UNIVERSITÄT LEIPZIG

Institut für Informatik
Lehrstuhl für Rechnernetze und Verteilte Systeme

Praktikum Computational Advertising
Prof. Dr.-Ing. Christoph Lindemann
Sommersemester 2011

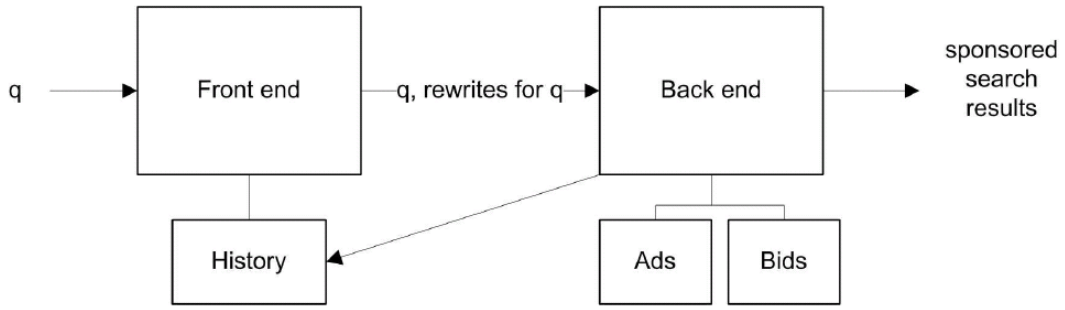
Betreuer: **Dipl.-Inform. Jan Friedrich, friedrich@rvs.informatik.uni-leipzig.de**
Dipl.-Inform. Michael Petrifke, petrifke@rvs.informatik.uni-leipzig.de

Aufgabe 2: Datenbank Backend

Übersicht

Nachdem Sie sich in der ersten Aufgabe mit den Grundlagen und dem Management eines AdServers vertraut machen durften konzentrieren Sie sich im Folgenden auf komplexere Methoden von Sponsored Search. Sponsored Search Ad-Retrieval-Systeme bestehen aus zwei Hauptkomponenten:

- **Frontend**: Zuständig für Query Rewriting. Dabei werden Teile der gestellten Suchanfrage ersetzt (z.B. um das Thema zu erweitern oder einzuschränken) bzw. die Suchanfrage durch alternative Queries und Features ergänzt. Das Ergebnis dieses 1.



  Schrittes wird an das Backend weitergereicht. Zusätzlich werden demographische Merkmale des Nutzers analysiert, falls diese nicht bekannt sind.

- **Backend**: Verwendet die Extended Queries um passende Ads aus dem Ad-Korpus auszuwählen (z.B. Exact Match der Query mit den zu den Ads hinterlegten Bid Phrases). Die Kandidaten werden nach definierten Kriterien geordnet (z.B. vermuteter Umsatz) und schließlich als Antwort auf die Query ausgegeben. Dazu wird über statistische Werte wie Anzahl der Impressions und Klicks Buch geführt.

Im Laufe des Praktikums wird sowohl ein Backend als auch ein Frontend entworfen und implementiert. Eine wichtige Rolle wird dabei der SimRank-Algorithmus [1] spielen (bzw. dessen optimierte Variante SimRank++) sowie Latent Dirichlet Allocation.

Die beiden Komponenten sollen dabei in jeweils einer eigenen C++-Bibliothek gekapselt werden um später als Modul des (gedachten) Web-Framework *AdWorks* arbeiten zu können. Im Praktikum wird dies eine triviale Konsolen-Applikation übernehmen.

Die Liste der Ads mit ihren Bid Phrases wird dabei in einer Datenbank abgelegt, auf die das Backend jederzeit zugreifen kann.

Im Frontend wird einmalig für Queries eine Liste „ähnlicher“ Queries erzeugt die derart gespeichert werden, dass ein späteres Lesen effizient möglich ist.

Beschreibung der Datenstrukturen

Die Interfaces der Klassen, die das Front- und Backend implementieren, lauten wie folgt:

```
class IQueryResult
{
	public:
		virtual std::string getTitle() const = 0;
		virtual std::string getCreative() const = 0;
		virtual uint32_t getAdID() const = 0;
};
enum Gender	{GENDER_NA, GENDER_MALE, GENDER_FEMALE};
enum Age	{AGE_NA, AGE_TEEN, AGE_YOUNG, AGE_OLD};
class IUser
{
	public:
		Gender getGender() const = 0;
		Age getAge() const = 0;
		std::list<std::string> getBrowsingHistory() const = 0;

};
```

```
class IFrontEnd
{
	public:
		// ermittelt das am besten passende Ad und erhöht die Impressionsanzahl
		virtual IQueryResult matchAd(std::string query, const IUser* user = NULL,  bool* foundAd = NULL) = 0;
		// ermittelt die Landing Page des Ads adID und erhöht seine Klickanzahl
		virtual std::string getAdURL(uint32_t adID) = 0;
		// Verwende des Log in file um den Klick-Graphen zu bauen und
		// ähnliche Queries zu finden
		virtual bool analyzeClickGraph(const std::string& file) = 0;
		// Berechne Wahrscheinlichkeiten demographischer Merkmale von
		// Webseiten
		virtual bool analyzeDemographicFeatures(const std::string& userFile, const std::string& visitFile) = 0;
		// setze das zu verwendende Backend
		virtual void setBackend(IBackEnd* backend) = 0;
};
class IBackEnd
{
	public:
		// siehe: IFrontEnd::matchAd
		virtual IQueryResult matchAdRewrites(std::list<std::string> rewriteList, const IUser* user = NULL,  bool* foundAd = NULL) = 0;
		// siehe: IFrontEnd::getAdURL
		virtual std::string getAdURL(uint32_t adID) = 0;
		// Datenbank mit Ads und Bid Phrases initialisieren
		virtual bool initDatabase(const string& adFile, const string& bidPhraseFile) = 0;
};
```

Das Backend pflegt eine Datenbank mit folgenden Tabellen:

Ads

Primary Key(AdID), Titel, Slogan, URL, Anzahl Impressions, Anzahl Klicks, Gender, Age

Queries

Primary Key(AdID, Bid Phrase), Gebot

Der folgende Ablauf liegt der Ad-Auslieferung in AdWorks zugrunde:

1. Nutzer verwendet Query q
2. Webseite w fragt AdWorks nach dem passenden Ad a für q
3. AdWorks-Frontend bestimmt Rewrites rw(q) für q und fragt Backend nach a
4. AdWorks-Backend nutzt rw(q) um passende Ads ma(rw(q)) zu bestimmen
5. AdWorks-Backend bewertet die gefundenen Ads und ordnet sie entsprechend
6. AdWorks liefert Titel und Werbeslogan, sowie AdID des besten Ads an w und erhöht die Impressionanzahl von a
7. Klickt der Nutzer auf das Ad wird AdWorks nach der Landing Page gefragt (getAdURL) und speichert im Backend den Klick für Ad a

Bitte dokumentieren Sie ihren Code angemessen. Dazu gehören Kommentare für jede Funktion einer Klasse sowie an Stellen im Code deren Funktionalität nicht offensichtlich ist. Verwenden Sie wo möglich einigermaßen deskriptive Variablennamen.

Aufgabe

Auf der Praktikumswebsite finden Sie einen Datensatz mit Ads und Queries. Implementieren Sie nun hierzu das Backend mit der beschriebenen Funktionalität. Verwenden Sie dazu ausschließlich C++ und mysql. Zusätzlich zum Standardsprachumfang von C++ (ISO 2003 + TR1) und den Datenbank-Bibliotheken sind nur Boost und Qt erlaubt.

Das Matching der Ads erfolgt in dieser ersten Aufgabe trivial per Exact Match: Die Extended Queries die das Frontend liefert (an `matchAdRewrites` übergeben werden) werden mit einem einfachen case-insensitiven Stringvergleich mit den Bid Phrases der Ads verglichen. Sollten Nutzer-Informationen wie Gender und Age gegeben sein, dürfen nur Ads berücksichtigt werden, die diesen Kriterien entsprechen.

Das Ranking der Ads erfolgt anhand der gemessenen Click Through Rate der Ads und den entsprechenden Geboten. Somit ergibt sich als Bewertungsfunktion: CTR*Gebot (erwarteter Umsatz). Das Ad mit dem höchsten erwarteten Umsatz wird von der `matchAdRewrites` Methode zurückgeliefert. Die CTR soll zunächst sehr einfach berechnet werden: Klicks / Impressions.

Wie sie vermutlich dem Ausdruck im obigen Text entnehmen konnten sind viele Teile des Backend austauschbar: Ranking, Matching, CTR-Berechnung. Gestalten Sie ihre Implementierung dementsprechend (z.B. Function Objects, Factory Design Pattern, ...). Im Code-Review müssen Sie zu dem Part Stelllung beziehen.

Um ihre Implementierung zu testen verwenden Sie ihre Backend-Bibliothek in einer Konsolen-Anwendung die folgende Programm-Parameter akzeptiert:

```
reload <adfile> <queryfile>

      Neuladen der Backend DB aus den gegebenen Dateien

matchad -q <query> [-q <query>]* [-a <age>] [-g <gender>]

      Eine Liste von Queries einlesen sowie die optionalen Nutzer-
      Eigenschaften (gender: m,f; age: 0,1,2) und das entsprechende
      optimale Ad für diese Rewrite Menge ausgeben (newline-separiert
      Titel, Werbeslogan und adID)

visit <adID>

      Erhöhe die Klick-Zahl von Ad um eins (getAdURL) und gib die URL
      aus
```

Abgabe

Bitte schicken Sie pro Gruppe genau eine Lösung bis zum **27.05.11, 23:59 Uhr** per Email an Jan Friedrich (friedrich*@rvs.informatik.uni-leipzig.de*). Die Abgabe sollte den gesamten Quellcode enthalten sowie ihre Build-Skripte (i.e. autotools, cmake, qmake, Custom Makefile, vcproj, ...). Geben Sie auch an in welcher Umgebung die Aufgabe angefertigt wurde (Betriebssystem, Compiler, inkl. Version). Wir empfehlen die Aufgabe mit GCC unter einem Unix-kompatiblem Betriebssystem oder der CygWin-Umgebung [3] zu bearbeiten. Sollten Sie dennoch unter Windows arbeiten wollen, können wir bei Problemen eventuell nur begrenzt Hilfe anbieten. Zudem müssen Sie zum Reviewgespräch einen Laptop mitbringen, auf dem der Code sowohl kompiliert als auch ausgeführt werden kann.

Die Termine für die Reviewgespräche sind am **30.05.**, **31.05.** und **01.06.** jeweils von **15 bis 17 Uhr**. Kommen Sie bitte innerhalb dieser Zeiten, wenn möglich mit eigenem Laptop, unangemeldet mit beiden Gruppenmitgliedern zu Jan Friedrich (Raum 04-06). Die Reviews werden ca. 15 Minuten dauern.

Referenzen

[1] Simrank++: Query Rewriting through Link Analysis of the Click Graph, I. Antonellis, H. Garcia-Molina, C. Chang, Proc. VLDB Endowment, 2008

[2] http://www.cplusplus.com/reference/

[3] http://www.cygwin.com/

Informationen zur Abgabe

- **Siehe Webseite: http://rvs.informatik.uni-leipzig.de/de/lehre/SS11/praktika/ca/**

Viel Erfolg!