2H2-3

多様なデータソースを活用するディベート型人工知能のための自然言語を核とするデータ表現

Representation based on Natural Language for Debating AI with Multiple Data Sources

柳井 孝介 *1　佐藤 美沙 *1　柳瀬 利彦 *1
Kohsuke Yanai　Misa Sato　Toshihiko Yanase

*1日立製作所基礎研究センタ
Center of Exploratory Research, Hitachi Ltd.

This paper proposes an system model for a intelligent software agent which uses various data sources and provides various functions. In the proposed model, instead of standardized schema and taxonomy, natural language plays an important role to combine mutilple data soruces and build up complex system with various modules. We implement a proto type system using the proposed model and investigate if the model works well.

1. はじめに

著者らの研究グループでは，人と議論ができるディベート型人工知能の研究を進めている [Sato 15]．ディベート型人工知能は，多様なテキスト情報をリサーチし，人の価値観に紐づけて，賛否両面から根拠を提示することで，政策決定者や経営者の判断を支援する。たとえば,「カジノを禁止すべきか」という議題が与えられると，賛成側の場合は「ギャンブル依存症の問題を引き起こす」，反対側の場合は「地域経済の活性化につながる」，などを理由として挙げ，続いてニュース記事や白書の中からその根拠となる事例を抽出して提示する．

これまでは，テキストから根拠を示す文を抽出するための根拠性認識に注力してきたが，本稿では別のアプローチとして，多様な分析機能を持つようにすることを検討する。具体的な例として,「X 社に投資すべきか」という論題を考える。この場合，これまでと同様に，過去のニュース記事や白書を解析して賛否の根拠を出すのに加えて，X 社の売上高のデータを過去 10 年分集めて，上昇傾向にあるか下降傾向にあるかを提示することで，より効果的な経営判断の材料になると考えられる。他にも以下のような分析が考えられる。

- X 社の業界ごとの投資額を調べ，有望な業界に投資ができているかを調べる。
- X 社が注力している地域の人口が増加傾向にあるか調べる。

このような分析のロジックは多数考えられる。このようなロジックをソフトウェアがデータや文書から自動的に習得するのは困難である。そこで本稿では，ロジックを自動的に習得する方法を考える替わりに，最初のステップとして，ロジックは人が考案して実装するものと仮定して，多数の分析ロジックを実装できるようなシステムの設計方針を考察する。10 程度のロジックであれば，通常のソフトウェアの設計で問題ないと思われる。しかしロジックの数が増えてくると，データやロジック間の関係が複雑になり，システムとして成立させるのが難しくなると思われる。本稿では，1 つのシステムに，かなりの種類のデータとロジックを実装することを,「機能をスケールさせる」と表現する。本稿では，機能をスケールさせられるシステムとはどのようなものかを議論する。

連絡先: 柳井 孝介，日立製作所基礎研究センタ，
kohsuke.yanai.cs@hitachi.com

本研究の背景としては，対話技術の応用が，従来の雑談や QA から，様々なシステムへのインターフェースとしての側面が強くなってきたことが挙げられる。著者らは対話システムの本質の 1 つは，対話システムで提供できる機能の数をスケールさせることであると捉えている。本稿では，機能をスケールさせるためのシステムのモデルを提案し，プロトシステムの実装を報告する。

2. 従来の設計とその問題点

ある企業に関して，過去 10 年分の売上高のデータを集めて傾向を分析するロジックを実装することを想定する。データソースとして，XBRL 形式で記述された有価証券報告書を使うことができる。XBRL とは財務情報向けの標準化された XML ベースの言語である。XBRL 形式の有価証券報告書からは 5 年分の売上高しか得られなかった場合，別のデータソースから売上高のデータを補完することが考えられる。商用で販売されているデータベースや，ニュース記事から抽出した数値データなどを使うことができる。またロジックによっては DBpedia などの知識ベースで整備されている数値データ/テキストデータを使うこともできる。ここで複数のデータソースにまたがって 1 つのロジックを実装するケースがありうることが分かる。START などの QA システムでも複数のデータソースを使っている [Katz 02]。複数のデータソースを組み合わせる場合，統合（SQL での Union，以下ユニオン）と結合（SQL での Join，以下ジョイン）がある。複数のデータソースにまたがって 1 つのロジックを実装するケース，逆に複数のロジックにまたがって，同一のデータソースが使われるケースで，以下の疑問が生じる。

疑問：「売上」の同一性 XBRL 形式の 5 年分の売上高と，商用データベースの別の数年分の売上高をユニオンすることを考える。この場合，売上と書かれていれば，本当に同じ意味を表していると考えてユニオンしてよいのか。そのユニオンはロジック固有のものなのか，それとも普遍的にユニオンしてよいのか。同一のデータソースであっても，XBRL においては売上高と解釈できるタグは 7 種類以上あり，それらは厳密には違う意味を持つが，いつ同じ意味を持つとして扱ってよいのか。たとえば，日本会計基準の売上高と国際会計基準の売上高では，業界によっては数倍異なることもあるが，10 年のうちに会計基

準が変わったら比較してよいのか。またベンチマークしたい 2 社で会計基準が異なる場合，比較してよいのか。ロジックによって，会計基準を無視してもよいものと，会計基準を区別すべきものがあるのではないか。

疑問：不完全なデータの扱い データソースとして用いる新聞や商用のデータベースに会計基準が記載されていない場合，別のデータソースで調べた会計基準をジョインし，会計基準付きの売上高データにして，安全にユニオンする方法が考えられる。この場合，利用可能なデータの範囲で，どの会計基準かわからなければ，そのデータは使わないようにすべきなのか。つきつめて考えると，すべてが揃っている完備なデータだけを使うべきなのか。それとも，不完全なデータであっても捨てずに，何とか推測や補完で補って使ったほうが有益なのか。

疑問：スキーマに関わるコスト 複数のデータソースを使う場合，そのすべての XML タグやデータフィールドの意味を理解しておき，かつ，その対応（どのタグとどのフィールドが同じ意味を持つか，あるいは持ちうるか）も理解しておく必要がある。しかし，それぞれのロジックでこれらを完全に管理しておくのは，多大なコストがかかるため，複雑なシステムを維持する場合には破たんするのではないか。一般に，共通のデータを複数のロジックで使う場合，複数のユースケースで使われる汎用的な定義を決めるのは困難が伴う。XBRL に関しては，2000 以上の XML タグが定義されているが，毎年更新されており，ある限定された領域であっても，スキーマの決定・更新には多大なコストがかかる。スキーマを使うシステム側にも，スキーマの意味づけ，対応付けにコストがかかる状況で，破たんなくシステムを維持することはできるのか。汎用的に使えるスキーマやタクソノミ，あるいは汎用の知識表現，それが意味することなどを，定義・更新・意味づけ・対応付けすることは，そもそも不可能なのではないか。

上記では売上高のデータを例としたが，数値かテキストに関わらず，多くのデータに対して本議論はあてはまると考えている。人が意思決定のための情報収集をするときに，厳密さを犠牲にしてうまく対処しているのであれば，人工知能も同様のアプローチで実現できないかという考えに至る。上記の疑問を出発点として考えると，自然言語を使ってデータをつなぐというのが自然な発想になると考えている。画像データなどに関しては，自然言語の注釈により検索できるようにする研究は行われている [Katz 06]。次章では，人が自然言語を使って，上記の問題にうまく対処しており，その唯一の成功事例であることから，自然言語を核としたシステムのモデルを提案する。

3. 提案モデル

図 1 に提案モデルの前提となるアーキテクチャを示す。「心の社会」[Minsky 09] にならって，1 つのロジックを実装するソフトウェアの単位をエージェントと呼ぶ。複数のエージェント全体で 1 つのシステムを構成する。データストアは 1 つでなくてもよいが，議論を簡単にするため，前処理で複数のデータソースを 1 つのデータストアに集約するものとする。

提案モデルの方針は以下の通りである。同一性に関しては，普遍的に同一かどうかを決めるのではなく，各エージェントが独自の類義辞書を持ち，そのタスクが実行されるコンテクストで同一と見なすかどうかを自然言語の表記で判定する。従って意味を事前に一意に決め，シンボル化するのとは別のアプローチとなる。また不完全なデータも，自然言語で理解できる限りで，推測や補完で補って使う。スキーマはできるだけ使わず，各エージェントが自然言語ベースでその持ち得る意味を推測する。具体的には，以下の方式でシステムを実装する。

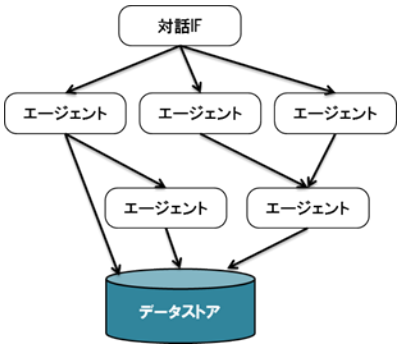

図 1: 提案モデルの前提となるアーキテクチャ

- XML であっても，表であっても，テキストであっても，データは自然言語で注釈をつけてデータストアに格納する。このとき，自然言語の注釈はデータを入れる人が，**自分が理解した範囲かつ他の人が見たときに理解してもらえると思う範囲で，自然に思いついた表現**で注釈を入れる。データの単位もデータを入れる人が自然と思う単位で入れる。注釈もデータの単位も，事前に決められたスキーマ，タクソノミ，その他の規則に縛られない。この自然言語の注釈がデータ検索のインデクスとなる。これは他の人に理解できる範囲で記述していれば，原理的にはデータを見つけてもらえるという発想に基づく。

- システムへのクエリは自然言語で入力する。クエリを普遍的に意味解釈するエージェントはなく，各エージェントが非同期に独立してクエリを解析して，自分がそのクエリを処理すべきかどうかを判断する。あるエージェント集団がすべて処理しなかったときに実行されるエージェントや，他のエージェントの実行を抑制，活性するエージェントがある。

- 各エージェントは，データにつけられた自然言語の注釈をインデクスとして処理に必要なデータを探す。たとえば，日本会計基準の売上高と国際会計基準の売上高を，同一と見なすかどうかは， 各エージェントが責任を持つ。クエリを意味解釈してシンボルに置き換えることなく，可能な限り，テキストのまま扱い，文字列一致による類似性判定をユニフィケーションのベースとする。

- 各エージェントは結果をレポートするための，それほど高度ではない言語合成能力やグラフ・表生成機能を持つ。

提案モデルのイメージがわくように，具体例を説明する。表 1 に，XBML 形式のデータ，表データ，ニュース記事が，それぞれどのようにデータストアに格納されるかを示す。#1 は新聞記事であり，言語解析により売上高の情報が記載されていることを認識し，企業名と会計年を抽出し，インデクスとする。@は「is」のような意味を表す。#2 は XBRL 形式の有価証券報告書から得られた売上高データであり，XBRL の定義書を

表 1: データストア中のデータ表現

#	インデクス	データの内容
1	{ id: 12112, source: "OX 新聞", receptor: ["売上", "日立製作所@企業", "2016@年"]}	日立製作所が昨日発表した２０１６年３月期連結決算は、売上高が 10 兆 343 億円で、１０兆円を超えたのは７年ぶりとなった。
2	{ id: 12113, source: "Edinet", receptor: ["収益", "株式会社日立製作所@企業", "IFRS@会計基準", "2015@年"]}	<jpcrp_cor:RevenueIFRSSummaryOfBusinessResults context Ref="CurrentYearDuration" unitRef="JPY" decimals="-6">9774930000000</jpcrp_cor:RevenueIFRSSummaryOfBusinessResults>
3	{ id: 12114, source: "XO データベース", receptor: ["売上", "株式会社日立製作所@企業", "2006@年"]}	9,464,801 百万円
4	{ id: 12115, source: "dbpedia", receptor: ["概要", "日立製作所@企業", "2016@年"]}	日本最大の総合電機メーカー。大手電機 8 社（日立製作所、パナソニック、東芝、三菱電機、ソニー、シャープ、NEC、富士通）の一角。重電 9 社...

参照して，収益が記載されていることを認識して，インデクスにいれる。#3 は商用データベースから抽出した売上高データ，#4 は DBPedia の企業概要を表す。

「X 社に投資すべきか」と質問が入力されると，まず構文的にパースして，売上高傾向分析エージェントにパースされたクエリがくる。本エージェントは，質問の構文構造「[企業名]に投資すべきか」から自分が活性化すべきと判断する。他にも業界別投資額分析エージェントなども活性化され，独立に処理が進む。同義語・類義語辞書により，「X 社」を，短縮表記や英語表記などに展開する。このときに，「X 社」を本質的にはシンボル化せずに，文字列一致により同義語辞書内から，同義語を探して展開する。よって，「X 社」はエージェント内では ID 表現などの意味表現には変換されない。「X 社」およびその同義語と，「売上 or 収益」[*1] をキーワードとして，データストアの注釈のインデクスを使ってデータを検索する。たとえば，データストアとして Solr を使う場合，以下のような検索クエリが発行される。

```
( ( receptor:"売上"              OR
    receptor:"収益"          )  AND
  ( receptor:"日立製作所@企業" OR
    receptor:"日立@企業"       OR
    receptor:"Hitachi@企業" ) )
```

検索結果として，表 1 の#1-3 のデータが得られる。これらを必要であれば，文字列一致ベースのユニフィケーションによるデータ選択，言語処理，分析，言語合成して，以下のような出力が得られる。シンボル化せずに，テキスト検索の延長として実現しようというモデルであることに留意頂きたい。

> 投資にすべき要因として，下記グラフが示す通り，売上額が過去数年で上昇傾向にあることが挙げられます。2006 年の数値は XO 企業情報データベースから，2010-2015 年の数値は有価証券報告書から，最新情報は OX 新聞から取得しています。

4. プロトシステムの実装

提案モデルがどの程度想定通りの効果を持つかを検証するため，プロトタイプシステムを実装した。プロトタイプシステムは，自然言語でなされた質問に対し，データを用いて分析して回答を示すものである。Vert.x を複数のエージェントが非同期分散実行するためのソフトウェア基盤として用い，Solr をデータストアとして用いて実装した。また Solr の本来の用途は全文インデクスであるため，Cassandra を補助的にデータストアをして用いた。以下のデータをそれぞれ日本語の注釈を付けてデータストアに格納した。

有価証券報告書の XBRL 対応データ 全 XML タグに対して，XBRL 仕様書から注釈を生成。

有価証券報告書の表データ 有価証券報告書の HTML 形式のデータの中の表を解析して，セグメントごとの売上高，営業利益，従業員数に対し，表のタイトルや表のヘッドの情報をもとに注釈を生成。

ニュース記事に書かれている売上高データ ニュース記事を自然言語解析して，売上高情報が書かれている部分を抽出し注釈を生成。

DBpedia DBpedia の企業に関するエントリの情報の一部に対し，注釈を生成。

エージェントとして，(1) 特定の業界に進出しようとしている企業を回答するロジック，(2) 売上高や純利益など，XBRL 形式の有価証券報告書のデータに対して質問回答するロジック，(3) ニュース記事の URL を提示すると，そこに書かれている売上高情報を言語処理により抽出してデータストアに格納し売上高の質問に回答できるようになるロジック，(4) 売上高増加の要因など，理由を回答するロジック，(5) 企業の概要を回答するロジック，などを実装した。

以下にシステムとのやり取りの例を示す。5 程度のロジックであるが，システムが正常に動作することを確認し，提案モデルの原理検証をすることができた。

*1 「収益」は売上高と同じ意味で使われることもあるが，一般的にはサービス収入なども含んだ広い概念。

ユーザ入力：日立製作所の 2016 年の情報・通信システムセグメントの利益は？
回答：
2016 年の情報・通信システムの利益は 109141 百万円です。

ユーザ入力：IoT 業界に参入しようとしているのは?
回答：
サン電子株式会社と，日本精機はと，などの 3 社です。たとえば，2016 のサン電子株式会社に関してはその他の事業 M2M 通信機器の開発では、モバイルルータ「Rooster」シリーズの開発で培った技術で、IoT ／ M2M 市場に参入し、継続してモバイル通信端末の開発を推進し、国内サードパーティ通信 Box マーケットにおいて 4 年連続シェア No. 1 を実現しております。（2016 年有価証券報告書より）

5. 考察

まだ対話システムと呼べる水準には達しておらず，原理実験レベルであるが，提案モデルの想定通りにシステムが動作することが分かった。提案モデルがどの程度有効に機能しているかを定量的に評価することは困難であるが，提案モデルにより機能をある程度まではスケールさせていける見込みを得ている。以下では，プロトシステムの実装を通して得たシステム設計に関する洞察をまとめる。

- 情報を構造化しようとした場合，合意がとれる構造（スキーマ）を定義するのには時間とコストがかかり，また頻繁に改定や別の似たスキーマとの不整合が生じる。また 2 つの異なるスキーマ間でマッピングをとる必要があり，そのマッピングは厳密な意味の同一性を考慮して完全に行うことはほぼ不可能であり，かつ誤差を許したとしても極めて困難な作業となる。そこで，情報は可能な限り構造化しないという思い切った選択があり得る。一方で，自然言語は，複雑な世界で 2 つのオブジェクトがやり取りする唯一の実現例とも考えられる。そこで，十分な機能を持つ複雑なシステムを構築する場合，自然言語を核として構築する方針は手段の 1 つとなり得る。

- その場合，自然言語は普遍的には，意味理解できないと考えるアプローチがある。この立場では，言語はあくまで記号であり，自然発生的には意味は宿らないと考える。受け取ったときに解釈が生じるだけとみなせば，システムの入力部で意味を決定するのではなく，各々のロジックを実行するエージェント部で，そのエージェントが実行されるコンテクストでのみ，解釈をすることができると考える。

- 上記の立場に立てば，自然言語で記述されたクエリおよび情報は，ロジックとは独立に意味を持てなくなるため，大局的に ID 化（グラウンディング）することができなくなる。従って意味は相対的なものとなり，意味を一つに決定するグラウンディングや Dis-ambiguation ではなく，同一性/類似性を判定するユニフィケーションのような考え方の方が望ましいという議論に至る。同一性/類似性の判定は，Word Embedding などの手法も使えるが，結局は，自然言語で表記されたときの「文字列」にかなりの部分は依存することになる。

- 情報は，極論すれば，それを探すためのインデクスと，情報の内容自体の 2 つからなるとみなせる。インデクス部分は可能な限り構造を排除し，かつインデクスの構造に規則を設けず，インデクスを自然言語で記述することで柔軟性が上がる。「他の人が読んでも理解できる」ようにインデクスを書いておけば，原理的には検索でそのデータを見つけられる可能性がある。情報の内容自体は，自然言語で書いておけば，その情報のコンテクストが失われずに情報を保存できる。もしその中にある数値情報を使って統計処理をする場合は，データをデータストアから取り出した後で，言語処理をして数値情報を取り出す。

- エージェントが事前に意味理解されていない自然言語のクエリを受け取り，エージェントが独自のその解釈をするという考え方は，ある機能（API）を呼び出すときに，その引数は自然言語で書かれたコンテクストだけを渡し，API 呼び出しの変数を分割して意味づけしないという考え方に発展する。この考え方を推し進めることで，システム設計が容易になり，API の更新や，API 仕様の伝達ミス，仕様のアンマッチに対する頑健性が増す。具体的な例で示せば，getRevenue(company=”X 社”, year=2016) は，getRevenue(context=”2016 年の X 社”) としても，context を十分に解釈できるように実装すれば，getReveneu を実行することができる。別の言い方をすると，変数の値の内容を十分に解釈すれば，機能はその情報の使い方をある程度，推測できるため，事前に変数の意味づけが十分になされていなくても，機能を実行することができる。

6. おわりに

本稿では，破たんなく機能をスケールさせるためのシステムのモデルを提案し，プロトシステムの実装を通して，提案モデルの原理検証をした。今後の研究課題として，定量的な評価方法を確立すること，機能の数を増やして提案モデルの検証を行うこと，ロジックを自動で学習できる仕組みの考案することがある。

Solr のインデクシング機能やデータストアとしての性能は不十分であり，現状，データストアとして利用可能な適切なテクノロジがないことも課題である。本稿での提案モデルは全文検索の考え方に寄せたモデルであるが，より推論エンジンの考え方を取り入れたモデルも構成可能と思われる。

参考文献

[Katz 02] Katz, B., Felshin, S., Yuret, D., Ibrahim, A., Lin, J. J., Marton, G., McFarland, A. J., and Temelkuran, B.: Omnibase: Uniform Access to Heterogeneous Data for Question Answering, in *NLDB*, pp. 230–234 (2002)

[Katz 06] Katz, B., Borchardt, G. C., and Felshin, S.: Natural Language Annotations for Question Answering., in *FLAIRS Conference*, pp. 303–306, AAAI Press (2006)

[Minsky 09] Minsky, M.: ミンスキー博士の脳の探検: 常識・感情・自己とは, 共立出版 (2009)

[Sato 15] Sato, M., Yanai, K., Miyoshi, T., Yanase, T., Iwayama, M., Sun, Q., and Niwa, Y.: End-to-end Argument Generation System in Debating, in *Proceedings of ACL-IJCNLP 2015 System Demonstrations* (2015)