

# 部分文書検索を用いた高精度なモバイル情報検索

櫻 惇志† 宮崎 純† 波多野賢治††

† 東京工業大学 〒152-8552 東京都目黒区大岡山 2-12-1

†† 同志社大学文化情報学部 〒610-0394 京都府京田辺市多々羅都谷 1-3

E-mail: †keyaki@lsc.cs.titech.ac.jp, ††miyazaki@cs.titech.ac.jp, †††khatano@mail.doshisha.ac.jp

あらまし 本研究では、モバイル端末をはじめとしたディスプレイサイズの小さな端末における情報アクセスにおいて、部分文書検索を用いた高精度化を行う。近年増大するモバイル端末からの情報検索では、従来の文書単位の検索よりも細かな粒度における情報検索の実現が期待される。その解決策の一つとして、ユーザの発行するクエリに対して適合しつつ、理解可能な最小単位のテキストの断片である iUnit を粒度とした情報検索システムである。iUnit を検索単位としたモバイル情報検索タスクである MobileClick では、検索結果を二層構造で提示することを想定している。特に重要な iUnit のみを提示する一層目ページと、個別の意図 (intent) ごとに詳細な iUnit を提示する二層目のページによる情報提示を行うことで、ユーザのページ遷移回数を最小限に留めつつ、有用な情報の提示を行う。本稿では、まず文中の適合箇所を発見するために部分文書検索を行い、高スコアを付与された部分文書と類似度の高い iUnit を上位にランキングすることによって高精度なモバイル情報検索手法を提案する。評価実験の結果、提案手法はベースライン手法と比較して 9.7% 高精度に検索可能であることを示した。

キーワード モバイル検索, 部分文書検索, MobileClick, iUnit

## 1. はじめに

Google 社によれば、アメリカや日本を含む 10 ヶ国においてモバイル端末からの情報検索の件数は PC からの情報検索の件数を上回ったと報告されている<sup>(注1)</sup> ことから、近年、モバイル端末からの情報検索の機会が増加している。そのため、モバイル端末において適切な情報提示手法を提案することは重要な課題である。モバイル端末には小型のディスプレイが搭載されているため、提示できる情報は少なく、情報の取捨選択の必要性はより高い。言い換えれば、従来の文書単位による情報検索では効率的に情報検索活動を行うことが困難である。

国立情報学研究所主催の情報アクセス技術のための国際的ワークショップである NTCIR の MobileClick [10] では、モバイル端末を利用するユーザに対して、適切な検索結果の提示に取り組んでいる。MobileClick では、ユーザがクエリを入力すれば、検索結果を二層構造に整形して、不必要なページやサイトを遷移することなく、直接的に情報要求を満たすことを目指している。その際、図 1 に示す通り、まず一層目のページには、ユーザの情報要求に対して、より重要かつ概要的な内容についての情報と、意図 (intent) ごとの詳細ページ (二層目) へのリンクを提示する。これにより、ユーザは限られたディスプレイサイズにおいても効率的に情報検索を行うことが可能となる。なお、MobileClick における情報提示では、information unit (iUnit) と呼ばれる、クエリに適合しており、かつ、意味を理解する上での最小単位を用いる<sup>(注2)</sup>。iUnit は、2, 3 語程度の語句、フレーズ、文といった、任意の情報粒度で構成される。一層目の概要と二層目の詳細は、重要語の順に並べられた複数

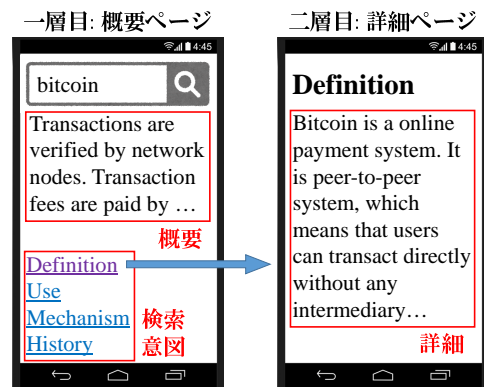


図 1 二層構造によるモバイル情報検索

の iUnit から構成される。

MobileClick は、さまざまなアプローチで取り組まれているが [4] [5] [27] [15] [3] [2] [19], 我々は部分文書検索を用いたアプローチ [7] [14] を採用する。部分文書検索の検索単位は XML 文書や HTML 文書などの構造化文書中の要素ノードであり、より厳密には、ある開始タグから終了タグの間に存在するプレーンテキストを結合したテキストを部分文書と呼ぶ。つまり、部分文書検索は、MobileClick と同様に、文書単位よりも細かな粒度である部分文書を検索単位とした情報検索である。部分文書検索では、構造化文書の文書構造を利用して、情報要求を満たす最適な箇所を文書中から特定することを目指す。MobileClick と部分文書検索その検索粒度に違いはあるものの、同じ方向性を目指す研究として流用可能な知見も多いと考えられる。

本研究では、下記の仮説の下に、部分文書検索を MobileClick へ適用する手順の概要を示す。

[Hypothesis 1] 適合箇所, すなわち、重要な iUnit は部分文

(注1): <https://adwords.googleblog.com/2015/05/building-for-next-moment.html> (accessed: December 3rd, 2016)

(注2): 文献 [25] においては全てのクエリキーワードを含む部分グラフであると定義しているものの、MobileClick ではそのような制約は存在しない。

書検索によって高スコアを付与された部分文書中に含まれる。まず既存の部分文書検索の枠組みで、文書集合に対してスコアリングを行う。これによって、クエリごとに、ランク付けされた部分文書リストが得られる。続いて、これらリストを用いて iUnit のランキング及び二層構造の検索結果の構築を行う。その際、提案手法の直感的なアイデアとしては、より上位にランキングされた部分文書と類似度の高い iUnit に対して高いスコアを付与する。

以降、本稿の構成について述べる。まず、2. 節にて MobileClick の概要について、3. 節にて部分文書検索の概要について述べる。それらを踏まえ 4. 節にて提案手法の説明を行い、その有用性検証のための評価実験を 5. 節にて行う。その後 6. 節にて他の MobileClick 参加者のアプローチについてレビューを行い、最後に 7. 節にて本研究を纏める。

## 2. MobileClick タスクの概要

MobileClick は、NTCIR-9 及び NTCIR-10 において取り組まれた 1CLICK-1 [24] と 1CLICK-2 [8] の流れを汲み、NTCIR-11 における MobileClick-1 [9] と、NTCIR-12 における MobileClick-2 [10] から構成される。MobileClick-1 と MobileClick-2 では、オーガナイザから提供されるデータの種類が異なり、完全に同一のタスクではない。本稿では主に MobileClick-2 に焦点を当てた説明を行い、断りが無い限り MobileClick-2 を指すこととする。

MobileClick は二つの手順から構成され、Step 1) iUnit のランキング、Step 2) ランキング結果を用いた二層構造の検索結果の構築である。各ステップは、iUnit Ranking サブタスクと iUnit Summarization サブタスクである。

### 2.1 テストコレクション

MobileClick では、オーガナイザから下記のデータが提供される。なお、対象言語は英語及び日本語であり、本稿では英語データセットを利用する。

**クエリ** 検索システムユーザのクエリログから、オーガナイザによって選別された、曖昧・不明瞭なクエリ 100 件を利用する。  
**文書コレクション** 各クエリごとに検索エンジン Bing を用いて最大上位 500 文書を取得したものを文書コレクションとして利用する。

**iUnit** iUnit は、適合した最小単位のテキストの断片である。従って、iUnit は、2, 3 語程度の語句、フレーズ、文といった、任意の情報粒度で構成される。例えば、クエリ “christopher nolan” の iUnit として、“runs the production company ‘Syn-copy Inc.’” や “married a film producer, Emma Thomas”, “Movie: ‘Interstellar’” などが存在する。iUnit は、オーガナイザによって雇用された複数名の評価者文書によって、コレクション中から網羅的に抽出されている。

**intent** intent は、NTCIR INTENT [23] や IMine [26] で用いられてきた、ある曖昧なクエリの特定の解釈や、不明瞭なクエリの特定の側面を表す。例えば、クエリ “jaguar” の intent は “Car” や “Animal” であり、クエリ “real madrid” の intent は “Players” や “Achievement”, “History” などである。MobileClick において、intent は iUnit の重要度を評価する際に用いられ、また、iUnit Summarization において、二層目へのリンクとして設定される。intent は、評価者により、手動で

クラスタリングされた iUnit に適切なラベルを付与することで作成されている<sup>(注3)</sup>。なお、各 intent には重要度 (意図確率) が設定されており、そのクエリによって検索した際にその意図に興味を持つと判断した投票者の割合が設定される。なお、1 度も投票されなかった意図は除去されている。

**iUnit の重要度** 手法の精度評価を行う際には、評価者によって付与された iUnit の重要度を利用する。iUnit には各 intent ごとに意図別重要度が評価されている。また、総合的重要度は、iUnit の意図別重要度と intent の意図確率から算出される。ある iUnit  $u$  の総合的重要度  $G_u$  は下記の通り算出される。

$$G(u) = \sum_{i \in I_q} P(i|q)g_i(u) \quad (1)$$

ただし、 $I_q$  はクエリ  $q$  の intent 集合、 $i$  は  $I_q$  に含まれる intent、 $P(i|q)$  は  $q$  における  $i$  の意図確率、 $g_i$  は  $i$  における  $u$  の意図別重要度とする。従って、ある intent においては極めて高い意図別重要度を持つ iUnit が、他の intent においては意図別重要度が低いため、低い総合的重要度を持つことが起こり得る。逆に、各 intent ごとの意図別重要度は必ずしも極めて高くない場合でも、多くの intent において少なからず重要と判断されれば、総合的重要度が高くなるが起こる。

### 2.2 iUnit Ranking サブタスク

二層構造の検索結果の構築に先立ち、iUnit は重要度順にランキングされている必要がある。なお、iUnit Ranking サブタスクでは、各 iUnit の重要度は総合的重要度から求められる。

iUnit Ranking の評価尺度は、nDCG@K ( $K=3, 5, 10, 20$ ) [1] と Q-measure [21] である。nDCG は文書検索の評価に多用される、順位ベースの多段階適合性判定データのための評価尺度である。それに対して、Q-measure は再現率ベースの多段階適合性判定データのための評価尺度である。従って、Q-measure は上位  $k$  件の iUnit だけでなくリスト全体の性能を評価することが可能であるため、公式尺度としては Q-measure が用いられる。

nDCG は式 (3) によって算出される。

$$DCG_K = \sum_{k=1}^K \frac{GG_{u_k}}{\log_2(k+1)} \quad (2)$$

$$nDCG_K = \frac{DCG_K}{iDCG_K} \quad (3)$$

ただし、 $u_k$  は  $k$  件目の iUnit、 $GG(u_k)$  は  $u_k$  の総合的重要度とする。 $iDCG$  は理想的なランキングにおける  $DCG$  であり、正規化の役割を果たす。

また、Q-measure は式 (4) によって算出される。

$$Q = \frac{1}{R} \sum_{r=1}^M IsRel(u_r) \frac{\sum_{r'=1}^r (GG(u_{r'}) + IsRel(u_{r'}))}{\sum_{r'=1}^r GG(u_{r'}^*) + r} \quad (4)$$

ただし、 $R$  は総合的重要度が 0 より大きい iUnit の個数、 $M$  はランク付きリスト長、 $u_{r'}^*$  は理想的なランク付きリストにおける  $r$  番目の iUnit である。

### 2.3 iUnit Summarization サブタスク

iUnit Summarization サブタスクでは、最も重要かつ概要的な情報を含む iUnit と intent から構成される一層目と、一層目の各 intent からリンクされ、intent に関しての詳細が記載

(注3) : クラスタリング結果は、intent のラベル付けにのみ利用される。

された iUnit から構成される二層目の構造へ検索結果の構築を行う (図 6 参照)。一層目と二層目の結果は、iUnit Ranking サブタスクで作成したリストを用いて構築する。二層構造の検索結果提示により、ユーザは情報要求を満たすためにページ (文書) の遷移を繰り返すことなく、最低限のページ遷移によって欲しい情報を発見することができる。

iUnit Summarization サブタスクではモバイル端末での情報検索を想定しているため、各層に表示可能な情報の文字数制限が設けられている。英語では 420 文字以下であり、記号や空白は文字数に含まれない。また、同一の iUnit が一層目と二層目に出現した場合、二層目の iUnit の重要度は 0 と判定される。

評価尺度である M-measure [10] は下記のユーザモデルに基づき設計されている。

- ユーザは意図確率  $P(i|q)$  に従い、ある一つの意図に興味を持つ。

- ユーザは以下のルールに従い要約を読み進める。
  - 一層目の冒頭から順番に検索結果を読み、記号や空白を除き  $L$  文字を読んだ時点で中断する。

- リンク  $l_i$  の終わりまで読んだとき、もしリンク  $l_i$  の intent に興味があれば、リンク先の二層目  $s_j$  を読む。

- 二層目  $s_j$  の終わりまで読んだ場合、リンク  $l_j$  の末尾から再度一層目を読み進める。

なお、ユーザは一つの意図にのみ興味があるため、ユーザが閲覧する可能性のあるテキスト (トレイルテキスト) は、意図の種類数存在する。各トレイルテキストの情報利得は、iUnit の出現位置が加味される U-measure [22] によって計測される。具体的には、重要な iUnit がトレイルテキストの冒頭に出現するほど、U-measure のスコアが高くなる。また、ユーザが検索結果を確率的に読むことを想定し、全てのトレイルテキストを考慮した情報利得の期待値は M-measure で計測される。

上記を踏まえて、M-measure は下記の式 (5) で算出される。

$$M = \sum_{i \in I_q} P(i|q) U_i(t_i) \quad (5)$$

ただし、 $I_q$  はクエリ  $q$  における intent 集合、 $P(i|q)$  は  $q$  における intent  $i$  の意図確率、 $U_i(t_i)$  は intent  $i$  のトレイルテキスト  $t_i$  の U-measure の値である。また、U-measure は下記の式 (6)、式 (7)、式 (8) によって算出される。

$$U_i(t) = \sum_{j=1}^{|t|} g_i(u_j) d(u_j) \quad (6)$$

$$d(u) = \max(0, 1 - \frac{pos_t(u)}{L}) \quad (7)$$

$$pos_t(u) = \sum_{j'=1}^j chars(u_{j'}) \quad (8)$$

ただし、 $g_i(u_i)$  は iUnit  $u$  の意図  $i$  の下での重要度、 $d$  は位置依存の減価関数、 $chars(u)$  は記号や空白を除いた iUnit  $u$  の文字数である。従って、一層目の iUnit の重要度の評価は総合的重要度によって行われ、二層目の iUnit は各 intent ごとの重要度によって評価されることになる。

### 3. 部分文書検索の基本事項

本節では、部分文書検索の概要について説明する。

#### 3.1 部分文書検索技術の目標

部分文書検索における最大の関心は、構造化文書中からクエリに適合する箇所、即ち適合部分文書を抽出し、それらをクエ

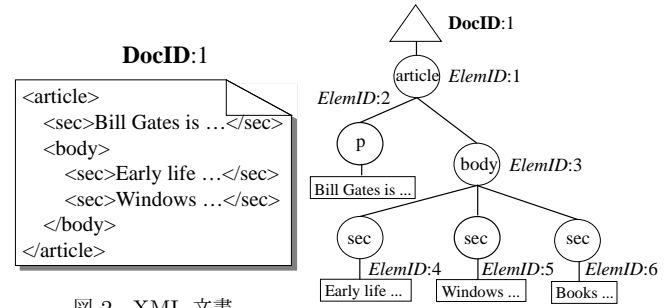


図 2 XML 文書

図 3 XML 木

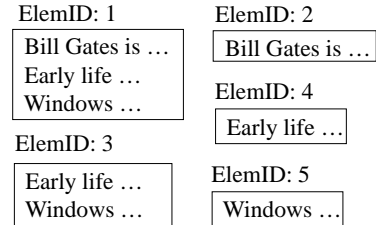


図 4 部分文書

リに対する適合度の降順でユーザへ提示することである。部分文書検索システムはクエリに適合する部分そのものをユーザに提示するため、ユーザは文書中から情報要求を満たす部分を自ら発見する必要がなくなり、情報検索を行う際のユーザの負担を大きく軽減することが可能である。

#### 3.2 部分文書とその重複関係

部分文書の概要について説明するために、図 2-図 4 を用いて具体例を示す。

まず、図 2 は文書番号 (DocID) 1 の XML 文書の例であり、図 3 は図 2 の XML 文書を木構造で表現した図である。構造化文書は一般的に木構造で表現することができ、文書構造の視認性の向上を目的として度々木構造で表現される。このとき、XML 文書のそれぞれの開始タグと終了タグが XML 木の各要素ノードのノード名に対応しており、タグの入れ子は要素ノードの親子関係によって表現されている。また、各要素ノードに対して、文書順に部分文書番号 (ElemID) を割り振る。図 4 の各部分文書は、図 3 の XML 木の各要素ノード以下に含まれるテキストノードを結合した文字列と対応する。つまり、文書全体を表す `article` ノードと対応する部分文書は子孫に存在するテキストノード全てを結合した文字列であり、`body` ノードと対応する部分文書はその子ノードである三つの `sec` ノードに含まれるそれぞれのテキストノードを結合した文字列である。包含関係 (先祖・子孫関係) を持つ部分文書間においてテキストノードの重複が発生するのはこのためである。なお、ある要素ノード (部分文書) に着目した際に、先祖にあたる要素ノードを粒度の大きなノード、逆に子孫にあたる要素ノードを粒度の小さなノードと表現する。

仮にユーザが “Early life ...”, “Windows...” に関する情報を求める場合、部分文書検索では ElemID 3 の部分文書を提示する。これは、部分文書中にユーザが求める情報全てを含み、余分な情報を含まないためである。

#### 3.3 部分文書検索における検索結果の提示方法

検索結果の構築において、複数の方式が存在する。いくつかの先行研究では重複が発生することを考慮せずに、クエリ処理

によって算出される部分文書の適合度に従ってランキングされた順位付きリストを提示していた。本論文では、このようなクエリ処理によって得られた、重複を排除するために特別な処理を行っていないリストを重複リストと呼ぶこととする。そして、文献[11]において重複リストを用いた場合の検索精度の低下が報告されて以来、多くの研究ではリスト中から重複を排除した非重複リストを用いている。XML 検索のためのプロジェクトである INEX (INitiative for Evaluation of XML retrieval) project<sup>(注4)</sup> の XML 情報検索用のトラック全般 [6] においても非重複リストを使用している。

重複リストから非重複リストを作成する際には下記の三種類の方式が存在する。

**OneElem** 一つの文書から文書中において最もスコアの高い部分文書を抽出する。

**MultiElem** 重複が起こらない限り、スコアの降順に一つの文書から複数の部分文書を抽出する。

**WholeDoc** 文書全体を提示する。文書全体を提示することで、文書中の適合箇所は網羅的に抽出可能となるものの、非適合箇所の抽出も行うという問題が存在する。文書検索と同義であるため、部分文書検索の本来の趣旨からは外れる。

### 3.4 高精度な部分文書検索

部分文書検索において利用されるスコアリング手法の多くは、文書検索用のスコアリング手法を基にしており、共通して用いられる統計量も存在する。文書検索と部分文書検索のスコアリング手法における最も大きな違いは、部分文書検索には属性の概念が導入されていることである。つまり、文書検索においては全ての文書は同じ属性を持つと見なして、全ての文書を母集団として大域的重みを算出する。それに対して部分文書においては、同じ属性を持つと見なす部分文書集合に分類し、それぞれを母集団として大域的重みの算出を行う。属性の分類は、同一タグを持つ部分文書は同じ属性を持つとするアプローチや、同一 path 式を持つ部分文書は同じ属性を持つとするアプローチなどが存在する。

部分文書検索用のスコアリング手法としては、TF-IPF [16]、BM25E [17]、部分文書検索用クエリ尤度モデル [18] など多数存在し、過去の研究において、BM25E が最も高精度検索に適切であるという結果が得られている [13]。従って、本稿では部分文書検索用のスコアリング手法として BM25E を用いる。BM25E の重み付けでは、属性  $a$  で表されるある部分文書  $e$  中の索引語  $t$  の重み  $w(a, e, t)$  は以下の数式で算出される。

$$w(a, e, t) = \frac{(k_1 + 1)tf_{e,t}}{k_1((1-b) + b\frac{el_e}{avel_a}) + tf_{e,t}} \cdot \log \frac{N_a - af_{a,t} + 0.5}{pf_{a,t} + 0.5} \quad (9)$$

ただし、 $tf_{e,t}$  を部分文書  $e$  中の索引語  $t$  の出現頻度、 $el_e$  を部分文書  $e$  の索引語数、 $avel_a$  を  $a$  で表される部分文書の平均索引語数、 $N_a$  を  $a$  で表される部分文書数、 $af_{a,t}$  を  $a$  で表され、かつ、 $t$  を含む部分文書数、 $k_1, b$  をパラメータとする。本稿では、一般的に利用されるパラメータ値である  $k_1 = 2.5$  と  $b = 0.85$  を設定する。

部分文書検索システム評価のための大規模なテストコレク

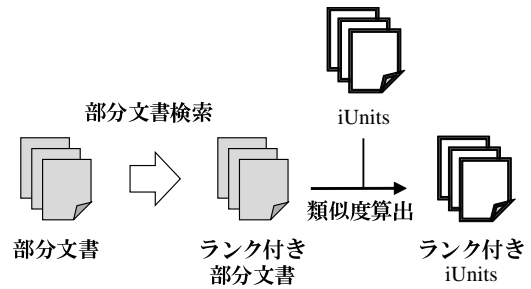


図5 iUnit のランキング手順

ションである INEX Wikipedia テストコレクション [7] においては、同一 path 式を持つ部分文書は同じ属性に分類して統計量を算出するアプローチが最も高精度と報告されている [20] ものの、統計量算出に用いる文書数が十分に確保されない状況においては path 式ベースの大域的重み算出では適切に統計量を算出できないということが報告されている [14]。2.1 節で述べた本研究の実験で利用するテストコレクションの文書集合は大域的統計量を算出する上で十分の分量が確保されていない可能性があるため、本稿では path 式ベースではなくタグベースのアプローチを採用する。つまり、式 (9) の属性  $a$  はタグとして統計量の算出を行う。

また、最終的に  $e$  のスコア  $Score(e)$  は、クエリキーワード集合を  $T$  として以下の式で算出される。

$$Score(e) = \sum_{t_i \in T} w(p, e, t_i) \quad (10)$$

## 4. 提案手法

2. 節と 3. 節より、重要な iUnit と適合部分文書はいずれも文書中の適合箇所を表現する。このことを踏まえ、本研究では、既存の部分文書検索技術を用いて抽出した、適合部分文書と考えられる部分文書から、重要な iUnit を特定することを目指す。換言すれば、Hypothesis 1: “適合箇所、すなわち、重要な iUnit は部分文書検索によって高スコアを付与された部分文書に含まれる” に基づき、手法の提案を行う。

以降、iUnit Ranking サブタスクと iUnit Summarization サブタスクそれぞれの取組みについて述べる。

### 4.1 iUnit Ranking サブタスクへの取組み

iUnit Ranking サブタスクでは、オーガナイザから提供される iUnit を総合的重要度の降順に並べる。

提案手法の概要を図5に示す。まず、文書集合である HTML 文書から部分文書を取り出す。続いて、クエリを用いて各部分文書に対して部分文書検索技術によってスコアリングを行い、ランキングされた部分文書のリストを取得する。更に、部分文書と iUnit の類似度を算出し、iUnit のランキングを行う。その際、上位にランキングされた部分文書と類似した iUnit により大きなスコアを付与する。iUnit  $u$  のスコア  $Score(u)$  は下記の式で算出される。

$$Score(u) = \sum_{e \in E} \frac{sim(u, e)}{norm(e)} \quad (11)$$

ただし、 $E$  はスコアリングに用いる部分文書集合、 $e$  は  $E$  に含まれる部分文書、 $sim(u, e)$  は  $u$  と  $e$  の類似度、 $norm(e)$  は  $e$  のランキングによる正規化とする。

以降、部分文書と iUnit の類似度算出方法  $sim(u, w)$ 、部分文書のランキングによる正規化方法  $norm(e)$ 、スコアリングに

(注4) : <http://inex.mmci.uni-saarland.de/>

用いる部分文書  $E$  の選別について述べる。

任意の部分文書と iUnit の類似度を算出する際に、いずれの類似度算出方法が適切であるのか不明であるため、三種類の類似度算出方法  $Freq$  (式 (12)),  $Ratio$  (式 (13)),  $Jaccard$  (式 (14)) を検討する。

$$sim_{Freq}(u, e) = count(u, e) \quad (12)$$

ただし、 $count(u, e)$  は部分文書  $e$  と iUnit  $u$  に含まれる語の共起語数とする。

$$sim_{Ratio}(u, e) = \frac{count(u, e)}{length(u)} \quad (13)$$

ただし、 $length(u)$  は  $u$  に含まれる語数とする。

$$sim_{Jaccard}(u, e) = \frac{|u \cap e|}{|u \cup e|} \quad (14)$$

iUnit のスコアリングにおいて、ランキング上位の部分文書の影響を大きく反映させるため、二種類の部分文書のランキングによる正規化手法  $Rank$  (式 (15)) と  $LogRank$  (式 (16)) を検討する。

$$norm_{Rank}(e) = rank(e) \quad (15)$$

ただし、 $rank(e)$  は  $e$  のランキングとする。

$$norm_{LogRank}(e) = 1 + \log(rank(e)) \quad (16)$$

$Rank$  ではランキングが下位になるに連れて急激に正規化される値が大きくなるため、 $LogRank$  では正規化の値の緩やかに補正する。

スコアリングを行う際に考慮する部分文書  $E$  は、下記の三種類の方針で取得する。

$E_{AllElem}$  リスト中全ての部分文書を用いる。

$E_{TopPerElem}$  リストの上位  $k$  パーセントの部分文書を用いる。

$E_{TopElem}$  リストの上位  $k$  件の部分文書を用いる。

#### 4.2 iUnit Summarization サブタスクへの取り組み

iUnit Summarization サブタスクでは、二層構造に結果を整形する。提案手法では、iUnit と intent の配置方針においては MobileClick-1 [9] のベースライン手法の方針に準拠しつつ、各層ごとに適切なクエリを用いた iUnit のランキング手法や intent のランキング手法を検討する。

まず、配置方針について、図 6 を用いて示す。

(1) 全ての intent を一層目の末尾に配置する。

(2) 文字数制限に達しない限り、一層目の冒頭から iUnit を配置する。

(3) 二層目の検索結果として、各 intent ごとに、文字数制限を超えない限り iUnit を配置する。ただし、一層目に出現した iUnit は二層目には配置しない。

以降、intent のランキング手法、一層目と二層目の iUnit ランキング方法について述べる。

##### 4.2.1 intent ランキング手法

評価尺度の M-measure では intent の位置も減価関数において考慮されているため、より意図確率の高い intent を上位に配置することでより高精度を実現することが可能となる。

提案する intent ランキング手法では、より多くの情報を取得できることが期待される intent に高いスコアを付与することを目指す。ランキング手法として、下記の 3 種類を提案する。

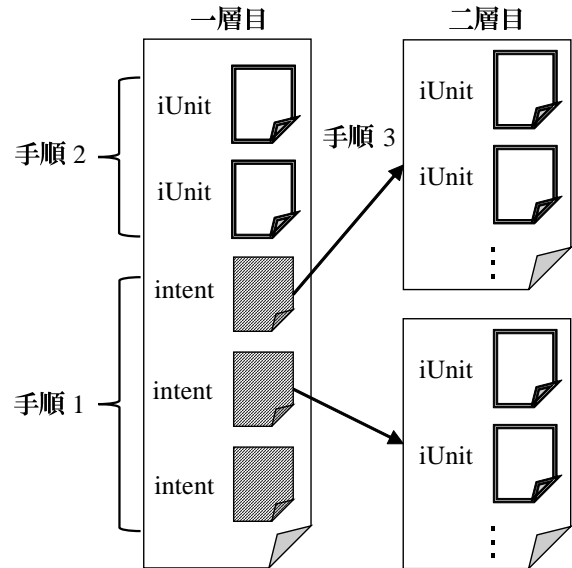


図 6 iUnit Summarization サブタスクにおける検索結果構築

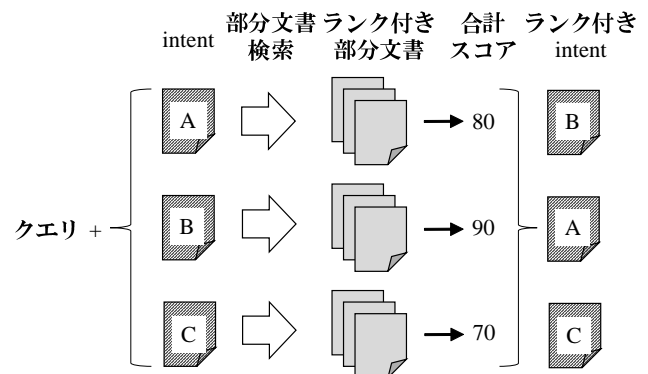


図 7 SumElemScore による intent のランキング

**SumElemScore** クエリに各 intent を追加した拡張クエリを用いて部分文書検索を行う。その後、得られたリストに含まれる部分文書の総スコアを各 intent のスコアとする。概要を図 7 に示す。

**NumResultElem**  $SumElemScore$  では部分文書のスコアの和を intent のスコアとしていたのに対して、リストに含まれる部分文書の件数をスコアとする。

**NumWebSearch** 前述の二つの手法はいずれもテストコレクション内の文書集合から得られる統計量を用いて intent のランキングを行う手法であるが、文書集合に含まれる文書数が各クエリごとに上限 500 件のみであるため、限られた情報のみしか反映できていない可能性が存在する。そこで、外部情報として、拡張クエリを用いて Web 検索を行った際の検索結果の件数の降順に intent のランキングを行う<sup>(注5)</sup>。

##### 4.2.2 iUnit ランキングに用いるクエリ

一層目に配置されることが望ましい iUnit は、多数の intent において重要な iUnit となる。その観点において下記の二種類の iUnit のランキングを提案する。

**AllIntent** クエリに全 intent を追加した全拡張クエリを用いて部分文書検索を行う。

(注5) : 文書集合の収集に用いた Web 検索エンジンと同様に Bing を利用した。

表 1 iUnit Ranking サブタスクの実験結果

	nDCG@3	nDCG@5	nDCG@10	nDCG@20	Q-measure
Freq	.7237	.7510	.7966	.8654	.8947
Ratio	.7325	.7499	.7995	.8707	.8969
Jaccard	.6849	.7192	.7717	.8511	.8800
Rank	.7443	.7632	.8060	.8748	.9005
LogRank	.7341	.7518	.8017	.8727	.8983
TopPerElem	.7453	.7627	.8065	.8751	.9007
TopElem	.7443	.7632	.8059	.8748	.9004
AllIntent	.7551	.7712	.8104	.8796	.9031
EachIntent	.7511	.7688	.8112	.8776	.9025
OneElem	.7641	.7764	.8140	.8815	.9047
MultiElem	.7604	.7742	.8132	.8806	.9044
WholeDoc	<b>.7613</b>	<b>.7750</b>	<b>.8154</b>	<b>.8815</b>	<b>.9050</b>

**EachIntent** 拡張クエリを用いて部分文書検索を行い、各部分文書のスコアは各拡張クエリごとの部分文書のスコアの和とする。

二層目の iUnit は各 intent に特化した iUnit が配置されることが期待される。この観点において、二種類の iUnit ランキングを提案する。

**ExtendedQuery** 拡張クエリを用いて部分文書検索を行う。

**IntentQuery** intent の比重をより強くするため、オリジナルクエリは使用せず intent のみを用いて部分文書検索を行う。

## 5. 評価実験

本節では、iUnit Ranking サブタスクと iUnit Summarization サブタスクそれぞれにおいて、提案手法の有用性評価のための行った評価実験について述べる。評価実験には 2. 節のデータセットを利用し、部分文書検索手法は 3.4 節で述べた BM25E を利用した。

### 5.1 iUnit Ranking サブタスクの実験結果

4.1 節にて、iUnit Ranking サブタスクのための、部分文書と iUnit の類似度算出方法 (式 (11) の  $sim(u, e)$ )、部分文書のランキングによる正規化方法 (式 (11) の  $norm(e)$ )、スコアリングに用いる部分文書の選別手法 (式 (11) の  $E$ ) の検討を行った。表 1 を参照しつつ各項目に関する評価実験を行う。なお、公式尺度は Q-measure である。

また、3.3 節で述べた通り部分文書検索においては検索結果の提示において非重複リストを用いることが一般的であるものの、現時点では iUnit のランキングにおいてはどのような挙動を示すのか明らかではないため、特記しない限り以降の実験では重複リストを用いる。

#### 5.1.1 提案手法の評価

類似度算出手法 *Freq*, *Ratio*, *Jaccard* の実験を行う。式 (11) において、部分文書による正規化は行わず ( $norm(e) = 1$ )、 $E_{AllElem}$  としてスコアリングに全ての部分文書を用いた。その結果、*Ratio* が最も高精度という結果が得られた。

続いて、部分文書のランキングによる正規化に関する評価を行うため、*Rank* と *LogRank* の実験を行う。式 (11) において、 $sim_{Ratio}(u, e)$ ,  $E_{AllElem}$  を使用した。

*LogRank* の底として 1.1, 1.5, 2, ネイピア数, 10 にて実験を行った結果、底が 2 において最も高精度を示した。また、シグモイド関数においてもパラメータを調整しつつ複数の関数において正規化を試みたものの、底が 2 の log による正規化が

表 2 iUnit Ranking サブタスクの他の参加者との比較

Team ID	Q-measure	improvement
WholeDoc	<b>.9050</b>	<b>1.008</b>
OneElem	.9047	1.008
cuis	.9042	1.007
IRIT	.9036	1.007
UHYG	.9028	1.006
IISR	.9004	1.003
ORG-L	.8975	1.000
RISAR	.8972	1.000
ALICA	.8959	.998
YJST	.8953	.998
ORG-R	.8859	.987
JUNLP	.8859	.987

最も高精度を示した。いずれの手法も精度の向上が確認され、*Rank* がより高精度という結果が得られた。

次に、スコアリングに用いる部分文書の選別結果について述べる。式 (11) の  $sim_{Ratio}(u, e)$  と  $norm_{Rank}$  を使用した。なお事前に行った閾値調査の結果、*TopPerElem* において上位 33% の部分文書を選択した場合に最も高精度であり、*TopElem* においては上位 2,000 件の部分文書を用いた場合に最も高精度であった。*TopPerElem* を利用することで精度は向上したものの、*TopElem* では却って精度が低下した。

なお、表 1 に掲載していない組合せの手法についても実験を行ったところ、式 (11) の類似度算出手法は  $sim_{Ratio}(u, e)$ 、部分文書のランキングによる正規化は  $norm_{Rank}$ 、スコアリングに用いる部分文書は  $E_{TopPerElem}$ 、すなわち *TopPerElem* が最も高精度であった。従って、以降の実験では iUnit のランキングには *TopPerElem* を用いることとする。

#### 5.1.2 追加の検証実験

次節の iUnit Summarization サブタスクにおける提案手法である、*AllIntent* と *EachIntent* のポテンシャルを確認するため、iUnit Ranking タスクにおいて評価を行った。その結果、いずれも精度が向上し、*AllIntent* がより高精度であるという結果を示した<sup>(注6)</sup>。

また、非重複リストを用いた場合の精度の検証を *AllIntent* に対して行った。その結果、いずれも重複リストを用いた場合よりも精度が向上したものの、部分文書を提示する *MultiElem* と *OneElem* より、文書検索と同義である *WholeDoc* が最も高精度であることを示した。

なお、以前著者らが行った Web 文書の文書構造に関する調査 [12] の結果、Web 文書では物理的な文書構造が、章立てなどの論理的な文書構造と一致していないということが示唆されており、この状況では部分文書検索の性能が最大限発揮できない可能性がある。従って、Web 文書に対して、論理的な文書構造を表すように文書構造の整形を行うことで、より部分文書検索の精度を向上することが期待される。

続いて、MobileClick iUnit Ranking サブタスクの他の参加者の公式ラン結果との比較結果を表 2 に示す<sup>(注7)</sup>。improvement

(注6) : intent の抽出においては MobileClick と同様 NTCIR のタスクである INTENT [23] や IMine [26] において取り組まれており、これらの研究成果を利用すれば取得可能である。

(注7) : 複数のランを提出した参加者に関しては、最も高精度のランのみ提示する。また、著者らが提出したラン (titec) は除外している。

表3 iUnit Summarization サブタスクの実験結果

	M-measure
Simple (TopPerElem)	18.105
AllIntent & ExtendedQuery	18.442
EachIntent & ExtendedQuery	18.315
AllIntent & IntentQuery	18.447
EachIntent & IntentQuery	18.313
SumElemScore	18.428
NumResultElem	18.365
NumWebSearch	18.469
OneElem	<b>18.530</b>
MultiElem	18.508
WholeDoc	18.363

は、ベースライン手法 (*ORG-L*) と比較した際の精度向上率を表す。state-of-the-art の手法を提案した *cuis* を上回る検索精度を達成した。また、符号検定の結果、提案手法はベースライン手法 (*ORG-L*) と比較して有意な差はなかった。

## 5.2 iUnit Summarization サブタスクの実験結果

4.2 節にて、iUnit Summarization サブタスクにおける、一層目及び二層目の iUnit のランキングで用いるクエリの検討と intent のランキング手法の提案を行った。それらの評価を行った結果を表3に示す。なお、公式尺度は M-measure である。

### 5.2.1 提案手法の評価

提案手法のシンプル手法として、*TopPerElem* の評価を行う。その際、一層目はオリジナルクエリ、二層目はオリジナルクエリと各 intent (拡張クエリ)、intent は初期の順序 (ランダム) である。

提案手法に関して、一層目のクエリの選択肢として *AllIntent* と *EachIntent*、二層目のクエリの選択肢として *ExtendedQuery* と *IntentQuery* が存在する。これらの全組合せ4通りにおいて、*AllIntent* と *IntentQuery* が最も高精度であった。

*EachIntent* に着目した場合に、*IntentQuery* よりも *ExtendedQuery* と組み合わせた場合により高精度を示した。これは一層目に *EachIntent* を用いた場合には、ある intent において重要度の高い iUnit が上位に配置されており、その結果として一層目に配置されるべき、多数の intent において重要度の高い iUnit の配置ができていないことを示唆する。従って、タスクの性質上、*AllIntent* と *IntentQuery* の組合せがより高精度を示したと予想される。以降の実験では、*AllIntent* と *IntentQuery* の組合せを利用する。

次に、intent のランキングに関する評価を行う。intent のランキング手法 *SumElemScore*、*NumResultElem*、*NumWebSearch* のうち、*SumElemScore* と *NumResultElem* は逆に精度が低下した。その一方で、*NumWebSearch* は精度が向上した。このことから、拡張クエリの結果の件数はその intent によって得られる情報利得の予想において有用であり、小規模の文書集合から得られる統計量では intent ごとの情報利得の期待値を正確に予測することは困難であると示唆された。

以上より、一層目は *AllIntent*、二層目は *IntentQuery*、intent ランキング手法は *NumWebSearch* の組合せが最も高精度であるという結果が得られた。

### 5.2.2 追加の検証実験

非重複リストを用いた場合の精度の検証を、*AllIntent*、*IntentQuery*、*NumWebSearch* の組合せに対して行った。その

表4 iUnit Summarization サブタスクの他の参加者との比較

Team ID	M-measure	improvement
OneElem	<b>18.530</b>	<b>1.097</b>
ORG-T	16.898	1.000
YJST	16.887	.999
IRIT	16.563	.980
cuis	16.419	.972
RISAR	16.047	.950
ORG-R	14.105	.835
UHYG	13.055	.773
JUNLP	11.703	.693
ALICA	8.497	.503

結果、iUnit Ranking と異なる結果が得られた。具体的には、*OneElem*、*MultiElem*、*WholeDoc* という順で高い精度を達成した。つまり、文書全体よりも部分文書検索を利用することで検索精度が向上した。iUnit Ranking サブタスクは提供される iUnit 全てをランキングするタスクであるのに対して、iUnit Summarization では420文字までの結果のみを提示する。これらを鑑みると、限られた文字数においてより網羅的に適合するテキストを提示する上では、部分文書検索の枠組みでスコアリングを行うことが適切であることが示唆された。

続いて、MobileClick iUnit Summarization サブタスクの他の参加者の公式ラン結果との比較結果を表4に示す<sup>(注8)</sup>。*OneElem* が最も高精度な手法であった。また、符号検定の結果、他グループのランのうち最も高精度な手法であるベースライン手法 (*ORG-T*) と比較して、提案手法 *OneElem* は統計的に有意に高精度であった ( $p < 0.01$ )。

以上より、MobileClick タスクにおいて部分文書検索を用いたアプローチによって、比較手法や文書検索手法と比較して高精度な検索を実現することができた。従って、Hypothesis 1: “適合箇所、すなわち、重要な iUnit は部分文書検索によって高スコアを付与された部分文書中に含まれる” は確からしいという結果が導き出された。また、文書検索アプローチは比較手法よりも高精度な結果を示した。

## 6. 関連研究

本節では、他の MobileClick 参加者のうち、ベースライン手法、及び、公式ラン結果上位の参加者の手法をレビューする。

まず、ベースライン手法 [10] では、odds ratio を用いて iUnit 中の語の重みの算出を行う。odd ratio による語の重み付けでは、クエリ語を含むページ中に多く出現して、クエリ語を含まないページ中に出現しないほど、語の重みが大きくなる。iUnit のスコアは、iUnit に含まれる語の重みの総和で求める。

公式ランの iUnit Ranking において最も高い精度を達成した Lai ら [15] は、LDA を用いた手法を提案した。その際、トピックは intent と見なす。LDA を用いてモデルが構築されれば、各クエリのトピックの確率値 (意図確率と同義) を求める。次に、各文書を最も類似度の高いトピックに割り振る。トピックごとに文書のリストが構築されれば、ラウンドロビン方式で、各トピックから一つずつ文書抽出してランキングを行う。全トピックから文書の抽出終われば、またはじめからトピックごとに文書の抽出を行う。文書のランキング完了すれば、iUnit と

(注8): 複数のランを提出した参加者に関しては、最も高精度のランのみ提示する。また、著者らが提出したラン (titec) は除外している。



文書の共起語の数をスコアとして iUnit に付与する。その際、文書のランキングで正規化を行う。

Chellal ら [2] は、シャノンのエントロピーをベースに語の出現頻度や文書頻度、Word2vec などを用いて拡張して語の重みを算出する。また、iUnit のスコアは、iUnit に含まれる語の重みの総和で表す。

Lai ら [15] の手法が高精度であったことから、iUnit を直接重み付けするアプローチよりも、まずは文書 (提案手法では部分文書) に対してスコアリングを行い、その結果を踏まえて iUnit のスコアリングを行うことがより適切であると考えられる。その理由として、iUnit の中には 2, 3 語しか含まないものも多く存在し、これらの iUnit に対しては適切に語の重み付けを行うことが困難であるためと考えられる。

また、Chellal ら [2] は odds ratio のように単語の出現の有無だけでなく、出現頻度やより複雑な統計量を利用しているため、他の odd ratio を利用する研究と比較して高精度を実現できたと考えられる。その点において、我々の採用している BM25E も多数の統計量を用いて語の重み付けを行っている。

## 7. おわりに

本稿では、モバイル環境における情報検索において高精度を実現するため、部分文書検索のアプローチを利用した。提案手法では、まず文書集合に対して部分文書検索を行い、部分文書のランキングを行う。次に部分文書と iUnit の類似度を計測し、より上位にランキングされた部分文書とより類似度の高い iUnit に高いスコアを付与する。

iUnit Ranking サブタスクにおいて、類似度算出手法、部分文書による正規化、スコアリングに用いる部分文書の選別を行うことで、state-of-the-art の手法を上回る精度を達成した。iUnit Summarization サブタスクでは、より高い情報利得が期待される intent を上位に配置したり、各層ごとに適切なクエリを用いた部分文書を行うことで、ベースラインの手法と比較して 9.7% の精度向上を達成した。

今後の課題として、本研究では提供された iUnit を利用したが、より実用的なモバイル検索システム実現のためには、部分文書から iUnit を自動抽出する手法の提案を行う必要がある。論理構造と合致するように Web 文書の構造整形も今後の課題の一つである。

## 謝 辞

本研究の一部は、JSPS 科研費 JP15K20990, JP26280115, JP15H02701, JP16H02908 の助成を受けたものである。ここに記して謝意を表す。

## 文 献

- [1] Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. Learning to Rank using Gradient Descent. In *Proc. of the 22th ICML*, pp. 89–96, 2005.
- [2] Abdelhamid Chellal and Mohand Boughanem. IRIT at the NTCIR-12 MobileClick-2 Task. In *Proc. of the 12th NTCIR*, pp. 143–146, 2016.
- [3] Monalisa Dey, Anupam Mondal, and Dipankar Das. NTCIR-12 MOBILECLICK: Sense-based Ranking and Summarization of English Queries. In *Proc. of the 12th NTCIR*, pp. 138–142, 2016.
- [4] Wen-Bin Han, Hung-Hsiang Wang, and Richard Tzong-Han Tsai. NCU IISR System for NTCIR-12 MobileClick2. In *Proc. of the 12th NTCIR*, pp. 115–117, 2016.
- [5] Sho Iizuka, Takayuki Yumoto, Manabu Nii, and Naotake Kamiura. UHYG at the NTCIR-12 MobileClick Task: Link-based Ranking on iUnit-Page Bipartite Graph. In *Proc. of the 12th NTCIR*, pp. 118–125, 2016.
- [6] Jaap Kamps, Shlomo Geva, Andrew Trotman, Alan Woodley, and Marijn Koolen. Overview of the INEX 2008 Ad Hoc Track. In *INEX 2008 Workshop Pre-proceedings*, pp. 1–28, 2008.
- [7] Jaap Kamps, Shlomo Geva, Andrew Trotman, Alan Woodley, and Marijn Koolen. Overview of the INEX 2008 Ad Hoc Track. In *Formal Proc. of INEX 2008 Workshop*, pp. 1–28, 2009.
- [8] Makoto P. Kato, Matthew Ekstrand-Abueg, Virgil Pavlu, Tetsuya Sakai, Takehiro Yamamoto, and Mayu Iwata. Overview of the NTCIR-10 1CLICK-2 Task. In *Proc. of the 10th NTCIR*, pp. 182–211, 2013.
- [9] Makoto P. Kato, Matthew Ekstrand-Abueg, Virgil Pavlu Tetsuya Sakai, Takehiro Yamamoto, and Mayu Iwata. Overview of the NTCIR-11 MobileClick Task. In *Proc. of the 11th NTCIR*, pp. 195–207, 2014.
- [10] Makoto P. Kato, Tetsuya Sakai, Takehiro Yamamoto, Virgil Pavlu, Hajime Morita, and Sumio Fujita. Overview of the NTCIR-12 MobileClick Task. In *Proc. of the 12th NTCIR*, pp. 104–114, 2016.
- [11] Gabriella Kazai, Mounia Lalmas, and Arjen P. de Vries. The Overlap Problem in Content-Oriented XML Retrieval Evaluation. In *Proc. of the 27th ACM SIGIR*, pp. 72–79, 2004.
- [12] Atsushi Keyaki, Jun Miyazaki, and Kenji Hatano. Applying XML Element Retrieval Techniques to Web Documents. *DBSJ Journal*, Vol. 13, No. 1, pp. 64–70, 2015.
- [13] Atsushi Keyaki, Jun Miyazaki, Kenji Hatano, Goshiro Yamamoto, Takafumi Taketomi, and Hirokazu Kato. A Path expression-Based Smoothing of Query Likelihood Model for XML Element Retrieval. In *Proc. of the 1st ACIT*, pp. 296–300, 2013.
- [14] Atsushi Keyaki, Jun Miyazaki, Kenji Hatano, Goshiro Yamamoto, Takafumi Taketomi, and Hirokazu Kato. Fast Incremental Indexing with Effective and Efficient Searching in XML Element Retrieval. *International Journal of Web Information Systems (IJWIS)*, Vol. 9, pp. 142–164, 2013.
- [15] Kwun Ping Lai, Wai Lam, and Lidong Bing. CUIS at the NTCIR-12 MobileClick2 Task. In *Proc. of the 12th NTCIR*, pp. 134–137, 2016.
- [16] Fang Liu, Clement Yu, Weiyi Meng, and Abdur Chowdhury. Effective Keyword search in Relational Databases. In *Proc. of ACM SIGMOD*, pp. 563–574, 2006.
- [17] Wei Liu, Stephen Robertson, and Andrew Macfarlane. Field-Weighted XML Retrieval Based on BM25. In *Formal Proc. of INEX 2005 Workshop*, pp. 161–171, 2006.
- [18] Paul Ogilvie and Jamie Callan. Parameter Estimation for a Simple Hierarchical Generative Model for XML Retrieval. In *Formal Proc. of INEX 2005 Workshop*, pp. 211–224, 2006.
- [19] Kevin Ong, Ruey-Cheng Chen, and Falk Scholer. RMIT at the NTCIR-12 MobileClick-2: iUnit Ranking and Summarization Subtasks. In *Proc. of the 12th NTCIR*, pp. 104–114, 2016.
- [20] Benjamin Piwowarski and Patrick Gallinari. A Bayesian Framework for XML Information Retrieval: Searching and Learning with the INEX Collection. *Journal of Information Retrieval*, Vol. 8, No. 4, pp. 655–681, 2005.
- [21] Tetsuya Sakai. The Reliability of Metrics Based on Graded Relevance. *Journal of Information Processing and Management*, Vol. 43, No. 2, pp. 531–548, 2007.
- [22] Tetsuya Sakai and Zhicheng Dou. Summaries, Ranked Retrieval and Sessions: a Unified Framework for Information Access Evaluation. In *Proc. of the 36th SIGIR*, pp. 473–482, 2013.
- [23] Tetsuya Sakai, Zhicheng Dou, Takehiro Yamamoto, Yiqun Liu, Min Zhang, and Ruihua Song. Overview of the NTCIR-10 INTENT-2 Task. In *Proc. of the 10th NTCIR*, pp. 94–123, 2013.
- [24] Tetsuya Sakai, Makoto P. Kato, and Young-In Song. Overview of NTCIR-9 1CLICK. In *Proc. of the 9th NTCIR*, pp. 180–201, 2011.
- [25] Keishi Tajima, Kenji Hatano, Takeshi Matsukura, Ryouichi Sano, and Katsumi Tanaka. Discovery and Retrieval of Logical Information Units in Web. In *Proc. of WOVIS in conjunction with the 4th DL*, pp. 180–201, 1999.
- [26] Takehiro Yamamoto, Yiqun Liu, Min Zhang, Zhicheng Dou, Ke Zhou, Ilya Markov, Makoto P. Kato, Hiroaki Ohshima, and Sumio Fujita. Overview of the NTCIR-12 IMine-2 Task. In *Proc. of the 12th NTCIR*, pp. 8–26, 2016.
- [27] Tatsunori Yoshioka and Takashi Yukawa. NUTKS at NTCIR-12 MobileClick2: iUnit Ranking Subtask Using Topic Model. In *Proc. of the 12th NTCIR*, pp. 131–133, 2016.