

常識推論を支援するための辞書（あるいはオントロジー）の構築方法

山田隆弘¹

¹CONOCIMISTA

tyamada99@yahoo.co.jp

概要

自然言語処理技術を用いて推論を行う際、文章の中で使われている語や句の意味を判定するために電子的に処理可能な辞書（あるいはオントロジー）が使用される。ところが、これらの辞書やオントロジーには、推論を行うために必要となる情報が十分に含まれていない場合がある。本稿では、推論（特に矛盾の検出）を支援するための辞書（あるいはオントロジー）を構築するための方法を提案する。推論において使用される知識には常識と世界知識とがあるが、本稿では、常識を使用した矛盾の検出を考察の対象とする。

1 はじめに

自然言語処理技術の用途の一つは、自然言語で書かれた文章について推論を行うことである。推論の際、文章の中で使われている語や句の意味を判定するために、電子的に処理可能な辞書（あるいはオントロジー）が使用される。代表的な辞書としては、FrameNet [1]や WordNet [2]が知られている。ところが、これらの辞書やオントロジーは、必ずしも特定の推論を支援するために作られたわけではないので、推論を行うために必要となる情報が十分に含まれていない場合がある（具体例については、2.1節で述べる）。

そもそも、辞書にしてもオントロジーにしても、具体的な使用目的を定め、その目的を達成できるように「設計」すべきものである。そこで、本稿では、推論を支援するための辞書（あるいはオントロジー）を設計するための方法の一つを具体的に提案する。

自然言語処理技術を利用した推論としては、いくつかの種類が考えられるが、本稿では、一つの文書内（あるいは複数の文書間）の矛盾の検出を取り上げる。その理由は、文書内（あるいは文書間）の矛盾の検出は、実用的に重要な課題であるからである。文書内（あるいは文書間）の矛盾を放置したままに

すると重要な損害がもたらされる場合は多い。例えば、一つの人工衛星に関する技術文書間の矛盾[3]、あるいは、企業が発行する文書と関連する法律の間の矛盾などがあげられる。

また、推論において使用される知識には、大きく分けて、常識と世界知識とがある。この両者を厳密に区別することは困難であるが、本稿では、概念レベルの知識を常識と考え、個々の事物レベルの知識を世界知識と考えることにする。すなわち、常識は、記述論理[4]の T-Box に相当し、オブジェクト指向モデリング[5]のクラスモデルに相当する。また、世界知識は、記述論理の A-Box に相当し、オブジェクト指向モデリングのインスタンスモデルに相当する。本稿では、常識を使用した矛盾の検出を考察の対象とする。

以上をまとめると、本稿では、文章中の矛盾の検出を行うための常識（概念レベルの知識）を記述するための辞書（あるいはオントロジー）の設計方法の提案を行う。辞書とオントロジーの区別は判然としないが、以降では、原則として、辞書とオントロジーの両者をまとめて「辞書」と呼ぶことにする。

なお、本稿の内容は、筆者の過去の研究[6]をさらに発展させたものである。

2 関連研究

2.1 矛盾の検出

自然言語処理技術を使用して文書の矛盾を検出する研究としては[7-10]等がある。これらの研究の大部分において、語の意味に起因する矛盾を検出するために WordNet [2]等における反義語の規定が使用されている。例えば、「大きい」と「小さい」は反義語であるから、「Xは大きい」と「Xは小さい」は矛盾しているとして判断される。

しかし、矛盾とは「同時には成立しない」ことであるから、必ずしも意味が反対でなくても矛盾していることはあり得る。例えば、「太郎は歩いている」

と「太郎は走っている」は矛盾している。なぜならば、「歩く」と「走る」を同時に行うことは困難であるからである。また、「太郎は昨日大阪から東京に戻った」と「太郎は今日は大阪に滞在している」は矛盾している。矛盾していないとすれば、「太郎は大阪から東京に戻った後で再び大阪に行った」という情報が不足していることになる。

文書中の矛盾を検出するために必要な情報を十分に含んでいる辞書は(筆者の知る限り)存在しない。文書中の矛盾を検出するためには、そのために必要な知識を蓄積した辞書を構築する必要がある。

2.2 辞書（あるいはオントロジー）の設計

特定の目的のためにオントロジーを設計した例として、Event and Situation Ontology (ESO) [11]を取り上げる。このオントロジーは、様々なニュース記事より事象を抽出し、抽出された事象を互いに関連づけるための道具として開発された。

この目的のために、このオントロジーでは、動的事象 (dynamic state) と静的事象 (static event) を定義する。ただし、この両者は一般的には事象(event) と状態 (state) と言われている概念にそれぞれ相当するので、以下では事象と状態という用語を使用することにする。

このオントロジーでは、変化を伴う事象は、その変化の内容を事前条件と事後条件によって規定する。事前条件と事後条件は、このオントロジーで定義されている状態に対応づけて規定する。例えば、「XはAからBに移動する」という事象の事前条件は「XはAに存在している」であり、事後条件は「XはBに存在している」である。ここで、「XはYに存在している」は、このオントロジーにおいて状態として定義されている。このように事象と状態を関連づけることにより、このオントロジーで定義されている概念の範囲内であれば、複数の事象や状態の間の関係を統一的な形で表現できる。

このオントロジーは矛盾検出用ではないので当然であるが、このオントロジーを矛盾検出に利用するには、いくつかの情報が不足している。例えば、複数の事象や状態が同時に成立するかどうかを判定するための情報が欠けている。

本稿の第3節以下では、矛盾検出のための辞書が満たすべき条件（あるいは設計原理）を明らかにしながら、そのような辞書を構築する方法を提示する。

3 ドメインモデルの構築

3.1 ドメインモデルの概念

本稿で構築する辞書でも ESO [11]と同様に状態 (state) と事象 (event) を定義するが(第4節参照)、本稿の方法では、状態と事象で参照される基本概念をドメインモデル[6]として定義する。ドメインモデルとは、特定の分野で使用される基本概念を一つのモデルとして定義したものである。ESOでは、事象の事前条件と事後条件を状態に準拠させることで全体の整合性を確保しているが、本稿では、事象と状態の双方をドメインモデルに準拠させることによって、さらに高いレベルの整合性の確保を狙っている。

3.2 ドメインモデルの定義

本稿ではオブジェクト指向モデリング[5]の手法を用いてドメインモデルを構築する。ドメインモデルの主要な構成要素を以下に示す。これらの要素は、Unified Modeling Language (UML) [12]のクラス図を用いても定義できる。オブジェクト指向モデリングの詳細は[5]や[12]を参照して頂きたい。以下で、オブジェクトとは、個々の物を属性値の集合として表現した仮想的な物である。

- クラス（共通の特徴を有するオブジェクトの集合）
- 属性（オブジェクトの特徴を表すパラメータ）
- 属性値（属性の取る値）
- 属性値型（属性の取り得る値の集合）
- 関係（複数のオブジェクト間に成立する関係）
- 属性の多重度（一つの属性がいくつの属性値を持てるか）
- 関係の多重度（一つのオブジェクトが他のいくつのオブジェクトと同一の関係を持てるか）

属性値型には、整数、実数、文字列、列挙型などがある。列挙型とは、いくつかの離散的な値（文字列で表される）の集合として規定され、それらのうち一つあるいは複数のものを値としてとることを示すものである。

このドメインモデルを矛盾の検出のために使用するためには、属性値型が列挙型の場合、列挙されている個々の値が互いに排他的である必要がある。列挙型の属性が二つ以上の異なる属性値を同時に持つことがある場合は、その属性の多重度を2以上にする必要があり。また、属性と関係の多重度の指定は

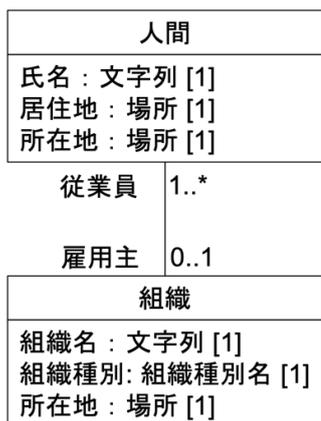


図 1 ドメインモデルの実例

必須である。これらは、矛盾検出のための辞書が満たすべき条件である。

3.3 ドメインモデルの実例

簡単なドメインモデルの例を UML のクラス図を用いて図 1 に示す。この図では、以下の要素が定義されている。

- クラス：人間，組織
- 属性：氏名，居住地，所在地，組織名，組織種別
- 属性値型：文字列，場所，組織種別名
- 関係：従業員—雇用主

図 1 で示されている属性値型のうち組織種別名は列挙型であり、その定義は以下になっている。

組織種別名 = {会社，役所，任意団体}

ここでは、会社，役所，任意団体の三者は互いに排他的であるとする。

さらに、この図では、多重度も示されている。属性の多重度は、属性値型の後の [] 内に示されているが、全ての属性について 1 である。すなわち、全ての属性は一つの属性値を常に持っていなければならない。関係の多重度は、以下のように示されている。

(このドメインでは) 一人の人間は、0 または一つの組織を雇用主として持ち得る。すなわち、一人の人間は、雇用主を持たないか、あるいは、一つの雇用主を持つかのどちらかである。一方、一つの組織は、一つ以上 (上限は設定されていない) の人間を従業員として持ち得る。

4 状態と事象の定義

ここでは、第 3 節で定義したドメインモデルに基づいて状態(state)と事象(event)を定義する方法を示

す。ここで登場するオブジェクトは、ドメインモデルで定義されている何らかのクラスに属しており、属性あるいは関係は、そのクラスで定義されている属性あるいは関係である。

4.1 状態の定義

本稿の方法では、各々の状態は以下の要素によって定義される。

〈属性に関する状態〉

- オブジェクト
- 属性 = 属性値
- [時刻あるいは時間間隔]
- [その他]

〈関係に関する状態〉

- オブジェクト
- 関係 [=関係先のオブジェクト]
- [時刻あるいは時間間隔]
- [その他]

上記で、[] 内の要素はオプションであり、必要がなければ示さなくてよい。

例えば、「太郎は大阪に滞在している」という文は属性に関する状態を表していて、以下の要素によって表される。これは、FrameNet の Being_located フレームに概ね相当する。

- 太郎 (人間クラス)
- 所在地 = 大阪
- 時刻：現在

また、「太郎は凸凹商会に勤めている」という文は関係に関する状態を表していて、以下の要素によって表される。これは、FrameNet の Being_employed フレームに概ね相当する。

- 太郎 (人間クラス)
- 雇用主 = 凸凹商会
- 時刻：現在

4.2 事象の定義

本稿の方法では、各々の事象は以下の要素によって定義される。

〈変化を伴う事象〉

- [変化を促すオブジェクト]
- 変化するオブジェクト
- 事前条件
- 事後条件
- [時刻あるいは時間間隔]
- [手段]

- [その他]
〈変化を伴わない事象〉
- [事象を促すオブジェクト]
- 事象の主題となるオブジェクト
- 事象の内容
- [時刻あるいは時間間隔]
- [手段]
- [その他]

上記で、変化（あるいは事象）を促すオブジェクトとは、使役を表すためのものであり、変化する（あるいは事象の主題となる）オブジェクト以外のオブジェクトが事象を引き起こす場合に、事象を引き起こすオブジェクトを示すものである。

事前条件と事後条件は、ESO [11]でも使用されている概念であり、事象の前後で変化する属性や関係を表現するものである。変化を伴わない事象の内容については、同時には成立しない事象の集合（例えば、「歩く」と「走る」等）を規定するものとする。手段等の他の項目についても、同時に使われることのない（すなわち排他的な）値の集合を定めるものとする。これらは、矛盾検出のための辞書が満たすべき条件である。

例えば、「太郎は昨日大阪から東京に移動した」という文は変化を伴う事象を表していて、以下の要素によって表される。これは、FrameNet の Self_motion フレームに概ね相当する。

- 変化するオブジェクト：太郎（人間クラス）
- 事前条件：所在地＝大阪
- 事後条件：所在地＝東京
- 時刻：昨日

5 矛盾検出の例

ここでは、本稿の方法によって検出できる矛盾の例を示す。

・「太郎は役所に勤めている」対「太郎は会社に勤めている」

この矛盾は、（このドメインにおいては）人間クラスのオブジェクトが雇用主を一つしか持てないことと組織クラスの組織種別名が排他的であることにより検出できる。

・「太郎は大阪に行った」対「太郎は東京にいる」

この矛盾は、（太郎が大阪から東京に戻ったという情報がないかぎり）人間クラスの所在地の多重度が1であることにより検出できる。ただし、厳密に言えば、この矛盾検出のためには、東京と大阪は排

他的な場所であるという世界知識が必要になる。

6 考察

6.1 他の辞書やオントロジーとの関係

本稿では、新しい辞書の構築方法を提示したが、既存の様々な辞書との関係も考慮する必要がある。一つの辞書であらゆる分野や用途をカバーできれば理想的であるが、それは現実的ではないので、以下のような方法が望ましい。

ドメインモデルで使われるクラス、属性、関係などの概念は、SUMO [13]等の上位オントロジーで定義されているものを使用するか、これらの上位オントロジーとの対応関係を示すべきである。状態と事象については、FrameNet [1]のフレームに概ね対応するが、厳密な一対一の対応ではないので、両者間の対応関係を詳しく示すべきである。

6.2 将来の課題

[7]や[9]で指摘されているように、A と B が矛盾しているかどうかを検出するためには、A と B が共に同一の事物について言及しているかどうかの確認が必要である。これは必ずしも容易な問題ではなく、辞書だけで解決できる問題でもないが、何らかの解決策が必要である。また、本稿では常識のみを利用したが、これに世界知識を組み込むための取り組みも必要である。本稿では、推論の一例として矛盾検出を取り上げたが、他の推論のタイプに適用できる手法も開発すべきである。さらに、ここで提示した辞書を機械で処理できる形式で表現する方法も必要であるが、このためには筆者の別の研究[14]等を利用する予定である。

7 おわりに

本稿では、矛盾の検出を支援するための辞書（あるいはオントロジー）の構築方法の一例を示した。本方式の特徴は、まずドメインモデルを構築し、それ以外の概念は全てドメインモデルに準拠して定義することである。ただし、本稿で示したのは大まかな構想のみであり、詳細は今後詰める必要がある。また、第6節で示したように今後解決すべき課題も多い。本稿で強調したいことは、辞書やオントロジーは使用目的に合致するように「設計」すべきであることである。実社会で役に立つような研究が今後ますます盛んになることを願っている。

参考文献

- [1] Josef Ruppenhofer, Michael Ellsworth, Miriam R. L. Petruck, Christopher R. Johnson, Collin F. Baker and Jan Scheffczyk, *FrameNet II: Extended Theory and Practice*, 2016.
- [2] Christiane Fellbaum, *WordNet: an electronic lexical database*, MIT Press, 1998.
- [3] 山田隆弘, “バックキャストによる技術開発のすすめ”, ISAS ニュース, JAXA 宇宙科学研究所, No. 488 (2021 年 11 月), p. 8, 2021.
- [4] Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniel Nardi and Peter F. Patel-Schneider, *The Description Logic Handbook: Theory, Implementation and Applications*, Second Edition, Cambridge University Press, 2010.
- [5] Michael Blaha and James Rumbaugh, *Object-Oriented Modeling and Design with UML*, Second Edition, Prentice Hall, 2005.
- [6] 山田隆弘, “公理としてのドメインモデルに基づく意味の規定について,” 第 31 回人工知能学会全国大会, 2017.
- [7] Sanda Harabagiu, Andrew Hickl and Finley Lacatusu, “Negation, Contrast and Contradiction in Text Processing,” *Proceedings of the Twenty-First National Conference on Artificial Intelligence (AAAI-06)*, pp. 755-762, 2006.
- [8] Marie-Catherine de Marneffe, Anna N. Rafferty and Christopher D. Manning, “Finding Contradictions in Text,” *Proceedings of ACL-08: HLT*, 2008.
- [9] Aikaterini-Lida Kalouli, Livy Real and Valeria de Paiva, “Correcting Contradictions,” *Proceedings of the Computing Natural Language Inference Workshop*, 2017.
- [10] 外園康智, 長谷川貴博, 渡邊知樹, 馬目華奈, 築有紀子, 谷中瞳, 田中リベカ, Pascual Martínez-Gómez, 峯島宏次, 戸次大介, “意味解析システム ccg2lambda による金融ドキュメント処理,” 第 32 回人工知能学会全国大会, 2018.
- [11] Roxane Segers, Piek Vossen, Marco Rospocher, Luciano Serafini, Egoitz Laparra and German Rigau, “ESO: a Frame based Ontology for Events and Implied Situations,” *Proceedings of MAPLEX 2015*, 2015.
- [12] Grady Booch, James Rumbaugh and Ivar Jacobson, “*The Unified Modeling Language User Guide*,” Second Edition, Addison-Wesley, 2005.
- [13] Ian Niles and Adam Pease, “Towards a Standard Upper Ontology,” *Proceedings of the 2nd International Conference on Formal Ontology in Information Systems*, 2001.
- [14] 山田隆弘, “Entity-Relationship モデルに基づいた知識グラフの構築方法,” 第 34 回人工知能学会全国大会, 2020.