自然言語の SPARQL クエリ変換に基づく大規模知識へのアクセス手法の開発

村山友理　(指導教員：小林一郎)

1　はじめに

現在，Linked Open Data (LOD) として様々な分野の膨大な知識がデータとして Web 上に公開されている．これらの大量の知識を自然言語処理システムの中に組み込むことができれば，自然言語理解における背景知識として利用可能になるなど，従来のシステムを超えた優れたシステムが実現されると考えられ，それらのデータを自然言語と結びつけるための様々な取り組みが進められている [1, 2, 3]．基本的な問題として，自然言語の表現から LOD のように形式的に記述されている知識にアクセスするために，自然言語文からデータレポジトリにアクセスするクエリ言語に変換できるようにすることを対象に研究が行われている．

本研究では，先行研究のアプローチを踏まえて，自然言語文から抽出した情報に知識のリレーションを紐づけることに重点を置いた手法の提案を行う．また，提案手法を評価する具体的な実験として，ロボットが喫茶店でお客さんから注文をとる場面を想定して，そこで発生する質問に回答するための知識を取得するためにユーザの質問文から SPARQL クエリを自動生成することを考える．

2　提案手法

本研究で提案する手法の概要を図 1 に示す．本研究の提案手法における，自然言語質問文から SPARQL クエリを自動生成する手続きについて以下に示す．

1. 自然言語質問文の構文解析に基づく RDF トリプル候補抽出
   京都大学黒橋河原研究室で開発された日本語構文・格・照応解析システム KNP[1] を用いて，自然言語質問を解析し，語彙の依存関係および述語項構造を抽出する．述語項構造の解析結果の内，述語に対して項を 2 つ持つものから候補の RDF トリプル（以下，トリプルと呼ぶ）として（述語，項 1，項 2）を抽出する．トリプルがとる各単語について WordNet を用いてその類義語を調べる．これは，Wang ら [1] の手法で用いているように，自然言語質問文中の表層表現や類義語がデータレポジトリ中の RDF トリプルを記述されている語彙になっている可能性があり，そのような知識に対して柔軟なアクセスができるように配慮するためである．
2. 述語に基づく質問タイプ判別
   候補となる RDF トリプルの述語に基づき，質問タイプを判別する．本研究で設定した述語の種類ごとの質問タイプを表 1 に示す（表 1 に示す質問タイプについては 3 章に詳述する）．この段階では，項 1，項 2 はまだ未確定なのでそれぞれ`?x0`,`?x1` とする．例えば，述語が “含む” であれば基本型と判定されることから，テンプレートとして {`?x0 ???? ?x1.`} が与えられる．
3. 項の属性判定
   候補の RDF トリプルの項に注目して，その属性がクラス，インスタンス，リテラルのいずれであるかを判定する．クラスとインスタンスの判定は，事前に作成した辞書との文字列一致で行い，辞書に存在しない場合はリテラルと判断する．クラスと判定された場合はトリプルパターン {`?x rdf:type :`クラス名 } を追加する．インスタンスだった場合は ’`?x` ’を ’`:`インスタンス名 ’に変換する．リテラルの場合は変数のままにしておく．
4. 語彙表現に基づく回答対象候補の特定
   質問文中の語彙表現によって質問の答えになる回答対象候補を特定する．候補を特定する語彙表現として，“どんな”，“はありますか”，“は何ですか” を採用する．質問文中に “どんな” が現れた場合はその直後に来る単語を，“はありますか” と “は何ですか” の場合は直前にくる単語を回答対象候補とし，それに対応する変数を `SELECT DISTINCT` 後の変数リストに加える．
5. 知識への問合せによるリレーション候補の決定
   上記の処理によって作成した SPARQL クエリの`????` の部分を`?rel` としてリレーションの候補を知識に問い合わせることにより決定する．
6. SPARQL クエリの生成
   リレーションの候補の検索結果の中で，WordNet を使って調べておいたトリプルの各単語の類義語すべてを対象に Jaccard 係数を計算し，最も類似度の高い候補を最適なリレーションとして決定する．最も類似度の高い候補が複数存在する場合は，ジップの法則に従ってより希少な方を最適なリレーションとして選択する．以上の手続きにより入力された自然言語質問文の内容を反映した SPARQL クエリを完成させる．

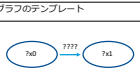
[1] http://nlp.ist.i.kyoto-u.ac.jp/index.php?KNP

3　質問タイプ

本研究で対象とする質問タイプの分類を表 1 に示す．

表 1: 質問タイプの分類

	述語	例文と候補のトリプル	SPARQLクエリのテンプレート	RDFグラフのテンプレート

基本型	含む
	使う
		カルボナーラはどんなアレルギー食材を含みますか？
		（含む, カルボナーラ, アレルギー食材）
		
		トマトを使ったパスタはありますか？
		（使う, パスタ, トマト）
			SELECT DISTINCT _ WHERE {
			?x0 ???? ?x1.
			}
				

Degree型	低い 高い	塩分の低い/高いパスタは何ですか？ （低い/高い, パスタ, 塩分）	SELECT DISTINCT _ ?amount WHERE { ?x0 ???? ?x1. ?x1 j.1552:amount ?amount. } ORDER BY ASC/DSEC(?amount) LIMIT 3	

List型	----	どんなパスタがありますか？
		--- 	
			SELECT DISTINCT ?x0 WHERE {
			?x0 rdf:type :クラス名.
			}
				

受け身型	----	どんなアレルギー食材がカルボナーラには含まれますか？
		（含む,アレルギー食材,カルボナーラ）
			SELECT DISTINCT _ WHERE {
			?x0 ???? ?x1.
			}
			⇩
			SELECT DISTINCT _ WHERE {
			?x1 ???? ?x0.
			}	

“含む”，“使う” などの述語は基本型としてテンプレートで {`?x0 ???? ?x1.`} を与える．“低い” のような「程度」を問合せる述語は Degree 型とし，テンプレートで {`?x0 ???? ?x1. ?x1 j.1552:amount ?amount.`} を与える．本研究では，知識のドメインを料理のメニューに限定しているため，程度に対する問合せは「量」への問合せであるとする．`ORDER BY ASC(?amount) LIMIT 3` により，量によって昇順に並

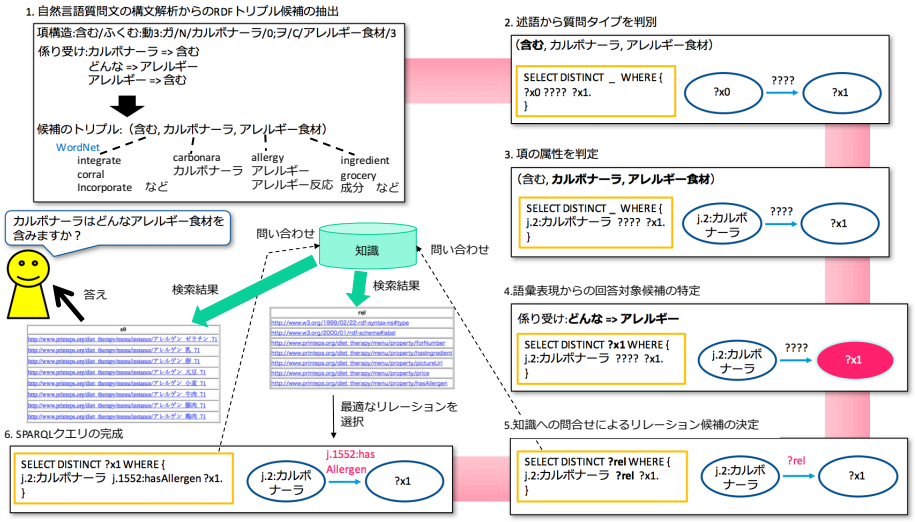


図 1: 提案手法の概要

べ上位 3 つを返すように設定している．“どんなパスタがありますか？“の場合は，KNP を用いてトリプルの情報を得られないため，提案手法のステップ 1，2，5 を除いて SPARQL クエリを生成する．ある単語が “どんな” と “がありますか” の 2 つのキーワードに挟まれた場合，該当する項目をすべて回答する List 型に分類されるとする．“～含まれますか？“のような受け身型の場合, 始めは能動態の述語と同じように項 1, 項 2 をそれぞれ?x0,?x1 として SPARQL クエリに変形してゆき, ステップ 5 でリレーションの候補を知識に問い合わせ答えが返ってこなかったら, リレーションの矢印の向きを反対にする, つまり {?x0 ?rel ?x1.} の部分を {?x1 ?rel ?x0.} に修正して再度リレーションの候補を知識に問い合わせる．

4 実験

喫茶店におけるメニューに関して，本研究で設定した 4 つの質問タイプに基づく自然言語より質問を受けた際の提案手法の有効性を検証する．

4.1 実験設定

図 2 に本研究で利用するメニューに関する知識を示す．

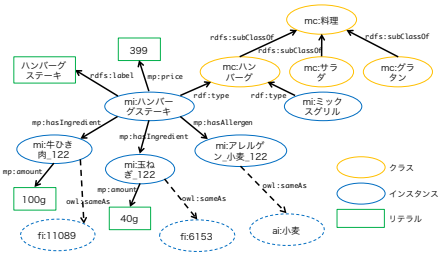


図 2: メニューに関する知識（一部）

4.2 実験結果と考察

提案手法の実装を行い，表 1 の質問タイプの分類に挙げた「カルボナーラはどんなアレルギー食材を含みますか？」「トマトを使ったパスタはありますか？」「塩分の低いパスタは何ですか？」「どんなパスタがありますか？」「どんなアレルギー食材がカルボナーラには含まれますか？」の 5 文に対して実験を行った．表 2 に示すように，それぞれ適切な SPARQL クエリを生成し, 知識から正確な答えを得ることができた．

表 2: 生成された SPARQL クエリ

型	自然言語質問文	生成された SPARQL クエリ
基本型	カルボナーラはどんなアレルギー食材を含みますか？	SELECT DISTINCT ?x1 WHERE { j.2:カルボナーラ j.1552:hasAllergen ?x1. }
	トマトを使ったパスタはありますか？	SELECT DISTINCT ?x0 WHERE { ?x0 j.1552:hasIngredient ?x1. ?x0 rdf:type j.1:パスタ. ?x1 rdf:type j.1:トマト. }
Degree 型	塩分の低い/高いパスタは何ですか？	SELECT DISTINCT ?x0 ?amount WHERE { ?x0 j.1552:hasIngredient ?x1. ?x0 rdf:type j.1:パスタ. ?x1 rdf:type j.1:塩. ?x1 j.1552:amount ?amount. } ORDER BY ASC/DSEC(?amount) LIMIT 3
List 型	どんなパスタがありますか？	SELECT DISTINCT ?x0 WHERE { ?x0 rdf:type j.1:パスタ. }
受け身型	どんなアレルギー食材がカルボナーラには含まれますか？	SELECT DISTINCT ?x0 WHERE { j.2:カルボナーラ j.1552:hasAllergen ?x0. }

5 おわりに

自然言語文から抽出した情報に知識のリレーションを紐づけることに重点を置いた，SPARQL クエリの自動生成のための手法を提案した．今後の課題として，質問タイプとして考えた 4 タイプ以外に Count 関数や Sum 関数を使ったもの，否定表現，接続詞が入った表現, サブクエリなども扱えるように拡張したい．質問タイプの例文以外の自然言語質問文を入力とした実験も行いたい．対話の中の文脈を考慮した SPARQL クエリの生成にも取り組みたい．

参考文献

[1] Chong WangMiao XiongQi ZhouYong Yu, PANTO: A Portable Natural Language Interface to Ontologies, European Semantic Web Conference ESWC 2007: The Semantic Web: Research and Applications pp.473-487, 2007.

[2] 鈴木遼司, 三輪誠, 佐々木裕： 交通オントロジーを対象とした質問文の SPARQL クエリ変換, 第 21 回言語処理学会年次大会, P2-13, pp.171-174, 京都, 2015.

[3] Saeedeh Shekarpour, Edgard Marx, Soren Auer, Amit Sheth, RQUERY: Rewriting Natural Language Queries on Knowledge Graphs to Alleviate the Vocabulary Mismatch Problem, the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17), San Francisco, CA, February 4-9, 2017.