Түвшний Нэвтрэлтэд Суурилсан Хайлтын Системийн Аалзны Тоог Сонгох Нь

П. Мөнхцэцэг
ШУТИС, КТМС, КУСалбар
Улаанбаатар, Монгол
se12e015@csms.edu.mn

Ч. Эрдэнэбат
ШУТИС, КТМС, КУСалбар
Улаанбаатар, Монгол
ch.erdenebat@must.edu.mn

Хураангуй—Дэлхийн өндөр хөгжилтэй орнуудын дийлэнх нь өөрсдийн гэсэн үндэсний хайлтын системтэй байдаг. Энэ нь тухайн орны хувьд, иргэдийн хувийн мэдээлэл, үндэсний аюулгүй байдал зэрэг олон талаараа чухал ач холбогдолтой байдаг. Энэхүү өгүүллийн зорилго нь түвшний нэвтрэлтийн алгоритм ашиглан зөвхөн монгол домэйнуудаас мэдээллээ цуглуулж, түүнийгээ санах ойн хувьд үр ашигтай байхаар индексжүүлэн хадгалах үндэсний хайлтын системийн аалзны тоог сонгоход чиглэгдэж байгаа юм. Туршилтын үр дүнд түвшний нэвтрэлтийн алгоритмын дагуу 2 – 4 аалзыг трийд байдлаар ажиллуулснаар өгөгдсөн хугацаанд илүү олон хуудсанд хэсэлт хийж байгаа нь ажиглагдсан.

Түлхүүр үгс—хайлтын систем, аалз, түвшний нэвтрэлт алгоритм, хувирсан индекс, терм, документ, постинг;

I. Удиртгал

Интернет нь асар их мэдээллийн үүр уурхай бөгөөд хүмүүсийн хамгийн их хэрэглэдэг үйлдэл нь хайлтын хэрэгсэл ашиглан мэдээлэл хайх процесс юм. Хайлтын хэрэгслийг ашиглан яг хэрэгтэй мэдээллээ богино хугацаанд хайж олох нь тийм ч амар хялбар зүйл биш. Интернетэд ихэнх мэдээлэл англи хэл дээр байдаг тул бусад орны хэрэглэгчдийн хувьд мэдээлэл хайх нь хүндрэлтэй байдаг. Мөн, хайлт хийхэд уншиж барахааргүй олон үр дүн гарч ирдэг. Эдгээрээс улбаалан хайсан мэдээллээ олж чадахгүй байх нь ч бий.

Монгол агуулга бүхий вэб хуудсууд нь .mn болон бусад домэйн (.com, .org, .net ... г.м.) нэрээр интернэтэд байрладаг. Судалгаагаар идэвхтэй ажиллаж буй 1500 гаруй монгол вэб сайт байгаагаас 73.6% нь .mn домэйн нэрийн дор дотоодын интернэтийн үйлчилгээ үзүүлэгчдийн вэб сервер дээр байрлаж байна [1]. Энэхүү өгүүллийн хүрээнд, хайлтын систем маань үндэсний гэдэг утгаараа зөвхөн .mn домэйнуудыг хамарна. Мөн Я. Цэвэлийн “Монгол хэлний товч тайлбар толь” –ны 30,000 орчим монгол үгийг терм болгон урьдчилан бэлтгэсэн болно. Ингэснээр түгээмэл хэрэглэгддэг эдгээр үгнүүдийн хүрээнд хайлт хийгдэнэ гэсэн үг юм.

Өгүүллийн хоёрдугаар хэсэгт судлагдсан байдлыг товч дурдана. Гуравдугаар хэсэгт аалзны ажиллах зарчмын талаар өгүүлнэ. Дөрөвдүгээр хэсэгт түвшний нэвтрэлтийн алгоритм ашигласан хайлтын системийн загвараа танилцуулна. Тавдугаар хэсэгт, аалзны тоог сонгох туршилт, түүний үр дүнг харуулна. Зургаадугаар хэсэгт судалгааны ажлын талаарх дүгнэлтээ өгнө.

II. Судлагдсан байдал

МТШХХГазраас [2] гаргасан 2014 онд хийх ажлын жагсаалтад үндэсний хайлтын систем хөгжүүлэх асуудал багтсан байдаг [3]. Гэвч, хайлтын систем хөгжүүлэлтийн талаарх эрдэм, судалгааны ажлууд Монголд төдийлөн хийгдэж байгаагүй нь дээрх зорилтыг амжилттай хэрэгжүүлэхэд хүндрэл учруулж болзошгүй байна.

Харин гадаадад хайлтын системийн талаар олон тооны судалгааны ажил хийгдсэн байдаг. Манай судалгааны ажилтай зарим талаараа төстэй ажлуудаас дурдвал, А. Булуч, К. Маддури нар тархсан санах ойтой үед хайлтын түвшний нэвтрэлтийн алгоритмыг параллель байдлаар зохиомжлох судалгааг хийхдээ [4], процессорын тоогоо нэмэх замаар туршилт хийсэн байдаг.

Д. Феттерли нарын судлаачид түвшний нэвтрэлтийн алгоритмын үед хайлтын үр дүнгийн ашигтай байдлыг хэмжихдээ [5] хэсэлт хийгдсэн документэд аалз дахин зочилход агуулгын хувьд өмнөхөөсөө хэр өөрчлөгдсан байгааг нь харьцуулсан байна.

Харин Монголын нөхцөлд энэ талын судалгааны ажил төдийлөн хийгдээгүй учир бид энэхүү өгүүллээрээ хайлтын системийн суурь судалгаа болох аалзыг загварчлан хөгжүүлэх талаар авч үзнэ.

III. Аалзны ажиллах зарчим

Хайлтын систем нь интернетийн орчинд мэдээлэл хайх зориулалтаар загварчлагдсан програм хангамж юм [6]. Хайлтын систем нь вэб хуудсуудын мэдээллийг цуглуулах, түүнийг индексжүүлж хадгалах, цугларсан мэдээллээсээ хайлт хийх гэсэн гурван үндсэн ажиллагаанаас бүрддэг.

A. Вэб хэсэлт хийх

Хуудасны мэдээллийг авах үүргийг вэб хэсэгч (crawler) буюу аалз гүйцэтгэнэ. Аалз нь URL –уудаар хэсэхдээ, хуудсан дээрх бүх линкүүдийг $L = \{l_1, l_2, \dots, l_n\}$ тодорхойлж, тухайн зочилж буй хуудсаа зочилсон URL –ын жагсаалтад (crawl frontier) $cf = cf \cap l$ нэмдэг.

Өөрийн гэсэн тодорхой дүрэм гаргахын тулд бүртгэлтэй URL –ууд руугаа рекурс маягаар ханддаг. Аалзны хувьд өгөгдсөн хугацаанд хязгаарлагдмал тооны вэб хуудсыг татаж авдаг учраас тэдгээрт эрэмбэ тогтоох хэрэгтэй. Устсан аль эсвэл агуулга нь их өөрчлөгдсөн байна гэдэг нь өндөр эрэмбэтэй хуудас гэсэн үг юм.

Зарим нэг URL –ууд хоорондоо өөр ч, яг адил агуулгыг үзүүлэх тохиолдол байдаг. Жишээ нь, олон тооны HTTP GET параметрүүдтэй үед яг аль параметр нь дахин давтагдашгүй вэбийн агуулгыг буцааж байна вэ гэдэг нь шууд мэдэгдэхгүй байх нь бий. Ийм асуудлыг аалз нь URL нормалчлал хийх замаар шийддэг.

Аалзын зан байдал гэдэг бол тухайн аалзын баримтлах дүрмүүдийн нэгдэл юм. Үүнд:

- Тухайн сайтны аль хуудсуудыг татах вэ?
- Хуудасны өөрчлөлтийг хэзээ шалгах вэ?
- Хэт их ачаалж байгаа хуудсын холболтыг хэрхэн таслах вэ?
- Тархсан аалзуудыг хэрхэн зохицуулах вэ?

Хуудсын шинэлэг байдлын хэмжигдэхүүнүүдийн талаар авч үзье [7].

1) Шинэлэг байдал

N элемент бүхий $S = \{e_1, e_2, \dots, e_n\}$ хуудас татагдсан байна гэж үзье. Өгөгдсөн хугацаанд бүх *N* элементүүд шинэчлэгдэх ёстой. Гэвч бодит амьдрал дээр зөвхөн *M (<N)* нь л шинэчлэгдэж амждаг. Өгөгдсөн *t* хугацаан дах *S* –ийн шинэлэг байдлыг (freshness) тодорхойлбол $\mathrm{F(S;t)} = \mathrm{M/N}$ болно. Энэ нь хэрвээ бүх элементүүд шинэчлэгдсэн байх юм бол 1 гэсэн утга авах ба бүгд шинэчлэгдээгүй байх юм бол 0 гэсэн утга авна. *t* хугацаан дах e_i элементийн шинэлэг байдлыг тодорхойлбол:

$$F(e_i;t) = \begin{cases} 1, t \text{ хугацаанд } e_i \text{ шинэчлэгдсэн байвал} \\ 0, \text{шинэчлэгдээгүй бол} \end{cases} \quad (1)$$

t хугацаан дах *S* –ийн шинэлэг байдлыг тодорхойлбол:

$$F(S;t) = \frac{1}{N}\sum_{i=1}^{N} F(e_i;t) \quad (2)$$

2) Насжилт

Насжилт гэдэг хэмжигдхүүн нь хуудсыг хэр зэрэг хуучирсан бэ гэдгийг тодорхойлдог. *t* хугацаан дах e_i элементийн насжилтыг тодорхойлбол:

$$A(e_i;t) = \begin{cases} 0, \ t \text{ хугацаанд } e_i \text{ шинэчлэгдсэн байвал} \\ t - e_i \text{ засварын хугацаа, эсвэл} \end{cases} \quad (3)$$

t хугацаан дах *S* –ийн насжилтыг тодорхойлбол:

$$A(S;t) = \frac{1}{N}\sum_{i=1}^{N} A(e_i;t) \quad (4)$$

Аалзны зарчим нь хуудсуудын дундаж шинэлэг байдлыг аль болох их буюу хуудсуудын дундаж насжилтыг аль болох бага байлгахад оршино.

3) Робот хориглох протокол

Вэб сайтны эзэмшигч нь robots.txt файлын тусламжтайгаар аалзад өөрийн сайтын талаарх удирдамжийг өгдөг [8]. Хэрвээ энэхүү файл үүсгэгдээгүй бол аалз сайтын админ ямар ч дүрмээр хангахыг хүсээгүй ба сайтаа бүхэлд нь хэсэлт хийлгүүлэхийг хүсч байгаа юм байна гэж ойлгоно.

```
User-agent: *
Disallow: /
```

User-agent нь хайлтын системүүдийг зааж, Disallow нь сайтны аль аль хуудсанд хэсэлт хийж болохгүйг заана.

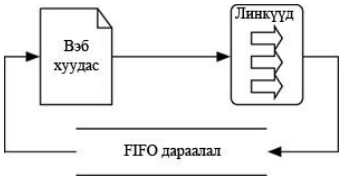
4) Түвшний нэвтрэлтийн алгоритм

Хайлтын алгоритмууд дотроос энгийнд тооцогддог түвшний нэвтрэлт (Breadth-First Search) нь 1994 оны эхээр зохиогдсон. Давуу тал нь дараалал ашигладаг. Түвшний нэвтрэлт нь LIFO –гийн оронд FIFO зарчмаар ажилладаг. Хязгаар нь дүүрэх үед хэсэлт хийгдсэн хуудаснаас ердөө 1 линк л нэмдэг [9],[10] байна.

```
insert_in_queue(seeds)
while more_links_in_queue do
    link := dequeue_first_inserted
    doc := fetch(link)
    out_links := extract_links(doc)
    insert_in_queue_last_pos(out_links)
end while
```

Код 1. Түвшний нэвтрэлт ашиглан вэб хэсэлт хийх псевдо код

Дээрх псевдо кодыг зургаар дүрсэлбэл:



Зураг 1. Түвшний нэвтрэлт ашиглан вэб хэсэлт хийх

B. Мэдээллийг индексжүүлэх

Аалз нь вэб хуудаснаас авсан мэдээллүүдээ индексжүүлэн хадгалдаг. Зорилго нь хурд болон асуулгатай холбоотой документийг олох гүйцэтгэлийг

оновчилох явдал юм. Ингэхдээ үндсэн үгсийг тодорхой зааж өгсөн тусгай тэмдэгтүүдээр салган, термүүдийн (term) олонлог $T = \{t_1,t_2...,t_n\}$ болгон хадгална. Тусгай тэмдэгтүүд гэдэгт нь хоосон зай, тодорхойлох цэг (:), цэг таслал (;), таслал (,), цэг (.), давхар хашилт (“), дан хашилт (‘), налуу зураас (/) ... г.м. байдаг. Үндсэн үгсийг хадгална гэсний учир нь үг болгон утга илэрхийлдэггүй. Жишээ нь, ба, бөгөөд, нь, л, ... г.м. тодорхой утгагүй үгнүүд байдаг. Эдгээр үгсийг стопвордс [11] гэж нэрлэдэг.

Хувирсан индекс (inverted index) нь мэдээлэл таталтын (Information Retrieval) үед өргөн хэрэглэгддэг өгөгдлийн бүтэц юм [12]. Документын цуглуулга $D = \{d_1, d_2, \dots, d_n\}$ дах терм болгон дээр хувирсан индексийн жагсаалт $t_i \rightarrow p(t_i)$ –г үүсгэнэ. Энэхүү жагсаалт нь тухайн термийг агуулсан документуудын жагсаалт юм. Хувирсан индексийн үндсэн санааг зураг 2 –т үзүүлсэн болно. Энэ нь документаас термүүд буюу үгсийн сангийн олонлогийг үүсгэж хадгална. Үгсийн санг заримдаа лексикон гэж нэрлэх нь бий.

Тухайн нэг термийг агуулсан документуудын жагсаалтыг постинг гээд $p(t_i) = \{d_j \mid t_i \in d_j\}$ гэж илэрхийлнэ. Бүх термүүдийн постингуудын жагсаалтыг бүхэлд нь постинг лист гээд $pl = \{p(t_1), .., p(t_n)\}$ гэж илэрхийлнэ. Термүүд нь үсгийн дарааллаар эрэмбэлэгдэж, постинг лист нь документын дугаараар эрэмбэлэгдсэн байгааг зураг 2 –оос харж болно. Энэ нь хожим бит үйлдэл хийх, хайлт хийх зэрэгт хэрэг болдог.

Документын цуглуулгатай байхад, document identifier (docID) гэх документ болгонд дахин давтагдашгүй сериал дугаарыг өгөх нь чухал ач холбогдолтой байдаг. Энэ нь индексийг байгуулж байх үед, шинэ документ тааралдвал түүнд оноох дараагийн тоон утгыг амархан тодорхойлох боломжийг олгоно.

Зураг 2. Хувирсан индексийн жишээ

Хэрвээ хувирсан индексээ өргөжүүлж, сайжруулъя гэвэл документ болгоны хувьд тухайн терм аль байршилд $p(t_i) = \{[d_j; pos(t_i)] \mid t_i \in d_j\}$ дээр байсныг нь хадгалж болно [13]. Мөн түүнчлэн термийн байршлын оронд документ болгон дээр хэдэн удаа орсон тоо буюу давтамжийг (term frequency) нь хадгалж $p(t_i) = \{[d_j; tf(t_i)] \mid t_i \in d_j\}$ болно. Зарим хайлтын системүүд 2 ч хувирсан индекс үүсгэсэн байдаг. Нэгэнд нь зөвхөн документ жагсаалтаа, нөгөөд нь термийн байршлын жагсаалтыг хадгалсан байдаг. Хайлтын түлхүүр үгнүүд хэдий чинээ энгийн байна тэр хэмжээний бага документ жагсаалт дээр боловсруулалт хийгдэнэ. Зарим хайлтын системийн хувьд мета өгөгдөл эсвэл өөр олон нэмэлт мэдээллийг хадгалж индексээ үүсгэсэн байдаг. Эдгээр мэдээллүүд нь хожим хайлтын үр дүндээ эрэмбэ тогтооход ашиглагддаг.

C. Хайлтын асуулгын төрөл

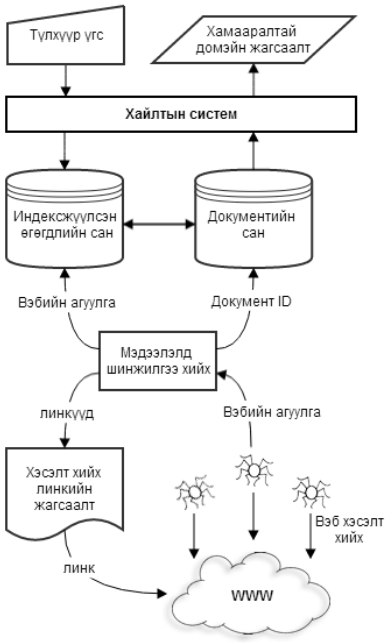
Чөлөөт текстэн асуулга гэдэг нь хоосон зайгаар тусгаарлагдсан үгнүүдийн дараалал бөгөөд үг болгоны голд нуугдсан логик AND үйлдэл байгаа гэж үзэх замаар хайлтыг хэрэгжүүлдэг. *N* ширхэг бүхий түлхүүр үгээр хайлт хийсэн гэж үзвэл, постингуудын огтлолцолоор $\{p_1 \cap p_2 \cap \dots \cap p_n\}$ хайлтын үр дүн гарна.

```
terms ← SortByIncreasingFrequency(t₁,…,tₙ)
result ← postings(first(terms))
terms ← rest(terms)
while terms ≠ NIL and result ≠ NIL do
  result ← INTERSECT(result, postings(first(terms)))
  terms ← rest(terms)
return result
```

Код 2. *N* ширхэг постингийн огтлолцол тооцоолох псевдо код

IV. СИСТЕМИЙН АРХИТЕКТУР

Бид өөрсдийн хөгжүүлж буй хайлтын системийнхээ архитектурыг зураг 3 дээрх байдлаар хэрэгжүүлсэн. Хамгийн түрүүнд аалзнууд маань эхлэл байдлаар зааж өгсөн нэг эсвэл хэд хэдэн хуудаснаас хэсэлтээ эхэлнэ. Хэссэн хуудасныхаа агуулгыг авч тэрхүү мэдээлэл дээрээ шинжилгээ хийнэ. Шинжилгээний үр дүнд 3 үйлдэл хийгдэхийн эхнийх нь тухайн хуудсандаа документын дугаар олгон документийн сандаа хадгална. Дараа нь документийн терм болгоныхоо постингийг шинэчлэн, индекстээ хадгална. Эцэст нь хуудсан дах линкүүдээс сарын хугацаанд хэсэлт хийгдээгүй линкүүдийг FIFO дараалалдаа нэмнэ. Түвшний нэвтрэлтийн алгоритмын дагуу аалзнууд ажлаа цааш үргэлжлүүлнэ.



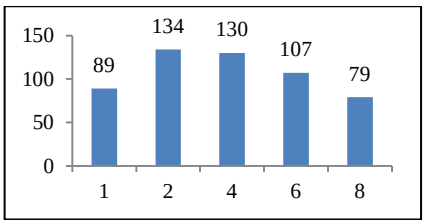
Зураг 3. Системийн архитектур

Хайлтын системээс ямар нэгэн түлхүүр үгүүдээр хайлт хийхэд, индексжүүлсэн өгөгдлийн сангаасаа шүүлт хийн, огтлолцолоор документын дугааруудыг нь гаргаж авна. Үүний дараа документийн сангаас дугааруудад харгалзах линкүүдийг нь авч дэлгэцэнд хайлтын үр дүнг хэвлэнэ.

V. Туршилт, үр дүн

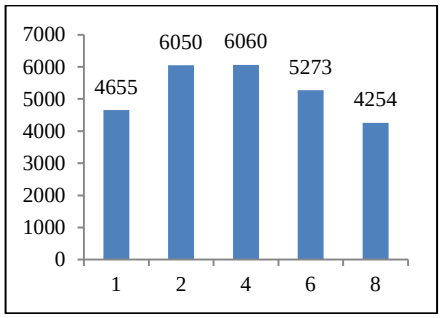
Туршилтыг Intel Core i5 (2.40GHz) процессортой, 8GB –ын рам бүхий компьютер дээр 3Mbps интернетийн хурдтайгаар гүйцэтгэсэн болно.

Өгөгдсөн хугацаанд хичнээн аалзыг трийд байдлаар ажиллуулахад хэдэн хуудсанд хэсэлт хийж байгааг шинжлэхийн тулд дараах туршилтыг бид хийсэн. 10 минутын хугацаанд ажиллуулахад гарсан үр дүнг зураг 4 –өөс харж болно. Баганын дагуу аалзны тоо, мөрийн дагуу хэсэлт хийсэн хуудсын тоог дүрслэв.



Зураг 4. Хэсэлт хийгдсэн хуудсын тоо

Мөн 10 минутын хугацаанд аалзыг ажиллуулахад хичнээн хуудас дараалалд нэмэгдэж байгааг туршсан үр дүнг зураг 5 –д харуулав.



Зураг 5. Хэсэлт хийх линкний тоо

VI. Дүгнэлт

Хайлтын системийн хувьд ердөө ганц аалзыг ажиллуулан мэдээлэл цуглуулах нь ажиллагааны хувьд удаан байдаг. Харин хэд хэдэн аалзыг зэрэг ажиллуулснаар их хэмжээний мэдээллийг богино хугацаанд цуглуулж чадна. Бид яг тэр тохиромжтой аалзны тоог сонгохын тулд дээрх туршилтыг хийсэн. Туршилтын үр дүнгээс харахад, хэт олон аалзыг ажиллуулахад эсрэгээрээ хугацаа их зарцуулж байгаа нь харагдаж байна. Харин 2 – 4 аалзыг трийд байдлаар ажиллуулснаар хамгийн өндөр үзүүлэлт буюу 10 минутанд дунджаар 130 – 134 хуудсанд хэсэлт хийж, дунджаар 6050 – 6060 шинэ линк дараалалд нэмэгдэж байсан.

Хайлтын систем болгоны хувьд серверын үзүүлэлт нь харилцан адилгүй учир орчин орчиндоо энэхүү туршилтыг хийн аалзны тоогоо сонгох нь зөв юм.

Дэлхийд тэргүүлэгч хайлтын системүүдийн дотоод ажиллагаа нь асар олон алгоритмуудыг нэгтгэсэн нарийн зохион байгуулалт бүхий нүсэр архитектуртай байдаг. Харин Монголын нөхцөлд энэ талын судалгааны ажил төдийлөн хийгдээгүй учир бид энэхүү өгүүллээрээ хайлтын системийн суурь судалгааны нэг болох аалзны тоог сонгох судалгаа болоод, туршилтыг хийж гүйцэтгэлээ.

Тиймээс цаашид энэхүү судалгааны ажлыг үргэлжлүүлэн аалзны хэсэлт хийх үеийн ажиллагааг илүү ухаалгаар зохицуулах, илүү хурдан хувирсан индексээ шинэчлэх, олон алгоритмыг хосолсон байдлаар гүйцэтгэл өндөртэйгөөр шийдэх боломжийг судлах зэрэг шаардлага урган гарч байна. Мөн цаашлаад хайлтын системийн хувьд авч үзвэл, термүүд дээр үг зүйн шинжилгээ хийх, цугларсан мэдээллээ шахаж санах ойг үр ашигтай зохион байгуулах г.м. өөр бусад сэдвүүдийг хамарснаар үндэсний хайлтын систем маань илүү хүчирхэг систем болж чадна гэж бид үзэж байна.

Ашигласан материал

[1] [1] *“Цахим Монгол” үндэсний хөтөлбөр*. Улаанбаатар: Засгийн газар, 2005.

[2] [2] “Мэдээллийн технологи, Шуудан, Харилцаа холбооны газар.” [Online]. Available: http://ict.mn/. [Accessed: 17-Apr-2014].

[3] [3] “2014 онд хийх ажлын санал,” *Мэдээллийн технологи, Шуудан, Харилцаа холбооны газар*, 2014. [Online]. Available: http://www.itpta.gov.mn/mn/249. [Accessed: 17-Apr-2014].

[4] [4] A. Buluç and K. Madduri, “Parallel Breadth-First Search on Distributed Memory Systems,” in *International Conference for High Performance Computing, Networking, Storage and Analysis on - SC ’11*, 2011, p. 1.

[5] [5] M. Boughanem, C. Berrut, J. Mothe, and C. Soule-Dupuy, Eds., *Advances in Information Retrieval*, vol. 5478. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 388–399.

[6] [6] “Search engine,” *Encyclopedia Britannica*. [Online]. Available: http://www.britannica.com/EBchecked/topic/1017484/search-engine. [Accessed: 06-Apr-2014].

[7] [7] J. Cho and H. Garcia-Molina, “Synchronizing a database to improve freshness,” *ACM SIGMOD Int. Conf. Manag. Data*, vol. 29, no. 2, pp. 117–128, Jun. 2000.

[8] [8] “About /robots.txt,” *The Web Robots Pages*. [Online]. Available: http://www.robotstxt.org/robotstxt.html. [Accessed: 06-Apr-2014].

[9] [9] I. G. Dorado, “Focused Crawling: algorithm survey and new approaches with a manual analysis,” Lund University, 2008.

[10] [10] F. Menczer, G. Pant, and P. Srinivasan, “Topical Web Crawlers: Evaluating Adaptive Algorithms,” *ACM Trans. Internet Technol.*, vol. 4, no. 4, pp. 378–419, Nov. 2004.

[11] [11] M. R. Falahati and Q. Fumani, “The Concept of Stopwords in Persian Chemistry Articles: A Discussion in Automatic Indexing,” *Glossa*, vol. 4, no. 1, pp. 173–191, Dec. 2008.

[12] [12] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge University Press; 1 edition, 2008, p. 496.

[13] [13] A. K. Mahapatra and S. Biswas, "Inverted indexes: Types and techniques," *Int. J. Comput. Sci. Issues*, vol. 8, no. 4, pp. 384–392, Jul. 2011.