

オープンドメイン質問応答における type とその資源構築

金山 博

日本アイ・ビー・エム株式会社 東京基礎研究所

hkana@jp.ibm.com

1 はじめに

米国のクイズ番組において人間のチャンピオンと対戦した質問応答システム Watson [1, 7] は、歴史、文学、芸能、スポーツ、言葉遊びなどを含む幅広い分野の問題に対して高い正解率を示し、勝利を収めた。このようなオープンドメインの状況では、答えるべき事物の型 (type) が非常に多岐にわたるため、個々の type に特化したシステムを設計して網羅することは不可能である。そこで、最初に意味的な type を確定させて解答に制約をかけるという特定分野向けの質問応答システムで取られてきたアプローチのかわりに、文書検索等によって解答の候補を列挙してから、それぞれの候補と表層上の type との一致度を測ってスコア付けをするという方針を採った。本論文では、type の一致度を多角的に検査する Type Coercion (TyCor) [2, 5] の仕組みのうち、Wikipedia など百科事典の本文から構築したリソースによる指標を中心に述べる。

2 節では質問応答システムの流れについて、3 節では type の一致を求めるコンポーネントについて解説する。4 節で、type と性別に関する情報を百科事典の本文から自動的に抽出する手法、5 節ではそれらのリソースを用いた type 一致の素性 IntroCor, GenderCor について述べ、6 節でその効果を検証する。

2 DeepQA アーキテクチャ

今回の題材である米国のクイズ番組 Jeopardy! の問題は、表 1 の例のように、疑問文ではなく、解答に関する事実を含む平叙文で書かれており、“this” を伴う語や “he” などの代名詞に相当する名詞や固有名詞が答えとなるのが典型である。

この種の問題、さらには Jeopardy! 特有の問題にも対応する質問応答システム Watson は、図 1 に示す DeepQA アーキテクチャ [1] により実現されている。DeepQA は、入力された質問文に対して、以下の 4 つの処理を実行し、解答を確信度付きで出力する。

質問の分析 入力された質問文およびカテゴリ (問題の分野) のテキストに対して、構文解析等の前処理を行う。ここで、表 1 で斜体字で示されている、表層上の解答の型 (Lexical Answer Type ; 以下 LAT) を定める。

解候補の生成 質問文に含まれる語句を大量の文書から検索し、高頻度で共起する語を列挙する。さらに正答となりやすい語句を拡張して、数百の解候

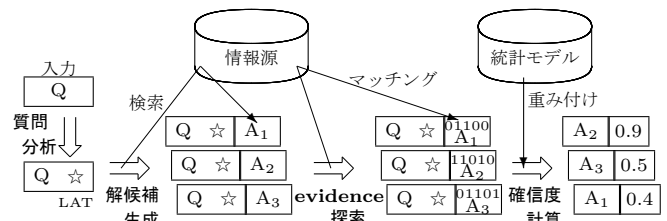


図 1: DeepQA アーキテクチャの概要

補を生成する。解候補の一部は、Wikipedia のエントリを由来とし、多義語の曖昧性が解消された形となっている。

evidence の探索 質問文は、解答が持つ性質の集合と捉えられる。各解候補がそれらを満たすことを示す evidence を情報源の中から探索して、その有無を素性として付与する。例えば、同等の論理関係が情報源に見つかるか、問題文中に現れる時間表現が解候補と関連しているか、といったものが evidence となる。次節で説明する TyCor もこの処理に含まれる。

確信度の計算 上記で付与した素性を用いて、過去のクイズ番組の問題と正答を学習データとした機械学習を行う。これにより、各 evidence に重みが与えられ、各候補の確信度が求まる。この結果、確信度が最大の候補を出力する。

3 TyCor (Type Coercion)

解候補が正答となることの evidence として最も基本的なものが、解候補の type と問題の type との一致である。しかし、今回の対象となるクイズ番組の問題では、type となるものの数が非常に多い。LAT を基にした調査 [1] によると、20,000 問の中に 2,500 種もの異なる LAT があり、頻出する 200 種の LAT では約半数の問題しかカバーできない。さらに、正解が同じ問題であっても、その問題文中の LAT は一意とは限らない。従って、特定のオントロジーの上で type の一致を検査することは本質的に困難である。

表 1 を見ると、1 は基本的な LAT であり語彙体系でも同様の記述が期待できるが、2 は “publisher” の他にも “writer” など様々な記述が可能である。3 は抽象的な LAT、4 は代名詞で男性の人物であることだけを示しており、5 はより抽象的なものとなっている。

表 1: クイズ番組の問題と正答、および LAT (斜体字部) の例

	問題文	正答
1	Japan's Akashi Kaikyo Bridge links the island of Awaji to the city of Kobe on this <i>island</i>	Honshu
2	This <i>publisher</i> who died in 1967 put the time in Time Warner	Henry Luce
3	1996 was a good year for Fox as 2 of its shows won: "The X-Files" and this animated <i>series</i>	The Simpsons
4	<i>He</i> hit his last, massively long home run o o pitcher Guy Bush in May 1935	Babe Ruth
5	When hit by electrons, a phosphor gives o electromagnetic energy in this <i>form</i>	Light (or Photons)

そこで、複数の情報源を用いて多角的に type の一致を調べることにする。この操作を、type を適合させられるかどうかの検査と見なし、Type Coercion、略して TyCor と呼ぶ。TyCor の各要素は、LAT と解答の候補を入力として、型が一致すれば 1、一致しなければ 0、またアルゴリズムによってはその中間の実数値を返す。主なものを以下に示す。

1. **WordNet:** WordNet の体系の中で、解候補が LAT の下位語になっているか
2. **YAGO:** Wikipedia のカテゴリを基に生成したオントロジーである YAGO [6] で LAT と解候補の間で Is-A 関係が認められるか
3. **語彙の制約:** LAT が規定する解答の制約、例えば “phrase”などを、解候補が満たしているか
4. **固有表現:** 約 100 の頻出 LAT に対して、Named Entity Detector の結果と一致するか
5. **Wikipedia カテゴリ:** LAT と関連している Wikipedia のカテゴリ体系の中に、解候補に関連するページが含まれるか
6. **Wikipedia リスト:** LAT に相当する Wikipedia のリストのページに解候補が入っているか
7. **導入文:** 百科事典本文の一文目 (Introductory Sentence) に書かれている type と一致するか
8. **性別:** 解候補が人名や人物を表す語の場合に、男性・女性の区別が LAT が示すものと一致するか

上記のうち、1~6 は、人手で整備されたリソースや規則を用いて type の一致を調べる。7 と 8 については、それらとは異なる観点で調べるために、Wikipedia をはじめとした百科事典のタイトルと本文を用いて、リソースを自動的に構築した。次節でその手法について述べる。

4 百科事典本文を用いた情報抽出

百科事典データは、以下の性質があることから、type と性別の情報を取得するのに好都合である。

項目のタイトル (見出し語) として、対象となる事物が決まっている。特に、多義語の場合には曖昧性が解消済である。

第一文の多くが、“X is Y” などの形で、見出し語 X の定義が書かれており、Y が type に相当する。

Henry Luce

Henry Robinson Luce (April 3, 1898 – February 28, 1967) was an influential American publisher. He launched and closely supervised a stable of magazines that transformed journalism and the reading habits of upscale Americans.

図 2: Wikipedia のエントリの例

一般の利用者向けの記述であるため、学術的な定義から事物を厳密に分類するというよりは、直感的な表現を使う傾向があり、問題文の LAT と一致する type が多いことが期待される。一文目に続く文も、見出し語についての記述が中心となっており、代名詞の先行詞が見出し語となることが多い。

例えば、Wikipedia で “Henry Luce” は図 2 のように記述されており、表 1 の例 2 の LAT と同一の表現が第一文に、男性を示す代名詞が第二文に存在する。

以下で、type と性別の情報を取得する方法について記述する。

4.1 第一文に含まれる type

固有表現抽出の辞書の作成を目的として、Wikipedia の本文から type を抽出する Kazama らの研究 [3] では、約 61% の記事において type を抽出することができ、既存の辞書や Wikipedia のカテゴリ情報を用いるよりもシステムの正解率向上に貢献したことが報告されている。

我々も同様の手法により、百科事典の第一文から type の抽出を行った。Kazama らの手法との違いは、tagger の結果に基づいたパターンマッチをする替わりに、高精度なパーザ ESG[4] による構文解析を行い、構文木上で type を抽出するパターンを適用したことで、等位接続詞や同格表現を用いて複数の type がある場合に網羅的に抽出している点である。

我々は、be 動詞を中心とした主節の構文に関するパターン 23 個、そこから抽出される語を具体的なものに変える名詞句パターン 16 個、等位接続詞・同格表現から複数の type を抽出する規則を再帰的に適用した。パターンの例を表 2 に示す。このうち、X が見出し語と一致 (部分一致を含む) するものに対して、Y を type として検出する。

これによって、2010 年 8 月の英語版 Wikipedia の 3,253,631 ページのうち、2,624,808 ページから一つ以上の type を抽出することができた。複数の type が取得されるものがあるため、延べ 3,226,793 件の type の情報が得られている。

表 2: Type 抽出に用いた構文パターンの例

主節パターン	名詞句パターン
X is Y	one of Y
X is founded as Y	name of Y
X is known as Y	mass of Y
X is classified as Y	any of Y
X is defined as Y	kind of Y
X is published as Y	style of Y
X is regarded as Y	sort of Y
X is considered to be Y	species of Y
X is elected as Y	name for Y
X, Y, was born	term for Y

表 3: 抽出された type の数と、正解に対して type が得られたものの割合（被覆率）

手法	抽出 type 数	被覆率	
		Wikipedia	merged
“X is Y” のみ	2,402,855	71.4%	72.3%
+主節パターン	2,491,909	73.6%	74.1%
+名詞句パターン	2,624,808	76.0%	77.1%
+等位接続	3,226,793		

表 3 に、各手法を取り入れたときの type の数、そして 20,000 問の正解に対して、何らかの type が割り当てられたものの割合を示す。Wikipedia のみを用いた時のほか、他の百科事典を組み合わせた場合 (merged) の被覆率も合わせて示した。主節パターンや名詞句パターンの導入により、被覆率が上がっている。

4.2 性別の判定

次に、主に人名の性別の判定をするためのデータを Wikipedia の本文から抽出する。以下にそのアルゴリズムを示す。

手法 A: 2 文目の主語が “he” または “she” であるか、主語が所有格の “his” または “her” で修飾されている時に、見出し語の性別をそれぞれ男性・女性とみなす

手法 B: 手法 A で性別が定まらない時に、最初の段落に含まれる代名詞 “he” (“his”, “him” を含む) と “she” (“her” を含む) の数のうち、多い側を見出し語の性別とみなす

手法 C: 手法 B で性別が定まらない時に、記事全体の “he” (“his”, “him” を含む) と “she” (“her” を含む) の多い側を見出し語の性別とみなす

人名の性別に関する網羅的なデータを持っていないなか、各手法の妥当性を検証したい。そこで、Wikipedia のエントリのうち、4.1 節の手法により抽出された type が女性にほぼ限定される “actress”, “goddess” と、男性が多いと思われる “actor” であるものについて、本手法で性別を推定したところ、表 4 のようになった。被覆率は、性別が推定されたエントリの割合、適合率は、“actress”, “goddess” が女性だと推定された割合を示す。手法を組み合わせることによって得られる情報が増加し、特に “actress” については、極めて高い適合率を保っていることがわかる。“actor”

が “she” と判定された例を見ると、‘*Traudl Stark was a child actor in German movies. She was born as ...*’ のように実際に女性である場合がほとんどであった。

この観察結果を受けて、以降では A、B、C の全手法を用いて作成したリソースを使うことにする。Wikipedia の 3,253,631 エントリのうち、1,383,410 エントリに対して性別の情報が得られた。なお、手法 C の性質上、性別と関連のないエントリについても、別の人称代名詞の影響で性別が振られてしまう場合があるが、5.2 節で対処する。

5 IntroCor & GenderCor

前節で構築した type と性別に関するリソースを用いて、TyCor の要素である素性を設計する。

5.1 IntroCor

4.1 節の手法により百科事典から抽出した type の情報を用いて、解候補の type のいずれかが、質問文の LAT と重なりを持つなら 1、そうでなければ 0 を返す関数が、IntroCor 素性である。但し、そのまま一致する場合は決して多くないので、複合語の部分一致や、WordNet の上位語による拡張などにより、一致条件を緩和する。その場合に僅かな減点がされるので、実際は実数値を取る素性となるが、以下では簡単のため 2 値の素性として考えてよい。

5.2 GenderCor・反 GenderCor

4.2 節の手法で得た性別の情報のうち、明らかに人間を表さないものを WordNet の上位語等を参照してフィルタリングしたのち、性別データとして用いる。その上で、以下の二つの素性を作成する。

GenderCor: LAT が代名詞 “he”, “she” や、“actress” など性別を持つ語の場合に、解候補の性別と一致するなら 1、そうでなければ 0 を返す素性。

反 GenderCor: LAT が上記と同様の条件の時、解候補の性別と矛盾するなら 1、そうでなければ 0 を返す素性。例えば、LAT が “actress” であるのに解候補の性別が男性である時に 1 となり、機械学習により強い負の重みが付与されることが想定されるものである。性別を持たないものの時には GenderCor 同様に 0 となる点で、GenderCor の補数とは異なる。

次節では、これらの素性が 0 以外の値を示すことを「素性が発火する」と表現する。

6 評価実験

上記の素性が、正答と誤答を弁別する際に、どのようにシステムに貢献するかを、素性値と正解の関係の観察、および質問応答の全プロセスを通した実験を通して確認する。

表 4: 手法 A~C による性別の自動判定の結果

type	頻度	性別推定結果						被覆率			適合率		
		女性 (she)			男性 (he)			A	A+B	A+B+C	A	A+B	A+B+C
		A	B	C	A	B	C						
actress	7,917	2,506	1,230	3,542	2	2	12	31.7%	47.2%	91.9%	99.9%	99.9%	99.8%
goddess	432	163	138	27	1	2	5	37.7%	69.7%	76.0%	99.4%	99.0%	97.6%
actor	11,214	125	38	208	3,763	1,522	4,653	32.0%	48.6%	91.9%	-	-	-

表 5: 主な TyCor 素性の発火と正答の関係

素性	正答発火	誤答発火	正答被覆	MI
WordNet	950	18,179	5.5%	2.4
YAGO	2,104	86,067	64.4%	1.3
IntroCor	1,517	42,557	46.4%	1.8
GenderCor	436	13,959	13.3%	1.6
反 GenderCor	8	3,123	0.2%	-1.9

表 6: 質問応答システムの正解率

手法	全体正解率	70% 正解率
TyCor 無し	65.1%	81.1%
WordNet	66.5%	82.9%
YAGO	67.8%	83.5%
List	66.8%	82.8%
IntroCor	66.8%	83.1%
GenderCor	65.7%	82.1%
全 TyCor	69.9%	87.1%

6.1 素性と正答の関係

IntroCor や GenderCor の素性は、解候補が正答の時に発火し、誤答の時に発火しないことが、反 GenderCor はその逆になることが望ましい。その効果を見るために、3,508 の質問と、それらに対して生成された 336,499 の解候補（うち 3,269 個が正答、残りが誤答）の上で、次の値を調べた。

正答発火: 解候補のうち、各素性が発火した正答の数

誤答発火: 解候補のうち、各素性が発火した誤答の数

正答被覆: 正答に対して各素性が発火する割合

MI: 正答と素性発火の間の相互情報量

表 5 から、IntroCor は WordNet より高い被覆率を持ち、YAGO よりも正誤との相互情報量が高いなど、語彙体系を利用した type の検査とは異なる性質を持つことがわかる。また、反 GenderCor は、性別の不一致と誤答との関連を GenderCor 以上に強く捉えている。

6.2 質問応答の正解率

Type 及び性別の素性が質問応答システム全体の正解率向上にどれだけ寄与するかを実験した。表 6 は、DeepQA の全プロセスを用いて Jeopardy! の 3,571 問を解かせたもので、TyCor を用いない場合と、一つずつの要素を用いた場合（GenderCor は反 GenderCor を含む）、すべてを組み合わせた場合の正解率を示す。70% 正解率とは、システムの確信度が 70% を超えた問題に限定した時の正解率で、クイズ番組の対戦における強さを近似する指標である。

百科事典本文から自動抽出した type であっても、WordNet や YAGO などの語彙体系を用いた場合と遜色ない結果が得られている。また、人名に関する問題の時にだけ有効である性別の情報も、70% 正解率を約 1 ポイント上げるなど、効果が示されている。TyCor の全要素を組み合わせた時にはさらに効果が大きくなり、複数の観点による type の検査により複雑な体系を補完できていることがわかる。

7 まとめ

本研究では、百科事典から抽出した知識を用いて、質問応答の解候補の正しさを推定する素性を設計し、それらが質問応答システム全体に寄与することを確認した。ここで自動抽出した type と性別の情報は、人手で構築した語彙体系と補完的である。また、百科事典の本文の記述をもとにリソースを自動的に作れることから、集合知の更なる拡がりを質問応答のための知識として取り入れられる点で、本手法の意義は大きい。

参考文献

- [1] David A. Ferrucci, Eric W. Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya Kalyanpur, Adam Lally, J. William Murdock, Eric Nyberg, John M. Prager, Nico Schlaefer, and Christopher A. Welty. Building Watson: An overview of the DeepQA project. *AI Magazine*, Vol. 31, No. 3, pp. 59–79, 2010.
- [2] Aditya Kalyanpur, J. William Murdock, James Fan, and Christopher A. Welty. Leveraging community-built knowledge for type coercion in question answering. In *International Semantic Web Conference (2)*, pp. 144–156, 2011.
- [3] Jun'ichi Kazama and Kentaro Torisawa. Exploiting Wikipedia as external knowledge for named entity recognition. In *Proceedings of EMNLP-CoNLL 2007*, pp. 698–707, 2007.
- [4] Michael C. McCord, J. William Murdock, and Branimir K. Boguraev. Deep parsing in watson. *IBM Journal of Research and Development*, Vol. 51, , 2012. to appear.
- [5] J. William Murdock, Aditya Kalyanpur, Chris Welty, James Fan, David Ferrucci, David Gondek, Lei Zhang, and Hiroshi Kanayama. Typing candidate answers using type coercion. *IBM Journal of Research and Development*, Vol. 51, , 2012. to appear.
- [6] F. M. Suchanek, G. Kasneci, and G. Weikum. YAGO: a core of semantic knowledge unifying WordNet and Wikipedia. In *Proc. of International Conference on World Wide Web*, pp. 697–706, 2007.
- [7] 金山博, 武田浩一. Watson: クイズ番組に挑戦する質問応答システム. *情報処理*, Vol. 52, No. 7, pp. 840–849, 2011.