情報型クエリのための
アンカーテキスト検索モデル

筑波大学

図書館情報メディア研究科

2019 年 3 月

CHEN TAIRUN

目次

図目次

表目次

第1章　はじめに

1.1　研究背景

情報検索とは、コンピュータシステムを使って、大量のデータ群から目的に合致したデータを探して取り出すことである。検索対象となるデータには文書や画像、音声、映像、その他さまざまなメディアやその組み合わせとして記録されたデータなどが含まれる。検索機能を提供しているそのコンピュータシステムを検索システム、検索エンジンという。検索システムは利用目的により機能も異なっている。読みたい本を探す蔵書検索システムや、過去の病例や判例などを検索するシステムも存在しているが、ウェブページを検索するウェブサーチエンジンが一番代表的なものである。

ウェブの発展とともに、サーチエンジンを利用して情報検索を行うユーザが増えている一方、ユーザの情報要求もだんだん複雑になっている。Broder はウェブ上の検索質問をユーザの情報要求に基づいてそれを情報型、案内型、取引型の 3 種類に分類した[1]。そして、ウェブページの数は幾何級数的に急増し、20 年前の百万単位から今の兆単位になっており、ウェブページに含まれているテキスト、画像、音声、映像などのメディアコンテンツも幾何級数的に増加している。ほぼ無限なウェブの中から、多種多様のユーザ情報要求を満たすウェブページを検索するのが情報検索領域において大きな課題になっていて、それに対する研究も盛んでいる。

ウェブ検索の研究はサーチエンジンの全てに対して行っている。サーチエンジンは、ユーザが入力したクエリにより、関連するウェブページを探し、ユーザに出力するシステムである。サーチエンジンは基本的に、クローラー、テキスト変換、索引、ユーザインタフェース、検索モデル、評価、六つのコンポーネント[2]に構成されている。その中で、一番重要なコンポーネントは検索モデルである。検索モデルは、ユーザが入力したクエリを、クエリと文書のマッチングルールにより関連文書を探し、関連文書の関連度によりランクするアルゴリズムをモデル化したものである。例として、ウェブページのテキストを単語にブレイクダウンして、単語の出現頻度と文章内頻度によりウェブページの関連度を計算する最も基本的なコンテンツ検索モデル BM25 がある[3]。

リンクはあるウェブページから他のウェブページに移動する参照で、リンクのアンカーテキストはいつもリンク先ウェブページに対して短く描写し、概括的に述べている特徴を持っていて、ユーザが他のウェブページに移動することを導いている。これは、クエリの探したい内容について簡略的に概括したキーワードという特徴と一致するため、アンカーテキストはクエリとウェブページを繋がる重要な鍵であると思われる。従って、アンカーテキストを用いてウェブページを検索するアンカーテキスト検索モデルの研究が行なわれてきた[4][5][6]。

今まで、アンカーテキスト検索モデルはよく研究されたが、ほぼ全てのアンカーテキスト検索モデルは案内型クエリには役立っている一方、情報型クエリには特に役立っていないことが示されている[4][5]。研究者達はアンカーテキスト検索モデルの研究を続いて行なってきたが、情報型クエリに役立つ成果はあまりない。Koolen らはその原因をコーパス内でのサイト内リンクとサイト間リンクの数が少ないことと示した[6]。しかし、近年ではより多くのデータ量を含む Clueweb12[1]などのデータセットが公開され、アンカーテキスト検索モデルの研究はブレイクスルーがあるかもしれない。

1 https://lemurproject.org/clueweb12/

1.2 研究目的

本研究の研究の目的はアンカーテキストの情報型クエリに対する提案する検索モデルの有用性を示すことである。そして、ウェブ検索において、サイト内リンクの有用性を示すことである。本研究はリンクとアンカーテキストを使って検索を行うアンカーテキスト検索モデルを提案し、提案した検索モデルを用いて実際の検索を行い、検索結果の評価に基づいて、以下のリサーチクエスチョン RQ に答える。

RQ1：アンカーテキストは情報型クエリに対して有用であるか。

RQ2：文書の関連スコアを計算において、アンカーテキストのスコアを計算する時、アンカーテキストの重要度により重み付けて取り扱うのは有効であるか。

RQ3：サイト間リンクとサイト内リンクのアンカーテキストの情報型クエリに対する有用性は異なるか。

RQ4：コンテンツ検索モデルと比べると、アンカーテキスト検索モデルの利点と欠点は何か。

1.3 本論文の構成

本論文は全部で 6 章から構成されている。第 1 章は、研究背景と研究目的について述べる。第 2 章は、関連研究について説明する。研究領域の全体像と他の研究者が提案した検索モデルについて説明する。第 3 章は、本論文で提案するアンカーテキスト検索モデルについて詳しく説明する。そして、第 4 章では、提案モデルに対して行う評価実験について述べる。第 5 章では、評価実験の評価結果に基づいて考察を行う。最後に、第 6 章で総括を行う。

第2章　関連研究

本章では、本研究の関連研究について説明する。2.1 節は、ウェブ検索領域の全体像と検索モデルを含めた研究方向について述べる。2.2 節は、検索モデルの研究において、検索モデルの種類について説明する。2.3 節は、他の研究者に提案されたアンカーテキスト検索モデルについて説明する。

2.1　ウェブ検索に関する研究

ウェブ検索に関する研究は主に 2 種類に分類できる。１つはシステム側の研究で、２つ目はヒューマンインタラクションの研究である。システム側の研究としては主に検索モデルの研究である。検索モデルはユーザが入力したクエリに対して、検索対象である文書データからなんらかの方法でクエリに関連する文書を探し、関連文書を上位にランクしてユーザに出力するものである。検索モデルの研究は多種多様であるが、節 2.2 で続いて説明する。

ヒューマンインタラクションの研究は、ユーザの情報検索行動とかユーザインタフェースなどがあって、評価指標の研究もこの方面で行う場合がある。ウェブ検索においてユーザの情報検索行動の研究でよく知られているのは Bates[7]の果実摘みモデルである。Bates はユーザがウェブ検索においての検索行動をモデル化して説明していた。Bates は人間の情報ニーズとクエリは検索システムの検索結果によって変えることと仮定し、人間の情報ニーズは一回の検索で満足されず、シリーズの検索行動に基づいて満足できると指摘した。ユーザインタフェースの研究は、検索ページで検索を支援するファセットに関して、デザインしたファセットの有用性を検討するものなどがある[2]。

2.2　検索モデルに関する研究

前節に述べたように、検索モデルはユーザが入力したクエリに対して、検索対象である文書データからなんらかの方法でクエリに関連する文書を探し、関連文書を上位にランクしてユーザに出力するものである。検索モデルの研究はこの何らかの方法に対して研究することである。関連文書として判定するルールを指定し、関連文書に順位つけるルールを決めることである。検索モデルは関連度によるモデルと重要度によるモデルに分けられる[8]。

関連度によるモデルはクエリと文書の関連度を計算し、関連度の高いものを検索結果として提示する。早期の関連度によるモデルとしてブーリアンモデル[9]がある。ブーリアンモデルは文書がクエリに対して関連するか否かを判定できるが、文書の関連度を計算することはできない。そして、ベクトル空間モデル[9]が提案された。このモデルはクエリと文書をベクトルに変換し、ベクトルのコサイン類似度により文書の関連度を計算した。ブーリアンモデルとベクトル空間モデルに異なり、確率に基づいた検索モデル BM25[3]と確率的言語モデル[10]が提案された。そして、ウェブの発展と共に、ウェブページ中のリンクとアンカーテキストが重視され、アンカーテキスト検索モデル[4][5]が提案されてきた。

重要度によるモデルは文書そのものの重要度を計算し、重要度の高いものを優先的に検索結果として採用する。一番有名な例としては Pagerank[11]がある。Pagerank はユーザがランダムにリンクをクリックするとき、特定のページに移動する確率を示しているものである。

機械学習の発展とともに、情報検索領域にも機械学習の手法を用いて検索モデルを研究する Learning to rank[8]、ランキング学習が近年からトレンドになっていた。一般的には情報検索の問題を機械学習の手法によって解決することを目指す一連のような手法をランキング学習と呼ばれる。例えば、機械学習によるパラメータの自動チューニングなどがランキング学習に含まれる。そして、複数の検索モデルを合併するときに各モデルに与える重みを自動

チューニングし、最適な値を計算することもランキング学習に含まれる。

2.3 アンカーテキスト検索モデルに関する研究

アンカーテキスト検索モデルの関連研究に対して Fujii[4]と Dou[5]らの研究を説明する。

まず、Fujii の研究研究について説明する。Fujii は検索質問分類、コンテンツ検索とアンカー検索の３つのコンポーネントで構成されているウェブ検索システムを提案した。検索質問分類はクエリを情報型と案内型に分類する分類器である。クエリが入力されると、クエリ分類器により、情報型と案内型に分類される。そして、クエリのタイプによらずに、コンテツ検索とアンカー検索を独立行って２つの検索結果を作成する。コンテンツモデルは BM25 を用いて、アンカーモデルは新たな手法を提案していた。最後に、２つの初期検索結果をクエリのタイプによって両方の検索結果に重みづけて合併して出力する。Fujii が提案いたアンカーモデルは、クエリ q が与えられた条件のもとでページ d が検索される確率 P(d | q)を計算し、この確率でページを順位つけるものである。Fujii の提案モデルはクエリを単語にブレイクダウンし、各単語に対して、文書に関連する確率を計算し、全ての単語の確率をかけてクエリの文書に対する関連確率とした。Fujii は SEO などのスパムに対する対策として、同一のサイト内のリンクとそれに対応したアンカーテキストは抽出しなかった。また、あるページから同一のページに対して複数のリンクがあった場合は、最初のリンクとそれに対応したアンカーテキストだけを抽出した。Fujii の検索システムは NTCIR-3[12]、NTCIR-4[13]と NTCIR-5[14]の３つテストコレクションを用いて評価実験を行った。評価結果として、案内型クエリに対してはベイスラインより評価結果が良かったが、情報型クエリに対しては大きな差はなかった。そして、NTCIR-5 テストコレクションは案内型クエリしかないため、情報型クエリの評価は NTCIR-3 と NTCIR-4 のテストコレクションにより行ったが、これは約１千万ページしか含まれていない小規模のデータセットである。従って、Fujii の提案モデルが大規模データセットにも適応できるか否かも検討すべき部分である。

次は、Dou らのアンカーテキスト検索モデルについて説明する。Fujii と異なり、Dou らはサイト間の関係に着目して検索モデルを構築した。クエリはブレイクダウンしなくて、クエリと全く同じタームを持っているアンカーテキストだけ扱っていた。検索方法はクエリとアンカーテキストをマッチングし点数を計算する一方で、あるサイトに同じ内容で複数回リンクされる状況や複数のミラーサイトにリンクされる状況を考慮し、このようなリンクを検出して点数を減らしながら検索を行っていた。Dou らのシステムはサイト間の関係に着目したため、サイト間のリンクだけ使って、サイト内リンクは抽出しなかった。評価結果として、案内型クエリに対してはベイスラインより評価結果が良かったが、情報型クエリに対しては逆にベイスライよりよくなかった。

両方の提案する検索モデルはサイト内リンクとそのアンカーテキストを取り扱わなかったため、ウェブ検索においてサイト内リンクの有用性まだは示されていない。そして、両方の提案モデルは情報型クエリに対して良い成果がないため、情報型クエリに対するアンカーテキストの有用性を示すことが課題になっている。

第3章　　提案手法

本章では本論文で提案するアンカーテキスト検索モデルについて説明し、提案モデルを実現した検索システムの構造について詳しく述べる。

3.1　　検索システム

本論文はアンカーテキスト検索モデルを提案し、提案した検索モデルを用いて実際の検索を行うために、検索システムを構築した。構築した検索システムには、四つのコンポーネントが含まれていて、それぞれ、文書データセット、テキスト変換、索引と検索モデルである。

文書データセットは検索対象になる文書の集合で、ウェブ検索にはウェブページを文書として取り扱う。本論文の検索システムが使うデータセットは Clueweb12[2]というデータセットで、約 7 億の英語ウェブページが含まれていて、現時点でデータ規模が一番大きい英語のデータセットである。

テキスト変換は索引を構築するために、文書の中から特徴量を抽出するコンポーネントである。アンカーテキスト検索モデルの索引はリンク情報とアンカーテキストが特徴量であるため、Clueweb12 データセットで HTML テキストに保存されているウェブページを読み込む、全てのリンクのタグを見つけて、リンク内のリンク先ページの URL とアンカーテキストを抽出するツールを構築した。

索引、或いは転置索引は、リンク先ページ、すなわち、リンクされらページをキーとして、キーのページに指している全てのリンクのアンカーテキストを値とする。そのキーの値であるアンカーテキスト集合は、キーのページの内容について概括して述べているため、キーのページ内容を代わりにキーのページを代表して検索に扱う。そして、そのアンカーテキスト集合はアンカーテキスト文書とも呼ばれる。アンカーテキスト文書は 3.5 節で詳しく説明する。

検索モデルは検索システムにおいて検索結果の質を直接に決めるものとして一番重要な部分である。検索アルゴリズムに細微な変更があっても検索結果が完全に異なる場合も珍しくない。本論文で提案する検索モデルは、サイト間リンクとサイト内リンク両方を扱うため、リンク数の多少による影響を最低限にするため、リンク数に応じた減損を与えクエリとアンカーテキストの内容に集中している。

3.2　　提案モデル

本論文で提案するアンカーテキスト検索モデルは、ユーザが入力したクエリにより関連するウェブページを探し、関連度によりランキングするアルゴリズムであり、ウェブにおいてウェブページ内のリンクとアンカーテキストの情報を取り扱う数学モデルである。

まず、クエリが入力されると、索引のアンカーテキスト文書にクエリのタームを一つ以上含んでいるウェブページを関連文書として検索する。検索されたウェブページに対してスコアを計算し、スコアに基づいてページに順位をつける。

提案するアンカーテキスト検索モデルの全体的なアルゴリズムは式（1）に示す。

$$Score(d,q) = I_d \times \left(\sum_{a \in D_a} g(a,q) \times dc(a) \times \beta + \sum_{a \in S_a} g(a,q) \times dc(a) \times (1-\beta) \right) \qquad (1)$$

2 https://lemurproject.org/clueweb12/

このアルゴリズムは、あるクエリ q に対する文書 d、すなわちウェブページ d の関連度スコアを計算するアルゴリズムである。式（1）において$\sum_{a\in D_a}$と$\sum_{a\in S_a}$は、それぞれアンカーテキスト文書の中で、サイト間リンクのアンカーテキスト文書D_aに含まれるアンカーテキストの得点の総合とサイト内リンクのアンカーテキスト文書S_aに含まれるアンカーテキストの得点の総合である。アンカーテキストは独立な存在で、各アンカーテキストスコアの計算はお互いに影響しない。すなわち、本論文でのアンカーテキスト文書は、複数のアンカーテキストで構成されている一つのテキスト文書ではなく、複数のアンカーテキストを含んでいるリストである。$g(a,q)$はクエリ q に対するアンカーテキストの得点を計算する方法で、具体的なアルゴリズムは式（2）に示す。

$$g(a,q)=\frac{sum\,(q\bigcap a)}{|a|}\times\left(\frac{num\,(q\bigcap a)}{|q|}\right)^{e}\times\prod_{t\in(q\bigcap a)}idf\,(t)\times I_a \qquad (2)$$

$sum(q\cap a)$ はアンカーテキストaの中でクエリqのタームが出現した回数の合計で、アンカーテキストaの単語数で割り、アンカーテキストaのクエリqに対する一致度を計算する。$sum(q\cap a)$はアンカーテキストにクエリの同じタームが複数回出現しても全部数える。$num(q\cap a)$はアンカーテキストaの中でクエリqのタームが出現した個数で、クエリqの単語数に割り、クエリqのアンカーテキストaに対する一致度を計算する。$num(q\cap a)$はアンカーテキストにクエリの同じタームが複数回出現しても一回だけ数える。そして、クエリのタームが出現した回数よりも、出現した個数をもっと重視するため、クエリqのアンカーテキストaに対する一致度を e 乗にする。$\prod_{t\in(q\cap a)}idf(t)$はアンカーテキスト$a$に出現したqのタームの$idf$値を連乗する。タームtの$idf$値は、アンカーテキスト文書を単位として、全てのアンカーテキスト文書の中タームtを含めているアンカーテキスト文書の比を対数で計算した。具体的な計算方法は式（3）に示す。idfの計算はサイト間とサイト内の索引に分けて計算を行なったため、同じ単語であってもサイト間とサイト内によりidfの値も異なる。前に述べたように、クエリのタームが出現した個数を重視するため、重ねているタームが多いほどスコアは高くなる、更にidf値が高いタームが多いほどスコアが高くなる。最後のI_aは、アンカーテキストaの重要度で、アンカーテキストaのリンク元ページの Pagerank 値とする。これは、全く同じ内容のアンカーテキストでリンクされても、訪問数が多いページからのリンクとほぼ訪問されないページのリンクの価値は異なると思うからだ。この仮定は比較評価を行う上で詳しく述べる。

$$idf(t)=\lg\left(\frac{|D|}{|\{d\in D:t\in d\}|}\right) \qquad (3)$$

$dc(a)$はアンカーテキストのクエリに対するスコアの総和に与える減損である。他のウェブページにリンクされる回数はあるウェブページの価値を評価する重要な指標である。他のウェブページに数回しかリンクされないページも存在し、数百万回リンクされるページも存在する。もちろん、数百万回リンクされるページの価値が高く、参照される確率も高い。しかし問題は、リンクは人手で簡単に作られるため、自分のウェブページの価値を向上するために有意にリンクをつけるのも珍しくない。そして、このような無意味なリンクを有効に識別する方法はまだ提案されてない。従って、リンクの量よりも質に集中するため、リンクの回数により適当な減損を与える。減損の計算方法は式（4）に示す。

$$dc(a) = \begin{cases} 1 & (sum(a) \leqslant AvgInGroup) \\ \frac{AvgInGroup}{sum(a)} & (sum\,(a) > AvgInGroup) \end{cases} \quad (4)$$

$dc(a)$は絶対的な数値ではなく、検索結果による相対的な数値である。$AvgInGroup$はクエリqに対する全ての関連文書を一つのグループとみて、グループ内でウェブページの平均リンク数を求め、平均リンク数を軸として減損を決める。各ウェブページのリンク数はこのウェブページが他のウェブページにリンクされた数であり、このウェブページのアンカーテキスト文書内に含まれているアンカーテキストの数と同様である。もし、あるウェブページのリンク数が平均リンク数より少ないと、このウェブページに減損を与えない。逆に、あるウェブページのリンク数が平均リンク数より多い時、平均リンク数をウェブページのリンク数に割った数値に相当する減損を与える。

βはサイト間のスコアとサイト内のスコアを合併する際、両方の数値を正規化するためのパラメータである。βの値は 0 以上 1 以下の範囲で取る。最後に、合併したスコアに文書 d の重要度であるI_dをかける。文書の重要度はこの文書自身の Pagerank 値とする。

図 1 は特定のクエリ q に対するある文書 d の関連度スコアの計算例である。クエリ q は「macbook air」二つのタームが含まれていて、それぞれの IDF 値は 10 と 5 である。IDF の値は単語の参照価値を代表している。多くの文書に出現している単語ほど IDF 値が低く、参照価値も低い。文書 d のクエリに対する関連度スコアを計算するためには、文書 d のアンカーテキスト文書 A_d が必要であり、それをサイト間リンクのアンカーテキスト文書 D_a とサイト内リンクのアンカーテキスト文書 S_a に分ける。文書 d のクエリに対する関連度スコアは両方のアンカーテキスト文書に分けて計算を行い、最後に両方のスコアを合計したスコアとする。続いて、両方のアンカーテキスト文書内の各アンカーテキストのクエリに対するスコアを計算する。一番のアンカーテキストのスコアを計算すると、アンカーテキストの二つのタームが全部クエリに含まれているので、$\frac{sum(q \cap a)}{|a|} = \frac{2}{2}$である。そして、クエリの二つのタームが全部アンカーテキストに出現されているので、$(\frac{num(q \cap a)}{|q|})^e = (\frac{2}{2})^e$である。従って、$\prod_{t \in (q \cap a)} idf(t) = 5 * 10 = 50$である。$I_{a1}$ はこのアンカーテキストのリンク元ページの Pagerank 値である。五番目のアンカーテキストのスコアを計算すると、アンカーテキストの二つのタームで macbook だけクエリに含まれているので、$\frac{sum(q \cap a)}{|a|} = \frac{1}{2}$である。そして、クエリの二つのタームで mabook だけアンカーテキストに出現されているので、$(\frac{num(q \cap a)}{|q|})^e = (\frac{1}{2})^e = 0.15$である。従って、$\prod_{t \in (q \cap a)} idf(t) = 10$である。上記のように両方のアンカーテキスト文書の各アンカーテキストのスコアを計算した後、スコアを合併する。合併したスコアに各アンカーテキスト文書に与えられる減損をかけ、βと$1 - \beta$をそれぞれかける。文書 d が含めているサイト間リンク数はサイト間アンカーテキスト文書内のアンカーテキスト数と相当で、7 個である。例え、クエリ q に関連する文書が 100 件あって、サイト間リンク数の平均値が 10 である場合、文書 d のサイト間アンカーテキストには減損を与えない。しかし、サイト内リンク数の平均値が 5 である場合、サイト内アンカーテキストに与える減損は

$\frac{AvgInGroup}{sum(a)} = \frac{5}{6}$である。最後に両方のスコアを合併して、文書 d の Pagerank にかけて文書 d の最終スコアになる。

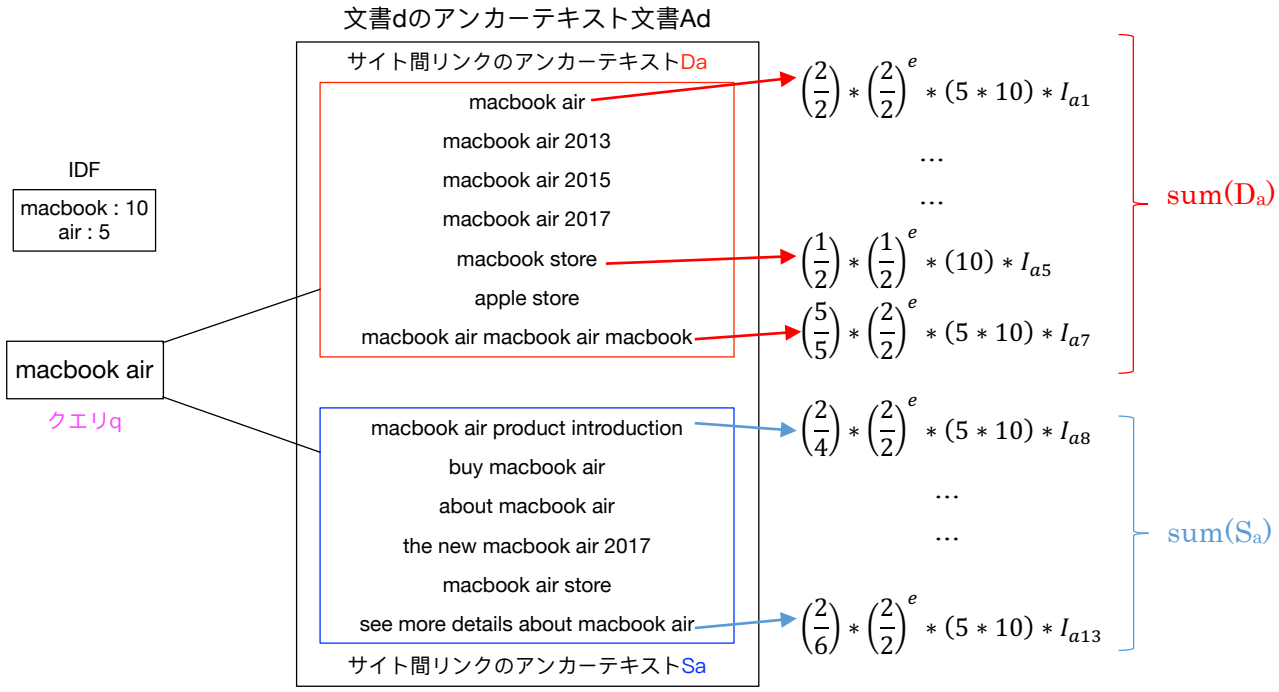

図 1　関連度スコアの計算例

3.3　　データセット

3.3.1　　文書データセット

本論文で提案する検索システムで、文書データとして取り扱う ClueWeb12 と言うデータセットは、LemurProject[3]の一部であり、カーネギーメロン大学の Jamie Callan のリサーチグループにより、情報検索の研究を支えるために作ったデータセットである。ClueWeb12 は 733,019,372 個の英語ウェブページが含まれていて、2012 年 2 月 10 日から 2012 年 3 月 10 日の間にネット上でクローラーを用いてウェブページを収集した。ClueWeb12 は ClueWeb12−B13 という小さいバージョンも提供している。このバージョンは ClueWeb12 データ量の 7%に相当するサンプルデータセットである。ClueWeb12−B13 には、33,447 個の WarcFile が含まれていて、合計 52,343,021 このウェブページが含まれている。

ClueWeb12 前のバージョンは ClueWeb09[4]というデータセットである。ClueWeb09 は情報検索領域において初めての大規模ウェブページデータセットである。ClueWeb09 は ClueWeb12 と異なり英語のウェブページだけでなく、10 種類言語のウェブページが含まれていて、合計約 10 億のウェブページが含まれている。その中で、英語、中国語、スペイン語、日本語、フランス語、ドイツ語、ポルトガル語、アラビア語、イタリア語、韓国語のウェブページがそれぞれ約 5 億、1 億 7 千万、8 千万、7 千万、5 千万、5 千万、3.7 千万、3 千万、2.7

3 https://lemurproject.org/index.php

4 https://lemurproject.org/clueweb09.php/

千万、1.8 千万個含まれている。このデータセットもカーネギーメロン大学の Jamie Callan のリサーチグループにより作られた。

ClueWeb12 のディレクトリ構成は図 2 にしめす。

Clueweb12

							WarcFiles	Records
Disk1								
	clueweb12_00	=>	clueweb12_0000	～	clueweb12_0013	=>	1389	45,278,522
	clueweb12_01	=>	clueweb12_0100	～	clueweb12_0112	=>	1270	44,389,316
	clueweb12_02	=>	clueweb12_0200	～	clueweb12_0212	=>	1233	44,069,951
	clueweb12_03	=>	clueweb12_0300	～	clueweb12_0312	=>	1234	42,491,359
	clueweb12_04	=>	clueweb12_0400	～	clueweb12_0412	=>	1236	36,026,724
Disk2								
	clueweb12_05	=>	clueweb12_0500	～	clueweb12_0512	=>	1267	21,720,416
	clueweb12_06	=>	clueweb12_0600	～	clueweb12_0612	=>	1283	23,101,855
	clueweb12_07	=>	clueweb12_0700	～	clueweb12_0717	=>	1751	30,503,029
	clueweb12_08	=>	clueweb12_0800	～	clueweb12_0819	=>	1911	39,712,288
	clueweb12_09	=>	clueweb12_0900	～	clueweb12_0920	=>	2001	38,540,335
Disk3								
	clueweb12_10	=>	clueweb12_1000	～	clueweb12_1021	=>	2122	39,802,260
	clueweb12_11	=>	clueweb12_1100	～	clueweb12_1118	=>	1883	40,754,618
	clueweb12_12	=>	clueweb12_1200	～	clueweb12_1218	=>	1854	38,606,284
	clueweb12_13	=>	clueweb12_1300	～	clueweb12_1316	=>	1653	31,329,242
	clueweb12_14	=>	clueweb12_1400	～	clueweb12_1416	=>	1680	32,95,0768
Disk4								
	clueweb12_15	=>	clueweb12_1500	～	clueweb12_1516	=>	1686	37,716,513
	clueweb12_16	=>	clueweb12_1600	～	clueweb12_1617	=>	1741	34,996,028
	clueweb12_17	=>	clueweb12_1700	～	clueweb12_1717	=>	1751	34,051,249
	clueweb12_18	=>	clueweb12_1800	～	clueweb12_1815	=>	1541	40,074,978
	clueweb12_19	=>	clueweb12_1900	～	clueweb12_1914	=>	1429	36,903,637

図 2　ClueWeb12 のディレクトリ構成

ClueWeb12 ディレクトリには四つのディスクがあって、それぞれ五つのセグメントが含まれ、合計で 20 個のセグメントが ClueWeb12 に存在してる。セグメント番号は 00 から 19 まで表記され各セグメント中には十数個のフォルダがある。各フォルダには数十個の WarcFile が含まれている。図のディレクトリの右側で書いている WarcFiles と Records の数値はそれぞれ、各セグメントに含まれている WarcFile 数の合計と Record 数の合計である。

ClueWeb12 データセットは、一つのウェブページを一つの Record として表記し、複数の Record を一つの WarcFile というファイルに保存している。WarcFile のファイル拡張子は.warc である。全ての WarcFile は gzip 形式で圧縮して保存している。

ClueWeb12 データセットには 33,447 個の WarcFile が保存されていて、全部 733,019,372 個の Record が含まれている。ClueWeb12 データセットの圧縮したサイズは 5.54TB であり、解凍したデータセットのサイズは 27.3TB である。

ClueWeb12 データセットについて更なる情報を取得するために、幾つかの項目に対して統計を示す。統計結果は表 1 に示す。

表 1 ClueWeb12 データセットの統計結果

総ウェブページ数	733,019,372
リンクされたウェブページ数	611,734,614
サイト内リンクにリンクされたウェブページ数	605,089,025
サイト間リンクにリンクされたウェブページ数	37,180,680
両方にリンクされたウェブページ数	30,535,898
総リンク数	31,398,703,748
サイト内リンク数	25,643,232,198
サイト間リンク数	5,755,278,368
アンカーテキストの平均文字数	3.9
サイト内リンクのアンカーテキスト平均文字数	3.91
サイト間リンクのアンカーテキスト平均文字数	4.28

表 1 の統計結果について説明する前に、本論文で定義する有効リンクと無効リンクについて明らかにする。有効リンクはリンクのアンカーテキストの内容があって、リンクするウェブページとリンクされるウェブページが全部データセットに Record として存在するものである。逆に、両者の条件を満たされないリンクは無効リンクである。表 1 の統計結果で書いているリンクは全て有効リンクを指している。

表 1 の統計結果によると ClueWeb12 データセットには 733,019,372 個のウェブページが含まれているが、611,734,614 個のウェブページが他のページにリンクされている。すなわち、約 1 億のウェブページが検索対象外になっているが、この数値はかなり高い方である。605,089,025 個のウェブページがサイト内リンクにリンクされているが、サイト内リンクを使わなく、アンカーテキスト検索モデルを構築すると膨大な有用のデータを失う。そして、37,180,680 個のウェブページがサイト間リンクにリンクされているが、サイト間リンクだけで検索を行うと検索対象が急激に縮減する。

Record は RcordHeader、HttpHeader、HTML テキスト、3 つの部分で構成されている。RcordHeader の WARC-TREC-ID 属性は当 Record の ID を表示し、WARC-Target-URI 属性は当 Record のウェブページの URL を表示している。それ以外も Record 内容の文字数などの情報が書いている。図 3 は RcordHeader の一つの例である。

```
WARC/1.0
WARC-Type: response
WARC-Date: 2012-02-10T20:51:40Z
WARC-TREC-ID: clueweb12-0000wb-00-00000
WARC-IP-Address: 100.42.50.192
WARC-Payload-Digest: sha1:2U2RINYC55ZCGDBLENIG4NYAMJ5PDZ6D
WARC-Target-URI: http://ahmetertug.com/ahmetertug.html
WARC-Record-ID: <urn:uuid:fe1d1f4e-a689-446b-8a39-e33ba376a335>
Content-Type: application/http; msgtype=response
Content-Length: 13805
```

図 3　RecordHeader の例

HttpHeader は当 Record のウェブページの Http メッセージが書いている。HttpHeader の例は図 4 で示す。

```
HTTP/1.1 200 OK
Date: Fri, 10 Feb 2012 20:51:42 GMT
Server: Apache/2.2.21 (Unix) mod_ssl/2.2.21 OpenSSL/0.9.8e-fips-rhel5 mod_auth_passthrough/2.1 mod_bwlimited/1.4 FrontPage/5.0.2.2635 mod_jk/1.2.30
Last-Modified: Sun, 05 Dec 2010 21:09:03 GMT
ETag: "ce38022-3475-496b02e5b79c0"
Accept-Ranges: bytes
Content-Length: 13429
Connection: close
Content-Type: text/html
```

図 4　HttpHeader の例

HTML テキストはウェブページを構成する HTML ソースコードであり、追加の説明はしない。しかし、注意すべきことは、HTML テキストはクローラーで獲得した時点での内容で、現時点にそのウェブページが存在しても、その内容は異なり、ウェブページの廃棄、サイト運営の停止によるウェブページの訪問が失敗することもある。

3.3.2　Pagerank データ

本研究で取り扱う全ての Pagerank は ClueWeb12 データセットで提供している Pagerank を使っている。この Pagerank は Galago というツールキットを使用して計算した結果である[5]。

3.4　テキスト変換

テキスト変換は Record の HTML テキスト部分の HTML ソースコードを解析し、必要な内容を抽出するツールである。本論文で提案するアンカーテキスト検索モデルはウェブページの本文を利用せず、リンクとアンカーテキストだけ利用するため、アンカーテキスト抽出ツールを用いてテキスト変換を行う。まず、本論文で取り扱うリンクとアンカーテキストの定義を明らかにする。リンクは HTML で、当ウェブページから他のウェブページに移動する参照で、<a>...を利用してリンクをつける。アンカーテキストは他のウェブページにリンクつける時、ターゲットページについて説明、概括する文字列である。

アンカーテキスト抽出ツールは既に良くリリースされたが、抽出ルールがお互いに違い、本論文のアンカーテキスト検索モデルに合うツールが特にないため、新たなアンカーテキスト抽出ツールを構築した。そして、構築したツールの抽出ルールを整理する

- <a>で始まり、で終わるテキストだけリンクとして扱う。

5 http://www.lemurproject.org/clueweb12/PageRank.php

- <a>からまでにアンカーテキストが書いてないリンクは抽出しない。アンカーテキストがないと、リンク自体は意味がない。
- <a>タグ内でリンク先の URL を代表する href 属性がない場合とか、href 属性の値が空いている場合はそのリンクを抽出しない。
- <a>タグ内の href 属性に「#」、「javascrip:0」など、どんなアクションも起こらないリンクも扱わない。
- <a>タグ内の href 属性に書いている URL の解析は Java の URL クラスの URL 解析ファンクションで行い、解析できない URL はそのリンクも扱わない。
- <a>タグ内の href 属性に書いている URL のエスケープコードを対応する特殊文字に変換する。これは、Record の URL はエスケープコードではなく、特殊文字を使っているので、索引を構築する際、リンクの URL と Record の URL を対応できるようにするためである。
- <a>との間に<img>タグで画像にリンクつける場合もあるが、この様なリンクは取り扱わない。

3.5 索引の構築

本論文で提案する検索システムは 3 種類の索引が構築されている。提案するアンカーテキスト検索モデルはサイト間リンクとサイト内リンクを分けて計算を行うため、索引も分離して構築した。各種類の索引はサイト間側とサイト内側二つの索引が含まれている。従って、合計 6 個の索引が構築されている。まず、一番重要なウェブページ索引から説明する。

ウェブページ索引は、検索アルゴリズムによりウェブページの関連度スコアを計算するための索引である。索引のキーはウェブページの URL である。ここで注意すべき点は、文書データセットを使った場合は、この文書データセットが全てのインターネットを代表することである。すなわち、あるリンクのリンク先ウェブページがデータセットに存在しない場合、このウェブページは存在しないと見られるはず、そのリンクは無効リンクとして扱う。従って、索引のキーであるウェブページの URL は全部データセットに Record として存在しているウェブページの URL である。もちろん、ウェブページがデータセットに存在しても、他のウェブページにリンクされないと、索引に書かれない場合もあるし、サイト内リンクばかりにリンクされると、サイト間側索引に書かれない場合もある。ウェブページ索引のバリューは索引のキーであるウェブページにリンクしている全てのサイト間、あるいはサイト内リンクのアンカーテキストとリンクするウェブページの URL の集合である。

テキスト変換を行い、1 つのアンカーテキストとリンクを抽出した結果は以下のような形になっている。

- リンクするページの URL → アンカーテキスト → リンク先ページの URL

ウェブページ索引の構築はこの結果を転置してリンク先ページの URL をキーとして、リンクするページの URL とアンカーテキストペアをバリューとする。そして、キーの URL が同じのバリューを集める。アンカーテキストはリンク先ページに対して述べる文字列、全てのアンカーテキストを集まるとリンク先ページを述べる文書が作られ、アンカーテキスト文書と呼ばれる。アンカーテキスト文書の例は図 5 に示す。

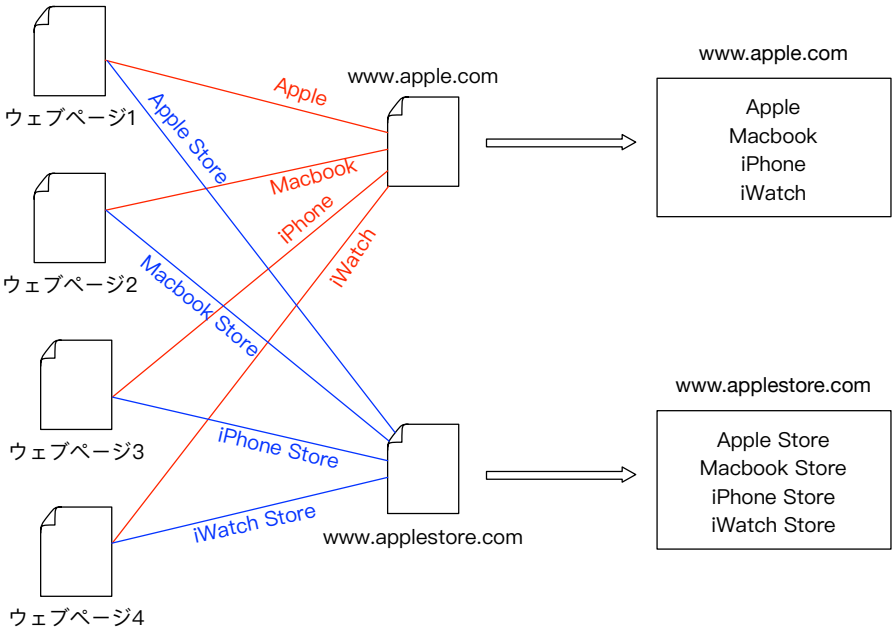

図 5　アンカーテキスト文書の例

図 5 の例でウェブページ 1 から 4 までの四つのページがそれぞれ、Apple、Macbook、iPhone、iWatch をアンカーテキストとして www.apple.com が URL であるウェブページにリンクしたとき、このウェブページのアンカーテキスト文書は右のようになる。

ウェブページ索引はデータの量が非常に膨大であるため、データベースを利用して保存すると処理する速度が遅くなる。本論文のシステムはウェブページ索引をテキストファイルに保存している。キーとバリューの間はタブで区切る。そして、各バリューの間もタブに区切る。索引の項目間には改行を使って区切る。索引を読む時、改行を読んで初めて読んだタブの間の内容がキーで、次のタブまで読んだ内容がバリューであり、キーとバリューを間違える場合はないようにした。

続いて単語索引と IDF 索引について説明する。単語索引はクエリに関連する文書を早く探すための索引である。索引のキーは一つの単語で、バリューはアンカーテキスト文書にこの単語を含んでいる全てのウェブページの集合である。すなわち、アンカーテキスト文書にクエリの単語が含んでいないと、実際にこのページがクエリと関連性があっても検索されないことである。実は、単語索引がなくても、ウェブページ索引で全ての索引項目を読み込んだ上で関連文書を探すことができるが、前に述べたように、ウェブページ索引のデータ量は非常に大きいため、単語索引でまず関連文書を探して、関連文書の URL とキーが一致するウェブページ索引を読み込むことで時間を短縮することができる。単語索引の例は図 6 に示す。

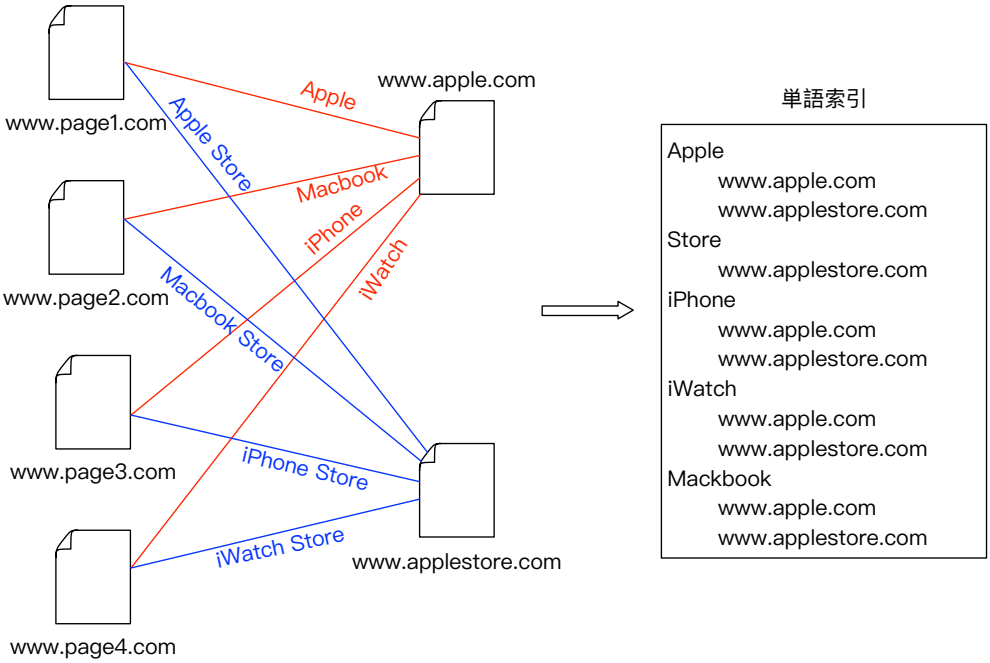

図 6　単語索引の例

IDF 索引は、全ての単語の IDF 値を取得するための索引である。IDF 索引のキーは一つの単語で、バリューはこの単語の IDF 値である。IDF 索引もサイト間側とサイト内側に分けていて、IDF 値の計算も分けて行なっていた。すなわち、同じ単語でも、サイト間とサイト内によりその値も異なっている。

3.6　　ストップワード

本論文で提案する検索システムは索引とクエリに対してストップワード処理を行っていた。ストップワードとは、情報検索において、意味を持ってない単語、あまり一般的である単語を検索から除外することである。例えば、日本語の「は」、「の」、「に」などがあげられる。英語の場合は、「a」、「an」、「the」などがある。ストップワードは特に権威的なバージョンがなくて、普通には自分の要求に応じて作られている。本論文で使うストップワードは MySQL[6] で使っているストップワードである。このストップワードには 36 個の単語が含まれていて、詳細は表 2 で示す。

表 2 ストップワード

a	about	an	are	as	at
be	by	com	de	en	for
from	how	i	in	is	it
la	of	on	or	that	the
this	to	was	what	when	where
who	will	with	und	the	www

6 https://dev.mysql.com/doc/refman/8.0/en/innodb-ft-default-stopword-table.html

第4章　　評価実験

本章では本論文で提案するアンカーテキスト検索モデルの有効性を評価するために行った評価実験について述べる。節 4.1 には検索モデルを評価する方法と、実際の評価に利用した評価指標について説明する。節 4.2 には評価するために利用したテストコレクションについて説明する。節 4.3 は提案した検索モデルの幾つかのバージョンと関連研究の提案モデルを評価対象として紹介し、比較評価により求められる結果について説明する。節 4.4 は実際の評価結果を示し分析を行う。

4.1　　評価方法

情報検索領域の研究は大まかに二種類に分けられる。一つは、システム側の研究で、検索モデルの研究はこのタイプに属している。システム側の研究は人間の主観思考を中心としなく、客観的な正解を求めることに集中している。言い換えると、システム側の研究には正解が存在し、この正解に接近するために努力を注ぐ。逆に、インタラクション側の研究は、人間の主観的な思いを一層深く理解するための研究である。このタイプの研究は、正解はなく、人間の情報検索において行う行動や、裏側の考えを明らかにするため努めている。

検索モデルの研究はシステム側の研究として、検索モデルの評価には正解がある。すなわち、あるクエリに対して、関連する文書は所与のものであり、各関連文書の関連度も判定されている。検索モデルの使命はより多くの関連文書を探して、関連度が高い文書を検索結果の上位にランクすることである。そして、評価指標により、検索モデルが自分の使命をどの程度完成したのかを測定する。

検索モデルの評価指標は多種多様である。例えば、MAP[15]、nDCG[16]、MRR[17]などがある。それ以外も色んな評価指標がある。評価指標はそれぞれ評価する中心も異なる。MAP は再現率を中心にした評価指標である一方、MRR は精度を中心にした評価指標である。従って、検索モデルを評価したい方面により評価指標を選ぶ必要がある。本論文は情報型クエリのためのアンカーテキスト検索モデルであるため、MAP と nDCG 二つの評価指標を使用する。これから、MAP と nDCG について簡単に説明する。

まずは MAP について説明する。MAP(mean average precision)は平均精度 AP の検索要求セットに関する平均である。AP は二値適合性基づく情報検索評価指標であるため、クエリに対して全ての文書は適合と非適合に判定される。AP の定義は式（５）に示す。

$$AP_{@k} = \frac{1}{k}\sum_{r=1}^{k} I(r)Prec(r) \qquad (5)$$

ここで、I(r)は第 r 位の文書が適合文書であるか否かを表すフラグで、Prec(r)は第 r 位の文書の精度である。すなわち、平均精度は、検索結果中に適合文書が見つかるたびにその時点の精度を計算し合計して、それを k に関して平均したものである。

続いて、多値適合性に基づいた情報検索評価指標 nDCG について説明する。nDCG の特徴は検索された適合文書のランクが上位にあるほど、その価値が上がり、適合性が高い文書のランクが上位にあるほど、その価値が上がると考えることである。nDCG は、文書の適合性グレードに応じた価値のことを利得という。利得の計算は主に２つの手法があるが、本論文はrel_iで利得の計算をする。そして、第 r 位における減損利得は、第 r 位の利得を$log_2(r+1)$で割ることで求める。nDCG の計算方法は式（６）に示す。

$$nDCG_{@p} = \frac{DCG_p}{iDCG_{@p}} = \frac{\sum_{i=1}^{p} \frac{rel_i}{log_2(i+1)}}{\sum_{i=1}^{|REL|p} \frac{rel_i}{log_2(i+1)}} \quad (6)$$

これから nDCG の計算例を示す。まず、高適合文書 2 件と部分適合文書 2 件を持つ検索クエリがある。高適合の利得を 2 とし、部分適合の利得を 1 とする。(a)の検索結果は高適合文書が第 1 位に、部分適合文書が第 4 位にランクされており、(c)の結果は部分適合文書が第 1 位に高適合文書が第 4 位にランクされている。多値適合性を考慮せずに部分適合以上を全て適合と見なして評価すると、(a)と(c)は同等と見なされてしまう。しかし、nDCG を使って評価すると両方の優劣が明らかに判別される。図 7 は nDCG の計算の例である。

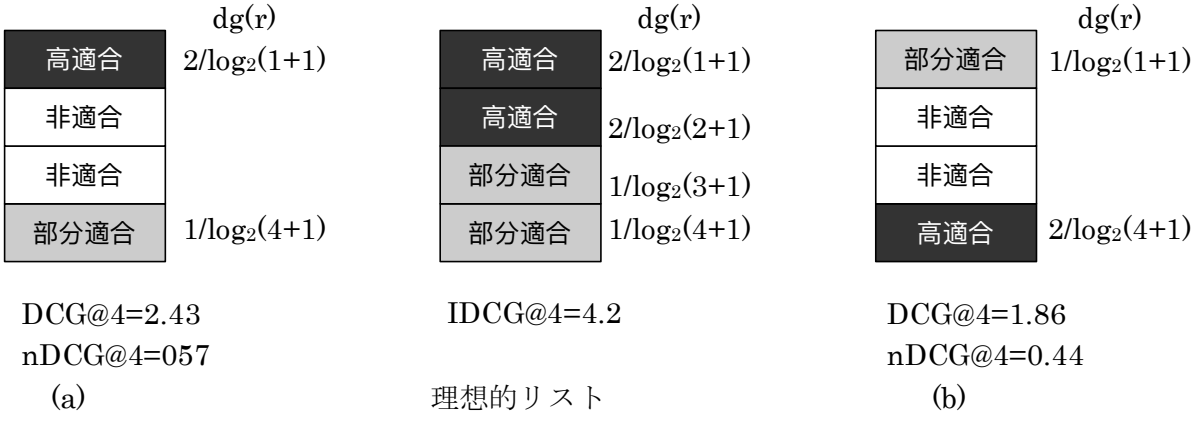

図 7　nDCG 計算例

4.2　テストコレクション

節 4.1 に述べたように、検索モデルを評価するためには、正解が必要である。すなわち、幾つかのクエリをあげて、すべてのクエリに対する文書の適合判定結果が必要である。テストコレクションはこのような正解を提供するものである。テストコレクションの選択に一番重要なのは、テストコレクションの文書適合判定を行う文書データは評価する検索モデルが使っている文書データと一致していることである。文書データが少しでも異なると検索モデルの評価の意味はない。

テストコレクションの正解を決める方法は多種多様であるが、本論文で使うテストコレクションは、情報検索領域で評価型国際会議である TREC[7]で作られた。本論文は TREC14WebTrack[8]のテストコレクションを使う。TREC14WebTrack は ClueWeb12 を文書データとして使って、ad hoc タスクと risk sensitive タスクがある。本論文の提案モデルは ad hoc タスクに該当する。TREC14WebTrack の両方のタスクは同じ 50 個のトピックを使う。1 個のトピックは、topic number、query、description、type 情報が必ず含まれていて、下記のような幾つかのトピックには subtopic の情報も含まれている。

- topic number：トピックの番号、クエリの番号に相当
- query：トピックのクエリ、検索する際に実際に入力するクエリ
- description：トピックに対する説明

7 https://trec.nist.gov/overview.html

8 https://trec.nist.gov/data/web2014.html

- type : single と faceted がある。single はサブトピックがなく、faceted はサブトピックがある。
- subtopic : サブトピックは複数あるが、トピックに対するユーザの異なる潜在的のニーズを表す。各サブトピックには type 属性があるが、inf と nav 二つの値が該当される。これは、サブトピックが情報型であるか、案内型であるかを示している。inf は情報型、nav は案内型である。

そして、図 8 と図 9 はトピックの type が single と faceted である時の例をあげる。

```
<topic number="251" type="single">
 <query>identifying spider bites</query>
 <description>
    Find data on how to identify spider bites.
 </description>
</topic>
```

図 8　type が single であるトピックの例

```
<topic number="253" type="faceted">
 <query>tooth abscess</query>
 <description>
    What treatments are available for a tooth
abscess?
 </description>
 <subtopic number="1" type="inf">
    What treatments are available for a tooth
abscess?
 </subtopic>
 <subtopic number="2" type="inf">
    What are the dangers/complications of leaving a
tooth abscess untreated?
 </subtopic>
 <subtopic number="3" type="inf">
    What are the concerns with extracting an
abscessed tooth?
 </subtopic>
 <subtopic number="4" type="inf">
    Which antibiotics are used to treat a tooth
abscess.
 </subtopic>
</topic>
```

図 9　type が faceted であるトピックの例

TREC14WebTrack のテストコレクションは二種類の適合性判定結果を提供している。第一種類は judgment file containing one aspect per topic という適合性判定結果で、この適合性判定結果は、サブトピックを考慮せず、トピックの description により適合性判定を行なっ

た。第二種類は、file containing all judgments という適合性判定結果で、各トピックにおいて全てのサブトピックに対して適合性判定を行なっていた。

TREC14WebTrack の ad hoc タスクにはランが提出されるまでに、クエリのキーワードのみが提供され、フルトピック情報は提供されない。従って、本論文は 50 個クエリだけ使用し、クエリのフルピックの情報は取り扱わない。そして、TREC14WebTrack によると ad hoc タスクに対して、文書の適合性判定は description により行なっていてと示し、第一番目のサブトピックは description と同様だと示した。従って、ad hoc タスクの評価は、テストコレクションの適合性判定結果の第一種類の judgment file containing one aspect per topic を使う。

そして、文書の適合性判定は以下のような 6 段階により行なっている。

- Nav（4）：このページはクエリが代表している実体のホームページである。
- Key（3）：このページは権威的、総合的なページである。
- HRel（2）：このページはかなり重要な情報を提供している。
- Rel（1）：このページは少し関連する情報を提供している。
- Non（0）：このページは有用な情報を提供していない。
- Junk（-2）：このページは全然関係がなく、その内容はゴミである。

テストコレクションのクエリに対する詳細の情報は表 3 と表 4 に示す。表 3 はテストコレクションの 50 個のトピックのクエリである。その中で、26 個のトピックの type が faceted であり、他の 24 個のトピックの type が single である。TREC14WebTrack により、ad hoc タスクに対して、第一番目のサブトピックは description と同一のものであるため、クエリの情報型と案内型の分類は第一番目のサブトピックの type により決める。もし type が inf になったらこのクエリは情報型であり、type が nav になったらクエリは案内型である。しかし、トピックの type が single であり、サブトピックがない場合は、そのトピックのクエリを情報型と見なす。そして、26 個の type が faceted であるトピックの中、23 個のトピックの第一番目のサブトピックの type が inf であって、3 個のトピックの第一番目のサブトピックの type が nav であった。従って、50 個のトピックのクエリで、三つのクエリ Q10、Q23、Q28 が案内型クエリである。しかし、Q10、Q23、Q28 のほかのサブトピックを見ると type が inf である場合も複数あるので、今回は Q10、Q23、Q28 も情報型クエリと見なす。表 4 は各クエリの適合判定結果の中で 6 グレードの各個数を示している表である。

TREC14WebTrack は研究者達の提案モデルの性能を表すためにベースラインを提供している。ベースラインはコンテンツ検索モデルである Indri[9]を使い、50 個のトピックに対する検索した検索結果である。

9 https://www.lemurproject.org/indri.php

表 3 テストコレクションのクエリ

Q1: identifying spider bites	Q2: history of orcas island	Q3: tooth abscess	Q4: barret's esopagus	Q5: teddy bears
Q6: patron saint of mental illness	Q7: holes by louis sachar	Q8: hip roof	Q9: carpenter bee	Q10: the american revolutionary
Q11: folk remedies sore throat	Q12: balding cure	Q13: evidence for evolution	Q14: tribe formerly living in alabama	Q15: F5 tornado
Q16: symptoms of heart attack	Q17: feliz navidad lyrics	Q18: benefits of running	Q19: marshall county schools	Q20: sun tzu
Q21: halloween activities of middle school	Q22: dreams interpretation	Q23: wilson's disease	Q24: golf instruction	Q25: uss cole
Q26: how has african american music influence history	Q27: bewitched cast	Q28: mister rogers	Q29: game theory	Q30: view my internet history
Q31: ketogenic diet	Q32: nasa interplanetary missions	Q33: hayrides in pa	Q34: where to find moral mushrooms	Q35: magnasium rich foods
Q36: common schizophphrenia drugs	Q37: carotid cavernous fistula treatment	Q38: fidel castro	Q39: benefits of yoga	Q40: norway spurce
Q41: sangre de cristo mountains	Q42: history of the electronic medical record	Q43: educational advantages of social networking sites	Q44: flowering plants	Q45: how to tie a windsor knot
Q46: recycling lead acid batteries	Q47: altitude sickness	Q48: medical care and jehovah's witness	Q49: pink slime in ground beef	Q50: how to find mean

表 4 クエリの各グレード判定個数

Q1 Nav: 0 Key: 0 HRel: 34 Rel: 147 Non: 93 Junk: 0	Q2 Nav: 0 Key: 1 HRel: 9 Rel: 112 Non: 154 Junk: 10	Q3 Nav: 0 Key: 10 HRel: 67 Rel: 69 Non: 41 Junk: 27	Q4 Nav: 0 Key: 4 HRel: 22 Rel: 107 Non: 109 Junk: 3	Q5 Nav: 0 Key: 0 HRel: 3 Rel: 2 Non: 353 Junk: 46
Q6 Nav: 0 Key: 9 HRel: 8 Rel: 72 Non: 198 Junk: 0	Q7 Nav: 1 Key: 4 HRel: 42 Rel: 58 Non: 91 Junk: 16	Q8 Nav: 0 Key: 0 HRel: 20 Rel: 50 Non: 156 Junk: 35	Q9 Nav: 0 Key: 0 HRel: 2 Rel: 110 Non: 135 Junk: 0	Q10 Nav: 0 Key: 4 HRel: 10 Rel: 10 Non: 302 Junk: 2
Q11 Nav: 0 Key: 0 HRel: 3 Rel: 18 Non: 259 Junk: 1	Q12 Nav: 0 Key: 0 HRel: 5 Rel: 205 Non: 114 Junk: 0	Q13 Nav: 0 Key: 0 HRel: 8 Rel: 145 Non: 171 Junk: 0	Q14 Nav: 0 Key: 0 HRel: 10 Rel: 60 Non: 141 Junk: 101	Q15 Nav: 1 Key: 3 HRel: 1 Rel: 16 Non: 208 Junk: 4
Q16 Nav: 4 Key: 10 HRel: 156 Rel: 71 Non: 101 Junk: 9	Q17 Nav: 0 Key: 41 HRel: 6 Rel: 38 Non: 64 Junk: 89	Q18 Nav: 0 Key: 4 HRel: 54 Rel: 73 Non: 200 Junk: 0	Q19 Nav: 1 Key: 6 HRel: 5 Rel: 15 Non: 289 Junk: 0	Q20 Nav: 0 Key: 2 HRel: 16 Rel: 48 Non: 175 Junk: 9
Q21 Nav: 0 Key: 0 HRel: 3 Rel: 12 Non: 248 Junk: 3	Q22 Nav: 0 Key: 4 HRel: 34 Rel: 189 Non: 122 Junk: 18	Q23 Nav: 5 Key: 22 HRel: 0 Rel: 13 Non: 199 Junk: 6	Q24 Nav: 0 Key: 6 HRel: 19 Rel: 75 Non: 207 Junk: 9	Q25 Nav: 0 Key: 0 HRel: 21 Rel: 3 Non: 194 Junk: 4
Q26 Nav: 0 Key: 17 HRel: 27 Rel: 155 Non: 92 Junk: 28	Q27 Nav: 0 Key: 1 HRel: 29 Rel: 32 Non: 219 Junk: 0	Q28 Nav: 0 Key: 0 HRel: 1 Rel: 3 Non: 222 Junk: 3	Q29 Nav: 0 Key: 6 HRel: 12 Rel: 80 Non: 181 Junk: 6	Q30 Nav: 0 Key: 1 HRel: 41 Rel: 142 Non: 217 Junk: 0
Q31 Nav: 0 Key: 0 HRel: 11 Rel: 94 Non: 120 Junk: 0	Q32 Nav: 0 Key: 0 HRel: 1 Rel: 201 Non: 132 Junk: 0	Q33 Nav: 0 Key: 10 HRel: 17 Rel: 69 Non: 123 Junk: 1	Q34 Nav: 0 Key: 10 HRel: 77 Rel: 142 Non: 39 Junk: 9	Q35 Nav: 1 Key: 10 HRel: 130 Rel: 74 Non: 60 Junk: 9
Q36 Nav: 0 Key: 5 HRel: 54 Rel: 64 Non: 192 Junk: 5	Q37 Nav: 0 Key: 7 HRel: 26 Rel: 14 Non: 117 Junk: 11	Q38 Nav: 0 Key: 1 HRel: 44 Rel: 98 Non: 106 Junk: 10	Q39 Nav: 0 Key: 2 HRel: 24 Rel: 21 Non: 340 Junk: 8	Q40 Nav: 0 Key: 0 HRel: 0 Rel: 30 Non: 241 Junk: 18
Q41 Nav: 0 Key: 1 HRel: 8 Rel: 13 Non: 203 Junk: 0	Q42 Nav: 0 Key: 4 HRel: 219 Rel: 85 Non: 56 Junk: 5	Q43 Nav: 0 Key: 9 HRel: 54 Rel: 58 Non: 244 Junk: 13	Q44 Nav: 0 Key: 0 HRel: 167 Rel: 94 Non: 159 Junk: 3	Q45 Nav: 0 Key: 0 HRel: 33 Rel: 52 Non: 130 Junk: 0
Q46 Nav: 0 Key: 10 HRel: 21 Rel: 125 Non: 75 Junk: 19	Q47 Nav: 0 Key: 0 HRel: 1 Rel: 163 Non: 54 Junk: 10	Q48 Nav: 20 Key: 6 HRel: 52 Rel: 73 Non: 92 Junk: 6	Q49 Nav: 0 Key: 0 HRel: 1 Rel: 132 Non: 108 Junk: 0	Q50 Nav: 0 Key: 0 HRel: 6 Rel: 56 Non: 395 Junk: 0

4.3 評価対象

本論文で提案するアンカーテキスト検索モデルの評価では、前節で説明したテストコレクションのクエリを用いて検索実験を行い、複数のバージョンの検索モデルと手法を nDCG で比較する。テストコレクションはグレード付き適合判定を行なっていたので nDCG をメイン評価指標とする。混乱を避けるために、まずは本実験において明らかにすべき点を整理する。

- RQ1：アンカーテキストは情報型クエリに対して有用であるか。
- RQ２：文書の関連スコアを計算において、アンカーテキストのスコアを計算する時、アンカーテキストの重要度により重み付けて取り扱うのは有効であるか。
- RQ3：サイト間リンクとサイト内リンクのアンカーテキストの情報型クエリに対する有用性は異なるか。
- RQ4：コンテンツ検索モデルと比べると、アンカーテキスト検索モデルの利点と欠点は何か。

以上の質問を念頭において、本実験で比較する 25 手法を紹介する。以下でサイト間或いはサイト内リンクだけ使うアンカーテキスト検索モデルをサイト間モデル或いはサイト内モデルと呼ぶ。

- A1：サイト間モデル、重みなし
- A2：サイト間モデル、重みあり
- A3：サイト内モデル、重みなし
- A4：サイト内モデル、重みあり
- $A5_{alpha0.1}$〜$A5_{alpha0.9}$：サイト間とサイト内の合併モデル、重みなし。βが 0.1 から 0.9 までの全部のモデル。
- $A6_{alpha0.1}$〜$A6_{alpha0.9}$：サイト間とサイト内の合併モデル、重みあり。βが 0.1 から 0.9 までの全部のモデル。
- F1：Fujii の提案モデル、サイト間モデル
- F2：Fujii の提案モデル、サイト内モデル
- B1：テストコレクションのベースライン(Indri)

4.3.1 Fujii のアンカーモデル

Fujii の提案モデルは、クエリに対する文書の関連度スコアを計算することではなく、クエリ q に対する文書 d が検索される確率 P(d|q)を計算し、確率により順位をつける。アンカーテキストにおけるクエリタームの頻度分布を用いて P(d|q)を推定する。ベイズ定理を用いて P(d|q)を式（７）のように変形する[4]。

$$P(d|q) = \frac{P(q|d) \times P(d)}{p(q)} \propto P(q|d) \times P(d) \qquad (7)$$

式（７）において P(q)は全ページに共通の定数であり、ページ間の相対的な順序に影響を与えないので無視する。P(d)は、クエリと関係なく、ページ d が選択される確率である。Fujii は P(d)の計算方法として、文書 d の Pagerank をつかう方法と、文書 d へのリンク数とウェブコレクションにおけるリンク総数の割合、すなわち、最尤推定によって計算する方法を使った。今回の評価実験は P(d)の計算方法として Pagerank の値だけを使う。そして P(q|d)の計算は式（８）に示す。

$$P(q|d) = \prod_{t \in q} P(t|d) = \prod_{t \in q} \sum_{a \in A_d} P(t|a) \times P(a|d) \qquad (8)$$

P(t | d)は、ページ d にリンクしている 1 つ以上のアンカーテキストから無作為に一つのタームを選択した時に、これが t である確率である。P(a | d)は、ページ d にリンクする際によく使われるアンカーテキストに対して大きな値をとる。P(t | a)は、アンカーテキスト a でよく使われるタームに対して大きな値をとる。P(t | d)の計算は、ターム t が文書 d のアンカーテキスト文書 A_d 中の各アンカーテキストに出現する確率とこのアンカーテキストがアンカーテキスト文書 A_d に出現する確率をかけて全てのアンカーテキストに合計する。図 10 は P(q | d)の計算例である。

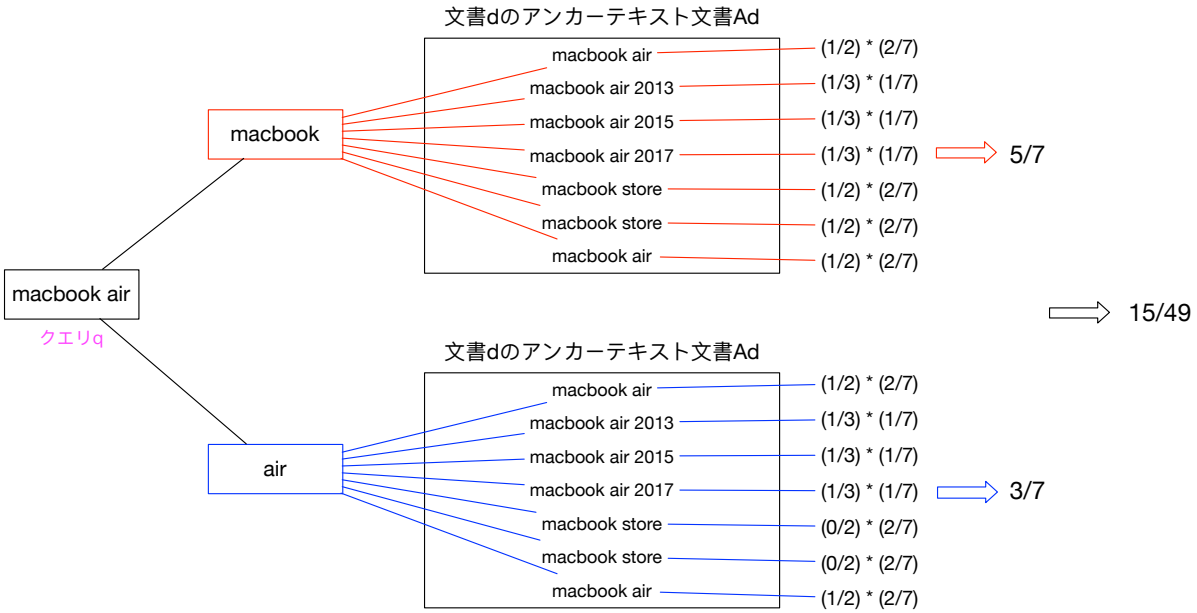

図 10　P(q | d)の計算例

例に示したようにクエリ q のタームが macbook と air であり、各タームの確率 P(t | d)はそれぞれ$\frac{5}{7}$と$\frac{3}{7}$である。従って、P(q | d)は$\frac{15}{49}$である。そして、P(d | q)は P(q | d)に文書 d の Pagerank 値をかけた点数である。

4.4　評価結果

前節で紹介した 25 手法について nDCG@10、nDCG@20、MAP@10、MAP@20 の評価を行なった結果を表 5 に示す。MAP の評価を行う際、文書の関連度グレードが 1 以上の文書を関連文書と見なした。

表 5 25 手法の nDCG@10、nDCG@20、MAP@10、MAP@20 スコア

	nDCG@10	nDCG@20	MAP@10	MAP@20
A1	0.1814	0.1508	0.1146	0.0711
A2	0.1939	0.1637	0.1186	0.0759
A3	0.2126	0.1945	0.1889	0.1411
A4	0.1856	0.1684	0.1642	0.1106
$A5_{alpha0.1}$	0.2200	0.1981	0.1926	0.1422
$A5_{alpha0.2}$	0.2313	0.2063	0.1962	0.1441
$A5_{alpha0.3}$	0.2261	0.2048	0.1960	0.1460
$A5_{alpha0.4}$	0.2284	0.2035	0.1953	0.1424
$A5_{alpha0.5}$	0.2293	0.2047	0.1905	0.1415
$A5_{alpha0.6}$	0.2519	0.2235	0.1905	0.1405
$A5_{alpha0.7}$	0.2448	0.2189	0.1869	0.1398
$A5_{alpha0.8}$	0.2491	0.2202	0.1805	0.1343
$A5_{alpha0.9}$	0.2288	0.2000	0.1607	0.1187
$A6_{alpha0.1}$	0.1912	0.1706	0.1688	0.1127
$A6_{alpha0.2}$	0.2007	0.1776	0.1777	0.1173
$A6_{alpha0.3}$	0.1985	0.1779	0.1725	0.1172
$A6_{alpha0.4}$	0.2071	0.1810	0.1746	0.1171
$A6_{alpha0.5}$	0.2202	0.1893	0.1767	0.1181
$A6_{alpha0.6}$	0.2316	0.1998	0.1747	0.1169
$A6_{alpha0.7}$	0.2352	0.2012	0.1693	0.1181
$A6_{alpha0.8}$	0.2218	0.1999	0.1518	0.1136
$A6_{alpha0.9}$	0.2196	0.1982	0.1475	0.1052
F1	0.0153	0.0127	0.0081	0.0045
F2	0.1054	0.0915	0.0979	0.0629
B1	0.2277	0.2475	0.2862	0.2780

表 5 のスコアは全てのクエリに対して評価を行い、評価結果を平均した結果である。

まず、nDCG の評価結果から説明する。本論文で提案する全てのモデルに対して、nDCG@10 の評価結果が nDCG@20 の評価結果より高い。すなわち、全てのモデルは 10 位から 20 位までの検索結果は上位 10 位までの検索結果より検索された関連文書数が少ない。サイト内モデルの 2 つの手法 A3 と A4 どちらもサイト間モデルの二つの手法 A1 と A2 より評価結果が良い。アンカーテキストの重み付けの有無に関わらずサイト内モデルはサイト間モデルより評価結果が良いとうことである。そして、サイト内モデルにおいて、nDCG@10 と nDCG@20 両方の評価結果が重みなしのモデルが重みありのモデルより高い。

合併モデル A5 と A6 は nDCG@10 と nDCG@20 において、両方共βとアンカーテキストの重み付けに関わらず全ての手法がサイト内モデルとサイト間モデル A1、A2、A3、A4 より評価結果が良い。すなわち、サイト内モデルとサイト間モデルを合併すると必ず個別のモデルより性能が高い。A5 と A6 において、alpha の値が同じである場合、nDCG@10 と nDCG@20 共に、A5 の評価結果はいつも A6 より良い。つまり、合併においても、アンカーテキストに重み付けない方がもっと性能が高いことを示している。nDCG@10 と nDCG@20 共に、A 5

において、alpha が 0.6 の値をとる時 nDCG の評価が一番良くて、A6 には、alpha が 0.7 の値をとる時 nDCG の評価結果が一番良い。そして、A5 と A6 両方の alpha による評価結果は同じパターンを持っている。その両方の手法の評価結果は、alpha が 0.1 から徐々上がって、それぞれ 0.6 と 0.7 になった時ピークになり、alpha が 0.9 になるまでだんだん下げって行く。A5 の評価が一番良い、alpha が 0.6 をとる時の評価結果と、A6 の評価が一番良い、alpha が 0.7 の値をとる時の評価結果を比較すると A5 の評価結果がもっと良い。

Fujii の手法である F1 と F2 は nDCG@10 と nDCG@20 において、A1 から A6 までの全ての手法より評価結果がよくなくて、評価点数の差もかなり大きい。F1 と F2 の評価結果を比較すると F1 の評価結果が F2 よりかなり高いことである。

本論文で提案するアンカーテキスト検索モデルとコンテンツモデルであるベースラインの評価結果を見ると、nDCG@10 において、全ての重みなしの合併モデルがベースラインより評価結果が良いし、重みありの合併モデルの中でも 2 つのモデルはベースラインより結果が良い。nDCG@20 の場合は、ベースラインの評価結果が全ての提案モデルより評価結果が良いが、この差は大きくない。

続いて MAP の評価結果を述べる。MAP の結果は nDCG とほぼ同様である。サイト内モデルの 2 つの手法 A3 と A4 はサイト間モデルの 2 つの手法 A1 と A2 より評価結果が良い。そして、アンカーテキストの重み付けに関わらずサイト内モデルはサイト間モデルより評価結果が良い。サイト間モデルは MAP@10 と MAP@20 両方が、アンカーテキストに重み付けた方が重み付けてない方より評価結果が良い。逆に、サイト内モデルは MAP@10 と MAP@20 両方が、重み付けてない方が重み付けた方より評価結果が良い。

nDCG と異なり、MAP はサイト内モデルが合併モデルの手法より評価結果が良い場合もある。MAP@10 と MAP@20 共に、重みづけたないサイト内モデル A3 は合併モデル A6 の全てのモデルより評価結果が高い。MAP は文書の関連度グレードに関わらず、単に上位のランクに置いている関連文書数に影響されるため、サイト内モデルは A6 の全てのモデルより関連文書の探す性能が高いと言える。サイト間モデルはアンカーテキストの重み付けとβに関わらず全ての合併モデルより評価結果がよくない。MAP の評価も nDCG と同様に、A5 と A6 において、alpha の値が同じである場合、A5 の評価結果はいつも A6 より高い。A5 は alpha の値が 0.2 をとるときに評価結果が一番高いし、A6 は alpha の値が 0.5 と 0.7 をとるときに評価結果が一番高い。A5 の評価が一番高い手法と A6 の評価が一番高い手法を比べると、A5 の評価がもっと良い。

本論文で提案するアンカーテキスト検索モデルとコンテンツモデルであるベースラインの MAP 評価結果を見ると、MAP@10 と MAP@20 両方において、ベースラインの評価結果が全ての提案モデルより高く、この差も大きい。

図 11 は nDCG@10 において A2、A3、$A5_{alpha0.6}$ の 50 個クエリの nDCG@10 の評価結果である。図の横座標はクエリ番号を示していて、縦座標は nDCG@10 の評価点数を示している。

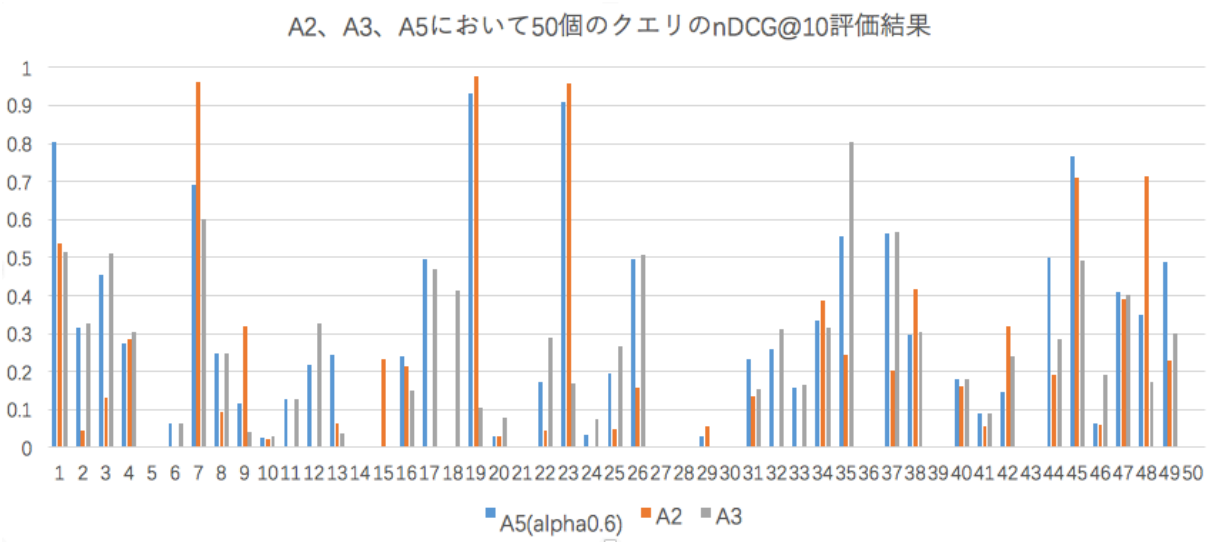

図 11　A2、A3、A5 において 50 個のクエリの nDCG@10 評価結果

A2、A3、$A5_{alpha0.6}$ 3 つのモデルを選んだ理由は、同じタイプのモデルで評価が一番高いためである。そして、nDCG@20 の結果は nDCG@10 の結果とほぼ同じパターンを持っているため本節で特に示さない。

サイト間モデル A2 において 50 個のクエリの nDCG@10 評価結果を見ると、評価点数が 0.1 以上になっているクエリが 22 個ある。その中で、0.5 以上は 6 個ある。評価点数が 0.1 以下になっている 28 個のクエリのうち 18 個の評価点数が 0 になっている。クエリの評価点数が 0 になっているのは、このクエリの検索結果のランク 10 位まで関連文書が 1 つもないことを示す。しかし、0.9 以上のクエリは 3 つあり、A3 と A5 より多いし、評価結果が一番高いモデルも A2 である。

続いて、サイト内モデル A3 において 50 個のクエリの nDCG@10 評価結果を見ると、評価点数が 0.1 以上になっているクエリが 31 個ある。これは A1 より 9 個多いの数値である。その中で、0.5 以上は 6 個ある。この数値は A2 と同じである。評価点数が 0.1 以下になっている 19 個のクエリのうち 12 個の評価点数が 0 になっている。これは A2 より 6 個少ない数値である。

合併モデルである A5 において 50 個のクエリの nDCG@10 評価結果については、評価点数が 0.1 以上になっているクエリが 31 個で A3 と同じである。その中で、0.5 以上も 7 個あって、一番多い。評価点数が 0.1 以下になっている 19 個のクエリのうち 12 個の評価点数が 0 以下になっている。これは A3 と同じ数値である。

合併モデル A5 は A1 と A3 を合併したものである。図 11 を見ると、A2、A3 と A5 が全部 0 より高い 28 個のクエリに対して、A5 の値が A2 と A3 両方より高いクエリは 6 個、A5 が A1 と A3 の間に値をとるクエリは 19 個、A5 の値が A1 と A3 両方より低いクエリは 3 個ある。

図 12 は Fujii のサイト間モデル F1 が 50 個のクエリに対して nDCG@10 評価を行なった結果を示す。

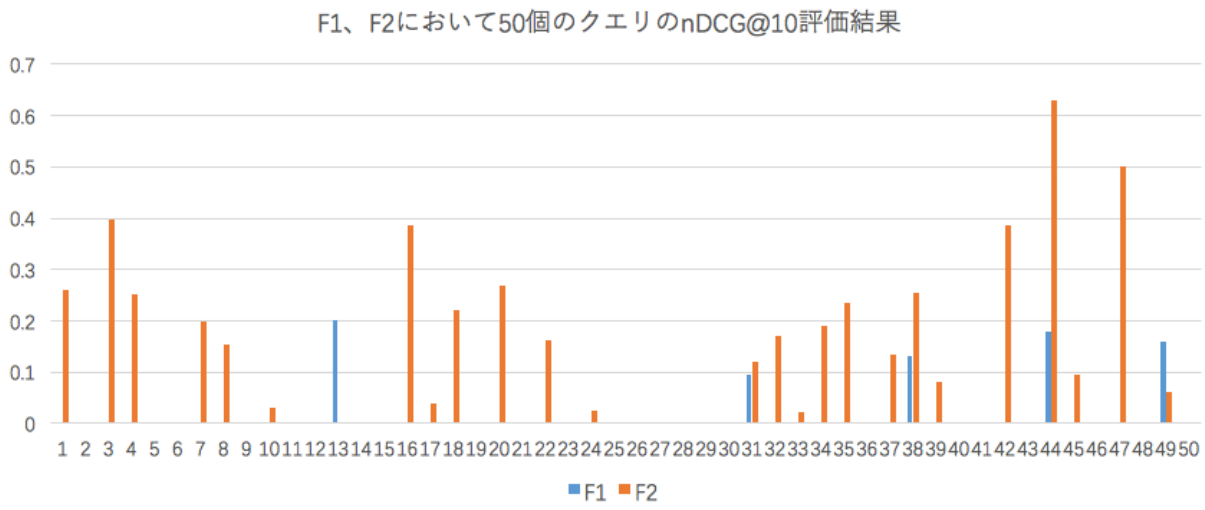

図 12　F1 において 50 個のクエリの nDCG@10 評価結果

図 12 を見ると、評価点数が 0 になっているクエリがすごく多い。これは前で述べたように、クエリの検索結果のランク 10 位まで関連文書が 1 つもないことを示す。サイト間モデル F1 において 0 点より高いクエリは 5 つしかなく、一番高い点数も 0.2 点しか取っていない。すなわち、検索結果のランク 10 位まで関連文書が検索されてもその個数が少ないことを示している。サイト内モデル F2 において、0 点より高いクエリは 25 個あり、F1 よりかなり高い数値である。

図 13 は MAP@10 において A2、A3、$A5_{alpha0.2}$ の 50 個クエリの MAP@10 の評価結果である。ここで、クエリの MAP@10 はそのクエリの AP@10 と相当する。は図の横座標はクエリ番号を示していて、縦座標は MAP@10 の評価点数を示している。

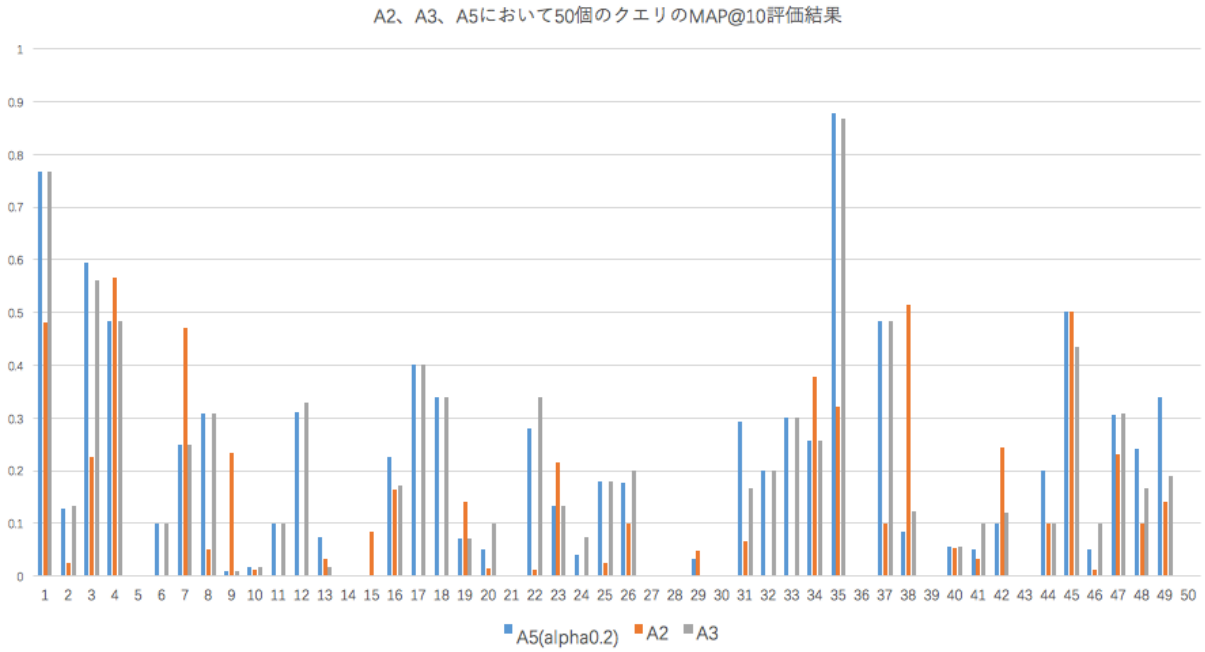

図 13　A2、A3、A5 において 50 個のクエリの MAP@10 評価結果

A2、A3、$A5_{alpha0.2}$はそれぞれ同じタイプのモデルで評価が一番高いため比較対象として選ばれた。

サイト間モデル A2 において 50 個のクエリの MAP@10 評価結果を見ると、評価点数が 0.1 以上になっているクエリが 19 個あり、その中で、0.4 以上は 5 つ、0.5 以上は 2 つあって、nDCG@10 の結果とほぼ同じである。評価点数が 0.1 以下になっているクエリが 31 個、評価点数が 0 になっているクエリが 18 個で、これも nDCG@10 の結果と相当する。評価点数が 0 になっているクエリ番号としては両方相当するが、検索結果の上位 10 位まで関連文書が検索されないと、nDCG と MAP 両方のスコアが 0 点になるべきである。そして、文書の適合判定が 0 と 1 にしているのでマイナスの評価点数がない。nDCG@10 の評価結果がマイナスであるクエリ 8 と 25 を見ると、両方とも 0 以上になっている。これは、2 つのクエリの検索結果に上位 10 位まで関連文書が検索されたが、ゴミ文書がより多く検索されたとか、ゴミ文書のランクが関連文書より高かったことである。

続いて、サイト内モデル A3 において 50 個のクエリの MAP@10 評価結果を見ると、評価点数が 0.1 以上になっているクエリが 32 個あり、その中で、0.4 以上は 7 つ、0.5 以上は 4 つあって、nDCG@10 の結果より高い方である。評価点数が 0.1 以下になっている 17 個で、nDCG@10 より 9 個少ない。しかし、点数が 0 以下になっているのは 12 個で nDCG@10 と同じ結果である。nDCG@10 に評価点数がマイナスであるクエリ 14 と 36 が MAP@10 において 0 点になっている。すなわち、2 つの検索結果の上位 10 位まで検索された関連文書はなく、ゴミ文書しかないことである。

合併モデルである A5 の MAP@10 の評価結果で、評価点数が 0.1 以上になっているクエリが 28 個 A3 より 4 個少ない。0.4 以上は 7 つ、0.5 以上は 4 つあって A3 と同じである。評価点数が 0.1 以下になっている 21 個で、A3 より 4 個多い。A2、A3 と A5 が全部 0 点より高い 30 個のクエリに対して、A5 の値が A2 と A3 両方より高いクエリは 7 個、A5 が A2 と A3 の間に値をとるクエリは 21 個、A5 の値が A2 と A3 両方より低いクエリは 2 個ある。

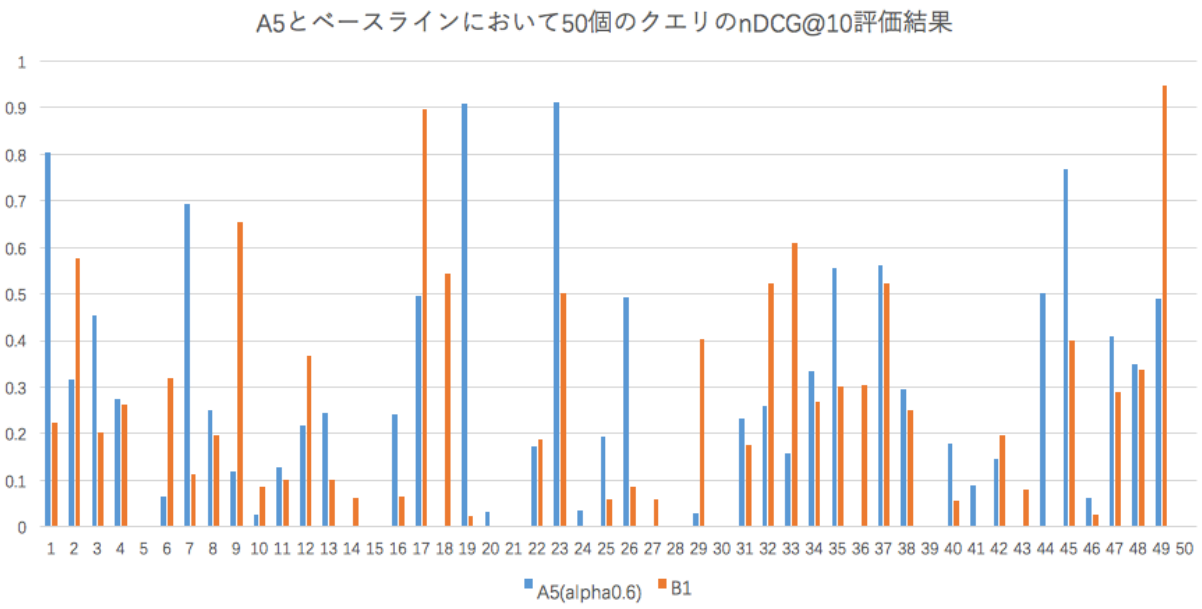

図 14 A5 とベースラインにおいて 50 個のクエリの nDCG@10 評価結果

図 14 は、本論文で提案するモデルの中で評価結果が一番高い A5(alpha0.6)とベースラインにおいて 50 個のクエリの nDCG@10 評価結果を示している。A5 において評価点数が 0.1 以上になっているクエリは 31 個、0.5 以上のクエリは 8 個、0.9 以上のクエリは 2 つ、0 になっているクエリは 12 個である。ベースラインの評価結果をみると、評価点数が 0.1 以上になっているクエリは 29 個で、A5 より少ない。評価点数が 0 .5 以上のクエリは 8 個で A5 と同一である。評価点数が 0.9 以上のクエリは 1 個で A5 より少ないが、評価点数が一番高いモデルは B1 である。評価点数が 0 になっているクエリは 11 個で A5 より少ない。A5 と B1 両方において評価点数が 0 になっているクエリは 7 個である。A5 の評価点数が B1 より高いクエリは 23 個であり、B1 の評価点数が A5 より高いクエリは 16 個ある。

第5章　　考察

本章では、前章で行った評価実験の結果に基づいて、4.3 節に示した５つのリサーチクエスションに応えるための議論を行う。節 5.1 はリサーチクエスチョン関する考察について述べる。節 5.2 は提案モデルの評価点数が０になっている検索に失敗したクエリに関する考察について述べる。5.3 節は本研究でやりたいことができなかったことについて説明する。

5.1　　提案モデルに関する考察

まず、4.3 節に示した５つのリサーチクエスションについて検討する。

RQ１：アンカーテキストは情報型クエリに対して有用であるか。

本論文で提案するアンカーテキスト検索モデルの A1 から A6 までの全ての手法はアンカーテキスト検索モデル同士である Fujii のモデル F1、F2 より評価結果が良い。そして、評価結果をかなり向上したため、アンカーテキスト検索モデルの情報型クエリに対する有効性を向上したと言える。

さらに、本論文で提案するモデルの中で評価結果が一番高い A5 とコンテンツモデルのベースライン B1 と比較すると、nDCG@20 の評価結果は B1 よりよくないが、nDCG@10 の評価結果は B1 より高いため、ある程度アンカーテキストの情報型クエリに対する有用性を示したと言える。

RQ２：文書の関連スコアを計算において、アンカーテキストのスコアを計算する時、アンカーテキストの重要度により重み付けて取り扱うのは有効であるか。

サイト内モデルでは、重みなしの A3 は A4 より結果が良いが、サイト間モデルでは、重みありの A2 は A1 より結果が良い。サイト内リンクは自分が自分に参照するもので、リンクの重み、すなわちリンク元ページの価値は同一であり、サイト間リンクは第三者が自分に参照することで、第三者によりリンク元ページの価値も異なっていると思われる。

合併モデル A5 と A6 において、βの値が相当である場合、nDCG も MAP も全ての重みなしの手法の評価結果が高い。従って、提案モデルのアンカーテキストのスコアの計算において、アンカーテキストの重要度として Pagerank の使用は有効とは言えない。

RQ３：サイト間リンクとサイト内リンクのアンカーテキストの情報型クエリに対する有用性は異なるか。

サイト間モデルとサイト内モデルの評価が一番高い A2 と A3 を比較すると、A3 がより高い。さらに、Fujii のモデルにおいて、サイト内モデル F2 がサイト間モデル F1 より評価結果が高い。従って、情報型クエリに対して、サイト内リンクの有用性はサイト間リンクより高いと言える。

RQ４：コンテンツ検索モデルと比べると、アンカーテキスト検索モデルの利点と欠点は何か。

評価結果が一番高い合併モデルの各クエリの nDCG 評価結果を見ると、50 個のクエリの中 12 個のクエリが 0 点になっている。これは、12 個のクエリに対して検索された適合文書が少ないためである。このような状況が出現した原因で、アンカーテキスト検索モデルの限界に関連するのは２つある。１つは、ウェブページのリンク数である。ウェブページのリンク数はあるウェブページがほかのウェブページにリンクされる回数で、ウェブページのリンク数は十数個から、数十万個までかなり大きなばらつきがある。リンク数が少ないと、アンカーテキストも少ない、クエリとマッチングされる確率も低くなる。マッチングされても、スコアの計算にアンカーテキスト数が少ないため、高い点数が取れなくなり、上位にランクされない。逆に、ウェブページのリンク数が数千個或いは数万個あっても、ウェブページの

関連度スコアを正確に計算できることでもない。これは、２つ目の原因であるスパムの制御不能である。ウェブページにリンクつけるのは誰でもできることで、個人や小規模な団体が行う場合には、自分たちが管理するページ間で相互リンクすることが多いが、これはほぼサイト内リンクである。そして、専門的な組織が行う場合は、サイト内リンクだけでなく、サイト間リンクも大量につけてあげる。しかし、ウェブページのリンクが多いのが単に、大勢の人が特定のウェブページが重要だと思うため、リンクつける場合がある。さらに、クエリによって、あるリンクがスパムであったり、有用なリンクであったりする場合も珍しくない。従って、リンクのスパム分類はすごく難しいことであるため、リンク数が多くても必ず正確に適合判定ができるとは言えない。しかし、図 14 に示したようにアンカーテキスト検索モデルはコンテンツ検索モデルより評価点数が高いクエリが多いである。提案モデル A5 とコンテンツモデル B1 を比較すると、nDCG@10 の評価結果は A5 が良いが、nDCG@20 の評価結果は B1 が良い。すなわち、アンカーテキスト検索モデルは、コンテンツモデルより上位に適合性が高い文書を検索する能力が高いが、コンテンツモデルはより多くの適合文書を検索する能力が高いと思われる。

最後に本論文で提案するモデルと Fujii のモデルの比較結果に対して考察する。

本論文で提案するアンカーテキスト検索モデルは Fujii のアンカーモデルより評価結果が良いことで、本論文で提案するモデルが Fujii のモデルより有効であると言える。Fujii のアンカーモデル B1 は NTCIR–4 と NTCIR–5 のテストコレクションを用いた検索で案内型クエリに評価結果が良かった。しかし、NTCIR–4 と NTCIR–5 のテストコレクションはそれぞれ約 1 千万ウェブページと 1 億ウェブページが含まれているが、7 億ウェブページが含まれている大規模データセット ClueWeb12 に比べるとデータ量が少ない方である。データ量の増加に従って、データノイズも大きくなるため Fujii のモデルは大規模データセットに良く適応できないと思われる。ウェブページとリンクのデータ量の増加に伴い、SEO などのスパムリンクも増えていき、検索質量にかなり大きな影響を与える。Fujii の提案モデルは、サイト内リンクによくある、SEO などの有意でリンク数を加える場合によるスパムを考慮し、サイト内リンクを使用しなかったため、検索される関連文書数がさらに少なくなったと思われる。Q3 に述べたように、サイト内リンクは大きな価値を持っており、サイト間リンクよりも有用性が高い。以上の２つの点が Fujii の提案モデルの評価が良くない理由であると見られる。本論文で提案する検索モデルは、サイト内とサイト間両方のリンクを取り扱う際、スパムリンクを考慮し、アンカーテキストに減損を与えていた。そして、アンカーテキストに減損を与えたことは役立ていると思われる。

5.2 提案モデルの失敗分析

本節では、nDCG の評価が一番良かった合併モデル $A5_{alpha0.6}$ の評価結果を基準として、評価結果の点数が 0 点以下になっているクエリを失敗クエリと見なす、考察する。まず、失敗クエリの情報を表 6 に示す。

表 6 失敗クエリ

クエリ番号	クエリ内容	クエリ情報	
Q5	teddy bears	Nav: 0 Hrel: 3	Key: 0 Rel: 2
Q14	tribe formerly living in alabama	Nav: 0 Hrel: 10	Key: 0 Rel: 60
Q15	F5 tornado	Nav: 1 Hrel: 1	Key: 3 Rel: 16
Q18	benefits of running	Nav: 0 Hrel: 54	Key: 4 Rel: 73
Q21	halloween activities of middle school	Nav: 0 Hrel: 3	Key: 0 Rel: 12
Q27	bewitched cast	Nav: 0 Hrel: 29	Key: 1 Rel: 32
Q28	mister rogers	Nav: 0 Hrel: 1	Key: 0 Rel: 3
Q30	view my internet history	Nav: 0 Hrel: 41	Key: 1 Rel: 142
Q36	common schizophrenia drugs	Nav: 0 Hrel: 54	Key: 5 Rel: 64
Q39	benefits of yoga	Nav: 0 Hrel: 24	Key: 2 Rel: 21
Q43	educational advantages of social networking sites	Nav: 0 Hrel: 59	Key: 9 Rel: 58
Q50	how to find the mean	Nav: 0 Hrel: 6	Key: 0 Rel: 56

クエリの検索が失敗した 1 つの原因は、クエリの適合判定された適合文書が少ないことであると思われる。クエリ Q5、Q15、 Q21、 Q28 は適合文書がそれぞれ 5 個、21 個、15 個、4 個である。この数値はほかのクエリと比べるとかなり少ない方である。特に Q5 と Q28 は適合文書が 5 つくらいしかないので、どう検索モデルを使ってこの 2 つのクエリに対して検索しても、検索結果の評価か高くなることは難しいと思う。

もう 1 つの主な原因は、クエリのステミング問題であると思われる。ステミングというのは、語形の変化を取り除き、同一の単語表現に変換する処理である。特に、英語は単語の変形が多いである。たとえ、swims、swimming、swimmer などの単語から語幹である swim を抽出することである。本論文の検索システムは索引に対しても、クエリに対してもステミングを行なっていなかった。実際に、クエリ Q5 の teddy bears に対して、適合文書がリンクされたアンカーテキストを見ると、ほぼ teddy bear をアンカーテキストとしてリンクされている。teddy bears 以外も benefits、activities、advantages など複数の単語についてステミングを行うことができる。もし、ステミングができたら、検索結果の評価がさらに高くなることが期待される。

最後に、情報検索においてずっと討論されているストップワードの問題。クエリ to be or not to be にストップワード処理を行うと not しか残ってないが、これは検索意図と全然違う

クエリになり、検索する意味がない。Q50 はこの問題と似たような場合である。how to find the mean に対してストップワード処理すると find mean しか残ってないが、これは元の how to find the mean と意味が少し異なっているため、元のクエリの適合文書が検索されないと思われる。

5.3 研究の限界について

本節には、本研究において、本来やりたいと計画したことが、時間や資源の制限のために現時点でできなかったことについて説明する。

まずは、前節の失敗分析で述べた索引とクエリのステミング処理である。ステミング処理ができたら、提案モデルの検索結果の評価は一層高くなれると思うが、英語の単語は非常に多くて、1 つの単語に対しても少なくとも 3、4 種類の変形があるので、ステミングの処理の実装はかなり時間かかる。

そして、クエリの拡張の実現も元のプランだったができなかったことである。クエリの拡張はステミングと異なり、あるクエリと同じ意味を持っている他の表現を用いて検索を行うことである。クエリの拡張により、検索範囲がもっと豊かになるため、リンク数が足りなかったことや、アンカーテキストとクエリがよくマッチングできなかった問題を解決できると思われる。例えば、クエリ Q33、hayrides in pa の中、pa は pennsylvania と略称である。実際の検索には pa を pennsylvania として検索しないので重要な情報が失ってしまうはずである。従って、クエリ拡張の実現によりこのような問題が解決できると提案モデルはもっと有効性が高くなる。

最後は、ウェブページ内容を用いて検索する、既存のコンテンツ検索モデルを実装し、提案するアンカーテキスト検索モデルとの合併モデルを提案する計画であった。コンテンツモデルの合併において、アンカーテキスト検索モデルの限界が削除されることで、評価結果がさらに高くなる検索モデルが作られると思われる。しかし、全てのウェブページのテキスト内容を抽出し、索引を構築することは時間もかかり、実装する際のプログラムのアルゴリズムとハードウェアに要求が厳しいであるため、現時点ではまだ実装されてなかった。

第6章　おわりに

6.1　結論

本論文では、アンカーテキスト検索モデルを提案し、検索システムを構築した。提案したアンカーテキスト検索モデルの有効性を評価するために、検索システムを用いて実際の検索を行なって、検索結果を用いて評価実験を行なた。

情報型クエリ向けの評価指標である nDCG と MAP 二種類の評価方法を用いて、評価実験を行なった結果、提案するモデルの中で、サイト間モデルとサイト内モデルの合併モデルは一番有効性が高かった。アンカーテキストを用いて検索を行う時は、サイト内リンクとサイト間リンク両方使って検索する方の結果がもっと良いことが確認された。そして、両方のリンクを使う時は、サイト内リンクにはサイト間リンクよりちょっとだけ大きく重みつける方が一番有効であることも確認された。

個別のモデルを見ると、サイト内モデルの評価結果がサイト間評価結果より高かった。すなわち、アンカーテキストを用いて検索を行う時は、サイト内のリンクがサイト間のリンクよりもっと有用性が高いことが知られた。そして、両方のリンクを使う時、サイト間リンクに対して重み付けることは有用であるが、サイト内リンクに対して重み付けるのは特に意味がない。

既存の案内型クエリに対して評価結果が良いアンカーテキスト検索モデルより、情報型クエリに対する評価結果が良いことと、コンテンツモデルのベースラインより評価結果が良いことにより、アンカーテキストの情報型クエリに対する有用性がある程度示されたと思われる。

6.2　今後の課題

本研究はアンカーテキストが検索に使用される時、お互いに持っている価値が異なっており、アンカーテキストが持っている価値により取り扱うともっと有効的に検索ができることを証明することが失敗した。本研究は Pagerank しか使ってないため、アンカーテキストの価値を決める他の手法を検討する必要がある。

本研究で提案するアンカーテキスト検索モデルは情報型クエリの検索結果の評価を上がるために構築したモデルであるが、案内型クエリに対しても有効であるかを評価実験に通じて検討する必要がある

本論文の評価実験では、提案する検索モデルを Fujii のアンカーモデルと比較して、アンカーテキスト検索モデルの有効性を考察した。そして、提案モデルをコンテンツモデルのベースラインと比較して、アンカーテキストの情報型クエリに対する有用性を考察した。提案モデルの情報型クエリに対する有効性を一層深く明らかにするためには、より多くの検索モデルと比較実験を行う必要がある。コンテンツモデルである BM25 をベースラインとして比較するとか、TREC WebTrack において他の研究者達が提案した検索モデルとの比較実験を行うことを今後の課題としたい。

参照文献

[1] A. Broder. A taxonomy of web search. ACM SIGIR Forum, 36(2):3-10, 2002.

[2] W. Croft, D. Metzler, T. Strohman. Search Engine Information Retrieval in Practice. Pearson Education, Inc. 2010.

[3] S. Robertson, S. Walker, S Jones, M. Beaulieu and M. Gatford. Okapi at TREC-3. In Proceedings of the Third Text Retrieval Conference. 1994.

[4] A. Fujii. Modeling Anchor Text and Classifying Queries to Enhance Web Document Retrieval. In Proceedings of WWW'08, ACM, 337-346. 2008.

[5] Z. Dou, R. Song, J. Nie and J. Wen. Using Anchor Texts with Their Hyperlink Structure for Web Search. In Proceedings of SIGIR'09, 227-235. 2009.

[6] M. Koolen, J. Kamps. The Importance of anchor text for ad hoc search revisited. In Proceedings of SIGIR'10, 122-129. 2010.

[7] M. J. Bates. The design of browsing and berry picking techniques for the online search interface. Graduate School of Library and Information Science, University of California at Los Angeles, CA 90024-1520. 1989.

[8] T.Y. Liu. Learning to Rank for Information Retrieval. Springer. 2011.

[9] R. Baeza, B. Ribeiro. Modern Information Retrieval. Addison-Wesley. 1999.

[10] W. Croft, J. Ponte. A language modeling approach to information retrieval. In Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Informational Retrieval, ACM, 275-281. 1998.

[11] S. Brin, L. Page. The Anatomy of a Large-Scale Hypertextual Web Search Engine. In Proceedings of WWW 1998, ACM. 1998.

[12] K. Eguchi, K. Oyama, E. Ishida, N. Knado, K. Kuriyama. Overview of the Web Retrieval Task at the Third NTCIR Workshop. Proc. 3rd NTCIR Workshop on Research in Information Retrieval, Automatic Text Summarization and Question Answering. 2003.

[13] K. Eguchi, K. Oyama, E.H. Ishikawa, A. Aizawa. Overview of NTCIR-4 Web Navigational Retrieval Task 1. Proc. 4th NTCIR Workshop on Research in Information Access Technologies Information Retrieval, Question Answering and Summarization. 2004.

[14] K. Oyama, M. Takaku, H. Ishikawa, A. Aizawa, H. Yamana. Overview of NTCIR-5 Web Navigational Retrieval Subtask 2. Proc. 5th NTCIR Workshop Meeting on Evaluation of Information Access Technologies Information Retrieval, Question Answering and Cross-lingual Information Access. 2005.

[15] 酒井哲也. 情報アクセス評価方法論. コロナ社出版. 2015.

[16] C. Clarke, M. Kolla, G. Cormack, O. Vechtomova, A. Ashkan, S. Buttcher, I. MacKinnon. Novelty and diversity in information retrieval evaluation. In Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval, SIGIR'08, 659-666. 2008.

[17] E. Voorhees. TREC-8 Questions Answering Track Report. In Proceedings of the 8th Text Retrieval Conference, 77-82.1999.