協調サーチエンジンにおける差分更新の評価

佐藤 永欣† 宇田川 稔† 上原 稔† 酒井 義文‡
東洋大学工学部情報工学科†
東北大学大学院農学研究科‡

1 はじめに

集中型サーチエンジンでは新鮮な情報の検索は困難である。そこで、我々は新鮮な情報の検索を実現するため、イントラネット向け分散型サーチエンジンである協調サーチエンジン (Cooperative Search Engine, CSE) を開発し [1]、新鮮情報検索の概念を提案した [2]。CSE は 10 分程度でインデックスを完全に更新できる。しかし、10 分の間に新規に作成・更新される文書は極めて僅かであり、毎回インデックスを完全に更新するのは無駄である。我々は以前、文書の変更点だけをインデックスに反映する単純更新を提案したが、文書や文書に含まれる語が減ることを考慮していなかった。そこで、文書や語の減少を反映する単純更新を差分更新という名で提案・実装し、評価を行った。

2 協調サーチエンジン

CSE は以下のコンポーネントからなる（Fig.1 参照）。

- Location Server (LS) は各 Web サイトに含まれるキーワードの表を管理する。LS は Site selection Cache (SC) を持つ。
- Cache Server (CS) は検索結果をキャッシュし、先読みを行う。CS は Retrieval Cache (RC) と SC (LS の SC のコピー) を持つ。
- Local Meta Search Engine (LMSE) はユーザと対話し、LSE の違いを隠蔽する。
- Local Search Engine (LSE) は検索、インデックス作成を行う。

CSE は以下のようにインデックスを更新する。

1. LSE の Gatherer が対象サイトの文書を収集する。
2. LSE の Indexer は Gatherer が収集した文書のインデックスを作成する。この時、並列処理を行う。
   (a) $LMSE_i$ の Engine I/F は LSE_i からキーワード、スコア情報を抽出して Forward Knowledge (FK) として LS に送信する。
   (b) LS は送信された情報を検索用に記録する。

†Nobuyoshi SATO, Minoru UDAGAWA, Minoru UEHARA, {jju, ti980039}@ds.cs.toyo.ac.jp,uehara@cs.toyo.ac.jp, Dept. of Information and Computer Science, Toyo Univ.
‡Yoshifumi SAKAI, sakai@biochem.tohoku.ac.jp, Graduate School of Agricultural Science, Tohoku Univ.

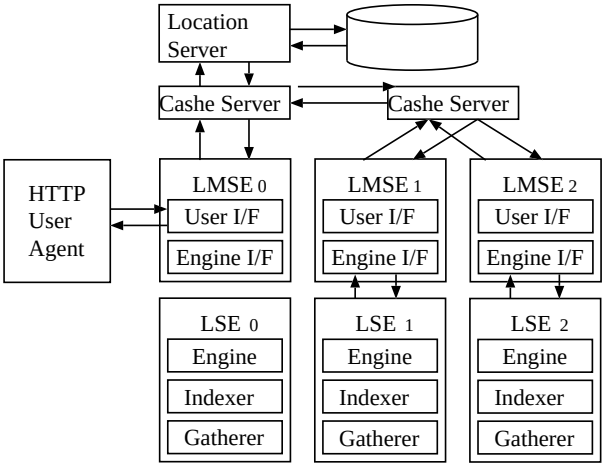

Fig. 1. CSE の構成と概要

以下では検索がどのように行われるかを述べる。

1. $LMSE_0$ はユーザーからクエリーを受け取り、CS に検索を依頼する。
2. CS は LS にクエリーを送り、どの LMSE がクエリーに適合する文書を持っているか尋ねる。
3. CS は LS がクエリーに適合する文書を持つと回答した LMSE にクエリーを送る。
4. 各 LMSE は LSE を用いてクエリーに適合する文書を検索し、CS に返答する
5. CS は検索結果をまとめて $LMSE_0$ に返す。
6. $LMSE_0$ は検索結果をユーザーのブラウザに表示する。

3 差分更新

Web サーバの文書は随時、作成・更新されるが、サーチエンジンのインデックス更新とは基本的に非同期であるため、ロボットによる文書の作成・更新の確認と収集が定期的に行われる。この間隔を更新間隔と呼び、CSE では 10 分程度である。インデックス全体を再作成する更新を完全更新と呼ぶが、10 分の間に作成・更新される文書はわずかであるため完全更新を毎回行うのは無駄である。従来、新たに作成された文書と更新された文書を対象としてインデックスを作成する単純更新を行っていた。単純更新では新たに現われた語はインデックスに反映されるが、消滅した語は反映され

なかった。語はインデックスからは消滅しているため検索結果に現われることはないが、FK には存在するため、無駄な検索要求が LMSE に対して送られ、検索応答速度の低下につながる。そこで、語の削除を FK に反映する単純更新である差分更新を導入する。

更新は以下のように行われる。3 から 6 までが差分更新である。

1. LMSE はインデックスを全て再作成し、FK を LS に送って完全更新を行う
2. LMSE は LS に送った FK を記録する
3. LMSE は更新された文書、新規文書をインデックスに追加する
4. LMSE はインデックス取り出した FK と前回 LS に送った FK を比較し、スコアに変更のあった語、新規に現われた語、消滅した語の差分を LS に送る
5. LMSE はインデックスから取り出した FK を記録する
6. 3 に戻る

以下では差分更新を実現するための、CSE の通信プロトコルの変更点について述べる。

CSE の検索・更新時のプロトコルは Generic Message Transfer Protocol (GMTP) と呼ばれる独自のプロトコルの上に Cooperative Search Protocol (CSP) と呼ばれる CSE での利用に特化したプロトコルを構築している。GMTP は一本のコネクション型通信路上で汎用メッセージの送受信を可能にする。GMTP は通常は TCP 上のプロトコルとして使用されるが、ファイヤーウォールを越える目的で HTTP 上にピギーバック形式で使用されることもある。

インデックス更新時の FK の転送は以下の CSP メッセージを用いて行われる。ここで、LMSE は FK を送信する LMSE の URL、LMSENumDocs は LMSE に登録されている文書の数である。FK は weightKeys は重み付けキーワード集合として送信される。

Update : (void) = Update(LMSE, LMSENumDocs, weightKeys)

weightKeys の内容を以下に示す。

```
weightKeys = *(Phrase SP TfMax SP TfMin
                              SP NumDocs CRLF)
   TfMax = 1*DIGIT
   TfMin = 1*DIGIT
   NumDocs = 1*DIGIT
```

Phrase はキーワード、TfMax はキーワードに対する各文書のスコアのうち、weightKeys を送信する LMSE における最大値であり、TfMin はその最小値である。NumDocs はそのキーワードを含む文書の数である。あるキーワードを含む文書が削除されるなどして、そのキーワードがインデックスから消滅した場合は、TfMax、TfMin、NumDocs の値として数値 0 を LMSE は送信する。これを受信した LS は当該キーワードを FK のデータベースより削除する。

Table 1. 差分更新の所要時間

	完全更新	差分更新
インデックス作成時間 [sec]	974	20
インデックス転送時間 [sec]		
読み込み	11.93	11.95
比較	3.616	4.967
転送	102.8	3.960
計	1093	40.88

以下では、メーリングリストをインデックスする場合の差分更新の評価を述べる。まず、ある時点での全メール 8381 通を完全更新でインデックスし、その後、1 日分の 32 通を差分更新で追加した。その結果を表 1 に示す。表 1 に示すように、インデックスの読み込みは完全更新と同様の時間を要している。1 日分の流量は変動するが、概ね 1 分程度で差分更新が可能と考えられる。また、この所要時間より、メールが到着する毎に 1 通ずつ差分更新することも可能と思われる。

4 まとめ

本稿では、CSE におけるインデックスの差分更新を提案した。差分更新はインデックス中の語の削除にも対応し、語を持たなくなった LMSE への検索時の問い合わせを避けることができる。更新された文書の量、転送すべき FK の量にもよるが、おおむね 1 分程度で差分更新を完了できる。CSE の更新間隔は概ね 10 分を想定しているので、LMSE をインストールした Web サーバやネットワークなどへの負荷を考慮しても、差分更新の所要時間は十分に短い。

参考文献

[1] 佐藤 永欣、上原 稔、酒井 義文、森 秀樹、“最新情報の検索のための分散型サーチエンジン”、情報処理学会論文誌、第 43 巻、第 2 号、pp.321–331、情報処理学会 (2002)

[2] N.Sato, M.Uehara, Y.Sakai, “Temporal Information Retrieval in Cooperative Search Engine”, in *Proc. of The 6th International Workshop on Network-Based Information Systems (NBiS2003)*, pp.215–220 (Prague, Czech Repulic