である点，採点者による客観的採点の可能性が高い点で有用である．また，その回答方式が “選択肢から選ぶ” ことで完結するため，一題に対する回答効率が高いことが考えられる [2]．そのため，大量かつ広範囲の分野にわたる出題に向いている問題形式である可能性があり，学習者にとっては広範囲にわたる学習が容易であり，出題者にとっては広範囲の単元に対する評価ができる可能性がある．

以上から，特に教科・科目間の関連を図った横断的な学習が求められていること，多肢選択式問題が大量かつ広範囲の出題に向いている形式であることから，学習者と出題者双方にとって，俯瞰的な多肢選択式問題が有用であると考えられる．一方，適切な多肢選択式問題を人手で生成・収集することはコストがかかる．

本稿では，俯瞰的な多肢選択式問題を自動生成する方法を提案し，生成問題に対する評価を行う．本手法では，Linked Data を知識ベースとして利用し，あらかじめ設定した正解に対する問題と不正解の選択肢の生成を可能にするアプローチを考案した．多肢選択式問題のうち，単純な単数選択形式に加えて，2 つ以上の指定個数の解が存在することが保証される複数選択形式を生成するアルゴリズムを

[1] 電気通信大学大学院
The University of Electro-Communications, Tokyo, Japan
[a)] okuhara.fumika@ohsuga.lab.uec.ac.jp
[b)] seiuny@uec.ac.jp
[c)] tahara@uec.ac.jp
[d)] ohsuga@uec.ac.jp

扱う．評価では，問題と選択肢のそれぞれに対して設定した要件に基づいて，俯瞰度や妥当性を含めた評価項目に対する評価実験を行った．第 2,3 章で背景と関連研究，第 4,5 章で研究目的と提案手法，第 6,7 章で実装と評価方法，第 8 章で評価実験の結果，第 9,10 章で考察とまとめを記す．

2. 背景

俯瞰的な学習活動に関して，学習指導要領 [3] には，“指導計画の作成にあたって配慮すべき事項” として，“各教科・科目等について相互の連携を図る” という記載がある．例として，地理歴史科目の第 3 款 “各科目にわたる指導計画の作成と内容の取扱い” では，“教科全体として調和のとれた指導” や “中学校社会科及び公民科との関連並びに地理歴史科に属する科目相互の関連に留意” をすることが記されている．したがって，教科や科目全体での俯瞰的な学習活動が求められていると言える．

問題生成には既存技術として Linked Data を用いる．Linked Data とは，Tim Berners-Lee により提唱された構造的グラフデータであり，Web の仕組みを用いてデータ同士を相互にリンクさせたものである．Linked Open Data (以下 LOD) は，Linkd Data をウェブ上でデータ公開したものであり，各自治体や機関によって Open Data として公開されているデータ同士を結びつけて，誰でも自由に利用できるように公開されている．LOD 同士のリンクを表す LOD クラウド[*1]には，2018 年 5 月時点で 1000 以上のデータセットが含まれている．

LOD は様々な分野で取り組まれており，中でも有名なものとして Wikipedia のデータを Linked Data 形式にした DBpedia[*2]がある．日本語版の DBpedia Japanese[*3]も存在し，日本語 Wikipedia の InfoBox 内の情報を Linked Data 形式にされている．

3. 関連研究

Linked Data を教育分野における教材生成リソースとして利用する試みが行われている．

ASSESS [4] では，LOD を利用した知識テスト自動生成の試みがなされている．エンティティの要約化と RDF(Resource Description Framework) の言語化を行うことにより，自然言語による出題を可能にしている．出題ドメインは人体循環システムに設定されており，各形式に対応する選択肢生成も LOD を利用することで実装されている．また，Papasalouros らは Semantic Web Rule Language rules から自然言語で多項選択問題を生成する方法を提示した [5] [6]．これは，“仮定 ⇒ 結論” で表現されるように、仮定が成立する場合に結果も成立するというシンプルな形式を示すものである．Rocha らは，データセットから特定のドメインまたはトピックに関連するリソースを持つ Question の生成を試み [7]、Afzal らは教師なしの関係抽出アプローチによりドメイン内に存在する重要な概念に関する Question 生成方法を提示した [8].

また，多肢選択式問題において作成された選択肢の品質評価モデルを作成した研究 [9] もある．作問者により手作業で作成された選択肢の品質を，自動評価することを可能にする品質評価モデルを構築することを目指したものである．モデルでは選択肢の品質に関わる要素として，選択肢間の構文的・意味的な類似度の大きさに着目している．一方，Patra らは，Distractors 生成のための提案システムにおいて，Web 情報を使用することによりキーワードと Distractors 間の関連性を考慮した [10]．さらに，歴史学習のための LOD を利用した歴史依存質問生成オントロジを構築した研究 [11] では，学習シナリオに基づく出題システムを視野に入れたものとなっている．複数の知識ベースと出題形式を分類した Grasser の分類法を用いることで，歴史問題の生成という利用目的に特化したオントロジを構築している．

これらの関連研究に，テスト問題の生成というテーマに対して，セマンティック Web 技術の利用可能性が検証されている．しかし，これらの手法ではキーワードに対して「X が生まれたのは次のうちどれか.」「Y の作品は次のうちどれか.」などといった画一的な問題や選択肢しか生成できず，多肢選択式問題として単純な内容になってしまう．また，上記の方法で生成された問題のリソースは関連性の高い内容からのみ構成されていると考えられるため、限られた狭い分野における出題となる可能性がある。

4. 目的

本研究では，俯瞰的な多肢選択式問題の自動生成手法を提案することを目的とする．ここで俯瞰的な問題とは，広い関連知識を含み全体像を捉えさせるような内容を指すものとする．問題生成システムの利用シーンにおいて，出題者はテスト作成のための時間や手間などと言ったコストの低減の可能性，学習者は自習用問題やテストのリハーサルとしての利用が挙げられる．

図 1 では，本手法により生成する多肢選択式問題の出力例を示している．カリキュラムと単元の選択を入力として，Wikipedia のデータを利用し，カリキュラム内に含まれるキーワード抽出，Answer に設定するためのキーワード選択，問題と不正解の選択肢の構成を行う．出力は，多肢選択式問題の構成要素として，問題文である Question Graph，正解の選択肢である Answer，不正解の選択肢である Distractors となる．なお，本研究ではグラフである Question Graph を問題として扱う．問題のグラフにおけるリンク構造から，Answer は「Philosopher」をクラスに

[*1] http://lod-cloud.net
[*2] http://dbpedia.org
[*3] http://ja.dbpedia.org

持ち，「無知の知」「ペルシャ戦争」という事項や，「古代哲学」の時代，「プラトン」が影響を受けた人物であるという関係性を持つということが分かる．この関係性を全て満たすような Answer に該当する語彙を，右の選択肢から選ぶと「ソクラテス」が正解の選択肢になり，残りの選択肢が不正解の選択肢となるという出題例である．

今回は，Answer を 1 つ回答する単数選択問題，2 つ以上の指定された個数を回答する複数選択問題を生成する．回答者は，各形式に応じて，Answer に当てはまる語彙を選択肢から指定個数回答することになる．このとき，Question Graph において Answer ノードの該当語彙が指定個数以上あることが保証される必要性がある．

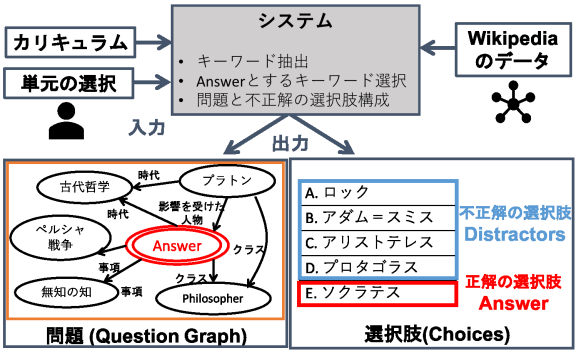

図 1 出力する多肢選択式問題の例

5. 提案手法

任意の Answer に対する Question Graph，Distractors の生成手法をそれぞれ示す．なお，知識ベースとして DBpedia, DBpedia Japanses 等を利用することが可能である．

5.1 Question Graph 生成アプローチ

Question Graph は，Answer に関するトリプルを探索することで生成する．まず，SPARQL を介して Answer まわりの情報を複数取得する．そして，RDF グラフで取得情報間の関係性を視覚化することで，出題とすることを考える．すなわち，グラフ中の Answer にあたる要素の表示を隠して，そのまわりの語彙やそれらのプロパティから，Answer を推察するという形式の出題である．そして，Answer という要素を隠したグラフ自体を Question Graph とする．なお，Question Graph で問う事柄は Answer まわりのデータとの関係性であるため，文章化する必要性はなく，図 1 のようにグラフの状態で出題する．複数データ間のサブグラフ構造を全探索するアルゴリズムを提案した研究 [12] があり，本アプローチでは以下の要件を考慮した探索手法を考案する．

5.1.1 Question Graph の要件

生成する Question Graph として，満たすべき要件を以下に設定する．特に，“(必須)” と記載する項目を必須の要件とする．

Question Graph の要件

(1) (必須) 各ノードはキーワード
(2) (必須) Answer まわりの情報の結合性を保証
(3) グラフ全体としてなるべく俯瞰的
(4) グラフの内容を把握できる程度の規模感
(5) Answer に該当する単語の数が少ない方が良い

5.1.2 Question Graph 生成手法

上述の要件を満たすような Question Graph を，DBpedia を利用して RDF グラフを元に生成する手法を考案する．

単数選択形式の問題生成アルゴリズム

単数選択形式の Question Graph 生成アルゴリズムとして，次の Algorithm 1,2 を考える．Answer ノードと隣接するノードからさらに隣接するノードを繰り返し探索し，ノード間のリンク構造を広げていくことで生成する．なお，リンク構造は IN と OUT の有向リンクにより形成される．探索過程において，要件 (4) の規模感に配慮したグラフにするため，探索して広げる範囲や回数を予め制限する．探索回数は，探索を開始する起点のノードである *Answer* を基準として，1-Hop ずつ隣接ノードを探索するとき，その Hop 数を探索の深さ h として定める．探索範囲は，同一の深さ h における IN,OUT それぞれの有向リンク構造の数を，探索の広さ w で定める．この h と w の制限を満たす規模感で Question Graph を生成する．特に，要件 (3) の俯瞰度を考慮し，Answer ノードを起点として以降すべての隣接ノード間の語彙の距離が常に遠くなるようなノードを抽出する．

Main_Standard は，*Answer* まわりの情報に関する全ノード集合を取得した後，それらの全リンク構造を取得して，サブグラフとして返すものである．入力として，知識ベースのグラフである KG(Knowledge Graph) の G，正解の選択肢 *Answer*，探索の深さ h，ノードの広さ w を与え，知識ベース上のサブグラフを出力として得るアルゴリズムである．サブグラフは，全ノードの集合 N_s とノード間のリンク構造の集合 M_s から成る．

ターゲットのノードに関するサブグラフの全ノード集合を返すアルゴリズム get_far_nodes においては，*target* の初期値として *Answer* を指定することで，最終的に *Answer* に関するグラフが得られる．line.6 から line.23 にかけて，*target* の各有向リンクに対して隣接ノードを探索することでグラフを広げていく．なお，*target* の隣接ノード集合を取得するために，line.7 において隣接ノード集合を返す *neighbors* 関数を定義して用いている．line.10 では，指定ノード間の距離を求める関数 *dist* により，ターゲットとの距離が最大になる隣接ノードが返却される．また，line.12 においては，隣接ノードの全祖先との距離を *target* と比

較することを，探索の深さ h まで再帰的に繰り返すことにより，*Answer* ノードから隣接ノードまでの全てのノード間の距離が常に遠くなることが保証される．最終的に，全ノード集合の返却により，全ノード集合とそのリンク構造から成るサブグラフを，提案の Question Graph として得られる．

Algorithm 1 Main_Standard

Input: KG G, *Answer*, depth h, width w
Output: KG
1: $N_S = get\_far\_nodes(Answer, \{\}, G, h, w)$
2: $M_S = get\_all\_links(N_S)$
3: **return** (N_S, M_S)

Algorithm 2 get_far_nodes

Input: Node *target*, Set of ancestor nodes *Ancestors*, KG G, depth h, width w
Output: Set of nodes
1: $N = \{target\}$
2: **if** $|Ancestors| == h$ **then**
3: **return** N
4: **end if**
5: $count = 0$
6: **for** $direction \in \{IN, OUT\}$ **do**
7: $B = neighbors(target, direction)$
8: **while** $count < w$ AND $0 < |B|$ **do**
9: $flg = True$
10: $n = \arg\max_{n' \in B} dist(target, n')$
11: $B = B \backslash \{n\}$
12: **for** $n_j \in Ancestors$ **do**
13: **if** $dist(n_j, n) < dist(n_j, target)$ **then**
14: $flg = False$
15: **end if**
16: **end for**
17: **if** flg **then**
18: $count = count + 1$
19: $N = N \cup get\_far\_nodes(n, Ancestors \cup \{target\}, G, h, w)$
20: **end if**
21: **end while**
22: $count = 0$
23: **end for**
24: **return** N

複数選択形式の問題生成アルゴリズム

次に，複数選択式のグラフ生成方法を述べる．選択肢から複数選んで回答する出題形式であるため，グラフ上で Answer ノードに該当する語彙が指定個数以上あることが保証されているような問題を生成する必要性がある．そこで，Question Graph の要件 (5) における Answer に該当する単語の数を調整する．

Algorithm3,4 では，get_far_nodes_h1 により，深さ $h = 1$ のときに *Answer* ノードに該当する語彙が指定個数 nA 以上であることを保証するようなアプローチを採用している．指定の範囲内において *neighbors* 関数により取得した隣接ノード全てを持つノードに該当する語彙が，知識ベース上に nA 個存在する場合，それをグラフ上の隣接ノードとして採用する．ただし，これを距離最大の語彙から順に確かめることで，なるべく俯瞰度を保つように作成する．line.9 では，各 $direction \in \{IN, OUT\}$ に関してそれぞれ広さ w 個の隣接ノードの全組み合せを返す関数として *combination* を定義し，*Answer* に関する全隣接ノードの組み合わせの集合を得る．また line.15 では，グラフ中心の Answer ノードに該当する全ての要素 *targets* を取得する関数 *get_all_targets* を定義して，*Answer* 同様に正解となるものを複数取得する．$h = 2$ 以降については，*get_all_nodes* の手順で隣接ノードをつなげていくことで問題を完成させる．

Algorithm 3 Main_MultipleAnswers

Input: KG G, *Answer*, depth h, width w, number of the answers nA
Output: KG
1: $(N_{Sh1}, targets) = get\_far\_nodes\_h1(Answer, \{\}, G, nA, w)$
2: $N_S = get\_far\_nodes(N_{Sh1}, Answer, G, h, w)$
3: $M_S = get\_all\_links(N_S)$
4: **return** $(N_S, M_S), targets$

Algorithm 4 get_far_nodes_h1

Input: Node *target*, Set of ancestor nodes *Ancestors*, KG G, number of the answers nA, width w
Output: Set of nodes
1: $N_{Sh1} = \{target\}$
2: $B_{combh1} = \{\}$
3: $targets = \{\}$
4: $count = 0$
5: **for** $direction \in \{IN, OUT\}$ **do**
6: $B\_w_{direction} = neighbors(target, direction)$
7: **end for**
8: **if** $|B\_w_{direction \in \{IN,OUT\}}| \neq 0$ **then**
9: $B_{combh1} = combination(B\_w_{direction \in \{IN,OUT\}}, w)$
10: **end if**
11: **while** $|targets| < nA$ **do**
12: **for** $b\_comb_{h1} \in B_{combh1}$ **do**
13: $N_{Sh1} = \arg\max_{n'_{h1} \in b\_comb_{h1}} dist(target, n'_{h1})$
14: $M_{h1} = get\_all\_links(N_{Sh1})$
15: $targets = get\_all\_targets(N_{Sh1}, M_{h1})$
16: **if** $|targets| < nA$ **then**
17: $B_{combh1} = B_{combh1} \backslash \{b\_comb_{h1}\}$
18: **end if**
19: **end for**
20: **end while**
21: **return** $N_{Sh1}, targets$

図 2 においては，*dist* 関数によるノードの抽出イメージを示している．グラフは $w = 2$ を想定し，知識ベースから *target* の隣接ノードを抽出していく．まず $h = 1$ のとき，*target* である *Answer* の全隣接ノード取得し，距離が最大の赤色で示す隣接ノード 2 つが抽出される．さらに $h = 2$ では，先に得られた *target* である赤色ノードとの距離が最大の隣接ノードを求め，祖先の赤色ノードとの距離について *target* ノードと比較して常に大きいとき，青色で示す隣接ノードとして採用する．これを指定した探索の深さ $h = i$ まで実行することで，起点の Answer から常に遠くなるようなグラフが作成できる．

また，今回は上記の基本アルゴリズムに加えて，語彙のメジャー度も考慮することとした．メジャー度の重みは，コーパス開発センターの『現代日本語書き言葉均衡コーパス』(BCCWJ)[*4]を利用し，教科書サブコーパス (OT) の語

[*4] http://pj.ninjal.ac.jp/corpus_center/bccwj/

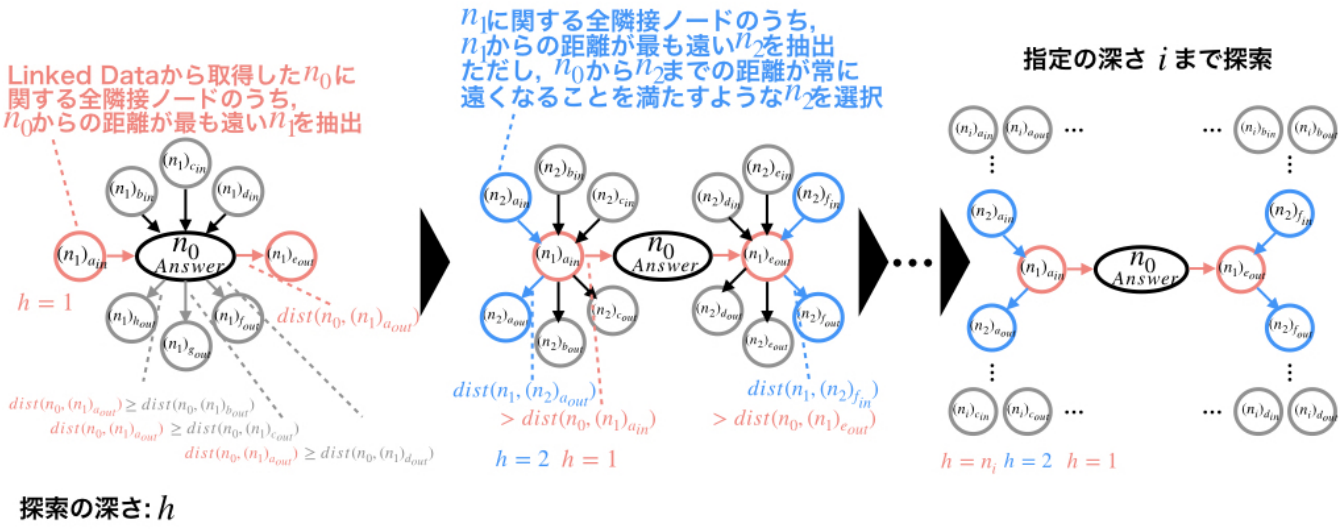

図 2 *dist* 関数による知識ベースからの隣接ノード抽出イメージ

彙を 1.0，図書館及び出版サブコーパス (LB,PB) のうち歴史と社会科学分類の語彙を 0.5，それ以外の語彙を 0.0 と設定した．これを *dist* 関数との和算により適用して，コーパスに存在する語彙がよりメジャーな語彙として採用されやすくなるようにした．ただし，深さ $h = 1$ の各ノードについて，知識ベースからの問い合わせ結果中にコーパス OT の語彙が存在する場合は優先的に採用するようにした．

以上のアルゴリズムに基づいてコンパクトな規模感で俯瞰度の高い Question Graph の生成を目指す．

5.2 Distractors 生成アプローチ

Distractors は，Question Graph において Answer に該当しないノードであり，すなわち Answer とのリンク構造のいずれかを満たしても全てを満たさないようなノードを探索することで生成可能である．したがって，ここでは上記で生成した Question Graph を利用した Distractors の生成方法を考える．

5.2.1 Distractors の要件

生成する Distractors として，満たすべき要件を以下に設定する．

Distractors の要件

(1) (必須) 各ノードはキーワード
(2) (必須) Question Graph に対して不正解であること
(3) 一見して不正解と明らかに分かるものを避けたい

5.2.2 Distractors 生成手法

要件 (3) より，一見して不正解と明らかに分からないような Distractors を生成するためには，Answer と類似した語彙集合を Distractors とする必要がある．また，Answer との類似語彙は，知識ベース上におけるリンク構造も Answer と類似するということが考えられる．このことから，前述の方法により生成した Question Graph を利用して，Answer の隣接リンク構造から Distractors 候補を生成する．

まず，Question Graph において，Answer の隣接リンク構造の集合を抽出する．抽出した集合について，1 つ以上の任意のリンクを削除したときに初めて，他のリンク構造の集合を満たすようなノードに該当する語彙集合を，Distractors の候補とみなすことができる．

さらに，Lorenz ら [9] の取得方法を参考にして，取得した候補のうち，DBpedia 上で Answer と同じクラスに属する語彙に絞り込む．なお，Answer の属するクラスは複数存在する場合があるが，それらのクラスのうち最下層のクラス C，すなわち，Answer がクラス C のインスタンスであるが，そのサブクラスのインスタンスではない直属のクラス C を取得する．その取得クラス C をプロパティとして得たオブジェクトに該当する候補を最終的に採用する．

候補から選択肢への採用方法としては，候補と隣接ノードとの距離と Answer との距離に関する差の小ささと，メジャー度の大きさの和が上位のものから採用するようにする．

6. 実装

ここで，上述のアプローチにより実装した例を示す．知識ベースとして DBpedia Japanese を利用し，SPARQL endpoint を “http://ja.dbpedia.org/sparql/” に設定する．

なお，問題生成のドメインとして社会科科目 (地理・歴史・公民等) を設定する．また，Answer となるキーワードは，Wikipedia 上のページ名またはカテゴリ名として定義された語彙を事前に複数選んで用いることとする．実装上は任意のカリキュラムを設定せず，DBpedia 上の語彙をキーワードとして設定するものとする．

6.1 Question Graph 生成

Question Graph は前項で述べた通り，Answer まわりの情報を SPARQL クエリを介して複数収集したのち，そのRDF グラフを描画して示す．

dist 関数を定義するにあたり，語彙間の構文的・意味的に比較可能な指標として，word2vec の事前学習済みモデル[*5] [13] による similarity を利用する．語彙間の similarity の値が小さいほど，語彙間の構文的・意味的な距離が大きくなるとみなして定義した．グラフの描画は，取得した Answer まわりの情報をリスト化し，各要素間の関係性としてプロパティを示したものを Graphviz により描く．

例として，Answer を “ソクラテス” に設定して，単数回答形式のアルゴリズムにより作成した Question Graph を次の図 3 に示す．なお，各ノードの色について，Answer ノードを赤色，コーパス OT の語彙を青色，コーパス LB 及び PB の語彙を緑色，いずれにも該当しない語彙を灰色で表している．

6.2 Distractors 生成

前述のアプローチの通り，事前に前項の方法で生成した Question Graph を利用して，Answer のノードとのリンク構造を利用して，Answer と同じクラス C に属するインスタンスを複数得ることで Distractors を生成する．

以下に，Answer を “ソクラテス” に設定した場合に生成された Distractors 候補を表 1 にまとめる．

表 1 Answer=“ソクラテス” に対する Distractors 候補

リンク集合	Answer/Distractors 候補 { 候補総数 }
(0,1,2,3,4,5,6)	ソクラテス {1}
(1,2)	ディオゲネス (犬儒学派) {1}
(1,3)	ヘラクレイトス {1}
(1)	アナクサゴラス, アリスティッポス, 他 {5}
(4)	イマヌエル・カント {1}

7. 評価

前述のアプローチにより生成した Question Graph と Distractors について，前述の各要件と照らした評価方法を以下に述べる．なお，今回の評価における Question Graph の探索の深さを $h = 2$，広さを $w = 2$ に設定した．また，実際には適切な Answer を選択するが，評価②のために，全ての被験者が正解を知らないと想定される語彙を Answer に設定した．

7.1 Question Graph の評価方法

Question Graph の要件を照らした評価項目を以下に挙げる．

[*5] http://www.cl.ecei.tohoku.ac.jp/~m-suzuki/jawiki_vector/

Question Graph の評価項目

① 整合性...[要件 (2)]
- Answer が真の解であること

② 俯瞰度...[要件 (3)]
- 各ノードのクラスの横断度
- カリキュラム上の単元の横断度
- 縦軸 (時間軸)/横軸 (同時期の関連事象) の横断度

③ 特定性...[要件 (5)]
- Answer に該当する語彙集合の小ささ

④ 可読性...[要件 (4)]
- グラフ規模のコンパクトさ

特に，要件 (3) のグラフ全体としての俯瞰度と要件 (5) の Answer に該当する単語の数に照らして，評価項目の 2 つ目の俯瞰度と 3 つ目の特定性について，それぞれ以下の評価実験を行う．

7.1.1 俯瞰度の評価

評価項目の 2 つ目については，特定のカリキュラムを設定していないため，今回は実施しない．1 つ目の項目は，Question Graph 生成に利用した知識ベース上のクラスをどれだけ横断しているかという指標である．クラスは，DBpedia 上の全クラスのうち Thing 以下のクラスから語彙の該当クラスを調べる．3 つ目の項目は，縦軸として時間軸の横断度，横軸として同時期における関連事象の横断度を検証するものである．縦軸については，各ノードの語彙について DBpedia 上の Abstract 内に含まれる西暦年を，その語彙の時間として採用し，1 つの Question Graph 上での横断度を調べる．横軸については，あらかじめ設定した目安の年から前後の時間に該当するノードの組み合わせ数とそのノードの個数を調べるものとする．なお，クラスと西暦年の両者とも，DBpedia 上から取得可能なノードのみを算出の対象とした．

7.1.2 特定性の評価

生成した Question Graph に対して，Answer ノードに該当するような Answer 以外の語彙が少ないどうかを調べる．これは，Answer が持つ全リンク構造を持つようなノードを数え上げることで把握できる．ただし，複数選択問題については，Answer ノードに該当する語彙が少なくとも指定個数あることが必須要件となる．

7.2 Distractors の評価方法

Distractors の要件を照らした評価項目を以下に挙げる．

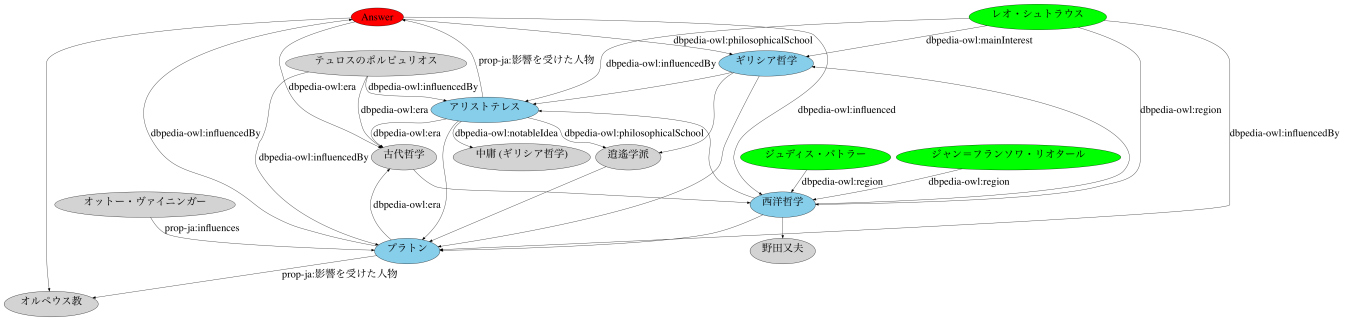

図 3 Answer=“ソクラテス” に設定したときの Question Graph

Distractors の評価項目

① Question Graph に対して不正解であること...[要件 (2)]
② 正解を知らない人の正答率が選択肢数分の 1 以下...[要件 (3)]
③ 選択肢間の類似度が大きい...[要件 (3)]

特に，要件 (3) の一見して不正解とは分からない選択肢であるかどうかを，評価実験を通して評価項目の 2 つ目と 3 つ目により検証する．

7.2.1 正答率の比較

被験者実験を通して実際の正答率を測り，項目 2 つ目の “正答率が選択肢数分の 1 以下” を満たすかどうかを検証する．

7.2.2 Answer との類似度比較

選択肢間の類似度比較を，構文的側面と意味的側面からの両者から行う．

- 構文的類似度
  選択肢間の品詞や構成パターンを含めて両者を比較する．構文解析ツールとして CaboCha[*6] を用い，係り受けと品詞の両者が一致するかを検証する．
- 意味的類似度
  選択肢間の意味的な類似度を比較する．Pho らの [9] における評価モデルの指標，あるいは [14] の項目分析の類似度指標より，語彙の種類 {Person，Location，Organization} の比較，DBpedia 上の注釈情報を利用した指標 “DBpedia Entity”，WordNet 上の語彙の階層構造を利用した指標 “wup similarity” などを利用し，類似度を算出して比較する方法がある．今回は，word2vec の事前学習済みモデルによる similarity の算出を行った．

8. 結果

実行環境として，PC は MacBook Pro, OS は macOS High Sierra を利用した．使用言語を Python として，ライブラリ SPARQLWrapper[*7] により SPARQL を実行した．また，設定した 10 個の Answer に対する問題生成にかかる実行時間を計測した．Question Graph は，プログラム実行時に一度だけ word2vec の事前学習済みモデル読込みに 76.66 sec 要し，その後の各問題に対するグラフ生成は平均 13.09 sec であった．Distractors は各問題で平均 3.575 sec であった．

実験では，1 つの Question Graph に対して 3 つの Distractors と 1 つの Answer を出題セットとして，単数回答形式 (以下 “単数形式”) を 10 問，複数回答形式 (以下 “複数形式”) を 5 問を実際の出題として扱った．

8.1 Question Graph の評価実験結果

8.1.1 俯瞰度の評価

俯瞰度に対する評価には，クラス横断度として 1 つの Question Graph 内におけるクラス数，縦軸の横断度として，西暦年の範囲とその幅 (year)，横軸の横断度として，目安前後 50 年間のノードの組み合わせ数とそのノードの個数の最大値 (MAX) をまとめた．以下表 2, 表 3 に示す．

表 2 クラスの横断度 (単数形式)

Q.	ノード数		クラス数	
	提案	random	提案	random
1	21	21	7	5
2	21	17	5	3
3	21	21	6	5
4	21	21	6	7
5	21	21	3	8
6	21	21	8	5
7	21	19	6	7
8	21	21	6	3
9	21	21	3	3
10	21	21	6	3
Ave.	21	20.4	5.6	4.9

8.1.2 特定性の評価

今回作成した単数形式における 10 個の Question Graph について，Answer ノードに該当する語彙集合を DBpedia

*6 https://taku910.github.io/cabocha/
*7 https://rdflib.github.io/sparqlwrapper/

表 3　縦軸/横軸の横断度 (単数形式)

Q.	縦軸 (years)		横軸 (MAX)	
	提案	random	提案	random
1	1782-1985(203)	1796-2013(217)	11 (5)	8(5)
2	1943-2013(70)	1841-1999(158)	9 (7)	4(2)
3	1190-2014(824)	1836-1957(121)	16 (7)	10(7)
4	1844-2006(162)	1825-2008(183)	11 (6)	16(8)
5	1856-1994(138)	1856-1969(113)	9 (4)	10(7)
6	1623-2001(378)	1568-2010(442)	9 (5)	9(5)
7	1603-1995(392)	1582-2005(423)	9 (3)	10(3)
8	1206-2015(809)	1079-1894(815)	13 (4)	11(2)
9	1156-1993(837)	1153-1951(798)	8 (4)	7(3)
10	1930-2014(84)	1919-2009(90)	11 (9)	14(12)
Ave.	****-****(389.7)	****-****(336)	10.6 (5.4)	9.9(5.4)

上で問い合わせたところ，Answer を除いて全て空であった．また，複数形式の 5 個についていずれも Answer の数を 2 つに指定したとき，該当語彙は 2 つのみのグラフが生成された．したがって，今回の実験においては，本アプローチによる特定性の評価項目は満たされていると言える．

8.2　Distractors の評価実験結果

8.2.1　被験者実験による正答率の比較

主観実験による結果を示す．表 4, 図 4 には，37 名の被験者から得られた単数形式の回答結果を，正答率を含めてまとめた．複数形式についても同様に，23 名の回答結果を表 5 にまとめた．

表 4　被験者実験の回答率 (単数形式)

Q.	正答率 [%]	Distractors 選択率 (降順)[%]		
		D1	D2	D3
1	60.9	24.3	8.1	2.7
2	21.6	59.4	3.61	5.4
3	21.6	48.6	18.9	10.8
4	13.5	70.3	13.5	2.7
5	40.5	32.5	18.9	8.1
6	13.6	54.0	21.6	10.8
7	91.9	8.1	0.0	0.0
8	56.8	35.1	5.4	2.7
9	21.6	40.5	29.7	8.1
10	18.9	35.1	29.7	16.2

表 5　被験者実験の回答率 (複数形式)

Q.	正答率 [%]			各選択肢の選択率 [%]			
	正解数 2	正解数 1	正解数 0	A1	A2	D1	D2
1	22	78	0	24	37	22	17
2	17	61	22	28	20	33	20
3	39	61	0	48	19	15	19
4	61	39	0	30	50	13	7
5	48	52	0	41	33	15	11

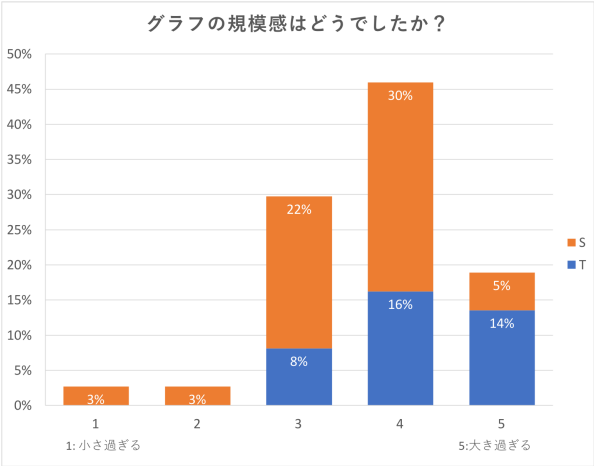

図 4　アンケート:「グラフの規模感について」
(T: 教職免許状所持者, S: 他学生被験者の回答とする．)

8.2.2　Answer との類似度比較

提案の dist 関数を利用した生成とランダム生成による各 Distractors について，構文パターンによる類似度比較 “pattern” と word2vec の事前学習済みモデルによる類似度比較 “ word2vec” の結果を表 6 に示す．“pattern” については，Answer のパターンと一致した場合に 1 を返したものであり，“ word2vec” については，Answer との similarity を算出したものである．なお，各値は各問あたりの値の平均を算出したものであり，すなわち，表の各値は 1-Distractor あたりの類似度の値となっている．

表 6　提案と random による Distractors と Answer との類似度比較 (単数形式)

Q.	pattern		word2vec	
	提案	random	提案	random
1	2	2	.613	.617
2	2	3	.305	.295
3	2	2	.451	.590
4	3	2	.359	.647
5	3	3	.801	.497
6	2	2	.438	.661
7	3	3	.354	.670
8	2	1	.557	.692
9	0	0	.475	.618
10	0	0	.671	.720
Ave.	1.9	1.8	.503	.601

9.　考察

まず，Question Graph について，必須要件 (1) の “各ノードはキーワード” と，要件 (2) の “情報の結合性” については，実装上はカリキュラムを設定せず，DBpedia 上の語彙をキーワードとして設定したため，要件を満たすことは自明である．実際にカリキュラムを設定する場合，利用する Linked Data 上にキーワードが存在するとき，要件を満たし本手法を利用可能である．また，Answer が特殊な

語彙の場合はリンクが見つからず理論上は問題作成に失敗する可能性がある．しかし，試験問題中に出現するようなメジャーな語彙やその関係性については，DBpedia のような LOD 上で概ねカバーしているとも考えられる．要件 (3) の “俯瞰的” の指標について，前項の表 2 のクラス横断度では，提案の方が平均で 0.5 クラス分上回った．また，表 3 の縦軸は提案が平均がで 53.7 年分上回り，横軸は前後 50 年間を同時期とした場合，時期数は提案の方が平均で 0.7 個分上回り，同時期の事象数の最大値は双方とも差がなかった．これより，今回の実験においては，*dist* 関数の利用がクラス横断度と縦軸の横断度について多少貢献したと言える．要件 (4) の “グラフの規模感” については，図 4 の被験者実験のアンケート結果より，今回の実験においては 5 段階中 4 の評価が最も多く，被験者によってやや大きめに感じる程度の規模感であったと言える．最後に，要件 (5) の “Answer に該当する単語の数の少なさ” については，前項の特定性に関する結果から，評価実験における 15 個のグラフについて，いずれも Answer ノードに該当する語彙が Answer 以外は存在せず，要件をクリアしていると言える．

Distractors について，必須要件 (1) の “各ノードはキーワード” と要件 (2) の “Question Graph に対して真に不正解” であることについては，生成アプローチより満たしていることは自明である．そして，要件 (3) の “一見して不正解と明らかに分からないこと” について，まず，評価項目の 2 つ目について述べる．“選択肢数分の 1”=1/4=約 25% であり，全ての被験者は正解の語彙を知らないという想定の下で，各問題に対する正答率が 25%以下であることが望ましい．表 4 より，単数形式において Q.1,5,7,8 を除く 6 問が指標を満たすという結果であった．満たさない 4 問のうち特に Q.7 は，Distractors について 3 つ中 1 つしか選択されなかったことから，要件を満たさない選択肢が生成された例として顕著な結果になった．複数形式の回答率は表 5 より，特に 2 つの Distractors 間の選択率の差は 13%以下であり，単数形式と比べて偏りのある例が見られなかった．次に，表 6 について，構文的類似度は両者に大きな差が見られなかったが，意味的類似度はランダムの方が 10% 程度大きい結果となった．これは，Distractors の生成アプローチでは隣接ノードを利用して生成されるが，隣接ノードについて提案では遠いもの，ランダムでは遠近両方あり得ることから，ランダムの方が Answer と意味的に近い語彙になるケースが多かったためと考えられる．

今回，教員免許状所持者 14 名に対して，Answer が同じ既存の問題集 [15] [16] の問題との比較アンケートを行い，生成した 5 つの単数形式の問題に対して主に “俯瞰度” に関する回答を得た．まず，質問 1 として俯瞰度の大きさ比較では，平均 72.8%が生成問題の方が高いと回答した．また，その俯瞰度の差や内容の差を 5 段階評価で得た結果は図 A·1,A·2 のようになり，内容の差も全体的に見て大きい印象を抱いたと言える．

10. まとめ

本稿では，俯瞰的なカリキュラムベースの多肢選択式問題自動生成手法を提案した．評価実験において単数形式と複数形式の問題を生成し，設定した要件に対する評価を実施したところ，俯瞰度に関する指標の一部に関して提案がランダムを上回るケースが見られた．今後の展望を以下に挙げる．

Question Graph 生成について，提案では “俯瞰” を叶えるための方法としてノード間の距離のみ考慮したが，ノード間のリンクの意味について考慮していなかった．出題として意味のある問題を生成するためには，ノードだけでなくリンクの種類についても考慮すべきだと考えられる．また，今回の評価では “俯瞰度” に対する評価指標として前項の 3 つの項目に基づいて評価したが，これらの指標のみで俯瞰度を叶えられるものではない．評価実験において行ったアンケート結果も参考として，現行の評価指標を今後見直していく必要がある．

Distractors 生成の改善として，本手法では Answer が属するクラスに属する語彙を採用しているが，Answer の属するクラスが存在しない，もしくは複数存在するケースにおいて，一見して明らかに Answer と異なる種類の候補が生成される可能性があるため，改善すべきである．また，選択肢間の同義語への対応のため，WordNet[*8]を利用した検証フェーズを設ける予定である．

また，本提案の新たな出題形式適用として，組み合わせ問題が考えられる．組み合わせ問題は大学入試センター[*9]の日本史や世界史の試験でもよく見られる “ 別々の問題における Answer の組み合わせを選択肢から回答する形式” であり，需要がある形式だと考えられる．

以上の展望を踏まえて，生成アプローチと評価方法の見直しに加えて他の出題形式への応用を実行し，より有効な俯瞰的問題の生成を目指す．

謝辞 本研究は JSPS 科研費 16K00419, 16K12411, 17H04705, 18H03229, 18H03340, 18K19835 の助成を受けたものです．本研究を遂行するにあたり，研究の機会と議論・研鑽の場を提供して頂き，御指導頂いた早稲田大学本位田 真一 教授をはじめ，活発な議論と貴重な御意見を頂いた研究グループの皆様に感謝いたします．

参考文献

[1] 文部科学省初等中等教育局初等中等教育企画課教育制度改革室：初等中等教育分科会 (第 100 回) 配布資料 1-1 4. 学習

*8 https://wordnet.princeton.edu
*9 https://www.dnc.ac.jp/center/

指導要領等の理念を実現するために必要な方策 (登録：平成27年11月), http://www.mext.go.jp/b_menu/shingi/chukyo/chukyo3/siryo/attach/1364319.htm (2015).
[2] 池上真人：多肢選択文法問題の設問形式に関する研究：択一式と複数選択式の解答プロセスに焦点をあてて，言語文化研究，Vol. 35, No. 1, pp. 55–72（オンライン），入手先 ⟨https://ci.nii.ac.jp/naid/120005677564/⟩ (2015).
[3] 文部科学省：高等学校学習指導要領，http://www.mext.go.jp/component/a_menu/education/micro_detail/__icsFiles/afieldfile/2011/03/30/1304427_002.pdf (2009).
[4] Bühmann, L., Usbeck, R. and Ngomo, A.-C. N.: ASSESS—Automatic Self-Assessment Using Linked Data, *International Semantic Web Conference*, Springer, pp. 76–89 (2015).
[5] Papasalouros, A., Kanaris, K. and Kotis, K.: Automatic Generation Of Multiple Choice Questions From Domain Ontologies., *e-Learning*, Citeseer, pp. 427–434 (2008).
[6] Zoumpatianos, K., Papasalouros, A. and Kotis, K.: Automated Transformation of SWRL Rules into Multiple-Choice Questions., *FLAIRS conference*, Vol. 11, p. 570Á575 (2011).
[7] Rocha, O. R., Zucker, C. F. and Giboin, A.: Extraction of Relevant Resources and Questions from DBpedia to Automatically Generate Quizzes on Specific Domains, *International Conference on Intelligent Tutoring Systems*, Springer, pp. 380–385 (2018).
[8] Afzal, N. and Mitkov, R.: Automatic generation of multiple choice questions using dependency-based semantic relations, *Soft Computing*, Vol. 18, No. 7, pp. 1269–1281 (2014).
[9] Pho, V.-M., Ligozat, A.-L. and Grau, B.: Distractor quality evaluation in multiple choice questions, *International Conference on Artificial Intelligence in Education*, Springer, pp. 377–386 (2015).
[10] Patra, R. and Saha, S. K.: A hybrid approach for automatic generation of named entity distractors for multiple choice questions, *Education and Information Technologies*, pp. 1–21 (2018).
[11] Jouault, C., Seta, K. and Hayashi, Y.: Content-Dependent Question Generation using LOD for History Learning in Open Learning Space, *New Generation Computing*, Vol. 34, No. 4, pp. 367–394 (2016).
[12] Fionda, V. and Pirrò, G.: Meta Structures in Knowledge Graphs, *International Semantic Web Conference*, Springer, pp. 296–312 (2017).
[13] 鈴木正敏，松田耕史，関根　聡，岡崎直観，乾健太郎：Wikipedia 記事に対する拡張固有表現ラベルの多重付与，言語処理学会第 22 回年次大会，pp. 797–800 (2016).
[14] Mitkov, R., Ha, L. A., Varga, A. and Rello, L.: Semantic similarity of distractors in multiple-choice tests: extrinsic evaluation, *Proceedings of the Workshop on Geometrical Models of Natural Language Semantics*, Association for Computational Linguistics, pp. 49–56 (2009).
[15] 歴史能力検定協会：歴史能力検定 2016 年実施 第 35 回全級問題集 pp.29-67 (2017).
[16] 歴史能力検定協会：歴史能力検定 2017 年実施 第 36 回全級問題集 pp.33-pp.75 (2018).

付　　録

A.1　評価実験のアンケート結果

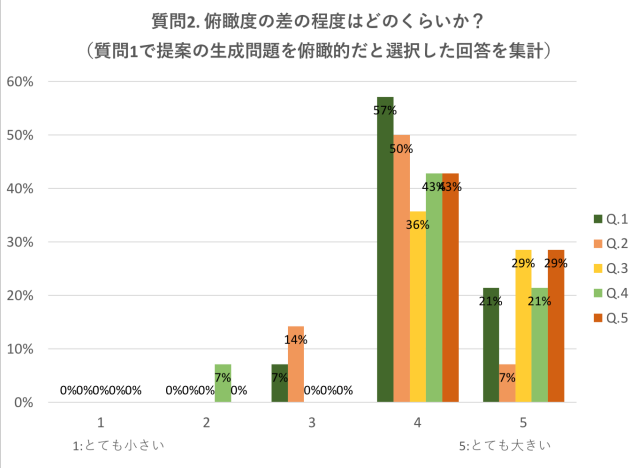

図 A·1　質問 2:「俯瞰度の差の比較」

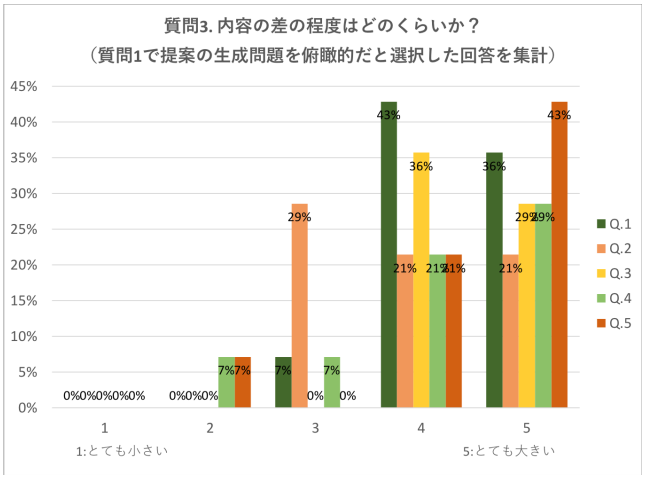

図 A·2　質問 3:「内容の差の程度の比較」