パターンマイニングを用いて「なぜ」に答えるシステム

磯崎 秀樹　　東中 竜一郎
NTT コミュニケーション科学基礎研究所

我々は「なぜ...」という質問の答を検索によって探しだすシステム NAZEQA を作成している。NAZEQA の特長は、多種多様な原因表現のパターンを、人手ではなく、コーパスからマイニングで獲得する点と、回答候補の採点を、質問回答セットから機械学習した採点関数によって行なうところである。本稿では、BACT を用いて因果表現パターンをコーパスからマイニングし、得られたパターンや質問文との各種類似度を素性として SVM で採点関数を学習する方法を提案する。その結果、IJCNLP-2008 で報告したシステムより、精度を大幅に向上させることができたので報告する。

1　はじめに

本稿では、「なぜ」という質問を受けつけ、大量の文書から検索によってその回答を探し出すシステムを作成したので、これについて報告する。

これまでの質問応答システムは、人名などの固有表現や数値表現など、簡単な言葉で答えられる、いわゆるファクトイド型のものが多かったが、最近は、もっと多様な質問に答えられるシステムの研究に関心が移ってきている [21]。とくに、「なぜ」については、Verberne らが研究をすすめており [22]、NTCIR-6 QAC-4[4] においても、「なぜ」を中心とした質問応答が取り上げられた。

昔から、因果や時間変化に関する知識表現や推論 [1, 20] は、自然言語研究者の興味を引いてきているが、なかなか研究が進展しない。その理由として、知識表現には、変化自体の完全な記述が困難であるという問題 (frame/qualification/ramification problems) があり、どんな場合にも例外や膨大な可能性があるため、時間や知識を含む論理的推論には計算量爆発の問題があり、期待されるような有意義な結論がほとんど得られない。我々も効率のよい因果関係推論に関する研究 [12, 11, 9] を行なってきたが、これら論理的アプローチの限界を回避することはできなかった。

本稿では、これらの反省に基づき、因果に関する知識や推論は直接扱わず、情報検索や情報抽出の延長で「なぜ」に答える。

このシステムを構築する意義は 2 つある。ひとつは、既存の質問応答システムのカバーする範囲を広げ、多様な質問に答えられるようにするという実用的側面である。もうひとつは、長年課題となっていながら、なかなか進展しない意味理解実現の手がかりを得るという学術的側面である。

本システムは、ファクトイド型質問応答システム SAIQA[10] をベースとしており、「質問解析」「文書検索」「回答抽出・評価」という標準的なフローになっている。「質問解析」「文書検索」は SAIQA と同一であり、詳細は省略する。

また、本稿では文を回答の単位とする。1 文では情報が少なすぎて、正解かどうかわかりにくいことがあるが、必要に応じて前後の文を提示することは容易である。

IJCNLP-2008 [6] での我々の実験によれば、成績は段落にした方がかなりよい。これは、段落の方が文よりはるかに数が少なく、当たりやすいからと考えられる。しかし、段落の中には非常に長いものがあり、回答の単位を段落にして提示すると読みづらい。理由を簡潔に提示する、という観点から、本稿では文を単位とする。

各回答候補は、以下の観点から採点する。

- 質問文と回答候補の類似度：質問文と共通の語彙が多いほど、質問の回答である可能性が高い。
- 回答候補中の原因表現の有無：原因は原因らしい表現で書かれているだろう。

ファクトイド型質問応答システムでは、パターンを例題から獲得する方法が提案 [18, 17] されている。一方、原因表現のパターンの獲得については、ほとんど研究されていない。

原因表現のパターンを獲得する手法として、我々は、BACT[14] を用いる方法 [5] と、機能語パターンを用いる方法 [6] を提案しているが、ここでは情報の多い BACT を利用する。そして、マイニングによって得られたパターンや各種類似度の他、様々な素性を利用して、回答候補を採点する。IJCNLP-2008 版 [6] では、採点に Ranking SVM [13] を用いていたが、今回は、リニアカーネル SVM の学習が非常に速い Pegasos [19] を利用することにより、様々な素性の組み合わせを試すことで、成績の向上を試みた。

2　提案手法

以下では、採点に用いる素性について説明する。

2.1　質問文との類似度

類似度としては、質問文中の単語、単語 N グラム、固有表現などが、回答候補中にどれくらい出現しているかで表せる。また、重複のない set として扱うか、bag として扱うか、IDF のような重みを別途計算して用いるか、などの

バリエーションが考えられる。単語 N グラムが入っていれば、固有表現を分けて考える必要は少ないように思われるかもしれないが、固有表現は情報の特定性が他の一般的表現よりも高いので、分けて考える。

我々は、ファクトイド QA に関する経験から、文中の頻度を考慮した bag ではなく、表現の set を用いることにした。検索でよく使われる TF·IDF では、頻度の多い語が高く評価される。しかし、我々の実験 [10] によれば、TF 項の影響は小さいほどよい。同じ検索語の頻度がいくら高くても、不足している検索語のかわりにはならないからであると考えられる。

したがって、質問文中のある種の表現の set を Q、回答候補中の同種の表現の set を C としたときに、カバー率を以下で定義する。

$$\mathrm{cov}(C|Q) = |Q \cap C|/|Q|$$

cosine 類似度を始めとして、多くの類似度の定義では、C の大きさも分母に入っている。しかし、こうした類似度は、長い回答候補が正しい原因を含んでいる場合にスコアが下ってしまう。そこで、C は分母に入れていない。

また、SAIQA の検索エンジンは、検索時に各語の IDF を出力するので、IDF による重みの利用が容易である。そこで、単語については、以下の IDF カバー率を用いる。

$$\mathrm{idfcov}(C|Q) = \sum_{w \in Q \cap C} \mathrm{idf}(w) / \sum_{w \in Q} \mathrm{idf}(w)$$

2.2 原因表現のマイニング

乾ら [8] が指摘する通り、原因の表現は多様であるうえ、明示的な手がかりのないものも多い。しかし、因果関係らしい表現を含んでいない文をユーザに提示しても、回答として説得力に欠ける。また、システムとしても、因果関係らしいパターンをまったく含まない文を不正解と区別して、高く評価するのは困難である。そこで、明示的な手がかり表現はなくても、人間が因果関係の記述である、と認識できる程度のパターンを保持していることが好ましい。

そのような、明示的でないパターンを人手で見つけてプログラムとして書き下すのは困難をきわめる。そこで、我々は、EDR コーパス[1] を利用して、原因表現のパターンをマイニングすることにした。EDR コーパスは単文の集合であり、各文に様々な解析結果が人手で付与されている。とくに今回重要なのは、その中で cause と purpose というタグである。

たとえば「(前略) その骨折りで命拾いをした。」という文であれば、以下のように原因表現である「その骨折りで」が cause タグにより明示されている。

```
[cause [[main 37:骨折り:3cf47a]
        [modifier 36:その:...]]
```

[1] http://www2.nict.go.jp/r/r312/EDR/

我々は、EDR コーパス 208,157 文のうち、cause と purpose を含む 18,524 文を正例、それ以外の文を負例として、原因表現を特徴づけるパターンをマイニングする。

IJCNLP 版では、EDR コーパス中の cause タグのついた原因表現から、機能語だけを残したパターンを作成して、その有無を素性とした。たとえば「声が出なくなって」という原因表現があれば、「＊ が ＊ て」というパターンが得られた。

本稿では、このような機能語パターンのかわりに、BACT によって得たパターンの有無を素性として利用する。

マイニングのため、まず、EDR の各文を平文に戻したあと、cabocha [15] で依存構造解析しなおす。そして、なるべく一般性のあるルールを獲得するため、機能語以外の語 (内容語) を品詞名と日本語語彙大系 [7] の意味カテゴリで置き換えた依存構造木を作成する。意味カテゴリは jtag という形態素解析システム [2] により得る。たとえば、「連勝」という単語は、「名詞-サ変接続」という品詞名と、N-1758 (勝利) という一般名詞カテゴリ、V-22 (結果), V-7 (相対関係) という用言カテゴリで置き換えられる。

BACT は boosting による木の分類器であるが、ここでは分類に有効なパターンをマイニングするためのツールと考える。得られるパターンの長さの上限 (L オプション) は 3 とした。BACT が出力する各パターンの係数や BACT のスコアは利用しない。

2.3 質問と回答のマッチング

回答候補の採点は難しい問題である。NTCIR-6 QAC-4 参加システム [5] では、回答候補の BACT スコアと、質問との類似度に基づく ad hoc な関数によって回答候補の採点を行なった。

IJCNLP-2008 のシステムでは、1998 年から 2001 年の CD-ROM 版毎日新聞 (QAC-4 と同じ設定) の 106,614 記事から 1,000 問の質問とその回答を作成し、機械学習により採点関数を学習した。たとえば：

> 質問：日本でのサミットの開催地にどうして沖縄が選ばれたのか？
> 回答：000116017,L25, 警備上の問題を押し切り首相が決断したのは、「[米軍基地整理・縮小問題を抱える沖縄に光をあて、戦後から脱皮させる道筋をつける」（外務省幹部）という狙いから] で、そこに沖縄開催の意義があったはずだ。

回答は各問題につき複数ありうる。回答の先頭の数字は、文書番号と行番号であり、[ ] は、とくに原因と考えられる部分である。

類似度としては、質問と回答候補の単語頻度ベクトルの cosine 類似度や、検索時の文書のランクやスコア、人手による原因パターン、因果関係にある語の有無などを用いた。そして、学習には svm-light で利用可能な Ranking SVM

[13] を用いた。

しかし、svm-light では学習に時間がかかりすぎるため、今回は Pegasos というアルゴリズム [19] を利用した。訓練データは

{(+1, 正解素性ベクトル−不正解素性ベクトル)}

とした。10-fold 交差検定の場合、1 fold あたり 220 万前後のベクトル数になる。正例しかないが、Pegasos は $f(\boldsymbol{x}) = \boldsymbol{w} \cdot \boldsymbol{x}$ の形の線形関数を学習し、$f(\boldsymbol{x}) = \boldsymbol{w} \cdot \boldsymbol{x} + b$ ではないので、$\boldsymbol{w} = \boldsymbol{0}, b > 0$ という自明な解はない。

Pegasos は、Perceptron によく似た学習アルゴリズムの一種であり、Perceptron との違いは、以下の通り。

- 1 度に 1 つのサンプルではなく、ランダムに選んだ k 個のサンプルをまとめてチェックし、ロスのあるサンプルで $\boldsymbol{w}$ を更新。
- $\boldsymbol{w}$ が大きくなりすぎたら、小さくする。

今回の実験では、$k = 1000$ としたほかは、公開されているプログラムの標準設定を利用した。

なお、カーネルを用いれば、成績の向上が期待できるが、今回は用いなかった。

3 実験結果

まず、マイニングによって得られた上位パターンの例を以下に示す。ただし、BACT のパターンなので、逆から読む。「より に」は「に より」のことである。上位には、人間が書くような明確な手がかり表現が多い。

・動詞-自立 ため N-2456(目的)
・ため
・べく
・こと 動詞-自立
・より に
・により
・によって
・よう 動詞-自立
・ので
・による

各回答候補 (文) を採点するための素性として、以下のものを利用した。

- BACT で得られた各パターンが候補文に含まれるか否か
- 候補文の IDF カバー率 (idfcov)、単語バイグラムカバー率 (cov_{BG})、文末機能語列、文頭の接続詞
- 直前 2 文の上記各素性
- 候補文の前半の idfcov, 後半の idfcov
- 候補文と直前 2 文を合わせた固有表現カバー率 (cov_{NE})
- 質問文中の単語との同義語の有無
- 文書検索時の文書スコア (DIDF [10])

候補文を前半と後半に分けたのは、「〜のため○○した」とか、「○○したのは〜のせいだ」のようなケースを考慮したためである。

表 1: IJCNLP 版との比較 (10-fold 交差検定)

評価	MRR		正解カバー率	
システム	提案手法	IJCNLP 版	提案手法	IJCNLP 版
1 位回答	**0.153**	0.113	**0.153**	0.113
5 位以内	**0.257**	0.196	**0.445**	0.354
10 位以内	**0.276**	0.216	**0.591**	0.504

表 2: 各素性の影響 (10-fold 交差検定)

評価	MRR		正解カバー率	
システム	提案手法	NE なし	提案手法	NE なし
1 位回答	**0.153**	0.146	**0.153**	0.146
5 位以内	**0.257**	0.244	**0.445**	0.422
10 位以内	**0.276**	0.266	**0.591**	0.584
システム	提案手法	前 2 文なし	提案手法	前 2 文なし
1 位回答	**0.153**	0.125	**0.153**	0.125
5 位以内	**0.257**	0.225	**0.445**	0.411
10 位以内	**0.276**	0.246	**0.591**	0.564

また、候補文にすべての必要な固有表現が含まれていることは考えにくい。必要な情報は少しずつ説明をしていくだろう、と考えて、候補文のみの cov_{NE}、候補文と直前の文を合わせた 2 文の cov_{NE}、2 文前も含めた 3 文の cov_{NE} をそれぞれ素性とした。固有表現は cabocha の出力を利用した。

本システムのスコアを IJCNLP 版 [6] と比較した結果を表 1 に示す。「MRR」は、最上位の正解の順位の逆数の平均であり、「正解カバー率」は、回答の中に正解が含まれている問題数の割合である。どちらも 0 が下限、1 が上限で、大きいほどよい。

表から、提案手法は、IJCNLP 版より大きく改善されていることがわかる。提案手法と IJCNLP の成績の差は、符号検定で、1 位正解について $p = 0.05$、5 位以内正解と 10 位以内正解について $p = 0.01$ で有意である。ちなみに IJCNLP 版は、単純な cosine 類似度や、人手パターンによる方法 [3] よりも、統計的に有意な差を持ってよいことが示されている。

固有表現素性 cov_{NE} をすべて削除すると、成績は表 2 の上半分のようになり、固有表現を分けた効果があったと考えられる。

また前 2 文の素性をすべて削除すると、表 2 の下半分のようになり、前 2 文の素性も重要であることがわかる。

提案手法では、候補文のあとの文を素性として入れていないが、あとの文を加えた実験も行ってみたい。

文末機能語列は、水野ら [16] も質問応答で利用している。Pegasos の出力したモデルファイルを見ると、候補文中の文末機能語列でとくに重みの大きかったものとして、

「たからだ」「ではないか」「られなかった」「たんです」「そうです」などがあり、「なぜ」に対する回答らしい表現が得られている。

4 おわりに

今回、いくつもの素性を追加することで、大きく精度を向上させることができた。成績向上に貢献しそうないろいろな素性が、他にも考えられるので、フルペーパーでは、これら様々な素性の効果について調べていきたい。

最後に、Pegasos を教えてくださった MIT の Michael Collins 准教授に感謝します。アドバイスをくれた鈴木潤氏や平尾努氏をはじめとする知識処理研究グループのメンバーにも感謝します。

参考文献

[1] Fagin, R., Halpern, J. Y., Moses, Y., and Vardi, M. Y.: *Reasoning About Knowledge*, MIT Press (1995).

[2] Fuchi, T. and Takagi, S.: Japanese Morphological Analyzer using Word Co-occurrence – JTAG, *Proceedings of COLING-ACL*, pp.409–413 (1998).

[3] Fukumoto, J.: Question Answering System for Non-factoid Type Questions and Automatic Evaluation based on BE Method, *Proceedings of the 6th Workshop Meeting on Evaluation of Information Access Technologies: Information Retrieval, Question Answering and Cross-Lingual Information Access*, pp.441–447 (2007).

[4] Fukumoto, J., Kato, T., Masui, F., and Mori, T.: An Overview of the 4th Question Answering Challenge (QAC-4) at NTCIR Workshop 6, *Proceedings of the 6th Workshop Meeting on Evaluation of Information Access Technologies: Information Retrieval, Question Answering and Cross-Lingual Information Access*, pp.433–440 (2007).

[5] Higashinaka, R. and Isozaki, H.: NTT's Question Answering System for NTCIR-6 QAC-4, *Proceedings of the 6th Workshop Meeting on Evaluation of Information Access Technologies: Information Retrieval, Question Answering and Cross-Lingual Information Access*, pp.460–463 (2007).

[6] Higashinaka, R. and Isozaki, H.: Corpus-based Question Answering for why-Questions, *Proceedings of the Third International Joint Conference on Natural Language Processing*, pp.418–425 (2008).

[7] 池原悟, 宮崎正弘, 白井諭, 横尾昭男, 中岩浩巳, 大山芳史, 林良彦: 日本語語彙大系, 岩波書店 (1997), http://www.kecl.ntt.co.jp/mtg/resources/GoiTaikei/.

[8] 乾孝司, 奥村学: 文書内に現れる因果関係の出現特徴調査, 情報処理学会自然言語研究会 NL-167 (2005).

[9] 磯崎秀樹: マルチエージェント環境で他者の信念の変遷を推定する前進的アルゴリズム, 情報処理学会論文誌, **40** No. 9, pp.3358–3372 (1999).

[10] Isozaki, H.: An Analysis of a High Performance Japanese Question Answering System, *ACM Transaction on Asian Language Information Processing*, **4** No. 3, pp.263–279 (2005).

[11] Isozaki, H. and Katsuno, H.: A Semantic Characterization of an Algorithm for Estimating Others' Beliefs from Observation, *Proceedings of the National Conference on Artificial Intelligence*, pp.543–549, MIT Press (1996).

[12] Isozaki, H. and Shoham, Y.: A mechanism for reasoning about time and belief, *Proceedings of the International Conference on Fifth Generation Computer Systems*, pp.694–701, Ohmsha (1992).

[13] Joachims, T.: Optimizing Search Engines using Clickthrough Data, *Proceedings of the ACM Conference on Knowledge Discovery and Data Mining* (2002).

[14] Kudo, T. and Matsumoto, Y.: A boosting algorithm for classification of semi-structured text, *Proceedings of EMNLP*, pp.301–308 (2004), http://chasen.org/~taku/software/bact/.

[15] 工藤拓, 松本裕治: カーネル法を用いた言語解析における高速化手法, 情報処理学会論文誌, **45** No. 9, pp.2177–2185 (2004), http://chasen.org/~taku/software/cabocha/.

[16] 水野淳太, 秋葉友良: 任意の回答を対象とする質問応答のための実世界質問の分析と回答タイプ判定法の検討, 言語処理学会年次大会, pp.1002–1005 (2007).

[17] Pasca, M., Lin, D., Bigham, J., Lifchits, A., and Jain, A.: Names and Similarities on the Web: Fact Extraction in the Fast Lane, *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pp.809–816 (2006).

[18] Ravichandran, D. and Hovy, E.: Learning Surface Text Patterns for a Question Answering System, *Proceedings of ACL-2002*, pp.41–47 (2002).

[19] Shalev-Shwartz, S., Singer, Y., and Srebro, N.: Pegasos: Primal Estimated sub-GrAdient SOlver for SVM, *Proceedings of the International Conference on Machine Learning*, pp.807–814 (2007), http://ttic.uchicago.edu/~shai/code/.

[20] Shoham, Y.: *Reasoning about Change*, MIT Press (1988).

[21] Soricut, R. and Brill, E.: Automatic Question Answering: Beyond the Factoid, *Proceedings of the Human Language Technology and North American Association for Computational Linguistics Conference*, pp.149–156 (2003).

[22] Verberne, S., Boves, L., Oostdijk, N., and Coppen, P.-A.: Evaluating Discourse-based Answer Extraction for Why-Question Answering, *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp.735–736 (2007).