

類似語の例示を用いた多義的クエリの曖昧性解消手法

橋本 泰平[†] 田島 敬史^{††}

[†] 京都大学工学部情報学科 〒606-8501 京都府京都市左京区吉田本町

^{††} 京都大学大学院情報学研究科 〒606-8501 京都府京都市左京区吉田本町

E-mail: †hashimoto@dl.kuis.kyoto-u.ac.jp, ††tajima@i.kyoto-u.ac.jp

あらまし 多くの検索エンジンでは、ユーザの検索意図は複数の検索語または検索フレーズからなるクエリで表現されるが、そのようなクエリの中には複数の解釈が可能な多義的クエリが多く存在する。多義的クエリの意味を絞り込むには、検索語を追加する、語をフレーズにする、などの方法が用いられる。しかし、ユーザの望む検索意図に適切に絞り込めるような語やフレーズを見つけるのは必ずしも容易ではない。そこで本研究では、多義性クエリの意味を的確に絞り込めるクエリの候補を提示する手法を提案する。提案手法では、多義性を解消するための情報として、ユーザがクエリ中の検索語の類似語を指定する。ユーザの検索意図に適合する Web ページにおいてその検索語の周辺に出現する語は、同じ文脈で類似語が現れる Web ページにおいて類似語の周辺にも頻繁に出現すると期待できる。本研究では、この仮定に基づき、類似語の周辺に出現する語と元のクエリを組み合わせることで、ユーザの検索意図に合うクエリの候補を生成する。

キーワード Web 検索, クエリ推薦, クエリ拡張, クエリ意図, 検索意図, 多義語
もれてしまう。

1. はじめに

現在、様々な情報がインターネット上に存在し、多くの人々がそれらの情報を日々検索・閲覧している。インターネット上の情報を検索したいユーザーは、Google や Yahoo, Bing などの検索エンジンからユーザーの検索意図に関係するキーワードを入力することで必要な情報を得るのが一般的である。

一つ概念に関するページだけでもその数は膨大であり、ユーザーはその中から自分の必要とする情報を得るために適切なクエリを入力しなければならない。しかしユーザーが自分の検索意図をクエリに完璧に反映することは難しく、多くの場合ユーザーのクエリは様々な意図を持つ曖昧なものである。このように、検索意図が十分に反映されていないために複数の解釈が可能なクエリを多義的クエリと呼ぶ。

多義的クエリの解釈にはメジャーな解釈とマイナーな解釈があるが、一般的にメジャーな解釈に適合するウェブページの方がマイナーな解釈に適合するウェブページよりも圧倒的に多いため、ユーザーの検索意図がマイナーな解釈の方である場合、その解釈に適合するウェブページの発見は難しい。さらにこれらの解釈の分野が似ている場合、その違いをクエリで表現することが難しく、マイナーな解釈に適合するウェブページの発見はより困難なものになる。

例えばユーザーが「トマトを肥料として利用する方法について知りたい」という検索意図を持つ場合、クエリとして考えられるのは「トマト 肥料」などであるが、一般的にトマトと肥料に関連する話題では「トマトに肥料を与える」話が多く、検索をしてもこちらの話題に関するページばかりが検索結果として表示される。このようにクエリに関連する複数の話題がある場合、そのクエリの検索結果にはメジャーな話題のウェブページばかりが現れ、マイナーな話題のウェブページは其中に埋

一般的に、多義的クエリにおいて求める検索結果が得られない場合、ユーザーはキーワードを追加したりフレーズを用いたりしてクエリを改良しようとする。しかし上記のような多義的クエリの場合、その中に含まれる話題の分野が似ているために容易にそれらの違いを表現することができず、ユーザーの検索意図がマイナーであると、ユーザーが自力で、求めるウェブページを発見するクエリを生成することは困難である。また、ユーザーが検索したい情報についての知識を十分に持たない場合、適当なクエリ候補を発見することはさらに難しくなる。

このような場合にユーザーの検索をクエリ生成の面から支援する技術が研究されてきた。既存の手法としては、多くのユーザーが検索しているクエリをユーザーにも提案するクエリ推薦や、ユーザーの入力したクエリがよくあるスペルミスにあてはまる場合やよく検索されているクエリに似ている場合に、そのクエリを提案するクエリ修正の技術などが挙げられる。これらの技術は、多くのユーザーのクエリログを解析し、頻繁に検索されるクエリをユーザーに提示することで、ユーザーのクエリ作成を支援する。これらは、多くのユーザーが検索するクエリは他のユーザーが検索したいことに合致している可能性が高いという考えに基づくものであるが、ユーザーの検索意図は多様であり、必ずしも多くの人々が検索するようなことを調べたいとは限らない。先ほど述べたように多義的クエリの持つ解釈にはメジャーな解釈とマイナーな解釈が存在するが、クエリログを用いた既存の手法はメジャーな解釈に適合するクエリの推薦を行うのみで、マイナーな解釈のクエリを推薦することはできない。そこで本研究では、多義的クエリの曖昧性を解消し、メジャーな情報に埋もれてしまうマイナーな情報を発見できるクエリことを目的とする。

本研究では、クエリの意図を区別するための情報源として、

そのクエリ中の検索語の類似語を用いる。ユーザーが発見したいウェブページの中の検索語の周辺で用いられる語は、同じ文脈で用いられる類似語の周辺にも用いられると仮定し、ユーザーが入力した類似語の周辺の語と元のクエリを組み合わせることで、元のクエリの曖昧性を解消し、検索意図を明確に反映したクエリを生成する。

例えば、先ほど挙げた例のようにユーザーの検索意図が「トマトを肥料として利用する方法について知りたい」という場合、トマトは肥料として用いるものなので、トマトの類似語として「石灰」「リン」「糞尿」など、通常肥料として用いるものが考えられる。これら類似語のページには、例えば「石灰を肥料として使う」「糞尿が肥料になる」などの表現が含まれると考えられ、その周辺フレーズとして「...を肥料として使う」「...が肥料になる」などの表現を抽出する。このような類似語の周辺フレーズは、肥料として用いられるものに対して使用される語であると期待できるため、この周辺フレーズと元のクエリである「トマト 肥料」を組み合わせることによって、「トマトを肥料として使う」や「トマトが肥料になる」などのように、曖昧性を解消したクエリを生成できる。本研究では、このような曖昧性を解消したクエリ候補を複数提示するクエリ推薦を行うシステムの開発を行う。

本研究で提案するシステムの評価は二段階に分けて行う。まず一段階目では、システムへの入力方法と推薦クエリのランキング手法を複数用意し、実験データの多義的クエリを用いて実験を行い、用意した入力方法およびランキング手法の中で最も良い精度のものを採用する。入力方法としては、一方は類似語のみを入力し、もう一方では類似語に加え非類似語を入力する。類似語がユーザーの検索意図に合致する文脈で使用される語であるのに対し、非類似語はユーザーの検索意図と別の文脈、すなわちユーザーの多義的クエリの解釈の中でユーザーの検索意図に合致しない解釈に類似した語である。この非類似語を入力することで、ユーザーのクエリは、類似語の文脈に近く、かつ非類似語の文脈からは遠い文脈のクエリを生成することができるようになり、生成クエリの精度が向上すると期待できる。例えば「トマト 肥料」という多義的クエリでは、「トマトに肥料を与える話」という解釈が存在するが、これはユーザーの検索意図に合致しない。この文脈での類似語は「みかん」「チューリップ」など肥料を与える対象であり、これらを非類似語としてシステムに入力する。さらに提案手法では、推薦クエリを生成した後、クエリのランキングを行う。まず推薦クエリを構成する周辺フレーズとユーザーが提示した類似語・非類似語を組み合わせたクエリを生成し、そのクエリの検索結果数を得る。類似語から生成したクエリ集合の検索結果数と、非類似語から生成したクエリ集合の検索結果数それぞれの平均、 \log の平均、相乗平均を計算し、類似語と非類似語の比をその推薦クエリのスコアとする。この三つの平均の計算手法の中で最も精度の高いものを提案手法として採用する。

二段階目では、一段階目で採用した手法を用いてユーザー実験を行う。ユーザー実験では、一段階目の実験で用いたものと同じ多義的クエリを使用し、ユーザーが自力でクエリの曖昧性

を解消するクエリを考案する方法と、提案手法が推薦したクエリを用いて検索を行う方法を比較する。ユーザーの検索意図に適合するページを発見することのできるクエリの生成数およびそれらのクエリで発見できる正解ページ数によって提案手法の評価を行う。

本稿の構成は下記の通りである。まず2章で本研究と関連のあるクエリ推薦やクエリ拡張、およびクエリの曖昧性解消についての研究を紹介し、本研究の位置付けを行う。続いて3章では類似語の説明および類似語の周辺フレーズと検索意図について具体的に説明し、4章では提案手法について詳しく述べる。

2. 関連研究

本節では本研究に関連する研究について述べる。クエリ推薦やクエリ拡張に関する研究は古くから行われており、中でもクエリログやクリックログなどを用いた推薦手法が数多く提案されている[1][2][3]。これらは、ユーザーの入力したクエリやそのクエリの検索結果でどのURLをクリックしたかなどのユーザーの行動履歴から、ユーザーの検索意図に適合するクエリを推薦する手法である。

Mei[3]らや今井[4]らは、クリックログから得られる一般ユーザーの入力クエリとそのクエリでのURLの閲覧履歴をもとに、クエリとURLの二部グラフを構成し、クラスタリングを行うことでユーザーの曖昧な入力クエリに関連する多様な分野のクエリ推薦を提案している。このようなユーザーの行動履歴を用いた推薦手法は現在主流な検索エンジンでも用いられている。しかしこの手法ではあくまでユーザーの大規模な行動履歴からユーザーのクエリに関連するクエリの推薦を行うだけであり、ユーザーの検索意図が明確に表現された曖昧性のないクエリを生成し、推薦するわけではない。またユーザーの検索意図が、他ユーザーの検索意図から大きく外れたマイナーなものになると、適切なクエリ候補の推薦は難しい。しかし本研究は、類似語によってユーザーの検索意図を明示的に指定することができる。そのため、クエリログやクリックログなど他のユーザーの行動履歴を用いる方法よりもよりユーザーの検索意図に合致したクエリ候補の推薦が可能であると考えられる。

また、検索結果をクラスタリングすることによって、多義的クエリから得られる情報を分類する研究もある[6][7][8]。これらは、理想的なクラスタでは、曖昧なクエリに含まれる複数の意味が、それぞれ別のクラスタに分類されるという仮定のもとで、分類した検索結果からユーザーの検索意図に適合するページを発見する方法である。しかし本研究で扱うようなマイナーな検索意図に適合するウェブページは、メジャーな検索意図に適合するウェブページに比べ非常に数が少ないため、うまく一つのクラスタにならないと考えられる。また、例えば与えられたクエリの解の上位1000件をクラスタリングする場合、そもそも上位1000件中にマイナーな検索意図に適合するウェブページが全く含まれていない場合も多い。

本研究同様、クエリ自体の曖昧性の解消を目的とした研究も存在する。Allan[5]らは、大規模コーパスを用いてユーザーの検索語の周辺に現れる品詞のパターンを解析することで、クエ

りの曖昧性を解消する手法を提案している。彼らは検索語の周辺の頻出パターンからクエリを生成し、そこにユーザーの元のクエリを当てはめることでクエリの曖昧性の解消を試みている。

また若木 [9] らは、曖昧なクエリの検索結果の文書集合中の単語の共起関係に注目し、特定の単語群との共起度が高い単語がそのクエリに関連するトピックを明確に分離する語であると考へた。検索結果に含まれる多様なトピックを網羅するようにそのようなトピックの代表的な単語を発見し、ユーザーに提示することでユーザーのクエリ拡張の支援を行う手法を提案している。

これらはクエリの曖昧性を扱うという点では本研究と類似しているが、クエリの類似語に関連する文書集合を利用するという点で本研究はこれらとは異なる。また、コーパスや曖昧なクエリの検索結果の文書集合中には様々な話題の文書が存在するが、その中にユーザーの検索意図に適合する文書が含まれるとは限らない。したがって既存研究の手法では、ユーザーは提示された候補の中から自分の検索意図に適合しそうな候補を選択する必要があり、また最悪の場合、適合するような候補が提示されないということも考えられる。一方提案手法では、ユーザーの検索意図を明示的に指定し、クエリに反映するため、より精度の高いクエリ候補の推薦が可能であると期待できる。

また Anand [10] らは、Term Graph の中でクラスタリングを行うことによって、クエリとより密接に関係のある語のクラスタを作成し、より精度の高いクエリ拡張モデルを提案している。Term Graph とは、語と語の関係をその意味の繋がり強さで重み付けしたグラフである。この研究は、検索意図を絞り込むために多くの検索語が必要となる場合に、どの検索語が重要であるかがわかりにくくなるという問題を扱うもので、本研究が扱うようなメジャーな話題に埋もれてしまうマイナーな話題を発見する困難さを扱うものとは異なる。

3. 問題設定

本章では本研究で扱う多義的クエリの曖昧性、クエリの類似語とその周辺フレーズについて説明し、さらに本研究での問題設定について述べる。

3.1 多義的クエリの曖昧性

1章でも述べたように、ユーザーの検索意図がクエリに十分反映できていない場合、そのクエリは複数の意味を持つ多義的クエリと呼ばれる。多義的クエリには複数の分野の意図が含まれるため、その分野で頻出する単語などをキーワードとして追加することでクエリの意図を絞り込める場合も多い。しかし中には、異なる意図ではあるが分野が類似しているために、そのような単語があまり存在しない場合や、ユーザーがそれらを思いつくのが困難な場合もある。例えば一章で挙げた「トマト肥料」というクエリに含まれる「トマトを肥料にする方法」「トマトに与える肥料」という2つの意図は、トマトが肥料になるのかトマトに肥料を与えるかというトマトと肥料の関係性が異なるだけなので、これらを分離するようなキーワードをユーザーが発見するのは困難である。本研究で扱う多義的クエリとは、このようにユーザーが容易に解消できないような曖昧性を

持つクエリである。また、一般的にユーザーが生成するクエリは短いものが多いため [11]、今回本研究では二つの検索語から成るクエリのみを扱う。

3.2 クエリの類似語と周辺フレーズ

1章でも述べたように、本研究ではユーザーの検索意図をクエリに反映するために、クエリの中の検索語の類似語を入力し、類似語が含まれる文書の中で、類似語の周辺に出現するフレーズを用いる。ただし本研究では、ユーザーが入力したクエリに含まれる検索語のうち、片方の検索語の類似語のみを入力する。本論文で述べる類似語は、ユーザーのクエリが「検索語1 検索語2」から構成される場合、検索語1の類似語であるとする。

ここで類似語とは、検索意図に適合する文脈で使用されるクエリと同じ使い方をされる語のことである。例えば「トマト肥料」というクエリにおけるトマトの類似語として石灰、糞尿が考えられるため、これらを使って「石灰 肥料」「糞尿 肥料」というクエリが考えられる。ユーザーの元のクエリでは、「トマトに与える肥料」「トマトを肥料にする方法」という意図が考えられ、ユーザーの検索意図は後者である。ここで類似語として挙げた石灰や糞尿は、このクエリからは「肥料として用いるもの」という使い方以外は考え難い。このように元のクエリが含む意図のうち、ユーザーの検索意図に適合する場合と同じ文脈で用いられる語を類似語とする。

本手法ではまず、ユーザーが入力した各類似語と、ユーザーのクエリのうち類似語を考慮しない方の検索語2から「類似語 検索語2」というクエリを生成する。このようにして生成したクエリ集合を $A = \{a_1, a_2, \dots, a_n\}$ とすると、 A の各クエリから得られたウェブページ集合から、類似語の周辺フレーズ $P = \{p_1, p_2, \dots, p_m\}$ を抽出する。この周辺フレーズはウェブページ中の類似語の前後二文節に含まれるフレーズから抽出されるものであり、類似語に関する文書集合の中で頻繁に出現する周辺フレーズはその文脈に特有のものであると期待できる。これより、周辺フレーズ P とユーザーの元のクエリを組み合わせることで、文脈を限定した曖昧性のない推薦クエリ候補 $R = \{r_1, r_2, \dots, r_{20}\}$ を生成する。周辺フレーズの詳しい抽出方法は4章で述べる。

3.3 問題設定

本研究の目的は曖昧な多義的クエリからユーザーの検索意図に適合した文書集合を発見することであり、ユーザーから類似語集合が与えられた時に、各類似語に関する文書集合から周辺フレーズ集合 P を抽出し、 P を用いて多義的クエリの検索意図を絞り込んだクエリ集合 R を生成・推薦することが本研究で扱う課題である。

4. 提案手法

本章では本研究で提案するクエリ推薦手法について、類似語に関する文書集合からの周辺フレーズの抽出、推薦クエリの生成、推薦クエリのランキング手法の三つの段階に分けて説明する。

4.1 周辺フレーズの抽出

提案手法ではまず、ユーザーが入力した類似語を含む文書集

合から、その周辺フレーズを抽出する。本研究では、類似語を用いて生成したクエリ集合 A の各クエリを Bing に入力し、その上位 1000 件のウェブページのスニペットを各類似語を含む文書集合として用いる。本研究で扱う多義的クエリは二つの検索語から成るクエリであり、類似語を用いて生成するクエリ集合 A の各クエリも同様に二つの検索語から成るクエリである。類似語から生成したクエリ a_i が検索語 k_1, k_2 から成る場合、まず a_i に関する文書集合の中で、 k_1 と k_2 を共に含む一文を抽出する。ここでこの一文を文節に区切ると、この文は次のように表すことができる。

$$c_1 / \cdots / c_{k_1-2} / c_{k_1-1} / c_{k_1} / c_{k_1+1} / c_{k_1+2} / \cdots \\ \cdots / c_{k_2-2} / c_{k_2-1} / c_{k_2} / c_{k_2+1} / c_{k_2+2} / \cdots / c_n$$

あるいは

$$c_1 / \cdots / c_{k_2-2} / c_{k_2-1} / c_{k_2} / c_{k_2+1} / c_{k_2+2} / \cdots \\ \cdots / c_{k_1-2} / c_{k_1-1} / c_{k_1} / c_{k_1+1} / c_{k_1+2} / \cdots / c_n$$

ここで c_i は各文節で、特に c_{k_1}, c_{k_2} はそれぞれ k_1, k_2 を含む文節である。次に周辺フレーズはクエリを構成する検索語を含む文節の前後二文節から構成されるので、周辺フレーズの構成要素として、上式の中の c_{k_1} の前後二文節と c_{k_2} の前後二文節及び c_{k_1}, c_{k_2} を抜き出し、周辺フレーズの構成要素候補の集合とする。ただし実際の文書では、類似語を含む一文の中で、検索語が文章の冒頭付近や末尾付近に現れるために前後に二文節未満しか存在しない場合や、二つの検索語間の距離が近いために検索語間に存在する文節が重複する場合もある。したがって様々な場合を簡単に表現するため、周辺フレーズの構成要素集合 C は以下のように表す。

$$C = \{Comp_f, c_{k_1}, Comp_m, c_{k_2}, Comp_b\}$$

$Comp_f, Comp_m, Comp_b$ はそれぞれ、 k_1 の前にある文節、 k_1 と k_2 の間にある文節、 k_2 の後ろにある文節、をまとめたものである。

次に、先ほど抽出した類似語の周辺の文節から周辺フレーズを構成する。本研究ではまず、集合 C の文節のすべての組み合わせを生成し、それらの文節を元の文章の順番と同じように接続することで様々なフレーズを生成する。ただし、どの組み合わせも検索語を含む文節である c_{k_1} と c_{k_2} を含むように構成する。また、接続の際に、各文節を形態素解析し、周辺フレーズに用いる品詞の選択を行う。これは後ほど説明する。したがって生成されるクエリを構成する文節の組み合わせは表 1 のようになる。このようにして、ユーザーが入力した類似語それぞれについて、その検索結果のスニペットから類似語を含む文を探し出し、それぞれの文からすべての文節の組み合わせの周辺フレーズを生成する。このようにして、各類似語について、類似語を含む一文を構成する文節のすべての組み合わせから周辺フレーズを生成する。

4.2 推薦クエリの生成

ここからは、生成された周辺フレーズとユーザーの元のクエリを用いてユーザーに推薦するクエリを生成する方法を説明す

表 1 周辺フレーズを構成する文節の全組み合わせ
文節の組み合わせ

c_{k_1}, c_{k_2}
$Comp_f, c_{k_1}, c_{k_2}$
$c_{k_1}, Comp_m, c_{k_2}$
$c_{k_1}, c_{k_2}, Comp_b$
$Comp_f, c_{k_1}, Comp_m, c_{k_2}$
$Comp_f, c_{k_1}, c_{k_2}, Comp_b$
$c_{k_1}, Comp_m, c_{k_2}, Comp_b$
$Comp_f, c_{k_1}, Comp_m, c_{k_2}, Comp_b$

る。4.1 節で紹介した方法で生成される周辺フレーズの候補数は非常に多いため、それらすべてをユーザーに推薦することはできない。したがってまずは、生成した周辺フレーズをスコア付けし、スコアの高いもののみを用いて推薦クエリを生成する。

スコア付けのために、各類似語に関する文書集合から、どの周辺フレーズが何個生成されたかの情報を用いる。ユーザーが入力した類似語集合において、そのそれぞれの類似語に関する文書集合から、周辺フレーズ f_i がそれぞれ $H_i = \{h_1, h_2, \dots, h_n\}$ 個生成された場合、 f_i のスコア S は次のような相乗平均の式で計算される。

$$S = \sqrt[n]{h_1 \times h_2 \times \cdots \times h_n}$$

ただし、 H の要素のうち 1 つでも 0 が存在すると S は 0 となってしまうため、本研究では H 集合中の 0 は 0.1 に置き換える。通常の算術平均では、周辺フレーズがある特定の類似語の文書集合からのみ生成個数が多いような場合でもスコアが高くなるが、相乗平均を用いることで、全ての類似語の文書集合から均等に生成される周辺フレーズのスコアが高くなる。こうすることで、ある類似語特有の表現のスコアが高くなるのを防ぎ、すべての類似語で用いられる、すなわちユーザーの検索意図に適合する文脈で広く一般に用いられるような周辺フレーズのスコアを高くすることができる。本研究では、このようにスコア付けを行った後、スコアの高い上位 50 件の周辺フレーズのみを用いて推薦クエリの生成を行う。

類似語を用いて生成したクエリを構成する周辺フレーズは、類似語を含む文節を必ず含むように生成しているため、類似語を構成する検索語を含んでいる。したがって推薦クエリの生成は単純に、周辺フレーズの中の類似語の部分、ユーザーの元のクエリを構成する検索語に置き換えるだけである。例えば、ユーザーの元のクエリが「トマト 肥料」である場合、「糞尿は肥料になる」という周辺フレーズが得られたとすると、このフレーズ中の類似語は「糞尿」「肥料」であるので、これらを「トマト」「肥料」で置き換えることで「トマトは肥料になる」というクエリが生成される。

4.3 推薦クエリのランキング

4.2 節で周辺フレーズから 50 件の推薦クエリを生成したが、50 件ものクエリからユーザーがクエリを選別するのは非効率的である。したがって推薦クエリ候補から、ユーザーの検索意図をより反映したクエリを選別する。

提案手法では、生成した推薦クエリ候補の中からユーザーに

推薦すべきものを選別するため、クエリのランキングを行う。本節ではランキングのためのクエリのスコア付けの方法について説明する。また 1 章で述べたように、本研究ではユーザーが入力する情報を二通りと、ランキングの計算手法を三通り実験し、それらの組み合わせのうち最も精度の良い手法を採用しているが、ここでは採用された手法について説明する。この実験の詳細については 5.1 節で述べる。

採用した手法では、ユーザーは類似語だけでなく非類似語を入力する。1 章でも説明したが、非類似語はユーザーの多義的クエリが含む検索意図の中で、ユーザーの検索意図に適合しない検索意図の文脈における検索語と類似する語である。まず生成したクエリと類似語及び非類似語を組み合わせたクエリを生成する。先ほどと同じで、生成クエリ中の検索語の部分を類似語 $A = \{a_1, a_2, \dots, a_n\}$ 、非類似語 $U = \{u_1, u_2, \dots, u_m\}$ で置き換えたクエリをすべてのクエリについて生成する。推薦クエリ R_i から検索語を A, U で置き換えて生成されたすべてのクエリを用いて Bing で検索を行い、各クエリの検索結果数の集合 $RA_i = \{ra_1, ra_2, \dots, ra_n\}$ 、 $RU_i = \{ru_1, ru_2, \dots, ru_m\}$ を得た後、それぞれの集合内の相乗平均 RA_{score} 、 RU_{score} を以下のように計算する。

$$RA_{score} = \sqrt[n]{ra_1 \times ra_2 \times \dots \times ra_n}$$

$$RU_{score} = \sqrt[m]{ru_1 \times ru_2 \times \dots \times ru_m}$$

本研究では、推薦クエリを類似語で置き換えたクエリの検索結果数が多いほどそのクエリはユーザーの検索意図に適合し、非類似語で置き換えたクエリの検索結果数が多いほどそのクエリがユーザーの検索意図に不適合であるという考えに基づき、 RA_{score} が大きく、かつ RU_{score} が大きいクエリほどスコアが大きくなるようにランキングを行う。したがって以下の式のように RA_{score} と RU_{score} の比をとった値を各推薦クエリのスコア R_{score} とする。

$$R_{score} = \frac{RA_{score}}{RU_{score}}$$

このようにして、前節で生成した推薦クエリ候補 50 件すべてにスコアを与え、その値に基づいてクエリのランキングを行い、その上位 20 件を推薦クエリとしてユーザーに提示する。表 2 は実際に「トマト 肥料」の例を用いてシステムが生成した推薦クエリである。

5. 評価実験

本章では、本研究で開発したクエリ推薦システムの性能評価を行う。第 1 章で述べたように、システムは二段階に分けて評価を行う。

まず一段階目では、システムへの入力として類似語のみを用いる場合と、類似語と非類似語の両方を用いる場合の 2 通りの手法を用意し、さらに生成したクエリ候補からの推薦クエリのランキングを行う際の各クエリのスコアの計算手法を 3 通り用意した。これらの手法のすべての組み合わせの 6 通りの手法を比較し、最も精度の高いクエリを推薦できる手法を選別した。

また二段階目では、システムを実際にユーザーに使用しても

表 2 推薦クエリ例：トマトを肥料にする方法

番号	クエリ	スコア
1	+”トマトは肥料として”	351.8944889
2	+”トマトを肥料として” +”使う”	310.8267905
3	+”肥料としての” +”トマト”	233.8107929
4	+”トマトを肥料”	220.1526705
5	+”トマト肥料が”	122.0147603
6	+”トマトを肥料と”	86.44025365
7	+”トマトを肥料として”	85.67092173
8	+”出る” +”トマトを肥料に”	80.67143230
9	+”トマトの肥料としての”	61.26925675
10	+”トマトを肥料に” +”した”	38.19821263
11	+”の” +”トマトを肥料に”	36.18067918
12	+”トマトが” +”肥料として”	35.65546229
13	+”トマトを肥料に”	33.35489743
14	+”トマトで肥料”	32.14182153
15	+”トマトを肥料に” +”して”	30.89100379
16	+”トマト肥料の”	28.13053832
17	+”トマトが肥料”	20.65432748
18	+”トマトは肥料と”	20.23505908
19	+”トマトの肥料成分”	17.16748438
20	+”花の” +”肥料にトマト”	16.33703369

らい、システムが生成したクエリを用いて検索を行う場合と、ユーザーが自力で検索を行う場合に、どちらがより多くの適合ページを発見できるかという比較実験を行った。

5.1 第一段階

第一段階では、システムに対してユーザーが類似語のみを入力するか、類似語および非類似語の両方を入力するかの 2 通りと、生成した推薦クエリ候補のランキングのために、各クエリのスコア付けする方法を 3 通り用意し、これらを組み合わせた合計 6 通りの方法を比較した。これらの手法の性能は、表 3 の検索要求、類似語および非類似語を想定した場合に、各方法で推薦されるクエリの精度を比較することで行った。以下ではまず、ユーザーの入力と、クエリのスコアの計算方法についてそれぞれ説明する。

5.1.1 ユーザーの入力

ユーザの入力として以下の 2 通りを用意した。

- 類似語 3 個
- 類似語 3 個、非類似語 2 個

5.1.2 クエリのスコア付け

次に、ユーザーの入力を元にして推薦クエリ候補にスコアを与え、クエリのランキングを行う。まず、抽出した周辺フレーズと類似語および非類似語を組み合わせて新しいクエリを生成する。この新たに生成したクエリを Bing を用いて検索し、その検索結果数を得る。ユーザーが入力する類似語は 3 個、非類似語は 2 個であるので、類似語と周辺フレーズからなるクエリの検索結果数は $A_{num} = \{an_1, an_2, an_3\}$ 、非類似語と周辺フレーズからなるクエリの検索結果数は $U_{num} = \{un_1, un_2\}$ と表せる。実験では、各検索結果数の集合に対して、以下の 3 通りの計算方法を行った。

- 集合の各要素の平均

- 集合の各要素の平均の log
- 集合の各要素の相乗平均

ユーザーの入力が類似語のみの場合、 A_{num} に対して上記の計算を行った結果の値を、その推薦クエリのスコア S とする。ユーザーの入力が類似語および非類似語の場合は、 A_{num}, U_{num} それぞれの集合内で上記の計算を行い、その結果の値の比 $\frac{A_{num}}{U_{num}}$ がその推薦クエリのスコアとなる。

以上のように、ユーザの入力 2 通り、クエリのスコア計算方法 3 通りを組み合わせた全 6 通りの手法を用意した。次に、各手法の性能を比較する方法について述べる。

表 3 実験に使用した問題

問題番号	検索意図	類似語	非類似語
1	りんごを肥料として利用する方法	石灰, 糞尿, リン	みかん, なし
2	じゃがいもを肥料として利用する方法	石灰, 糞尿, リン	みかん, なし
3	トマトを肥料として利用する方法	石灰, 糞尿, リン	みかん, なし
4	Twitter というサービスに関する話題	安納芋, 火花, 青色 LED	ネット, SNS
5	Facebook というサービスに関する話題	安納芋, 火花, 青色 LED	ネット, SNS
6	Instagram というサービスに関する話題	安納芋, 火花, 青色 LED	ネット, SNS
7	サメを餌として食べる生物	ドングリ, ドッグフード, プランクトン	犬, カンガルー
8	ピラニアを餌として食べる生物	ドングリ, ドッグフード, プランクトン	犬, カンガルー
9	イルカを餌として食べる生物	ドングリ, ドッグフード, プランクトン	犬, カンガルー
10	yum のアンインストール方法	skype, iTunes, emacs	apt-get, pip
11	apt-get のアンインストール方法	skype, iTunes, emacs	yum, pip
12	pip のアンインストール方法	skype, iTunes, emacs	apt-get, yum
13	アイスというダンサーについて	マイケルジャクソン, KENZOU, TAKAHIRO	ケーキ, ゼリー
14	世界というダンサーについて	マイケルジャクソン, KENZOU, TAKAHIRO	日本, 国外

5.1.3 各手法の性能比較

まず、表 3 の各問題において各手法で推薦されるクエリを用いて Bing で検索を行い、その検索結果上位 20 件中にユーザーの検索意図に適合するページが存在する場合そのクエリのスコアを 1、適合するページが存在しない場合はそのクエリのスコアは 0 とする。推薦クエリ 20 件をすべてスコア付けし、そのスコアの合計を、その問題における各手法のスコアと考える。すべての問題・手法についてこのスコア付けを行い、各手法ごとにその手法が持つ各問題に対するスコアの相乗平均を計算し、その結果を手法の最終的なスコアにする。

以下の表 4 に各手法および各問題のスコアと、手法毎のスコアをまとめた。手法毎のスコアは、1 つの手法が持つすべての問題に対するスコアについて、それらの平均をとった場合と相乗平均をとった場合の結果の値をそれぞれ算出した。ただし、手法 1~6 については、表 5 の通りである。

表 4 各手法の性能比較

問題番号	手法 1	手法 2	手法 3	手法 4	手法 5	手法 6
1	3	3	3	2	2	1
2	13	13	14	12	11	12
3	5	6	7	0	0	0
4	12	12	12	12	12	12
5	10	10	9	11	11	12
6	13	13	13	8	8	8
7	4	2	4	6	6	6
8	10	10	9	7	7	7
9	6	7	8	4	4	3
10	5	6	8	5	5	5
11	5	5	5	3	3	4
12	7	7	7	7	7	7
13	3	3	3	3	3	3
14	3	3	2	3	3	3
平均	7.071429	7.142857	7.428571	5.935714	5.864286	5.935714
相乗平均	6.166610	6.090289	6.414644	4.117328	4.091818	3.942870

表 5 各手法の入力および計算方法

手法	入力	計算方法
手法 1	類似語, 非類似語	平均
手法 2	類似語, 非類似語	平均の log
手法 3	類似語, 非類似語	相乗平均
手法 4	類似語のみ	平均
手法 5	類似語のみ	平均の log
手法 6	類似語のみ	相乗平均

表 4 から、平均・相乗平均ともに手法 3 が最も高いスコアであることがわかる。したがって、本実験で比較した 6 つの手法の中で、ユーザーが類似語・非類似語の両方を入力し、推薦クエリのスコア付けに相乗平均を用いる手法 3 が最も良い性能を発揮すると期待できる。したがって本研究では手法 3 を採用した。

5.2 第二段階

本節では、第一段階で採用した手法を用いて実際にユーザーにシステムを使用してもらい、システムを用いた検索とユーザーが自力で検索を行う場合を比較し、システムの性能の測定を行う。

5.2.1 実験データ

本実験は、無作為に選んだ 12 人のユーザーに対してシステムを用いた検索、および自力での検索を行った。実験では各ユーザーに 4 つの検索要求を提示し、そのうち 2 題をシステムが生成したクエリを用いて検索し、残りの 2 題をユーザーが自力でクエリを生成して検索を行ってもらった。

実験に使用した検索要求は表 6 の通りである。本実験では、異なる種類の検索意図を扱う問題を 4 種類用意し、それらの種類記号をそれぞれ A, B, C, D とした。さらにそれぞれの種類の中で、同種の検索意図を持つが、検索語の異なる問題を 3 種類用意した。例えば、表 6 を見ると、種類 A は通常肥料になるとは考え難いものを肥料として利用する方法に関するものであり、種類 A の問題の中にはそれらの一例として「トマト」「ジャガイモ」「りんご」という別の検索語を使用している。

また、各ユーザーへ提示する問題の選び方として、次のような規則を定めた。

- 各種類から 1 問ずつ提示し、全種類の問題を行う。
- 一つの問題について、その問題を提案手法で行うユーザーと自力で行うユーザーが各々 2 人となるように選ぶ。

以上の規則を満たすような問題の選び方は表 7 のようになる。ただし、表中の「X/1」のような表記方法は、X, Y がそれぞれ提案手法、比較手法を表し、右側の数値が各種類の中の問題番号を表している。この規則に従うことで、各問題が異なるユーザーに各手法で 2 回ずつ行われ、比較手法の条件を同等に保つことができる。

5.2.2 実験の流れ

ここでは、具体的な実験方法について述べる。実験の手順は次のようになっている。

- 検索要求 4 題のうち、システムを用いて検索する 2 題をユーザーに提示し、ユーザ自身がシステムに類似語及び非類似

表 6 実験に使用した検索要求

番号	検索要求
A-1	トマトを肥料として利用する方法が知りたい
A-2	ジャガイモを肥料として利用する方法が知りたい
A-3	りんごを肥料として利用する方法が知りたい
B-1	ツイッター(自己)に関する様々な話題を調べたい
B-2	Facebook(自己)に関する様々な話題を調べたい
B-3	Instagram(自己)に関する様々な話題を調べたい
C-1	サメを餌として食べる生物について調べたい
C-2	ピラニアを餌として食べる生物について調べたい
C-3	イルカを餌として食べる生物について調べたい
D-1	yum のアンインストール方法が知りたい
D-2	apt-get のアンインストール方法が知りたい
D-3	pip のアンインストール方法が知りたい

表 7 各ユーザーに提示した問題の組み合わせ

ユーザー番号	問題 A	問題 B	問題 C	問題 D
1	X/1	X/1	Y/1	Y/1
2	X/2	X/2	Y/2	Y/2
3	X/3	Y/3	X/3	Y/3
4	X/1	Y/1	X/1	Y/1
5	X/2	Y/2	Y/2	X/2
6	X/3	Y/3	Y/3	X/3
7	Y/1	Y/1	X/1	X/1
8	Y/2	Y/2	X/2	X/2
9	Y/3	X/3	Y/3	X/3
10	Y/1	X/1	Y/1	X/1
11	Y/2	X/2	X/2	Y/2
12	Y/3	X/3	X/3	Y/3

語を入力する。

- ユーザーの入力した類似語・非類似語から推薦クエリを 20 個生成する。
- システムを用いて検索する問題について、ユーザーは推薦クエリから 10 件を選び、検索を行う。
- 同様に自力で検索する検索要求 2 題についても、異なるクエリ 10 個をユーザ自身が生成して検索する
- 各手法で取得できた適合ページ数を記録し、それを用いて両手法の性能を比較する。

5.2.3 提案手法について

提案手法では最終的に 20 個の推薦クエリをユーザーに提示する。本実験ではユーザーが自力で検索をする手法との比較を行うために、両手法で検索を行うクエリの数を同じにしなければならないが、ユーザーが自力で 20 個ものクエリを生成するのは困難である。したがって本実験では両手法ともに 10 個のクエリを試行することとした。ただしユーザーは、システムが推薦した 20 個のクエリの中から 10 個を選ぶという方法とした。

5.2.4 性能評価

本実験では、各手法の性能を precision@k、および nDCG で評価する。

5.2.5 precision@k

本実験では、表 7 からわかるように、各手法、各問題ごとに 2 人の異なるユーザーが実験を行っている。それぞれの問題の、

それぞれの手法ごとに、その手法で試行したクエリ 10 個から各検索要求に適合するページが得られた数をその問題・手法でのスコアとする。ただし、ユーザーは通常検索を行う場合検索結果の上位のページのみを閲覧し、上位ページに適合するページが存在しない場合はクエリを変更するなど、下位の検索結果ページまで閲覧することは少ない。したがってこの仮定に基づき、本研究では各クエリの検索結果において、上位 20 件のみをそのクエリから得られた全ページとみなす。本実験を 12 のユーザーに対して行った結果、2 つの手法の各問題でのスコアは表 8 のようになった。

表 8 各手法の precision@k

手法	1	2	3	4	5	6	7	8	9	10	11	12
提案手法	7	4	9	9	10	28	23	45	9	16	0	18
比較手法	5	8	13	135	133	81	125	66	53	28	0	13

表 7 を見ると、一部の問題において比較手法が提案手法よりも圧倒的に多くの適合ページを発見している。特に twitter や facebook, instagram 自体に関する話題を調べる問題においては、検索意図自体に曖昧性があったことが原因だと考えられる。すなわち、話題には、新機能、買収、新アプリなど、様々な種類の話題が存在するため、ユーザーはそれらの 1 つの話題に関して検索を行えば、20 件中多くのページが適合ページとなるようなクエリが生成でき、結果として適合ページを大量に発見することができた。一方で提案手法は、ユーザーが入力した類似語の周辺に一般的に現れる語を用いてクエリを生成するため、話題の中の一つに特化した語を生成しにくい。したがって、話題一般に対するクエリばかりが生成され、少ない適合ページしか発見できなかったものと考えられる。

一方で、問題 1~3 (種類 A) や問題 10~12 (種類 D) では、両手法とも同等程度か、一部では提案手法の性能が比較手法を上回った。これらの問題では、2 つの異なる文脈のページが、どちらも同じ分野に属しているため、それらを区別するような検索語の発見がユーザーにとって困難であったと考えられる。

5.2.6 nDCG を用いた性能評価

次に、実験によって得られた各クエリの検索結果数とそのうちの適合ページ数のデータから、各問題における nDCG を算出し、両手法を比較する。

まず初めに、各問題において両手法が生成したクエリの検索結果から、それぞれの nDCG を算出する。両手法ともに 10 個のクエリを生成した順序が存在するので、さきほどの nDCG を各クエリのゲイン値と見なし、各手法のクエリのゲイン値およびクエリのランキングから手法の nDCG を計算し、これを各手法の性能と見なす。ただし、各クエリの検索結果数の上限は 20、すなわち各問題のすべてのクエリの検索結果数の合計の最大値は 200 である。

以下の表が算出した nDCG 値の一覧である。ただし、各問題種別ごとの nDCG の平均を載せる。

表を見ると、問題 B では提案手法が比較手法を大きく下回るが、それ以外の問題では提案手法が比較手法と同等以上の性能

表 9 問題種別ごとの nDCG

問題 手法	問題 A	問題 B	問題 C	問題 D
提案手法	0.518618709	0.520386369	0.597585458	0.432070922
比較手法	0.453051287	0.700255329	0.609625074	0.384977668

であることがわかる。

問題 B は、本来の予想よりもユーザーにとって適合ページの発見が容易であったためユーザーが自力で検索を行う方がより効率的に適合ページを見つけられたと考えられる。例えば、「facebook 話題」というクエリには、「facebook 上で流行の話題」という意図と「facebook というサービスに関する話題」という意図が考えられ、ユーザにとって後者の発見が難しいものであると予想していたが、話題には「新機能」「買収」など様々なものが考えられ、これらを用いて検索すると 20 件すべてが適合ページと見なせる。ユーザにとってこのような検索語の発見は容易であったため、比較手法が圧倒的に提案手法を上回ったものと考えられる。

一方問題 A や問題 D は、りんごと肥料の関係性、yum とアンインストールの関係性が異なるだけであるため、二つの検索意図を区別する検索語の発見がユーザにとって容易ではなかった。したがって、このような多義的クエリの検索においては、本手法で生成するような検索意図を絞り込んだフレーズクエリの方がより効率的に検索意図に適合するページを発見できたと考えられる。

6. 結 論

本論文では、多義的クエリが含むマイナーな話題に関するページを発見するために、多義的クエリを構成する検索語の類似語に着目し、曖昧性を解消したクエリの生成および推薦手法を提案した。マイナーな話題はメジャーな話題に比べて圧倒的に数が少なく、またそれらを差別化する検索語を発見するのもユーザーにとっては容易ではない。そこで、類似語というよりユーザーが発見しやすいものに着目し、類似語の周辺フレーズがその言葉が用いられる文脈を決定するものであるという仮定のもとで、周辺フレーズを多義的クエリに適用することでクエリの曖昧性の解消を図った。提案手法の評価として、12 人のユーザーに提案手法を使用してもらい、ユーザーが自力で検索を行う場合と比較し、性能を評価した。結果として、提案手法では効率的に適合ページを発見できる場合が多かったが、最終的に適合ページをより多く発見できたのはユーザーが自力で検索を行う場合であった。しかし、ユーザー数、問題数ともに少なかつたこともあり、本来の性能を発揮できていないと考えられる。今後は、問題、ユーザーともに数を増やし、問題を精練した上で正しい性能を評価したいと考える。また本研究では、2 語の検索語からなるクエリのみを扱ったが、3 語以上の検索語からなるクエリにも本研究で扱ったような難しい多義的クエリが存在すると考えられる。

文 献

- [1] Baeza-Yates, Ricardo and Hurtado, Carlos and Mendoza, Marcelo, “Query recommendation using query logs in search engines”, Current Trends in Database Technology-EDBT 2004 Workshops, pp.588–596, 2005.
- [2] Cao, Huanhuan and Jiang, Daxin and Pei, Jian and He, Qi and Liao, Zhen and Chen, Enhong and Li, Hang, “Context-aware query suggestion by mining click-through and session data”, Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, pp.875–883, 2008.
- [3] Qiaozhu Mei, Dengyong Zhou, Kenneth Church, “Query suggestion using hitting time,” In Proceedings of the 17th ACM conference on Information and knowledge management, pp.469478, 2008.
- [4] 今井良太, 戸田浩之, 関口裕一郎, 望月崇由, 鈴木智也, 今井桂子, “Web 検索サービスにおける多義的なクエリ推薦手法” DBSJ Journal, pp.1-6, 2010.
- [5] Allan, James and Raghavan, Hema, “Using part-of-speech patterns to reduce query ambiguity” Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval, pp.307-314, 2002.
- [6] Leuski, Anton, “Evaluating document clustering for interactive information retrieval”, Proceedings of the tenth international conference on Information and knowledge management, pp.33–40, 2001.
- [7] Leuski, Anton, “Interactive information organization: Techniques and evaluation”, Diss. University of Massachusetts Amherst, 2001.
- [8] Allan, James and Leuski, Anton and Swan, Russell and Byrd, Donald, “Evaluating combinations of ranked lists and visualizations of inter-document similarity”, Information Processing & Management, pp.435–458, 2001.
- [9] 若木裕美, 正田備也, 高須淳宏, 安達淳, “検索語の曖昧性解消のためのトピック指向単語抽出および単語クラスタリング”, 情報処理学会論文誌. データベース, pp.72–85, 2006.
- [10] Rajul Anand and Alexander Kotov, “Improving Difficult Queries by Leveraging Clusters in Term Graph”, In Proc. of Asia Information Retrieval Societies Conference, Dec, 2015.
- [11] Franzen, Kristofer and Karlgren, Jussi, “Verbosity and interface design”, SICS Research Report, 2000.