日本語テキストからのオントロジー構築

兼岩研究室　　1731102 田邉憲太朗

背景・目的

人工知能開発の発展とともにその知識をどう表現するか，どう用意するかが重要になってきた

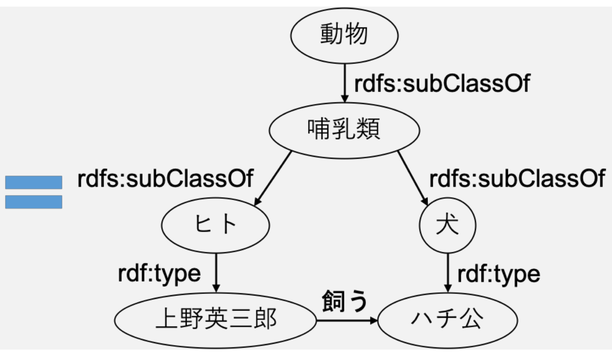
RDF

- 知識表現 (データの表し方) の方法の1つ
- (S: subject, P: predicate, O: object) の3つ組で構成

オントロジー

- 情報 (リソース) を構造的にまとめたデータの集合
  - コンピュータや人工知能のための辞書のようなもの
- RDF等を使って書かれる

主語 (S)	述語 (P)	目的語 (O)
犬	rdfs:subClassOf	哺乳類
ヒト	rdfs:subClassOf	哺乳類
哺乳類	rdfs:subClassOf	動物
ハチ公	rdf:type	犬
上野英三郎	rdf:type	ヒト
上野英三郎	飼う	ハチ公

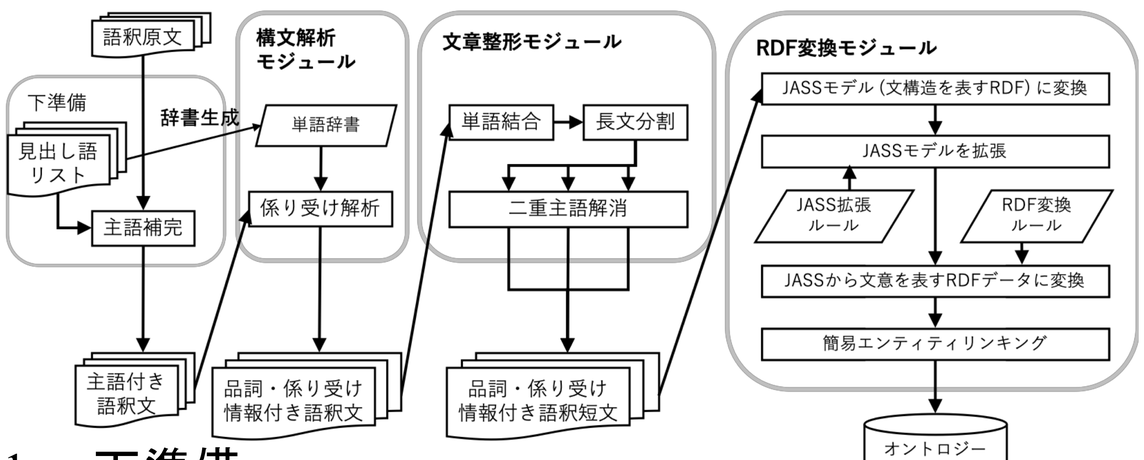


文章をRDFに変換できれば、コンピュータに物事を教えることができる

手法

国語辞書の文章 (語釈) を入力対象に

- 正しい日本語
- 客観的事実が書かれている

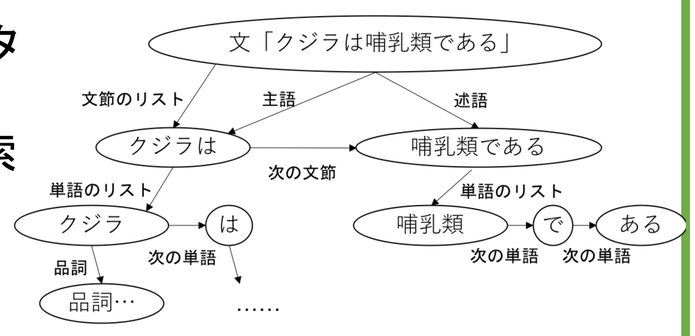


1. 下準備
   - 語釈は主語がないことが多いので補完
   - 単語辞書を用意し構文解析の精度向上
2. 構文解析モジュール
   - 係り受け解析器CaboChaを使用
   - 文の文節の区切り・品詞等を解析
3. 文章整形モジュール
   - 文章を簡潔にする
     - 複数の文節からなる名詞句をまとめる
     - 複数の述語が含まれる文を分割
4. RDF変換モジュール
   - まず文構造をRDFデータで表す (=JASSモデル)
   - 各JASSモデルがRDFルールにマッチした場合，文章の意味を表すRDFデータに変換

JASSモデルとRDFルール

JASS (JApanese Sentence Structure) モデル

- 文章の構造をRDFデータにしたもの
- 既存のRDFデータを検索する技術が応用できる



RDFルール

- IF部
  - JASSモデルとマッチさせる部分
  - 複数のRDFトリプルで構成．変数を使える
  - 「述語は名詞か」など文章に当てはまる条件を書く
- THEN部
  - IF部がマッチした場合に出力するRDFデータ
  - 同じく複数のRDFトリプルで書かれる
  - 変数はIF部と共有されマッチ時のリソース・値が代入される

右の例では、IF部に「文の述語が形容詞なら、その主語のリソースは?rsrcS，形容詞を表すリソースは?rsrcP」と書かれている．
その2つの変数に入っている値を元にTHEN部が出力される．

```
IF (subClassOf) {
 ?stc a jass:Sentence ;
   jass:subject/jass:categorem/
     jass:means ?rsrcS;
   jass:predicate ?clsP .
 ?clsP jass:consistsOfCategorem ?wrdP .
 ?wrdP jass:mainPoS "名詞" ;
   jass:means ?rsrcP .
} THEN {
 ?rsrcS rdfs:subClassOf ?rsrcP .
};
```

実験・結果

実行環境

- OS: macOS 10.14.1
- CPU: Intel Core i7 2.3GHz
- メモリ: 16GB

入力: goo国語辞典動物カテゴリの語釈 15,956文

出力: RDFトリプル 34,828組

出力されたRDFデータの一部をルールごとに正しいか (元の文意に沿うか) を判定した．

グループ	ルール	生成トリプル数	正解率 (%)
名詞グループ	数値と単位	4473	94.12
	同値関係	1211	96.39
	形容的名詞	1757	81.63
	概念階層	5629	59.32
動詞グループ	部分含有	1314	71.43
	動詞	14436	52.17
	サ変動詞	2172	69.05
形容詞グループ	形容詞	3836	75.45
	合計	34828	73.56

考察

名詞の概念階層ルールと動詞ルールがやや精度が悪い．なぜなら名詞の場合，数値・同値・形容はIF部の条件が厳しい一方，概念階層ルールは「述語が名詞」という最低限の条件しかないため．それはある文がどのルールにもマッチせず出力が無いことを避けるように設計した結果なので問題はない．

オントロジーとしての有用性を上げるために出力したリソースを既存のデータセットとリンクさせることも視野に入れる．