

係り受け解析に基づくグラフ構造を用いた質問応答システムの構築とその評価

Question answering system with graph structure based on dependency analysis and its evaluation

倉田 岳人 岡崎 直観 石塚 満
Gakuto KURATA Naoaki OKAZAKI Mitsuru ISHIZUKA

東京大学大学院 情報理工学系研究科

Graduate Schhol of Information Science and Technology, University of Tokyo

Question Answering(QA) is the hot research topic. In the task of QA, queries are written in natural language. Then, the system returns the correct answer from newspaper articles. Ranking answers is very difficult in QA task, and there have been no sophisticated algorithm yet. Our method using Graph Structure from Dependency Analysis is superior to former approaches. In the end, the result on NTCIR-4 QAC2 is shown.

1. はじめに

近年、計算機性能の向上や様々な電子化された文書の整備により、自然言語処理に関する研究が盛んに行なわれている。質問応答とは、自然言語で与えられた質問文に対して大量文書中から適切な解答を導き出す技術であり、盛んに研究されている。

本報告では、まず2において、従来手法のついでに、それらの問題点を指摘する。次に3において、今回提案する手法について説明する。その後、4において、構築した質問応答システムを用いて行なった評価実験の結果についてまとめる。最後に6で、本報告をまとめ、今後の課題を述べる。

2. 日本語質問応答に関する従来手法

2.1 質問応答の流れ

日本語質問応答を実現するための一般的な流れを図1に示す。

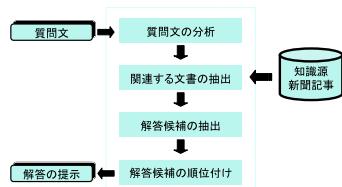


図1: 質問応答システムの一般的な流れ

従来のシステムでは、図のような4段階に基づき質問応答は実現されていた。この4段階の処理は妥当であると考えられるので、本報告で構築するシステムにおいても踏襲することとする。しかし、各々の処理においてはいくつかの問題点がある。これらを以下に指摘する。

2.2 従来手法の問題点

質問文の過分類 解答の分類を多数にした場合、それに対応した固有表現抽出器が必要となる。しかし、現状で分類数を非常に多くして、それに対応できる固有表現抽出器の実現は困難である。

namazu の利用 namazu は全文検索システムとしては非常に優れたシステムである。しかし、namazu を用いた場合、処理の多くの部分がブラックボックス化してしまう。また、検索語の選択に関しても柔軟な処理を行うことができる、とは言いがたい。

また、高木らは一般的な検索と、質問応答システムにおける関連文書の検索では、効果的な検索語の設定が異なる、ということを主張している [1]。

このような観点から、質問応答システムのための検索、というタスクに適した検索エンジンを構築し、検索語をより柔軟に扱うことができるようにするべきである、ということが出来る。

単純な単語間距離の利用 「質問文に含まれる検索語と、質問に対する解答は近い位置に現れる」という前提は、非常に有効である。しかし、実際にその前提に従った処理を行って、高精度の結果は得られていない。これは、単語と単語が何文字分離しているという尺度や、何バイト離れているという尺度の様な単純な単語間距離を用いていることが原因となっていると考えられる。日本語の場合、例えば主語と述語の様な関係の強い文節間に他の文節が挿入される、ということが多発する。このような観点から、単純な単語間距離を用いて、順位付けを行うことには問題がある、ということが言える。

3. 提案手法

3.1 検索エンジンの構築

汎用連想計算エンジン GETA[2] を用いた検索システム(以下、GBSE:Geta Based Saerch Engine と呼称する。)を構築した。

GBSE を用いた検索では、最初に新聞記事に対して、形態素レベルでの索引付けを行なう。

次に、索引付けされた知識源から、質問文と関連する文書を抽出する方法を述べる。GBSE を用いた検索の特徴を以下に簡単にまとめる。

- 検索語を用いて、各抽出単位に対し TF・IDF に基づくスコアを与え、そのスコアに従って上位から順に出力する。
- 検索語に対して、優先度を与えることができる。優先度は「高」「低」の二値で表現され、優先度が「高」の形態素は、出力される抽出単位中に必ず含まれていないといけない。それに対して、優先度が「低」の形態素は、必ずしも含まれる必要はない。ただし、含まれている方がその抽出単位に対するスコアは大きくなる。

GBSE を用いると、検索語となる形態素に優先度を与えることができ、より柔軟な検索を行うことができる。GBSE を用いた検索を行う場合、質問文の形態素解析結果から得られる形態素を、検索時の優先度の設定の尺度として用いるために、必須検索語と任意検索語に分類し、優先度の設定の尺度とする。

実際の検索を行なう際には、最初はすべての検索語の優先度を「高」とする。この状態で検索に失敗した場合は、任意検索語の優先度を TF の多い順に「低」に変更しながら、検索を繰り返すこととする。

GBSE を用いた検索には、以下の様な利点がある。

連絡先: 倉田 岳人, 現在 日本 IBM 株式会社勤務, kurata@miv.t.u-tokyo.ac.jp

- 索引付けに用いる形態素、日付表現などの柔軟な設定ができる
- 索引付け、質問文の解析ともに統一した形態素解析器の結果に基づく形態素レベルで行う(日付表現を除く)ことにより、複合語等の問題を考慮する必要がなく、一貫性の取れた高精度の検索が実現できる。
- 検索語に優先度という尺度を導入することにより、質問応答システムに適した検索をすることができる。
- GBSE によるでの索引付けは *namazu* を用いる索引付けよりも遥かに高速であり、システムのチューニングが容易である。例えば、新聞記事一年分のデータをパラグラフごとに切り分け、“茶釜”で形態素解析し、索引付けをするために要する時間は、CPU:Pentium4 2.8GHz、メモリ:1GBのマシンを用いて、およそ1日程度であるが、GBSEでの索引付けの場合、同等の処理を行うのに要する時間は1時間程度である。

3.2 推定される解答の形に基づく質問文の分類

本報告では質問文を以下の4種類に分類することとした。

- TYPE 1** 「何銀行ですか」「何メートル」ですかのような形の質問で、解答の接尾語もしくは単位がわかるもの
- TYPE 2** 「誰ですか」「どこですか」のような形の質問で、解答として「人名」「地名」「組織名」「時刻」のような固有表現を求めているもの。
- TYPE 3** 「どのくらいですか」「いくらですか」のような形の質問文で、解答として数値表現を求めているもの。
- TYPE 4** 上の3種類に分類されないもの。解答の形に対する情報が少ない質問であり、解答を提示することが困難である。

3.3 グラフ構造に基づく解答候補の順位付け

抽出された解答候補に対して、どの解答候補が最も解答らしいかという点に関して順位付けを行う。提案手法では以下の様にして順位付けを行った。

1. 検索された文に対して係り受け解析を行う。今回は係り受け解析に *CaboCha* を用いた。
2. 複数の文から得られた文節間の係り受け関係に従い、各文節をノードとするグラフ構造を作成する。
3. グラフ内で、質問文から抜き出された検索語を含むノードに関しては、検索語とその他に分割する。具体的には、キーワードに「発明」があり、グラフ中に「発明品」というノードがあれば、「発明 → 品」という形にする。
4. 係り受け関係から作成されたグラフは有向グラフであるが、これらをすべて無向グラフにする。
5. ノード間のリンク数に従って、隣接するノード間のコストを定める。ここで隣接するノード A, B 間のコスト $Cost(A, B)$ は式1に従って定めた。

$$Cost(A, B) = 1/(N_{link(A,B)})^2 \quad (1)$$

ただし、 $N_{link(A,B)}$ はノード A, B 間のリンク数とする。

6. Dijkstra のアルゴリズムに従い、解答候補と検索語の最短距離を算出する。そして、ある解答候補とすべての検索語との距離の和を、その解答候補のスコアとし、そのスコアに従って順位付けを行った。

$$Score(Candidate) =$$

$$\sum_{All\ keywords} Distance(Candidate, Keyword) \quad (2)$$

ここで $Candidate$ は特定の解答候補、 $Keyword$ は検索語を表し、ノード X, Y の最短距離 $Distance(X, Y)$ はダイクストラのアルゴリズムにより、式3の様定められる。

$$Distance(X, Y) = \min \sum Cost \quad (3)$$

4. 評価実験とその結果

4.1 実験条件

昨年の12月に行なわれた NTCIR-4 QAC2 のデータに基づく実験を行なった。表1に実験の条件を示した。

表1: 評価実験の条件

知識源	毎日新聞 98年, 99年 読売新聞 98年, 99年
質問数	200問

4.2 評価方法

表1に示した様に、今回は Task 1 の条件に従って評価を行った。ここで、Task 1 の評価方法について簡単に述べる。

Task 1 では、システムは一つの質問に対して、順位を付けて5個の解答を返す。ここで、正解を返した最も上位の順位の逆数 RR をその設問の得点とする。そしてその平均値 MRR をシステムの評価とする。

$$MRR = \frac{\sum_{i=1}^n RR_i}{n}, \quad RR_i = \frac{1}{Rank}$$

4.3 実験結果

表2に、3.2での分類タイプごとの MRR を示した。

表2: 分類タイプごとの結果

	Type 1	Type 2	Type 3	Type 4	Total
MRR	0.49	0.50	0.56	0.28	0.425

Type 4 に質問が分類された場合の MRR が非常に低くなった。これは、Type 4 の場合、解答の形に対する情報が全く得られないため、抽出される解答候補の数が非常に大きくなる。また、解答候補の中に無意味な名詞の接続などが多数含まれるようになる。この結果、 MRR が低下したと考えられる。

また、全体の MRR に関しても、著しく優れているとはいえない。これは、検索に失敗した場合に解答を提示しない、という方針をとったためである。

5. まとめと今後の課題

本報告では、我々が構築した質問応答システムの概要について述べた。グラフ構造に基づく順位付けアルゴリズムの詳細に関しては別稿を参照されたい[3]。また、構築したシステムに対して、NTCIR-4 QAC2 のデータに基づき評価を行なった。

Type 4 に分類される質問の数が減れば、全体の MRR の向上が期待される。よって、質問文から解答の形に関する情報をより多く抽出できる枠組みを作ることが今後の課題といえる。また、検索に失敗した場合の処理を検索エンジンに導入することも重要な課題である。

参考文献

- [1] Toru TAKAKI, Yoshio ERIGUCHI. "NTT DATA Question-Answering Experiment at the NTCIR-3 QAC". *Proceedings of the Third NTCIR Workshop*, 2003.
- [2] 情報処理技術振興事業協会 (IPA), <http://geta.ex.nii.ac.jp/>. 汎用連想計算エンジン GETA.
- [3] 倉田岳人, 岡崎直観, 石塚満. 係り受け関係に基づくグラフ構造を用いた質問応答システム. 電子情報通信学会技術研究報告, Vol. 103, No. 408, pp. 1-6, 11 2003.