情報処理学会論文誌 Vol.60 No.10 1738–1756 (Oct. 2019)

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)]

受付日 2019年1月25日, 採録日 2019年7月3日

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必要とされていること，多肢選択式問題が大量かつ広範囲の出題に向いており多分野の出題に対応する形式であることから，学習者と出題者の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．“俯瞰的な問題” とは，幅広い関連情報を含み全体像をとらえさせるような内容である．一方，俯瞰的な問題を人手で生成・収集することはコストがかかる．そこで本研究では，出題時に幅広い関連情報を提示することで俯瞰的な視点で問題をとらえさせるような，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，Linked Dataを利用した自動問題生成アプローチを考案する．Linked Dataとは，構造化されたデータどうしをリンクさせることができるグラフデータである．出題形式として，単一の正解が存在する単数回答形式，複数の正解が存在する複数回答形式，複数の問題に対して単一の正解の組合せが存在する組合せ回答形式の3種類の生成アルゴリズムを提案した．生成問題に対する評価では，被験者として教職免許状所持者と学生に出題したうえで，要件に対応して設定した評価項目を満たすことを確かめた．

キーワード：俯瞰的な問題，Linked Data，多肢選択式問題，セマンティックウェブ

Generation of Multiple Choice Questions for Learning in a Broad Perspective Using Linked Data

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

Received: January 25, 2019, Accepted: July 3, 2019

Abstract: In recent years, just about all subjects require students to learn in a broad perspective. Because the need exists for cross-curriculum learning aimed at relating subject areas, it is useful for multiple choice questions to include panoramic information for learners. Panoramic information means comprehensive information that gives us macro-perspective; through which us look down at the whole learning subjects. A question including panoramic information refers to content that includes transverse related information and makes respondents grasp the whole knowledge. However, it is costly to manually generate and collect appropriate multiple-choice questions for learners and exam preparers. Therefore, in this research, we propose a method for automatic generation of multiple choice questions including panoramic information using Linked Data. Linked Data is graphical data that can link structured data, and it is used as a technology for data integration and utilization. In this paper, we aim to realize a system for automatically generating three types of multiple choice questions by implementing an approach to generating questions and choices. An evaluation method for the generation of questions and choices involves setting indicators for each evaluation item, such as validity and the degree of the inclusion of panoramic information.

Keywords: questions on a broad perspective, linked data, multiple choice question, semantic web

[1] 電気通信大学大学院情報理工学研究科情報学専攻
Department of Informatics, Graduate School of Informatics and Engineering, The University of Electro-Communications, Chofu, Tokyo 182–8585, Japan

[a)] okuhara.fumika@ohsuga.lab.uec.ac.jp
[b)] seiuny@uec.ac.jp
[c)] tahara@uec.ac.jp
[d)] ohsuga@uec.ac.jp

1. はじめに

近年の学習活動において，教科や科目全体を通した俯瞰的な学びが求められている．“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” が重要な鍵であり [1]，また，俯瞰的な学習活動に関して学習指導要領 [2] 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．

このような学習活動の成果を計測するため，また，学習意欲を高めるために，学習者に試験問題を解かせることが一般的に行われている．試験問題には様々な形式があるが，なかでも多肢選択式問題は，資格試験や検定試験等で広く用いられている形式の 1 つである．多肢選択問題とは複数の（一般に 3 個以上の）選択肢の中から正解を選択する出題形式であり，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．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，1 題に対する回答効率が高いことが考えられる [3]．

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．

本論文では，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．提案手法では，Web 上で公開されている膨大なグラフデータを知識ベースとして，Linked Data 技術により正解の語句に関連する情報を取得して抽出することで俯瞰的な問題を生成する．多肢選択式問題のうち，単一の正解が存在する単数回答形式，複数の正解が存在することが保証される複数回答形式，複数の問題に対して単一の正解の組合せが存在する組合せ回答形式の 3 種類の生成アルゴリズムを扱う．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．ここで俯瞰度とは “正解について関連情報を横断的に含む度合い” と定義する．

Linked Data とは，構造化されたデータどうしをリンクさせる技術であり，Web 上でオープンデータとして公開された Linked Open Data はグラフデータとして様々な分野で利活用される．教育分野においても，Linked Data を利用して学習問題を生成する既存研究は複数存在する．たとえば，Linked Data 上のグラフ構造より語句間の単純な関係性を抽出することで知識問題を自動生成した研究 [4] や，多肢選択式形式の問題において選択肢の品質評価モデルを作成した研究 [5]，歴史学習問題の作成のための歴史依存質問生成オントロジを構築した研究 [6] 等がある．しかし，これらの研究で提案された生成手法では，対象の学習語句を正解として，たとえば「X 年に生まれた人物は次のうち誰か」や「Y の作品は次のうちどれか」等といった問題と，正解の関連カテゴリに属する類似語句のみからなる不正解の選択肢等，単純な内容しか生成できず，正解との関連性が非常に高く自明な内容からのみ構成されると考えられるため，多肢選択式問題としてきわめて限定的な分野における単純な内容の出題となってしまう．

本研究では，Web 上で公開されている膨大なグラフデータを知識ベースとして，Linked Data の技術により正解の語句に関連する情報を取得して抽出することで俯瞰的な問題を生成する．**図 1** 右上や**図 A·1**～**図 A·4** に，本研究で生成する多肢選択式問題の例を示す．本研究ではこれらの図のように問題をグラフで表現し，以下このグラフを **Question Graph** と呼ぶ（文献 [4] 等の手法を使い，Question Graph を自動的に文章に変換することも可能である）．また，多肢選択式問題の選択肢を **Choices**，正解を **Answer**，不正解を **Distractors** と呼ぶ．知識ベース上のグラフデータは膨大であり，グラフ規模が小さい方が可視性が高いため，問題に利用するための情報抽出のアプローチが肝になる．本研究ではグラフの可読性を維持したまま俯瞰度の高い問題を生成するために，Question Graph に出現する語句間との距離がなるべく遠くなるように抽出する．一方，問題に対する選択肢についても，抽出したグラフデータにおける正解の関連語句とその関係性の情報を利用して，正解に近いが正解ではない語句を探索すること

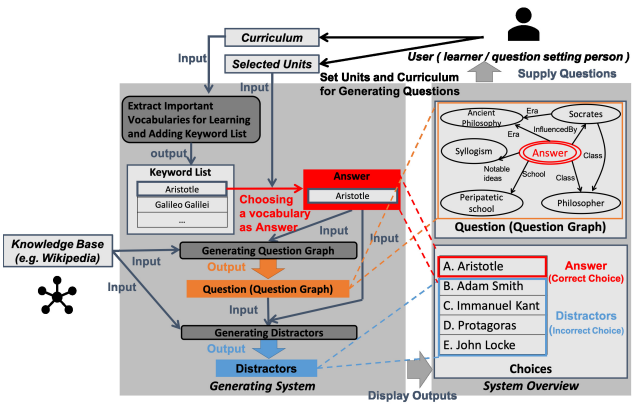

図 1 提案システムによる多肢選択式問題の出力例

Fig. 1 An examples of output image of multiple choice questions generated by proposed system.

で，一見して不正解とは分からない不正解の選択肢の生成を目指した手法を考案する．そのほか，語彙コーパスを利用した語句のメジャーな度合いの重み付けや表記ゆれ対応も含めて考慮した実装を行い，俯瞰度の高い多肢選択式問題の生成を目指している．

なお，本論文は国際会議 ICAART の論文 [7] を拡張したものである．以下では，2 章で関連研究，3，4 章で研究目的と提案手法，5 章で実装，6 章で評価項目と評価結果，7，8 章で考察とまとめを記す．

2. 関連研究

2.1 Linked Data

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どうしを相互にリンクさせたものである．Linked Data を Web 上で公開したものを特に Linked Open Data（LOD）と呼ぶ．LOD のような Web 上のリソースを記述するための形式として RDF（Resource Description Framework）があり，これはデータを「主語（subject）」「述語（property）」「目的語（object）」の 3 つの要素で構成されるトリプルの組合せで表現し，それぞれのトリプルを有向グラフで表すものである．たとえば，“日本の首都は東京都である．” という文章があった場合，トリプルとして「主語」：“日本”，「述語」：“首都”，「目的語」：“東京都” で表現し，有向グラフで表すと「“日本” $\xrightarrow{\text{“首都”}}$ “東京都”」となる．

LOD は様々な分野で取り組まれており，なかでも有名なものとして Wikipedia のデータを Linked Data 形式にした DBpedia[*1] があり，日本語版の DBpedia Japanese[*2] も存在する．DBpedia 上には “Person” や “Place”，“TimePeriod” 等語句の種類としてクラスが定義されており，各語句はそれらのうち特定のクラスに属する．

Iijima ら [8] は，LOD を利用して複数データ間の意外性のあるつながりを抽出する手法を推薦システムに適用した．大量の関連データが存在する LOD Cloud からの情報抽出方法としては，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 [9] や，ターゲットデータに関するリンクデータの階層構造やデータの持つクラス情報の類似性に着目してターゲットと類似ドメインの関連情報を抜き出すアルゴリズムを提案した研究 [10] がある．さらに，Demarchi ら [11] によるセマンティック Web 技術をエージェントへ適用する研究がある．

2.2 問題の自動生成

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も行われている．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 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．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 [12], [13]．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 [14]，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 [15]．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 [5] 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かかわる要素として，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 [16]．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 [6] では，学習シナリオに基づく出題システムを視野に入れ，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．問題の難易度推定に関する研究 [17] では，大量の事前データが必要である一般的な項目応答理論（IRT）に対して，問題と選択肢間で含まれる語の類似度をもとに各選択肢の選択率を推測するアプローチが考案されている．

以上のように，教育・学習分野のテスト問題の生成というテーマに対して，様々なアプローチでセマンティック 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 年に生まれた人物は次のうち誰か」や「Y の作品は次のうちどれか」等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おり，俯瞰的な問題生成は考慮されていない．

3. 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” について定義する．関連事項として，中教審は “「合教科・科目型」や「総合型」を導入する必要性” を述べており，“複数の教科・科目の知識・技能等を教科横断的・総合的に組み合わせることが必要” としている [18]．また “文理横断，学修の幅を広げる教育” として，“高度な専門知識を持ちつつ普遍的な見方のできる能力” や “分野を越えた専門知の組合せ” が求められている旨が記されている [19]．これらをふまえて，本研究における “俯瞰的” とは，「任意の語句や概念について幅広い関連情報からマク

[*1] http://dbpedia.org
[*2] http://ja.dbpedia.org

ロな視点でとらえるような」という意味として，“俯瞰的な問題” とは，「正解を中心として幅広い関連情報を含み全体像をとらえさせるような内容の出題」として定義する．正解の関連情報を問題の中で俯瞰的・総合的に組み合わせることで，分野を越えた専門知の組合せによるマクロな視点をもって問題全体をとらえて正解を導かせるような内容の出題とする．教育現場で必要とされる “幅広い分野からなる文理横断的なカリキュラム” の中で，試験問題を通じてその教育や学習の効果を評価するとき，正解を導くために “高度な専門知識” だけでなく “普遍的な見方のできる能力” が求められるような “俯瞰的な問題” の必要性が考えられる．

なお，本論文では多肢選択式として生成される問題の中に，幅広い関連情報を含む必要性を考えたうえで問題生成アプローチをとる．たとえば，「問 1：X 年に生まれた人物は次のうち誰か．問 2：Y の作品は次のうちどれか」という複数の問題が提示されたとき，この人物や作品がまったく異なる時代や性質の話である場合は，幅広い情報を含むことは可能である．しかし，上述の「俯瞰的・総合的に組み合わせる」「分野を越えた専門知の組合せ」は問題を解く際に必要とされず，何ら組み合わせることなく独立して解けてしまう．したがって，「俯瞰的・総合的に組み合わせる」「分野を越えた専門知の組合せ」を目的とした場合，複数の問題提示を独立に行うだけでは達成できず，多肢選択式問題内容として幅広い情報を含む必要がある．

図 1 に，本手法により生成する多肢選択式問題の出力例を示す．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 に設定するキーワードの選択，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．問題のグラフにおけるリンク構造から，**Answer** は「Philosopher（哲学者）」のクラスに属していて，「Ancient Philosophy（古代哲学）」の時代で，有名な思想として「Syllogism（三段論法）」があり，学派は「Peripatetic school（逍遥学派）」であり，「Socrates（ソクラテス）」に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．この関係性をすべて満たすような Answer に該当する最も適当な語句を図 1 右下の選択肢から選ぶと，「Aristotle（アリストテレス）」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．

生成する問題形式として，Answer に該当する語句を選択肢から 1 つ回答させる「単数回答形式（図 A·1）」[20], [21]，Answer に該当する語句を選択肢から 2 つ以上の指定個数回答させる「複数回答形式（**図 A·2**）」[22]，2 つの別々の問題について各 Answer に該当する語句の正しい組合せを選択肢から 1 つ回答させる「組合せ回答形式（**図 A·3**）」を生成する．これらに加えて，Question Graph から正解を導き出すために参考にした関連ノード（言葉）が多いほど，より俯瞰的な視点から正解を考えて回答した，と考えられることから，回答者に俯瞰的な視点で問題を解かせることを目指して，問題中に提示する正解の関連情報を一部隠して出題する「虫食い形式（図 A·4）」も生成する．

4. 提案手法

任意の Answer 語句に対する 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 Japanese 等を利用することが可能である．

4.1 Question Graph 生成アプローチ

Question Graph は，Answer 語句の関連情報を RDF グラフ上のトリプル構造から探索し，それらの情報を Answer 語句の関係グラフとして視覚化することで生成する．まず，DBpedia 等の膨大な知識ベースから Answer ノード周りの情報を取得して抽出する．Answer ノード周りの情報として，有向グラフである知識ベース上の Answer 語句に該当するノードと 1-hop でリンクする隣接ノード，それ以降 hop 数を増やしたときの語句の全リンク構造を，指定したコストの範囲内で全取得し，それら抽出情報のうちその中の少ないノード数で俯瞰度が高くなるような語句のリンク構造を抽出する．具体的に表すと，Question Graph において Answer ノードからの 1-hop 隣接ノードの集合を N_1^Q，それ以降 hop 数を増やしたときの i-hop 目のノード集合を N_i^Q と表すとき，Answer ノードを *Answer* とし，語句間の意味的類似度等で求められる距離を返す *dist* 関数を用いて，$n_i \in N_i^Q$ から $n_{i+1} \in N_{i+1}^Q$ へのリンクがあるようなときに次を満たすようなグラフである．

$$dist(Answer, n_i) < dist(Answer, n_{i+1})$$

その後，抽出したリンク構造を Answer 語句の関連語句と語句間の関係性として視覚化することで，出題とすることを考える．すなわち，グラフ中の Answer 語句の表示を隠して，その周りの語彙やそれらの関係性から，Answer を推察するという形式の出題である．そして，Answer 語句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 ノード周りの語句との関係性から推測される Answer 語句であるため，文章化の必要性はない．図 1 のようにグラフの状態で出題とする．本アプローチでは以下の要件を考慮した探索および抽出手法を考案する．

4.1.1 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) グラフ構造の知識ベースの部分グラフである.
(2) 全体としてなるべく俯瞰的.
(3) Answer に該当する単語の数が少ない方がよい.
(4) 内容を把握できる程度の規模感.

4.1.2 Question Graph 生成手法

上述の要件を満たすような Question Graph を，知識ベースとして DBpedia を利用し，RDF グラフをもとに生成する手法を考案する.

単数回答形式の問題生成アルゴリズム

単数回答形式の Question Graph 生成アルゴリズムとして，次の Algorithm 1，2 を提案する．*Answer* 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あらかじめ制限する．探索回数は，Question Graph において，探索を開始する起点のノードである *Answer* を基準として，以降 1-hop ずつ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 (2) の俯瞰度を考慮して，Question Graph において，*Answer* ノードを起点として hop 数が増えるほどに，*Answer* ノード語句とその他ノード語句の距離がつねに遠くなるようなその他ノード語句を抽出する.

Main_Standard は，*Answer* 周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 からなる.

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 *dist* 関数により，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

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

Algorithm 3 $\backslash * * * \backslash$

1: **if** $count == 0$ **then**
2: $flg = True$
3: $n = \arg\max_{n' \in B} dist(target, n')$
4: $B = B \backslash \{n\}$
5: $nB = \{n\}$
6: **else**
7: $flg = True$
8: $n = \arg\max_{n' \in B} \{dist(target, n') + \sum_{n'' \in nB} dist(target, n'')\}$
9: $B = B \backslash \{n\}$
10: $nB = nB \cup \{n\}$
11: **end if**

においては，全祖先のノードについて *target* との距離の比較を，探索の深さ h まで再帰的に繰り返すことにより，*Answer* ノードからその他ノードまでのすべてのノード間の距離がつね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なるサブグラフを，提案の Question Graph として得られる.

なお，利用するモデルと提案アルゴリズムの性質上，グラフの広さ w が 2 以上の値で同一の深さにおけるノードが複数取得される場合，それらのノード間の *dist* による距離が近くなり似通った関連語句が抽出される可能性が考えられる．そのため，広さ w が 2 以上の場合に関しては，前述の Algorithm 2 および以降の提案アルゴリズムにおいて，コメント $\backslash * * * \backslash$ で示して囲んだ範囲の処理に置き換えて，実装上次の Algorithm 3 の処理を施した.

Algorithm 4 Main_Multiple

Input: KG G, Multiple Answers $Answers$, Depth h, Width w, Number of the answers nA
Output: KG
1: $N_{S01} = get\_far\_nodes\_h1(Answers, \{\}, G, w, nA)$
2: $N_S = N_S \cup N_{S01}$
3: $cnt = 0$
4: **for** $n_1 \in N_{S01}[1]$ **do**
5: $N_S = N_S \cup get\_far\_nodes\_hx(n_1, \{N_{S01}[0]\}, G, h, w, nA)$
6: **end for**
7: $M_S = get\_all\_links(N_S)$
8: **return** (N_S, M_S)

複数回答形式の問題生成アルゴリズム

次に，複数回答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 (3) における Answer に該当する語句の数を調整する．なお，以前の提案 [22] では，Answer に設定する 1 つの語句のみを入力として，Question Graph 上で Answer ノードにあてはまる語句が指定個以上になるまで探索する手法で生成した．今回は，指定個の語句を Answer として指定して Question Graph を生成するアプローチを新たに提案する．

まず，Algorithm 4 において指定個数 nA の $Answers$ を指定する．Algorithm 5 の get_far_nodes_h1 では，深さ $h = 1$ のときに指定個数のすべての $Answers$ について，共通する隣接ノード語句のみを返す関数を定義している．$neighbors$ 関数により，指定した各 $target$ について隣接ノード語句をすべて取得した後，共通するノードのみを抽出する．これらを距離最大の語句から順にグラフ上の隣接ノード語句として採用することで，なるべく俯瞰度を保つように作成する．$h = 2$ 以降については，Algorithm 6 の $get\_far\_links\_hx$ により，$Answers$ に指定した複数の語句についてすべてのノードの距離がつねに遠くなることを満たすように生成していく．

ただし，Algorithm 5 により，指定した複数の $Answers$ について共通の隣接ノード語句が発見されない場合，$Answer$ ノード以降のノードはつながらず，問題生成は失敗する．

組合せ回答形式の問題生成アルゴリズム

組合せ回答形式の Question Graph は，2 つの別々の問題を 1 つのグラフとして生成する．Question Graph の要件 (4) の “グラフの内容を把握できる程度の規模感” を考慮して，コンパクトな規模のグラフの生成を目指す．

まず，各問題が 1 つずつ持つ $Answer$ について，単数回答形式の生成アルゴリズムである Algorithm 1，2 の手順により，それぞれ 1 つのグラフを生成する．Algorithm 7 において，各問題が持つ $Answer$ を区別するために，$Answer\_0$ と $Answer\_1$ と表記する．その後，次の Algorithm 8 に示す get_pair_graph 関数により，$Answer\_0$ ノードから $Answer\_1$ ノードに至る道のりを探索するアルゴリズムを

Algorithm 5 get_far_nodes_h1

Input: Node $Targets$, Set of ancestor nodes $Ancestors$, KG G, Width w, Number of the answers nA
Output: Set of nodes N_{S01}
1: $k = 0$
2: **for** $target \in Targets$ **do**
3: $N_{S01}[0][k] = \{target\}$
4: $k = k + 1$
5: **end for**
6: **for** $direction \in \{IN, OUT\}$ **do**
7: $count = 0$
8: **for** $target \in Targets$ **do**
9: $B'_{count} = neighbors(target, direction)$
10: $count = count + 1$
11: **end for**
12: $B = \prod_{k=0}^{nA-1} B'_k$
13: $count' = 0$
14: **while** $count' < w$ AND $0 < |B|$ **do**
15: \ * * * \
16: $n = \arg\max_{n' \in B} \sum_{k=0}^{nA-1} dist(N_{S01}[0][k], n')$
17: $B = B \backslash \{n\}$
\ * * * \
18: $count' = count' + 1$
19: $N_{S01}[1] = N_{S01}[1] \cup n$
20: **end while**
21: **end for**
22: **return** N_{S01}

Algorithm 6 get_far_nodes_hx

Input: Node $target$, Set of ancestor nodes $Ancestors$, KG G, Depth h, Width w, Number of the answers nA
Output: Set of nodes N
1: $N = \{target\}$
2: **if** $|Ancestors| == h$ **then**
3: **return** N
4: **end if**
5: **for** $direction \in \{IN, OUT\}$ **do**
6: $count = 0$
7: $B = neighbors(target, direction)$
8: **while** $count < w$ AND $0 < |B|$ **do**
9: $flg = True$
10: \ * * * \
11: $n = \arg\max_{n' \in B} dist(target, n')$
12: $B = B \backslash \{n\}$
13: \ * * * \
14: $count' = 0$
15: **for** $n_j \in Ancestors$ **do**
16: **if** $count' \neq 0$ **then**
17: **if** $dist(n_j, n) < dist(n_j, target)$ **then**
18: $flg = False$
19: **end if**
20: **else**
21: **if** $dist(n_j, n) < \sum_{k=0}^{nA-1} dist(n_j, Ancestors[0][k])$ **then**
22: $flg = False$
23: **end if**
24: $count' = 1$
25: **end if**
26: **end for**
27: **if** flg **then**
28: $count = count + 1$
29: $N = N \cup get\_far\_nodes\_hx(n, Ancestors \cup \{target\}, G, h, w)$
30: **end if**
31: **end while**
32: **end for**
33: **return** N

通して，2 つの別々のグラフを 1 つのグラフに結合することを目指す．line.7 では，$Answer\_0, Answer\_1$ 間の最短ルートを求める $get\_shortest\_routes$ 関数を定義して，追加ノードができる限り少なくなるような距離最小の道のり RN_{S_01} を複数取得する．なお，RN_{S_01} として得られるのは，$Answer\_0$ と $Answer\_1$ の間の道のり上にある中間ノードの組合せの集合である．また，実装上は反復深化により探索し，初期値として道のりの最大距離を line.4 の探索の探索規模 $h' \times 2$ を超えない範囲で指定して全探索することにより，距離最小順に複数の道のりを取得しており，

Algorithm 7 Main_Pair

Input: KG G, $Answer\_0$, $Answer\_1$, depth h, width w, number of the answers nA
Output: KG
1: $N_{S\_0} = get\_far\_nodes(Answer\_0, \{\}, G, nA, w)$
2: $N_{S\_1} = get\_far\_nodes(Answer\_1, \{\}, G, nA, w)$
3: $M_{S\_0} = get\_all\_links(N_{S\_0})$
4: $M_{S\_1} = get\_all\_links(N_{S\_1})$
5: $(N_S, M_S) = get\_pair\_graph(Answer\_0, Answer\_1, N_{S\_0}, N_{S\_1}, M_{S\_0}, M_{S\_1})$
6: **return** (N_S, M_S)

Algorithm 8 get_pair_graph

Input: Each answers $Answer\_0$, $Answer\_1$, Each nodes N_{S_0}, N_{S_1}, Each links M_{S_0}, M_{S_1}
Output: (Ns, Ms)
1: **if** $get\_all\_links(N_{S\_0} \cup N_{S\_1}) \cap M_{S\_0} \neq \emptyset$ OR $get\_all\_links(N_{S\_0} \cup N_{S\_1}) \cap M_{S\_1} \neq \emptyset$ **then**
2: **return** $(N_{S\_0}, M_{S\_0}) \cup (N_{S\_1}, M_{S\_1})$
3: **else**
4: $h' = h + x, w' = y$* 道のりの探索規模をあらかじめ設定 *\
5: $N_{S\_0\_all} = get\_far\_nodes(Answer\_0, \{\}, G, h', w')$
6: $N_{S\_1\_all} = get\_far\_nodes(Answer\_1, \{\}, G, h', w')$
7: $RN_{S\_01} = get\_shortest\_routes(Answer\_0, Answer\_1, N_{S\_0\_all}, N_{S\_1\_all})$
8: $N_{S\_01} = \underset{rN_S \in RN_{S\_01}}{\arg\max} \{\sum_{n' \in rN_S} dist(Answer\_0, n') + dist(Answer\_1, n')\}$
9: $M_{S\_01} = get\_all\_links(N_{S\_01})$
10: **return** $(N_{S\_0} \cup N_{S\_1} \cup N_{S\_01}, M_{S\_0} \cup M_{S\_1} \cup M_{S\_01})$
11: **end if**

最小の道のりのみを抽出している．このとき，line.4 における道のりの探索規模 h' と w' は，想定する探索コストに応じて x と y に適宜設定しておく．その後，取得した距離最小となる複数の道のりのうち，実際に Question Graph として採用するものを 1 つに絞るために，RN_{S_01} の各道のりにおけるすべての中間ノードについて，$Answer\_0$ と $Answer\_1$ 間の $dist$ 関数による距離の総和で最も遠くなる道のり上にある中間ノードを採用する．

ただし，Algorithm 8 を通して各 $Answer$ 間の道のりが発見されない場合，1 つのグラフとして結合することができないため，問題生成は失敗する．

虫食い形式の問題生成アルゴリズム

虫食い形式は，前述 3 つの回答形式の Question Graph において，一部虫食いとして語句を隠したノードを埋め込んで生成する形式の出題である．この形式では，回答時に虫食いノードに該当する語句を推測することで，Answer ノードの語句の推測につながるという回答シーンを想定するものである．ただし，前述のアルゴリズムのような語句間の $dist$ 関数による最大距離のみで構成されるグラフのみでは，虫食いノードに該当する語句の推測が困難になることも考えられる．そこで，$dist$ 関数による距離最大と距離最小の語句について両者を均等に採用し，距離最小の語句のうち一部を虫食いノードとして隠すという生成アプローチを提案する．

ここでは，提案の 3 つの回答形式のうち，単純な回答形式である単数回答形式のアルゴリズムに虫食い形式を適用する場合のアルゴリズムとして Algorithm 9 を定義する．

Algorithm 9 Main_Standard_Blank

Input: KG G, $Answer$, Depth h, Width $w(w = 2i, i \geq 1)$
Output: KG
1: $N_{S\_far} = get\_far\_nodes(Answer, \{\}, G, h, w/2)$
2: $N_{S\_near} = get\_near\_nodes(Answer, \{\}, G, h, w/2)$
3: $N_S = N_{S\_far} \cup N_{S\_near}$
4: $M_S = get\_all\_links(N_S)$
5: **return** (N_S, M_S)

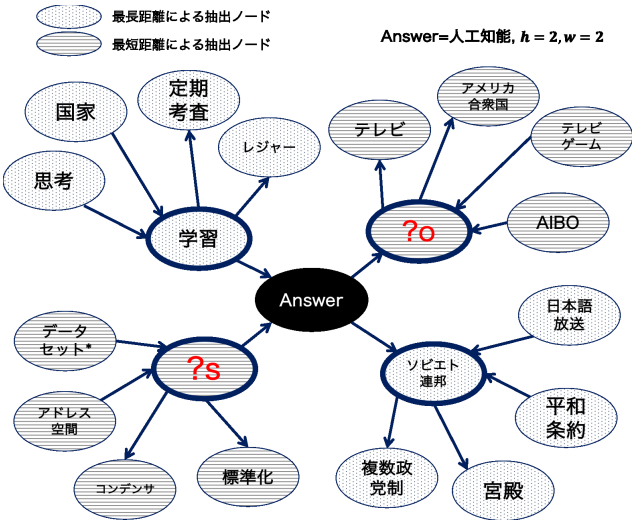

図 2 単数回答形式における虫食い形式のグラフイメージ（※）
Fig. 2 An example image of fill-in-the-blank questions in single answer.
（※）図中の隠しノードに入る語句として “?s”：「コンピュータ」，“?o”：「ロボット」が該当する．

虫食い形式では，まず広さ w は正の偶数を設定し，Algorithm 9 において，Algorithm 2 で定義した get_far_nodes と get_near_nodes の両関数を用いて，各々から同数のノードを採用する．get_near_nodes 関数は，Algorithm 2 の line.10 において，argmax ではなく argmin として $target$ の隣接ノード語句を探索することで，$dist$ 関数による最短距離の語句を求めるものである．

図 2 では，Algorithm 9，3 を用いて生成する単数回答形式における虫食い形式のグラフイメージを示している．実際の出題においては，Answer ノードの隣接ノード語句のうち距離最小の語句を隠すこととする．図のイメージ例のように，虫食いノードは該当語句の代わりに「?o」と「?s」等，「?」を用いて語句を隠して表記する．

図 3 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 $target$ の隣接ノード語句を抽出していく．まず $h = 1$ のとき，$target$ である $Answer$ の全隣接ノード語句を取得し，距離が最大の赤色で示す隣接ノード語句 2 つを抽出する．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語句を求め，祖先の赤色ノードとの距離について $target$ ノードと比較してつねに大きいとき，青色で示す隣接ノード語句として採用する．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

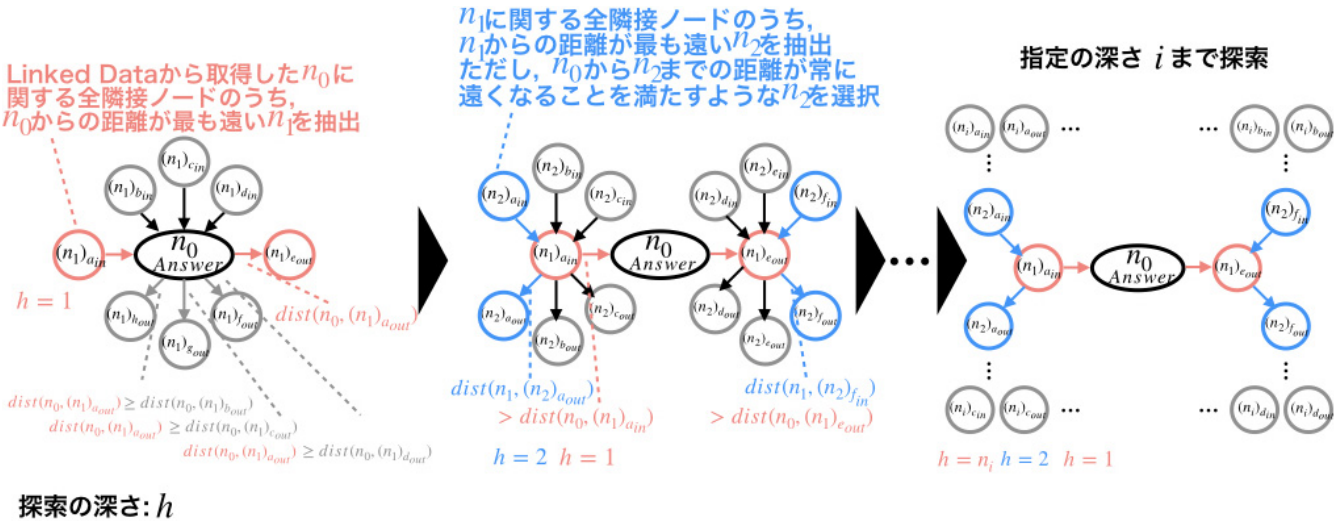

図 3 *dist* 関数を用いた知識ベースからのリンク構造抽出イメージ

Fig. 3 An image of extracting the link structures from the knowledge base using *dist* function.

Answer からつね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，語彙のメジャーな度合いも考慮することとした．メジャーな度合いを major 値として，そ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）[*3]を利用し，教科書サブコーパス（OT）の語句を 1.0，図書館および出版サブコーパス（LB，PB）の語句を 0.5，それ以外の語句を 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句がよりメジャーな語句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合せ結果中にコーパス OT の語句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．

4.2 Distractors 生成アプローチ

Distractors は，Question Graph において Answer ノードに該当しない語句であり，すなわち Answer とのリンク構造のいずれかを満たしてもすべてを満たさないようなノード語句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．

4.2.1 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）Question Graph に対して不正解であること．

(2) 一見して不正解と明らかに分かるものを避けたい．

4.2.2 Distractors 生成手法

要件 (2) 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 語句と類似した語彙集合を Distractors とする必要がある．ここで Answer の類似語句は，知識ベース上におけるリンクデータの構造も Answer 語句と類似していることが予想される．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 ノード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 ノード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 の候補と見なすことができる．

さらに，Lorenz ら [5] の取得方法を参考にして，取得した候補のうち，DBpedia 上で Answer 語句と同じクラスに属する語彙に絞り込む．なお，Answer 語句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 語句がクラス 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 property として得た object 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，Answer ノードが Question Graph 上のノードとして持つリンク構造と Dis-

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

tractors が Question Graph 上のノードとして持つリンク構造の類似度計算をする．候補と Answer ノードの隣接ノード語句との距離 Dd，Answer ノード語句と隣接ノード語句との距離 Da とを比較したときの差 $|Dd-Da|$ の小ささと，候補の語句の major 値 Md の大きさの和 $|Dd-Da|+Md$ が上位のものから採用するようにすることで，一見して不正解とは分かりにくい選択肢生成を目指す．

5. 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，Answer として設定する語句は，語彙コーパスの教科書（OT）上にある語句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しないため，DBpedia 上の語句がすべて未設定のカリキュラム上に存在するものと仮定して用いることとする．

5.1 Question Graph 生成

Question Graph は前章で述べたとおり，Answer 周りの関連情報として Answer 語句のリンク構造を知識ベース上から SPARQL クエリを介して複数取得して抽出したのち，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，語句間の構文的・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[*4] [23] 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見なして定義した．グラフの描画は，取得した Answer 周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 D3.js により描く．

生成アプローチ上利用する語彙コーパスや DBpedia，word2vec 学習済みモデルにおける表記ゆれについては，Wikipedia ページ主題名について Wikipedia によりあらかじめ定義された別名一覧を取得可能な Redirects を利用して，各々の表記ゆれに対応するリストを作成することでグラフ生成した．なお，Question Graph 上で実際に表示する語句は Wikipedia ページの主題名に統一した．また，選択肢の語句間の同義語対策として，日本語 WordNet による上位語・下位語の関係を利用した類似語検索を行い，Distractors 候補の語句間の類似語を候補から除いた．

さらに，出題で表示するグラフ上のリンクラベルの種類についても配慮した．リンクラベルはグラフ上のリンクノード間の関係性の種類や関連事項を表している．なるべく具体的な関係性が示されている方がその関連事項をもとに Answer に該当する語句を推測することになり，俯瞰的な視点で解くことにつながるため，問題として好ましいと

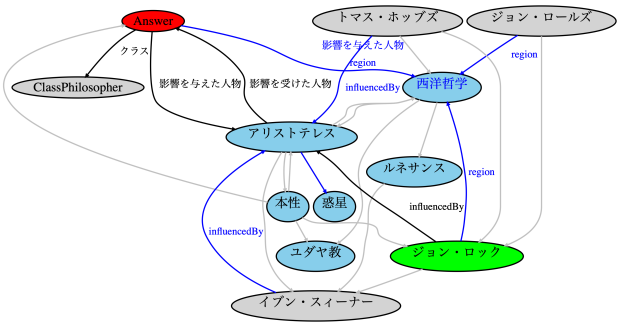

図 4 Answer=“ソクラテス” に設定したときの Question Graph
Fig. 4 The Question Graph when the when answer is “Socrates”.

考えられる．そのため，知識ベース上のリンクの種類を考慮して，“wikipagewikiLink” のような関係性が具体化されていないものを除いたリンクラベルの優先度を高くして抽出することとする．

また，虫食い回答形式の生成においては，一部の語句を隠した虫食いノードに入る語句を推測させるような問題にするために，*target* 語句との距離が最短の隣接ノード語句を採用する過程においては，DBpedia 上から取得される Abstract 情報に含まれる語句を優先的に採用するという処理を施す．

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 Graphviz により描画したものを**図 4** に示す．なお，各ノードの色について，Answer ノードを赤色，クラスノードを緑色，語彙コーパス上に存在する語句を青色および緑色，いずれにも該当しない語彙を灰色で表している．その他の複数回答形式，組合せ回答形式，虫食い単数回答形式の Question Graph 生成例についても，付録内のそれぞれ図 A·1，図 A·2，図 A·3，図 A·4 に示した．

5.2 Distractors 生成

前述のアプローチのとおり，事前に前項の方法で生成した Question Graph を利用して，Answer ノードのリンク構造を利用して，Answer 語句と同じクラス 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．

なお，Distractors 生成においても語彙コーパス，DBpedia，word2vec 学習済みモデルにおける表記ゆれについては，Wikipedia 上で定義された Redirects を利用した表記ゆれ対応リストを作成して用いた．Distractors とする語句についても Wikipedia ページの主題名に統一した．加えて，同義語への対処として，WordNet[*5] を利用した類義語検索による検証フェーズを設けた．

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．

なお，これらの提案手法により生成した本被験者実験用

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

*5 https://wordnet.princeton.edu

表 1 Answer=“ソクラテス” に対する Distractors 候補の抜粋
Table 1 An extract of candidates of the distractors when the answer is “Socrates”.

Distractors 候補	共通ノード数	構造の類似度（※）
ヘラクレイトス	3	5.63
カール・マルクス	3	3.95
バートランド・ラッセル	3	3.77
シャルル・ド・モンテスキュー	3	3.67
バールーフ・デ・スピノザ	3	3.64
イマヌエル・カント	2	3.63
セーレン・キェルケゴール	2	2.80
フリードリヒ・ニーチェ	2	2.78

（※）4 章の Distractors 生成アプローチにおける構造の類似度計算値．$\sum_{n \in comNodes}(1 - |Dd_n - Da_n|) + Md$

の問題は，評価実験用の出題システムとして Web 上で現在公開中である[*6]．

6. 評価と結果

前述のアプローチにより生成した Question Graph と Distractors の各要件と照らし合わせた評価項目とシステム全体の評価項目を示し，評価結果を以下にまとめる．なお，本評価における Question Graph について，単数回答形式および複数回答形式の深さを $h = 2$，広さを $w = 2$，組合せ回答形式は深さを $h = 1$，広さを $w = 2$ に設定して生成した．Answer に対する Distractors の数は，以下で述べる比較用の既存問題の形式にあわせて，“単数回答” が Answer1 つと Distractors3 つ，“複数回答” が Answer2 つと Distractors4 つ，“組合せ回答” が 2 つの問題について Answer 各 1 つ，Distractors 各 2 つに設定した．

なお，本評価実験においては，被験者実験と機械的実験を以下の内容で行った．

被験者実験における既存問題は，大学入試センター試験の過去 5 年分（平成 26 年～平成 30 年）[*7]より，対象の科目を地理歴史と理科として，提案の 3 つの出題形式の問題を抜粋して利用した．提案の各回答形式において抜粋した比較問題の合計は，単数回答形式 12 問，複数回答形式 3 問，組合せ回答形式 26 問である．また，生成する選択肢の数は，比較問題における選択肢の数に合わせた．実験において，抜粋した比較問題の解答一覧から被験者により知っている言葉を指定個数尋ねて選択させ，選択語句を Answer として生成した提案問題について，対応する比較問題とその俯瞰度を比較させた．なお，被験者 1 人あたりの比較数は，単数解答形式，複数回答形式，組合せ回答形式のそれぞれに対して 8 問，3 問，10 問に設定した．ただし，単数回答形式のうち半数を虫食い形式による出題とし，回答時の俯瞰度について参考ノード数を回答させて比較した．俯瞰度の主観評価に加えて，Distractors の評価項目②について，被験者が正解を知らないと想定される語句を Answer に設定した問題を出題して，その正答率を測定した．出題は組合せ回答形式を別に 10 問生成して出題した．今回，被検者として教職免許状所持者と学生に出題し，各要件について設定した評価項目を満たすことを確かめた．なお，今回の評価実験において，被験者の教職免許状に対する科目や学校区分は関係なく，「教職」という範囲での評価と見なして実施した．

機械的実験では，語彙コーパスの 551,136 語から表記ゆれも考慮したうえで，DBpedia と word2vec 学習済みモデル上に存在する語句のみをランダムに採用し，複数語句（“単数回答”：434 語，“複数回答形式”：168 語，“組合せ回答”：174 語）を Answer に設定した生成問題を評価対象として用いた．Question Graph については，提案の *dist* 関数を利用してリンク構造を抽出する手法で生成した問題（提案手法）と，*dist* 関数を利用せずランダムにリンク構造を抽出する方法（比較手法）により生成したグラフについて，各評価項目に沿って評価，比較した．Distractors については，提案の Answer ノードが持つリンク構造を利用して生成した Distractors（提案）と，Answer 語句の属するクラスからランダムに語句を選択して生成する方法（比較手法）について，各評価項目に沿って評価および比較をした．なお，本実験では提案と比較両手法において，生成問題のうち Question Graph 上の合計ノード数が各形式の最大ノード数の半数以上で，かつ Distractors が指定個数生成されたもののみを評価対象として用いた．

6.1 Question Graph の機械的実験による客観評価項目と評価結果

Question Graph の要件を照らし合わせた機械的実験による客観評価項目を以下にあげる．

Question Graph の客観評価項目

① 整合性...［要件 (1)］
- Answer が真の解であること

② 俯瞰度...［要件 (2)］
- 各語句が属するクラスの横断度
- 各語句が分類される科目の横断度
- 各語句が分類されるジャンルの横断度

③ 特定性...［要件 (3)］
- Answer に該当する語句が少ないこと

項目①では，生成した Question Graph 上の Answer ノードに Answer 語句が正しくあてはまるかどうかである．これはグラフの生成過程で必ず満たすことは自明である．

項目②では，俯瞰度を評価するために本論文において我々が設定した 3 つの指標に基づき，機械的実験により評価する．指標の 1 つ目 “各語句が属するクラスの横断度” は，Question Graph 生成に利用した知識ベース上のクラス

[*6] http://www.ohsuga.lab.uec.ac.jp/okuhara/
[*7] https://www.dnc.ac.jp/center/kakomondai.html

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．2 つ目の “各語句が分類される科目の横断度” と 3 つ目の “各語句が分類されるジャンルの横断度” については，前述の語彙コーパス中の教科書サブコーパス（OT）における 9 つの科目分類と出版サブコーパス（LB，PB）における NDC（日本十進分類法）の第 1 区分に基づいた 9 つの書籍分類に基づいて評価する．

項目③では，生成した Question Graph に対して，Answer ノードに該当するような Answer 以外の語句が少ないかどうかを調べる．これは，Question Graph 上で Answer ノードの持つ全リンク構造を，すべて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．

6.1.1 整合性

前述のとおり，Question Graph の生成過程より Answer の整合性は自明であるが，評価に際して Answer ノードの全隣接ノード語句の関係性を用いて，Answer ノードに該当する語句を DBpedia 上から取得した結果，評価対象の全問題について Answer 語句が含まれていたため，本実験において整合性が確認された．

6.1.2 俯瞰度

機械的実験による俯瞰度に関する 3 つの評価項目について，機械的に複数個生成した各 Question Graph 内における，クラスの横断数について**表 2** に示す．

Question Graph の要件 (2) “俯瞰度” に対する評価指標としてあげたクラス，科目，ジャンルの横断度の機械的実験における結果である表 2 について，クラス横断度は提案手法と比較手法のいずれも 1 グラフにおけるクラス横断数は平均でおよそ 1，2 個程度であり，大きな差が見られなかった．ただし，各語句は DBpedia 上の複数のクラスに所属するため，集計にあたり 1 つの語句について取得された複数の最下層のクラス一覧のうち一番はじめに取得されたクラスのみを比較に利用している．一方，語句コーパスを利用した科目横断数は，いずれの回答形式についても提案が比較より 0.94〜2.57 科目分大きい結果となっており，ジャンル横断度も提案が比較をすべて上回り，0.700〜2.13 ジャンル分大きくなった．提案手法である語句間の *dist* 関数によるノードの語句間の距離を考えたリンク構造抽出アプローチがより多くの科目およびジャンルを含む問題生成に貢献したと考えられる．

6.1.3 特定性

今回生成したすべての Question Graph について，Answer ノードに該当する語句を DBpedia 上で確認したところ，Answer 語句を除いた Answer ノードへの語句の該当数は**表 3** のとおりであった．提案において，いずれの回答形式についても “虫食い” ありの方が “虫食い” なしに比べておよそ 10 語分特定性が小さい結果となった．それに比べると，提案比較と比較手法とでは特定性にそれほど大きな差はみられなかった．

表 2 Question Graph の俯瞰度の各指標に基づく客観評価
Table 2 An objective evaluation result of Question Graph based on each indexes of “broad perspective”.

形式	クラス数		科目数		ジャンル数	
	提案	比較	提案	比較	提案	比較
単数回答	1.67	2.03	3.43	1.90	4.32	3.43
虫食い単数回答	2.12	2.03	4.47	1.90	5.56	3.43
複数回答	1.66	1.87	3.74	2.47	4.63	3.71
虫食い複数回答	2.16	1.87	4.72	2.47	5.70	3.71
組合せ回答	1.32	1.52	3.22	2.28	3.85	3.15
虫食い組合せ	1.39	1.52	3.74	2.28	4.46	3.15

（※）各指標において 1 問あたりの平均値を算出．

表 3 Question Graph の特定性の客観評価
Table 3 An objective evaluation result of specificity of Question Graph.

形式	Answer ノードの該当語句数（※）	
	提案	比較
単数回答	18.8	4.71
虫食い単数回答	3.58	4.71
複数回答	14.5	4.54
虫食い複数回答	5.11	4,54
組合せ回答	(13.9, 14.2)	(3.80, 4.96)
虫食い組合せ	(3.07, 4.15)	(3.80, 4.96)

（※）Question Graph 上で Answer ノードに該当する Answer 語句以外の語句の合計数を集計．各形式で 1-Question Graph あたりの平均値を算出．

6.2 Question Graph の被験者実験による主観評価項目と評価結果

Question Graph の要件を照らし合わせた被験者実験による主観評価項目を以下にあげる．

Question Graph の主観評価項目

① 俯瞰度...［要件 (2)］
- グラフの俯瞰度の主観評価

② 可読性...［要件 (4)］
- グラフの可視性

項目①の俯瞰度については，多肢選択式問題として俯瞰的な情報を包む度合いの差を測ることを目的として被験者実験に基づく主観評価も行い，提案の生成問題と既存問題について俯瞰度の 4 段階評価により比較させる．また，問題自体の俯瞰度だけでなく回答時の俯瞰度として，Answer を回答する際に参考にしたノード数を，出題形式における虫食いの有無で比較することで，実際の回答のための俯瞰的な視点の程度に差異があるかどうかも確かめる．

項目②の可読性の評価は，Question Graph の規模感を

主観評価により被験者に5段階評価させる．

6.2.1 俯瞰度

俯瞰度の主観評価の結果を**表 4** に示す．さらに，表4に示す“提案の問題”をより俯瞰度が大きいと評価した回答については，**表 5**，**表 6** において，それぞれ俯瞰度の差の程度と内容の差の程度について4段階評価の結果をまとめて示した．

俯瞰度の主観評価結果について，各回答形式における被験者ののべ回答数は，“単数回答”が76回答形式，“複数回答形式”が57回答，“組合せ回答形式”が189回答であった．表4では，いずれの回答形式においてもおよそ半数の被験者が，“提案の問題”の方が“比較の既存問題”よりも“より俯瞰度が大きい”と評価した．

表5，表6には，表4において提案の問題を俯瞰度が大きいと評価した被験者について，比較問題との“俯瞰度の差の程度”と“‘内容の差の程度”の4段階評価の結果をまとめた．いずれも7割〜8割の被験者が「差がある」と回答したことから，提案の問題の方がより俯瞰度が大きいと評価された生成問題は，全体としてその俯瞰度にも内容にも既存問題との差がおよそある印象をいだいたとみられる．

また，**表 7** において被験者が回答時に参考にしたノード数を見ると，各形式において3，4個程度の語句が参考にされていた．特に，“単数回答”と“虫食い単数回答”とを比較して，虫食いのノードがある方が参考ノード数が多く，より俯瞰的な視点で問題を解くことにつながったとみられる．“虫食い単数回答”の標準偏差が大きいのは，Answerを回答するために虫食いノードに入る語句を推測する必要のある問題については参考ノード数が多くなったが，虫食いノードを推測しなくても正解を導き出せるような虫食い問題は，参考ノードが少なくなりばらつきが出たと考えられる．

6.2.2 可読性の評価

要件(4)の“グラフの規模感”について，**図 5** の被験者実験における5段階の主観評価から，被験者の半数がちょうど良い規模であると回答している．一方，4割弱が大きいあるいは大きすぎるとの評価であり，被験者によってはやや大きめに感じる程度の規模感であった．要件(3)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，評価実験で生成したすべてのグラフについて，いずれもAnswerノードに該当する語句がAnswer語句以外存在せず，要件を満たしていた．

6.3 Distractorsの機械的実験による客観評価項目と評価結果

Distractorsの要件を照らし合わせた評価項目を以下にあげる．

表 4 俯瞰度の主観評価

Table 4 A subjective evaluation result of the degree of “broad perspective” in proposal and comparative questions.

出題形式	より俯瞰度が大きいと評価された割合（※）		
	提案の問題	変わらない	比較の問題
単数回答	0.48	0.26	0.26
虫食い単数回答	0.65	0.25	0.10
複数回答	0.68	0.13	0.18
組合せ回答	0.50	0.22	0.28

（※）収集できた回答のみ集計．各回答について全問題に対する平均値を算出．

表 5 提案の問題と比較の問題との俯瞰度の差の程度の主観評価

Table 5 A subjective evaluation result of the difference of degree of “broad perspective” between proposal and comparative questions.

出題形式	4段階評価の割合（※）			
	1	2	3	4
単数回答	0.00	0.27	0.28	0.45
虫食い単数回答	0.00	0.18	0.32	0.50
複数回答	0.00	0.14	0.44	0.42
組合せ回答	0.0056	0.16	0.57	0.26

（※）提案の問題がより俯瞰的であると評価した回答より集計．
質問「俯瞰度の差はどの程度であると感じますか？」に対する回答「1：まったく差がない，2：あまり差がない，3：すこし差がある，4：とても差がある」

表 6 提案の問題と比較の問題との内容の差の程度の主観評価

Table 6 A subjective evaluation result of the difference of degree contents between proposal and comparative questions.

出題形式	4段階評価の割合（※）			
	1	2	3	4
単数回答	0.00	0.23	0.36	0.41
虫食い単数回答	0.014	0.25	0.40	0.33
複数回答	0.00	0.17	0.33	0.51
組合せ回答	0.012	0.28	0.54	0.16

（※）提案の問題がより俯瞰的であると評価した回答より集計．
質問「内容の差はどの程度であると感じますか？」に対する回答「1：まったく差がない，2：あまり差がない，3：すこし差がある，4：とても差がある」

表 7 被験者実験における回答時の参考ノード数

Table 7 The number of nodes referred for answering Question Graph based on a subject experiment.

出題形式	回答時の参考ノード数（※）	
	平均	標準偏差
単数回答	3.22	0.32
虫食い単数回答	3.80	1.24
複数回答	3.74	0.73
組合せ回答	4.40	0.84

（※）収集できた回答のみ集計．各回答について全問題に対する平均値を算出．

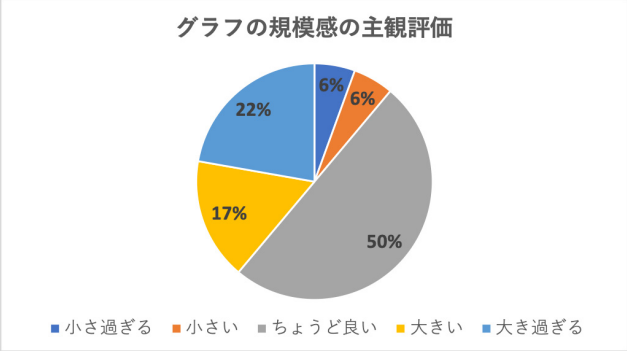

図 5 グラフの規模感の主観評価（※）
Fig. 5 A subjective evaluation of readability of Question Graph.

（※）質問「提案の「Question グラフ」の規模感はどうでしたか.」に対する 5 段階評価

Distractors の客観評価項目

① Question Graph に対して不正解であること... [要件 (1)]
② 選択肢間の類似度が大きい... [要件 (2)]

項目①の Question Graph に対して不正解であることは，Distractors の生成過程から自明である．

項目②の選択肢間の類似は，Distractors の要件 (3) の“一見して不正解とは分からない選択肢であること”についての検証である．検証では構文的側面と意味的側面の両者から行う．構文的検証では，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．すなわち，Answer 語句との字面的な類似度を測るためのものである．構文解析ツール CaboCha[*8]を用いて，係り受けと品詞の両者が一致するかを検証する．意味的検証では，選択肢間の意味的な類似度を比較する．関連研究において，Pho らの文献 [5] における評価モデルの指標，あるいは文献 [24] 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” 等を利用し，類似度を算出して比較する方法がある．今回は，問題生成の際に利用した word2vec の事前学習済みモデルによる similarity の算出を行う．

6.3.1 Question Graph に対して不正解であること

前述のとおり，Distractors の生成過程より Question Graph に対して不正解であることは自明であるが，Question Graph 上の Answer ノードとリンクを持つノード語句との関係性を用いて，Answer ノードに該当する語句を DBpedia 上から取得した結果，評価対象の全問題について Distractors 語句が含まれていなかったため，評価上は改めて Distractors が不正解であることが確認された．

[*8] https://taku910.github.io/cabocha/

表 8 Distractors の Answer との類似度比較による客観評価
Table 8 An objective evaluation of the distractors based on similarity to the answer.

Q.	pattern 一致率		similarity 平均	
	提案	比較	提案	比較
単数回答	0.553	0.265	0.0720	0.0863
虫食い単数回答	0.580	0.265	0.0986	0.0863
複数回答	(0.686, 0.753)	(0.353, 0.407)	(0.0483, 0.0183)	(0.103, 0.0453)
虫食い複数回答	(0.673, 0.738)	(0.353, 0.407)	(0.0672, 0.0273)	(0.103, 0.0453)
組合せ回答	(0.302, 0.288)	(0.207, 0.123)	(0.0824, 0.0812)	(0.0309, 0.0224)
虫食い組合せ	(0.302, 0.288)	(0.207, 0.123)	(0.0924, 0.0812)	(0.0309, 0.0224)

6.3.2 選択肢間の類似度

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，語句の形態素の品詞構成パターンの比較 “pattern 一致率” と word2vec の事前学習済みモデルによる類似度比較 “similarity 平均” の結果を**表 8** に示す．“pattern 一致率” については，1 つの問題において Answer 語句の係り受けと品詞の両者が一致する確率を，1 つあたりの Distractor について平均値を算出したものである．“similarity 平均” については，本手法の *dist* 距離計算で利用した word2vec 学習済みモデルにおける similarity を用いて，Answer 語句との類似度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，表の各値は 1-Distractor あたりの類似度の値となっている．

評価項目③の “選択肢間の類似度が大きい” かどうかについて，機械的実験により Answer 語句との構文パターンと word2vec 学習済みモデルによる similarity 値を比較して，表 8 のような結果となった．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．構文パターンとして “pattern 一致率” については，いずれの回答形式においても比較の方が提案より一致率が大きい結果となった．“similarity 平均” については形式によって結果が異なり，“単数回答” は比較の方が，“虫食い単数回答” は提案の方が大きく，“複数回答” と “虫食い複数回答” は比較の方が大きく，“組合せ回答” と “虫食い組合せ回答” は提案の方が大きくなった．特に，各回答形式について虫食いの有無で比較すると，いずれも虫食いありの形式の方が similarity 値が大きい結果であった．これについては，Distractors 生成には Question Graph における Answer ノードのリンクノードを利用していることから，虫食いの有無によるグラフ生成のアプローチの違いが起因していると考えられる．虫食いなしの方では Answer 語句からつねに *dist* 関数による距離最大のノード語句を抽出するが，虫食いありでは *target* の語句から距離最大と最小の隣接ノード語句を半数ずつ抽出して生成する．このことから，Question Graph において Answer ノードの隣接ノード語句と Answer 語句との距離が近いほど，提案手法のアプローチによりそのグラフから生成される Distractors は意味的に近い語句になる傾向にあったと考えられる．

6.4 Distractors の人を対象にした評価項目と評価結果

Distractors の要件を照らし合わせた評価項目を以下にあげる．

Distractors の人を対象にした評価項目

① 正解を知らない人の正答率が選択肢数分の 1 程度...［要件 (2)］

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

被験者実験において，被験者が正解を知らないと想定される語句を Answer に設定した出題を別に 10 題用意し，それらについて回答を得る．その際，被験者には回答後に正解を表示させたうえで「正解の語句自体を聞いたことがあったか」と「グラフを見て答えにその語句が入ることが分かったか」を 4 段階評価で尋ね，その回答もあわせて結果をまとめる．

6.4.1 正答率

被験者実験における回答結果を示す．今回の実験においては，合計 20 名の被験者から回答が得られ，うち 5 名が教職免許状所持者であった．なお，Distractor 必須要件 (1) 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．

6.4.2 被験者実験による正答率の比較

表 9 では各回答形式における各選択肢の選択率の全体を，**表 10** では，正解の語句に対する認知度に応じた正答率をまとめた．要件 (2) の “一見して不正解と明らかに分からないこと” の評価について項目②では，すべての被験者は正解を知らないという想定の下で，“各問題に対する正答率がおよそ選択肢数分の 1” すなわち $1/4 = 0.25$ 程度であることが望ましい．表 9 では，組合せ回答形式 10 問を被験者に出題したときの正答率を Answer の認知度別にまとめた．“語句自体の認知度” において，Answer の語句自体を「まったく知らない」と回答した被験者の正答率が 0.40 であり，正答率の評価指標 0.25 を 60%上回る結果となった．特に Q.4，5 問目の正答率が 9 割以上である一方で Q.2，8，9 は 2 割以下とばらつきがあることから，問題によっては一見して不正解だと分かる選択肢が生成された失敗例もあった．また，“グラフ上の認知度” において「まったく分からなかった」と回答した被験者の正答率は 0.26 であり，こちらは正答率の評価指標 0.25 を 4%わずかに上回った．組合せ問題の選択肢は比較問題の形式に合わせて 3 つの Distractors を生成しており，すなわち 2 つの各 Answer 語句に対応する Distractors 語句を 2 つずつ生成した．表 9 の正答率測定実験における各選択肢の選択率より，Q.4，5 は “(D0, D1)” の 2 つの Distractor どうしの組合せからなる選択肢の選択率が 0.00%であることから，

表 9 正答率測定評価実験における各選択肢の選択率（全体）
Table 9 The selectivity of each choices on proposal questions.

Q.	Answer 選択肢	Distractors 選択肢		
	(A0, A1)	(A0, D1)	(D0, A1)	(D0, D1)
1	0.44	0.50	0.00	0.056
2	0.39	0.17	0.33	0.11
3	0.61	0.22	0.17	0.00
4	0.84	0.053	0.11	0.00
5	0.95	0.053	0.00	0.00
6	0.26	0.16	0.47	0.11
7	0.37	0.53	0.053	0.053
8	0.16	0.21	0.32	0.32
9	0.47	0.11	0.16	0.26
10	0.32	0.16	0.32	0.21

(A0, A1)：2 つの正解の語句どうしの組合せ
(A0, D1)，(D0, A1)：正解の語句と不正解の語句の組合せ
(D0, D1)：2 つの不正解の語句どうしの組合せ

表 10 正答率測定実験における正解の語句に対する認知度別の正答率の割合
Table 10 The validity of proposal questions based on degree of the recognition of answer in subject experiment.

Q.	語句自体の認知度 (※ 1)				グラフ上での認知度 (※ 2)			
	1	2	3	4	1	2	3	4
1	0.40	0.40	0.50	1.0	0.00	0.71	0.50	NA
2	0.17	0.60	0.50	0.33	0.00	0.25	0.67	0.00
3	NA	0.44	0.86	0.50	NA	0.17	0.71	1.0
4	1.0	0.80	0.60	1.0	1.0	0.60	0.88	1.0
5	0.92	1.0	1.0	1.0	0.80	1.0	1.0	NA
6	0.33	0.11	0.50	NA	0.00	0.30	1.0	NA
7	0.25	0.40	0.40	NA	0.13	0.43	0.75	NA
8	0.00	0.00	0.25	0.33	0.00	0.08	0.67	NA
9	0.17	0.56	1.0	0.00	0.10	0.00	1.0	1.0
10	0.33	0.00	0.67	1.0	0.29	0.20	0.67	NA
Ave.	0.40	0.43	0.63	0.65	0.26	0.37	0.78	0.80

(※ 1) 語句自体の認知度：質問「この問題の答えである言葉自体はどの程度知っていますか？」に対する回答「1：まったく知らない，2：あまり知らない，3：まあ知っている，4：よく知っている」
(※ 2) グラフ上での認知度：質問「グラフを見て答えにその言葉が入ることが分かりましたか？」に対する回答「1：まったく分からなかった，2：あまり分からなかった，3：まあまあ分かった，4：よく分かった」

生成した 2 つの Distractors が両者とも不正解だと分かりやすいものであったために，特に該当の 2 問の正答率も高い結果となったとみられる．

6.5 提案システム全体の評価項目と評価結果

ここでは，提案の問題自動生成システムの性能評価として実行時間の測定や，その他被験者実験において得たアンケートによる出題内容全体の主観評価結果をまとめる．

6.5.1 問題生成の実行時間

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

表 11 問題生成の実行時間
Table 11 The time to generate proposal questions.

出題形式	1 問あたりの生成時間 [sec]	
	Question Graph	Distractors
単数回答	2.7	1.2
虫食い単数回答	4.6	7.3
複数回答	3.7	0.8
虫食い複数回答	5.3	2.2
組合せ回答	17.3	7.2
虫食い組合せ回答	19.2	1.8

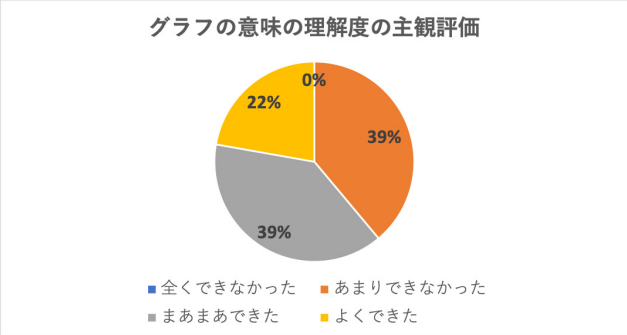

図 6 グラフの意味理解度の主観評価（※）
Fig. 6 The subjective evaluation of the level of understanding Question Graph.
（※）質問「提案の「Question グラフ」の規模感はどうでしたか」に対する 4 段階評価

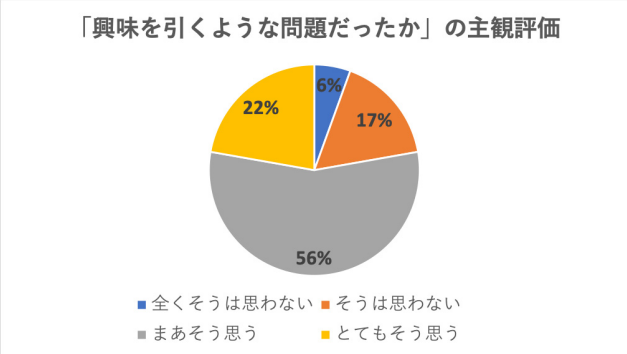

図 7 「興味をひくような問題だったか」の主観評価（※）
Fig. 7 The subjective evaluation of the degree shown by interest of proposal questions.
（※）質問「回答者の興味を引くことができるような問題でしたか」に対する 4 段階評価

ブラリ SPARQLWrapper[*9]により SPARQL を実行した．また，問題生成にかかる実行時間を計測し，Answer に 15 個の語句を設定したときの 1 問あたりの平均値を算出した．なお，プログラム実行時に 1 度だけ word2vec 事前学習済みモデル読み込みに約 18.00 sec 要し，その後の問題生成時間は以下の**表 11** のようになった．

6.5.2 被験者アンケートによる出題全体の主観評価

被験者実験において出題内容全体に対するアンケートの

[*9] https://rdflib.github.io/sparqlwrapper/

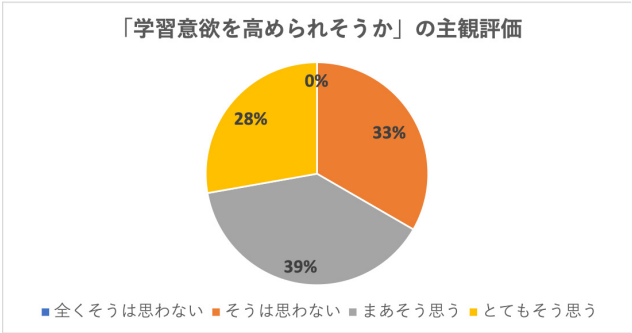

図 8 「学習意欲を高められそうか」の主観評価（※）
Fig. 8 The subjective evaluation about possibility of increasing an incentive to learning through proposal questions
（※）質問「学習者は問題を解くことで学習意欲を高められそうですか」に対する 4 段階評価

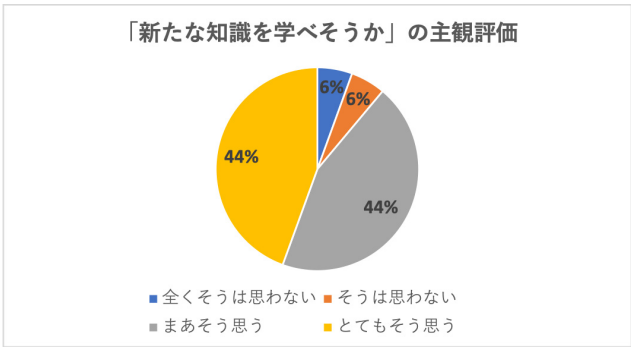

図 9 「新たな知識を学べそうか」の主観評価（※）
Fig. 9 The subjective evaluation about possibility of gaining some new knowledge through proposal questions.
（※）質問「学習者は問題を解くことで新たな知識を学べそうですか」に対する 4 段階評価

主観評価結果を，**図 6**，**図 7**，**図 8**，**図 9** にまとめた．特に「新たな知識を学べそうか」については，肯定する回答が 9 割弱であった．また，「グラフの意味の理解度」に対しては 4 割弱の被験者が “あまりできなかった” という回答である一方，そのうち 7 割超の人が「興味をひくような問題か」と「新たな知識を学べそうか」に対して “まあそう思う” あるいは “とてもそう思う” との回答であったことから，一定数の被験者は理解が多少難しい問題だと感じる一方で，新たな知識を学べそうであり興味をひくと問題でもあると感じていたと分かった．

7. 考察

まず，Question Graph の俯瞰度については，本実験における主観評価と科目およびジャンル横断度の指標による客観評価で比較を上回る結果となったため，提案手法である語句間の *dist* 関数を利用したリンク構造抽出アプローチが俯瞰的な問題生成に貢献したと考えられる．また，出題形式については “単数回答”，“複数回答”，“組合せ回答” に加えて “虫食い形式” により，グラフの生成方法において，

“俯瞰的” を叶えるためにノードの語句間の距離のみを考慮したが，そのリンクラベルの意味も考慮したアプローチを適用はしていない．出題として意味のある問題を生成するために，ノードだけでなくリンクの種類についても考慮することは有効であると考えられる．実際に提案アルゴリズム上で適用するためのハードルとして，利用した知識ベースから提案手法により抽出されたリンク構造には，具体的なリンクラベル情報を持つものが少ないことがあげられる．これは利用する知識ベースの制約によるものが大きいが，一方で Answer に設定する語句によって具体的なラベル情報を持つリンク構造の数に差がある．このことから，知識ベース上で Answer の設定語句について具体的なリンクラベルを持つリンク構造の数と，ユーザが求める俯瞰度の程度に合わせて，部分グラフの抽出過程における距離の条件を調整することで，よりグラフの意味に配慮した俯瞰度の高い問題生成が可能になると考えられる．また，知識ベース上のリンクノードの語句やラベル情報の語句には表記ゆれが存在しており，それは知識ベースの部分グラフとして抽出された Question Graph においてもいくつか見受けられた．なお，今回はノード語句に対しては DBpedia 上の Redirect 情報を利用して，知識ベースと語彙コーパス，word2vec 学習済みモデルの表記ゆれに対応した実装を施しており，リンクラベルの語句に対してもこれと同様の方法で対応することも一案として考えられる．しかしその場合は，Question Graph 上のリンクラベルが知識ベース上の部分グラフのリンクラベルと厳密には一致せず，4 章で設定した要件 (1) を満たさないケースが発生する可能性がある．また，今回は “俯瞰度” の評価項目として，被験者実験では主観評価，機械的実験では 3 つの評価指標（知識ベース上のクラス横断度，科目の横断度，ジャンルの横断度）をあげ，それに基づいて評価した．しかし，本来の利用シーンでは入力として任意のカリキュラムと単元情報があり，そのカリキュラム内の学習項目の横断度の測定を俯瞰度評価に盛り込むことが理想的であると考えられる．今回は，いずれの実験においても具体的に設定した学習カリキュラムはないが，設定することでより実際の利用シーンに即した評価ができる可能性がある．そのため，今後，実際に任意のカリキュラムと特定の単元情報を設定し，カリキュラム内の学習項目の横断度を俯瞰度の指標として追加して，評価を行う予定である．

Distractors の生成においては，“一見して明らかに不正解ではない” ような不正解の選択肢となるように，Question Graph のリンク構造との類似度を考慮した生成方法を適用した．結果として，客観評価である構文パターンについては，提案手法が比較手法を上回る結果となった．一方，人を介した評価である正答率については，結果にバラつきがあったことから，問題によっては一見して不正解だと分かる選択肢が生成されてしまったと考えられる．生成過程では，Answer 語句の属するクラスが DBpedia 上のすべての語句が属する最上位のクラス “Thing” であるケースがあり，固有のクラスが取得できないものも見られた．その場合，Question Graph 上の Answer ノードのリンク構造の類似度計算のみを利用した Distractors 生成となるが，固有のクラスから生成する場合と比較すると，結果として Answer 語句とは明らかに異なるジャンルの候補が生成されてしまう可能性があると考えられる．そこで，そのようなケースに対応するために，DBpedia 上では多くのカテゴリ階層が定義されていることから，クラス分類に加えてカテゴリ分類を利用して生成することは，追加のアプローチとして意義があると考える．

8. まとめ

本論文では，俯瞰的な多肢選択式問題の自動生成手法を提案した．本提案手法では，Linked Data 技術により生成されたグラフデータを知識ベースとして利用し，多肢選択式問題の「問題」として知識ベースの部分グラフである “Question Graph”,「不正解の選択肢」として “Distractors” を生成した．問題の Answer の関連情報として，知識ベース上の膨大なデータを利用してグラフの可視性を保った規模感の問題を生成する過程で，Answer 語句に関連する広範囲の分野からの情報を横断的に含むような問題を生成するためにデータ抽出方法を考えた．特に，グラフ構造である知識ベースにおいて，Answer 語句とその他ノード語句がつねに遠くなるようなリンク構造抽出アプローチを考案して実装した．出題形式として，“単数回答形式”，“複数回答形式”，“組合せ回答形式” を扱い，さらに問題内の一部のノードに該当する語句を隠した状態で出題する “虫食い回答形式” の生成方法も考案して，本論文でそれぞれの生成アルゴリズムを示した．事前に設定した生成する問題の要件に対する評価実験として，被験者の主観実験と機械的実験を実施したところ，特に俯瞰度の評価では，約半数の被験者が比較の既存問題より提案の生成問題の方が俯瞰度が高いと評価した．さらに，俯瞰度の指標として設定した科目横断度とジャンル横断度について，提案の手法である「*dist* 関数を用いたノード抽出手法」と比較の手法である「*dist* 関数を用いないランダムなノード抽出手法」とで比較したところ，全体として提案の手法による生成問題が比較を上回る結果となった．今後の展望として，Question Graph はリンクノードだけでなくリンクラベル情報の意味を考慮した生成手法を，Distractors は知識ベース上のクラスに加えてカテゴリ分類に従った候補取得を採用することで生成結果を改善し，さらに評価実験では，実際にカリキュラムを設定したときのカリキュラムの内容に即した俯瞰度の測定があげられる．また，今回は評価において教職免許状に対する科目や学校区分は関係なく，「教職」という範囲での評価と見なして実施したが，免許状所持者をユー

ザとして想定したときのユースケースに即する評価となりうるため，免許状の種類や区分を考慮した実験を行うことも今後の展望としてあげる．

以上の展望と本実験における評価結果をふまえて，より俯瞰度が大きく汎用性の高い Question Graph と Distractors 生成アプローチの工夫を追加し，評価方法を見直していくことで，さらに有効な問題生成手法の提案を目指す．

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いただき，ご指導いただ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ご意見をいただ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），入手先 ⟨http://www.mext.go.jp/b_menu/shingi/chukyo/chukyo3/siryo/attach/1364319.htm⟩ (2015).

[2] 文部科学省：高等学校学習指導要領，入手先 ⟨http://www.mext.go.jp/component/a_menu/education/micro_detail/__icsFiles/afieldfile/2011/03/30/1304427_002.pdf⟩(2009).

[3] 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).

[4] Bühmann, L., Usbeck, R. and Ngomo, A.-C.N.: ASSESS—Automatic Self-Assessment Using Linked Data, *International Semantic Web Conference*, pp.76–89, Springer (2015).

[5] Pho, V.-M., Ligozat, A.-L. and Grau, B.: Distractor quality evaluation in multiple choice questions, *International Conference on Artificial Intelligence in Education*, pp.377–386, Springer (2015).

[6] 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).

[7] Okuhara, F., Sei, Y., Tahara, Y. and Ohsuga, A.: Integration Between Agents and Remote Ontologies for the Use of Content on the Semantic Web, *Proc. 11th International Conference on Agents and Artificial Intelligence* (*ICAART 2019*), INSTICC, SciTePress (2019).

[8] Iijima, T., Kawamura, T., Sei, Y., Tahara, Y. and Ohsuga, A.: Sake Selection Support Application for Countryside Tourism, *Trans. Large-Scale Data- and Knowledge-Centered Systems XXVII*, pp.19–30, Springer (2016).

[9] Fionda, V. and Pirrò, G.: Meta Structures in Knowledge Graphs, *International Semantic Web Conference*, pp.296–312, Springer (2017).

[10] Maillot, P., Raimbault, T., Genest, D. and Loiseau, S.: Targeted Linked-Data Extractor, *Proc. 6th International Conference on Agents and Artificial Intelligence—Volume 1: ICAART*, INSTICC, pp.336–341, SciTePress (online), DOI: 10.5220/0004758503360341 (2014).

[11] Demarchi, F., Santos, E.R. and Silveira, R.A.: Integration Between Agents and Remote Ontologies for the Use of Content on the Semantic Web, *Proc. 10th International Conference on Agents and Artificial Intelligence—Volume 1: ICAART*, INSTICC, pp.125–132, SciTePress (online), DOI: 10.5220/0006718701250132 (2018).

[12] Papasalouros, A., Kanaris, K. and Kotis, K.: Automatic Generation Of Multiple Choice Questions From Domain Ontologies., *e-Learning*, Citeseer, pp.427–434 (2008).

[13] Zoumpatianos, K., Papasalouros, A. and Kotis, K.: Automated Transformation of SWRL Rules into Multiple-Choice Questions., *FLAIRS Conference*, Vol.11, pp.570–575 (2011).

[14] 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*, pp.380–385, Springer (2018).

[15] 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).

[16] 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).

[17] 池田信一，高木輝彦，高木正則，勅使河原可海：多肢選択式項目の出題パターンと選択肢の類似性に着目した難易度推定方法の提案と評価，情報処理学会論文誌，Vol.54, No.1, pp.33–44 (2013).

[18] 文部科学省中央教育審議会高大接続特別部会：高大接続特別部会（第 16 回）配付資料 1-2「合教科・科目型」や「総合型」を導入する必要性，入手先 ⟨http://www.mext.go.jp/b_menu/shingi/chukyo/chukyo12/shiryo/1349406.htm⟩ (2014).

[19] 文部科学省中央教育審議会：2040 年に向けた高等教育のグランドデザイン（答申）（中教審第 211 号）【概要】，入手先 ⟨http://www.mext.go.jp/component/b_menu/shingi/toushin/__icsFiles/afieldfile/2018/12/17/1411360_9_1_1.pdf⟩ (2018).

[20] 奥原史佳，清 雄一，田原康之，大須賀昭彦ほか：Linked Data を用いたカリキュラムベースの多肢選択式問題自動生成手法の提案，電子情報通信学会技術研究報告，人工知能と知識処理，Vol.2018, No.7, pp.35–40 (2018).

[21] 奥原史佳，清 雄一，田原康之，大須賀昭彦ほか：Linked Data を用いたカリキュラムベースの多肢選択式問題自動生成手法の提案，合同エージェントワークショップ＆シンポジウム 2018（JAWS-2018），Vol.2018, No.3B-1 (2018).

[22] 奥原史佳，清 雄一，田原康之，大須賀昭彦ほか：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，研究報告コンピュータと教育（CE），Vol.2018, No.6, pp.1–10 (2018).

[23] 奥原史佳，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).

[24] Mitkov, R., Ha, L.A., Varga, A. and Rello, L.: Semantic similarity of distractors in multiple-choice tests: extrinsic evaluation, *Proc. Workshop on Geometrical Models of Natural Language Semantics*, Association for Computational Linguistics, pp.49–56 (2009).

付　　録

A.1 Question Graph 生成例

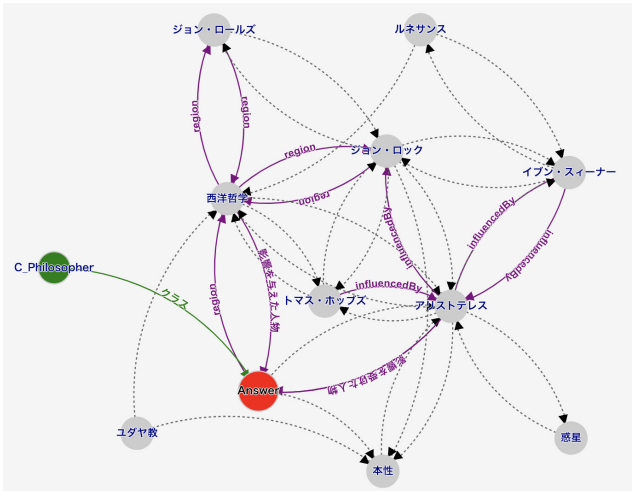

図 A·1 Answer=“ソクラテス” に設定したときの Question Graph（単数回答形式）

Fig. A·1 An example of Question Graph that has as the single answer “Socrates”.

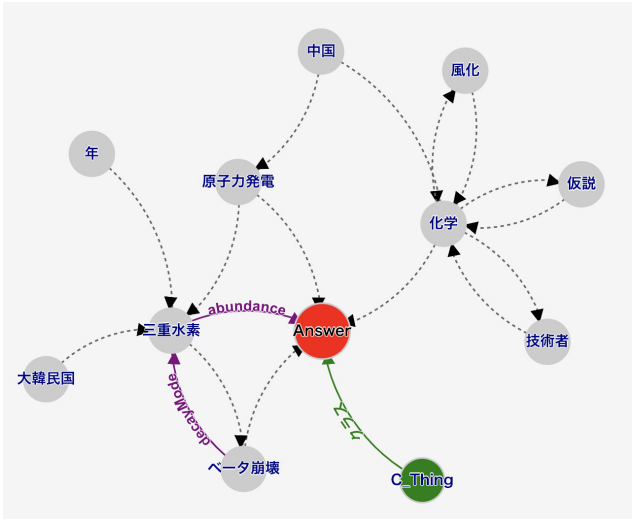

図 A·2 Answer=“放射性同位体”, “南北首脳会談” に設定したときの Question Graph（複数回答形式）

Fig. A·2 An example of Question Graph that has as the multiple answers “radioactive isotope (RI)” and “South-North prime ministers meeting”.

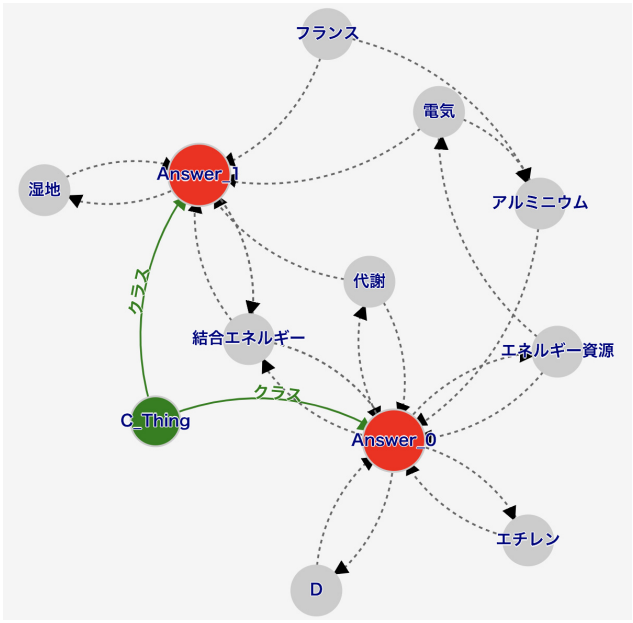

図 A·3 Answer_0=“フッ素”, Answer_1=“鉄” に設定したときの Question Graph（組合せ回答形式）

Fig. A·3 An example of Question Graph that has as the pair of answers “fluorine” and “iron”.

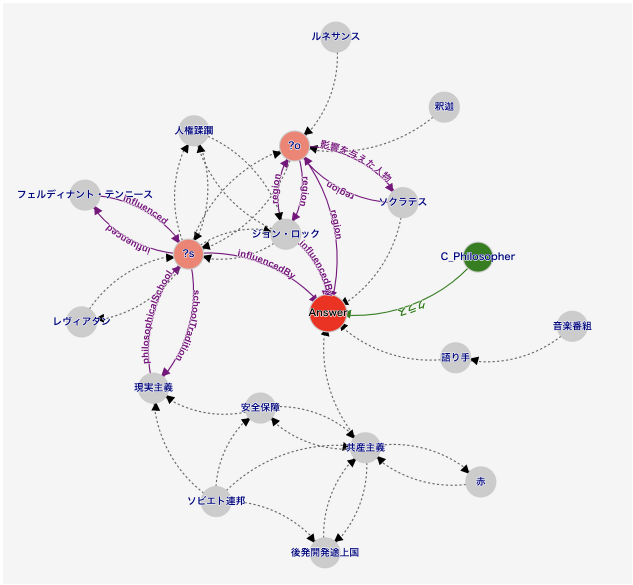

図 A·4 Answer=“プラトン” に設定したときの Question Graph（虫食い単数回答形式）

Fig. A·4 An example of Question Graph that has as the single answer “Plato” (fill-in-the-blank question).

奥原 史佳

1994 年生．2019 年電気通信大学大学院情報理工学研究科情報学専攻博士前期課程修了．同年（株）NTT ドコモ入社．主としてセマンティック Web，自然言語処理を用いた教育分野における学習問題生成に興味を持つ．

清 雄一 （正会員）

1981 年生．2009 年東京大学大学院情報理工学系研究科博士後期課程修了．同年（株）三菱総合研究所入社．2013 年より電気通信大学．現在，同大学大学院情報理工学研究科准教授．博士（情報理工学）．エージェント，プライバシ保護技術等の研究に従事．2016 年度土木学会水工学論文賞，情報処理学会論文賞受賞．電子情報通信学会，日本ソフトウェア科学会，IEEE Computer Society 各会員．

田原 康之 （正会員）

1966 年生．1991 年東京大学大学院理学系研究科数学専攻修士課程修了．同年（株）東芝入社．1993～1996 年情報処理振興事業協会に出向．1996～1997 年英国 City 大学客員研究員．1997～1998 年英国 Imperial College 客員研究員．2003 年国立情報学研究所着任．2008 年より電気通信大学准教授．博士（情報科学）（早稲田大学）．エージェント技術，およびソフトウェア工学等の研究に従事．日本ソフトウェア科学会会員．

大須賀 昭彦 （正会員）

1958 年生．1981 年上智大学理工学部数学科卒業．同年（株）東芝入社．同社研究開発センター，ソフトウェア技術センター等に所属．1985～1989 年（財）新世代コンピュータ技術開発機構（ICOT）出向．2007 年より電気通信大学．現在，同大学大学院情報理工学研究科教授．2017 年より同大学院情報システム学研究科研究科長併任．2012 年より国立情報学研究所客員教授兼任．工学博士（早稲田大学）．ソフトウェア工学，エージェント，人工知能の研究に従事．1986 年度および 2016 年度情報処理学会論文賞，2013 年度人工知能学会研究会優秀賞，2014 年度同学会功労賞受賞．IEEE Computer SocietyJapan Chapter Chair，人工知能学会理事，日本ソフトウェア科学会理事，同学会監事等を歴任．電子情報通信学会，人工知能学会，日本ソフトウェア科学会，電気学会，IEEE Computer Society 各会員．本会フェロー．