The Role of Documents vs. Queries in Extracting Class Attributes from Text

Marius Paşca
Google Inc.
1600 Amphitheatre Parkway
Mountain View, California 94043
mars@google.com

Benjamin Van Durme*
University of Rochester
734 Computer Studies Building
Rochester, New York 14627
vandurme@cs.rochester.edu

Nikesh Garera*
Johns Hopkins University
3400 North Charles Street
Baltimore, Maryland 21218
ngarera@cs.jhu.edu

ABSTRACT

Challenging the implicit reliance on document collections, this paper discusses the pros and cons of using query logs rather than document collections, as self-contained sources of data in textual information extraction. The differences are quantified as part of a large-scale study on extracting prominent attributes or quantifiable properties of classes (e.g., *top speed*, *price* and *fuel consumption* for *CarModel*) from unstructured text. In a head-to-head qualitative comparison, a lightweight extraction method produces class attributes that are 45% more accurate on average, when acquired from query logs rather than Web documents.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing; I.2.7 [**Artificial Intelligence**]: Natural Language Processing; I.2.6 [**Artificial Intelligence**]: Learning; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms

Algorithms, Experimentation

Keywords

Knowledge acquisition, class attribute extraction, textual data sources, query logs

1. INTRODUCTION

To acquire useful knowledge in the form of entities and relationships among those entities, existing work in information extraction taps on a variety of textual data sources. Whether domain-specific (e.g., collections of medical articles or job announcements) or general-purpose (e.g., news corpora or the Web), textual data sources are always assumed to be available as document collections [12]. This reliance on document collections is by no means a weakness. On the contrary, the availability of larger document collections is instrumental in the trend towards large-scale information extraction. But as extraction experiments on terabyte-sized document collections become less rare [4], they have yet to capitalize on an alternative resource of textual information (i.e., search queries) that millions of users generate daily, as they find information through Web search.

Characteristic	Data Source	
	Doc. Sentences	Queries
Type of medium	text	text
Purpose	convey info.	request info.
Available context	surrounding text	self-contained
Average quality	high (varies)	low
Grammatical style	natural language	keyword-based
Average length	12 to 25 words	2 words

Table 1: Textual documents vs. queries as data sources for information extraction

Table 1 compares document collections and query logs as potential sources of textual data for information extraction. On average, documents have textual content of higher quality, convey information directly in natural language rather than through sets of keywords, and contain more raw textual data. In contrast, queries are usually ambiguous, short, keyword-based approximations of often-underspecified user information needs. An intriguing aspect of queries is, however, their ability to indirectly capture human knowledge, precisely as they inquire about what is already known. Indeed, users formulate their queries based on the commonsense knowledge that they already possess at the time of the search. Therefore, search queries play two roles simultaneously. In addition to requesting new information, they also indirectly convey knowledge in the process. If knowledge is generally prominent or relevant, people will eventually ask about it [13], especially as the number of users and the quantity and breadth of the available knowledge increase, as it is the case with the Web as a whole. Query logs convey knowledge through requests that may be answered by knowledge asserted in expository text of document collections.

This paper is the first comparison of Web documents and Web query logs as separate, self-sufficient data sources for information extraction, through a large-scale study on extracting prominent attributes or quantifiable properties of classes (e.g., *top speed*, *price* and *fuel consumption* for *CarModel*) from unstructured text. The attributes correspond to useful

*Contributions made during internships at Google.

CIKM'07, November 6–8, 2007, Lisboa, Portugal.

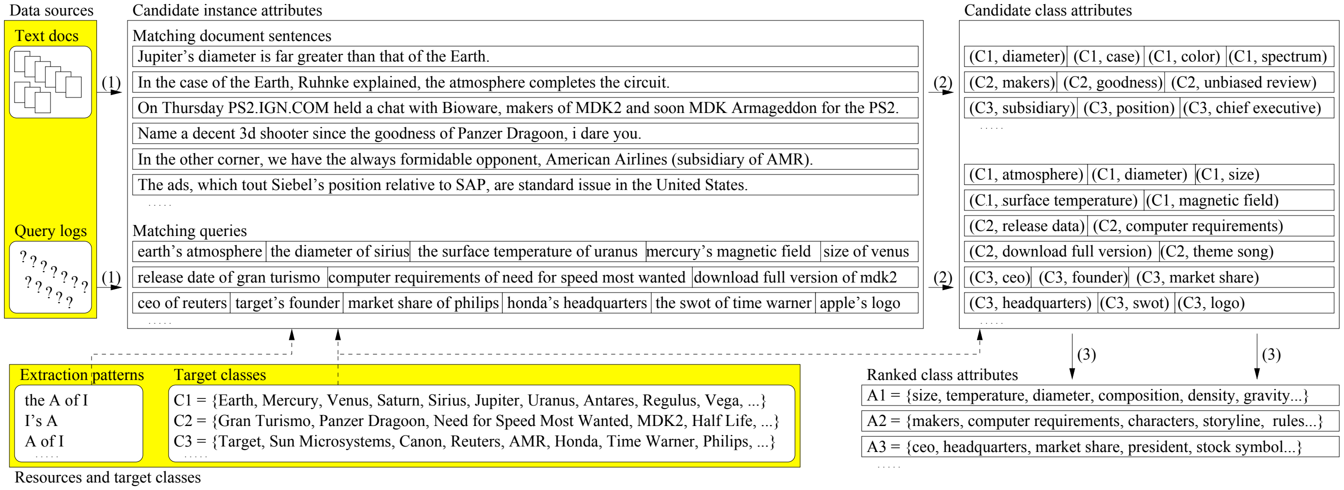
Data sources
Text docs
Query logs
Candidate instance attributes
Matching document sentences
Jupiter's diameter is far greater than that of the Earth.
In the case of the Earth, Ruhnke explained, the atmosphere completes the circuit.
On Thursday PS2.IGN.COM held a chat with Bioware, makers of MDK2 and soon MDK Armageddon for the PS2.
Name a decent 3d shooter since the goodness of Panzer Dragoon, i dare you.
In the other corner, we have the always formidable opponent, American Airlines (subsidiary of AMR).
The ads, which tout Siebel's position relative to SAP, are standard issue in the United States.
Matching queries
earth's atmosphere | the diameter of sirius | the surface temperature of uranus | mercury's magnetic field | size of venus
release date of gran turismo | computer requirements of need for speed most wanted | download full version of mdk2
ceo of reuters | target's founder | market share of philips | honda's headquarters | the swot of time warner | apple's logo
(1)
(2)
(3)
Candidate class attributes
(C1, diameter) (C1, case) (C1, color) (C1, spectrum)
(C2, makers) (C2, goodness) (C2, unbiased review)
(C3, subsidiary) (C3, position) (C3, chief executive)
(C1, atmosphere) (C1, diameter) (C1, size)
(C1, surface temperature) (C1, magnetic field)
(C2, release data) (C2, computer requirements)
(C2, download full version) (C2, theme song)
(C3, ceo) (C3, founder) (C3, market share)
(C3, headquarters) (C3, swot) (C3, logo)
Extraction patterns
the A of I
I's A
A of I
Target classes
C1 = {Earth, Mercury, Venus, Saturn, Sirius, Jupiter, Uranus, Antares, Regulus, Vega, ...}
C2 = {Gran Turismo, Panzer Dragoon, Need for Speed Most Wanted, MDK2, Half Life, ...}
C3 = {Target, Sun Microsystems, Canon, Reuters, AMR, Honda, Time Warner, Philips, ...}
Resources and target classes
Ranked class attributes
A1 = {size, temperature, diameter, composition, density, gravity...}
A2 = {makers, computer requirements, characters, storyline, rules...}
A3 = {ceo, headquarters, market share, president, stock symbol...}

Figure 1: Overview of data flow during class attribute extraction from textual data sources

relations among classes, which is a step beyond mining instances of a fixed target relation that is specified in advance. More importantly, class attributes have several applications. In knowledge acquisition, they represent building blocks towards the appealing, and yet elusive goal of constructing large-scale knowledge bases automatically [17]. They also constitute topics (e.g., *radius*, *surface gravity*, *orbital velocity* etc.) to be suggested automatically, as human contributors manually add new entries (e.g., for a newly discovered celestial body) to resources such as Wikipedia [16]. In open-domain question answering, the attributes are useful in expanding and calibrating existing answer type hierarchies [9] towards frequent information needs. In Web search, the results returned to a query that refers to a named entity (e.g., *Pink Floyd*) can be augmented with a compilation of specific facts, based on the set of attributes extracted in advance for the class to which the named entity belongs. Moreover, the original query can be refined into semantically-justified query suggestions, by concatenating it with one of the top extracted attributes for the corresponding class (e.g., *Pink Floyd albums* for *Pink Floyd*).

The remainder of the paper is structured as follows. Section 2 introduces a method for extracting quantifiable attributes of arbitrary classes from query logs and Web documents. The method relies on a small set of linguistically motivated extraction patterns to extract candidate attributes from sentences in documents, and from entries in query logs respectively. Section 4 is the first head-to-head comparison of the quality of information (in this case, class attributes) extracted from document collections vs. query logs. Results are described comparing attributes extracted from approximately 100 million Web documents vs. 50 million queries.

2. EXTRACTION OF CLASS ATTRIBUTES

2.1 Overview

The extraction method is designed to be simple, general and generic, allowing for robustness on large amounts of noisy data, the ability to operate on a wide range of open-domain target classes, and most importantly ensuring a fair, apple-to-apple comparison of results obtained from query logs vs. Web documents. As shown in Figure 1, given a set of target classes, the extraction method identifies relevant sentences and queries, collects candidate attributes for various instances of the classes, and ranks the candidate attributes within each class.

2.2 Pre-Processing of Textual Data Sources

The linguistic processing of document collections is limited to tokenization, sentence boundary detection and part-of-speech tagging. Comparatively, the queries from query logs are not pre-processed in any way. Thus, the input data source is available in the form of part-of-speech tagged document sentences with document collections, or query strings in isolation of other queries in the case of query logs.

2.3 Specification of Target Classes

Following the view that a class is a placeholder for a set of instances that share similar attributes or properties [7], a target class (e.g., *HeavenlyBody*) for which attributes must be extracted is specified through a set of representative instances (e.g., *Venus*, *Uranus*, *Sirius* etc.). It is straightforward to obtain high-quality sets of instances that belong to a common, arbitrary class by either a) acquiring a reasonably large set of instances through bootstrapping from a small set of manually specified instances [2]; or b) selecting instances from available lexicons, gazetteers and Web-derived lists of names; or c) acquiring the instances automatically from a large text collection (including the Web), based on the class name alone [19]; or d) selecting prominent clusters of instances from distributionally similar phrases acquired from a large text collection [10]; or e) simply assembling instance sets manually, from Web-based lists.

2.4 Selection of Class Attributes

For robustness and scalability, a small set of linguistically-motivated patterns extract potential pairs of a class instance and an attribute from the textual data source. Although the patterns are the same, their matching onto text is slightly different on document sentences vs. queries.

With document sentences, each pattern is matched partially against the text, allowing other words to occur around the match. When a pattern matches a sentence, the outer boundaries of the match are checked and computed heuris-

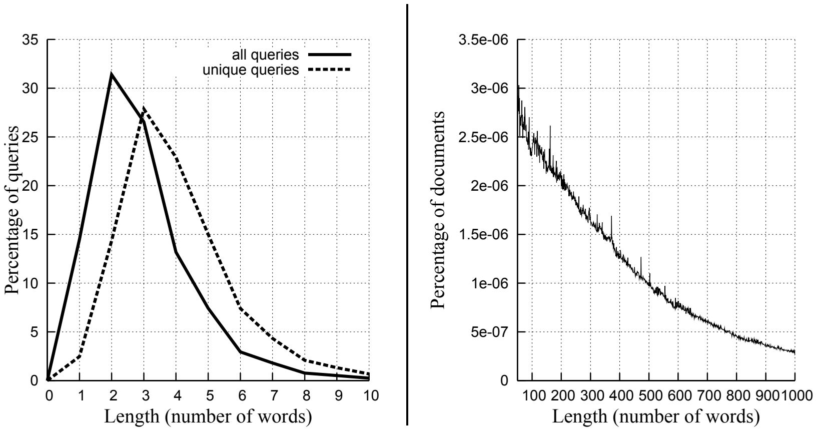

Figure 2: Percentage of input queries of various lengths, computed over all queries (including duplicates) and over unique queries (first graph); percentage of input documents of various lengths (second graph)

tically based on the part of speech tags. For example, one of the extraction patterns matches the sentence *"Human activity has affected Earth's surface temperature during the last 130 years"* via the instance *Earth*, producing the candidate attribute *surface temperature*. In contrast, although the sentence *"The market share of France Telecom for local traffic was 80.9% in December 2002"* matches one of the patterns via the instance *France*, it does not produce any attribute because the instance is part of a longer sequence of proper nouns, namely *France Telecom*.

With queries, patterns are matched fully, with no additional words allowed around the match and with no additional checks. Thus, the outer boundaries of the candidate attributes are approximated trivially through an extremity of the query, producing the candidate attributes *size* and *download full version* from the queries *size of venus* and *download full version of mdk2* respectively.

With the exception of how the patterns are matched onto text (i.e., fully vs. partially), the extraction method operates identically on both documents vs. queries.

2.5 Ranking of Class Attributes

A candidate attribute selected for an instance from the input text (e.g., *diameter* for *Jupiter* from the first sentence, or *atmosphere* for *earth* from the first query in Figure 1) is in turn a candidate attribute of the class(es) to which the instances belong. For example, *diameter* and *atmosphere* become associated to the class $\mathcal{C}_1$ in Figure 1 because *Jupiter* and *Earth* are instances of that class. The score of a candidate attribute $\mathcal{A}$ within a class $\mathcal{C}$ is higher if the attribute is associated to more of the instances $\mathcal{I}$ of $\mathcal{C}$:

$$S_{freq}(Att(\mathcal{C},\mathcal{A})) = \frac{|\{\mathcal{I}_i : ((\mathcal{I}_i \in \mathcal{C}) \wedge Att(\mathcal{I}_i,\mathcal{A}))\}|}{|\{\mathcal{I}_j : \mathcal{I}_j \in \mathcal{C}\}|}$$

Candidates simultaneously associated to many classes are either less useful because they are generic (e.g., *history*, *meaning*, *definition*), or incorrect because they are extracted from constructs that occur frequently in natural language sentences (e.g., *case* and *position* extracted from the second and sixth sentence of Figure 1 respectively). An alternative scoring formula demotes such attributes accordingly:

$$S_{norm}(Att(\mathcal{C},\mathcal{A})) = \frac{S_{freq}(Att(\mathcal{C},\mathcal{A}))}{\log(1 + |\{\mathcal{C}_k : Att(\mathcal{C}_k,\mathcal{A})\}|)}$$

The scores determine the relative ranking of candidate attributes within a class. The ranked list is passed through a filter that aims at reducing the number of attributes that are semantically close to one another, thus increasing the diversity and usefulness of the overall list of attributes for that class. For the sake of simplicity, we prefer a fast heuristic that flags attributes as potentially redundant if they have a low edit distance to, or share the same head word with, another attribute already encountered in the list. With moderate effort and added complexity, this heuristic could be combined with one of the popular semantic distance metrics based on WordNet. After discarding redundant attributes, the resulting ranked lists of attributes constitute the output of the extraction method.

3. EXPERIMENTAL SETTING

3.1 Textual Data Sources

Two sets of experiments acquire attributes separately from Web documents maintained by and search queries submitted to the Google search engine. The document collection (D) consists of approximately 100 million Web documents in English, as available in a Web repository snapshot in 2006. The textual portion of the documents is cleaned of HTML, tokenized, split into sentences and part-of-speech tagged using the TnT tagger [3].

The collection of queries (Q) is a random sample of fully-anonymized queries in English submitted by Web users in 2006. The sample contains around 50 million unique queries. Each query is accompanied by its frequency of occurrence in the logs.

The first graph in Figure 2 shows the distribution of the queries from the random sample, according to the number of words in each query. Despite the differences in the distributions of unique (dotted line) vs. all (solid line) queries, the first graph in Figure 2 confirms that most search queries in Q are relatively short. Therefore, the amount of input data that is actually usable by the extraction method from query logs is only a fraction of the available 50 million queries, since an attribute cannot be extracted for a given class unless it occurs together with a class instance in an input query, which is a condition that is less likely to be satisfied in the case of short queries.

Class	Size	Examples of Instances
Actor	1500	Mel Gibson, Julia Roberts, Tom Cruise, Jack Black, Jennifer Lopez, Sharon Stone, Samuel L. Jackson, Halle Berry, Julia Roberts, Bruce Willis, Morgan Freeman
BasicFood	155	fish, turkey, rice, milk, chicken, cheese, eggs, ice cream, corn, duck, peas, ginger, cocoa, tuna, garlic, cereal, cucumber, kale, celery, sea bass, okra, butternut squash
CarModel	368	Honda Accord, Audi A4, Ford Focus, Porsche 911, Ford Explorer, Chrysler Crossfire, Toyota Corolla, Chevrolet Corvette, Jeep Grand Cherokee, Volkswagen Passat
CartoonCharacter	50	Mighty Mouse, Road Runner, Bugs Bunny, Scooby-Doo, Homer Simpson, Popeye, Donald Duck, Tom and Jerry, Butthead, Woody Woodpecker, Wile E. Coyote
City	589	San Francisco, London, Boston, Ottawa, Dubai, Tucson, Amsterdam, Buenos Aires, Seoul, Rio de Janeiro, Lyon, Frankfurt, Casablanca, Delhi, Osaka, Reykjavik
Company	738	Adobe Systems, Macromedia, HP, Gateway, Target, Apple Computer, Reuters, Intel, New York Times, Sun, Delta, Sony, Ford, Nokia, Reuters, Canon
Country	197	Canada, Japan, Australia, India, Liechtenstein, Italy, South Korea, Monaco, Grenada, Namibia, Dominican Republic, Somalia, Monaco, Mongolia, Nicaragua, Cyprus, Haiti
Drug	345	Vicodin, Soma, Hydrocodone, Xanax, Vioxx, Tramadol, Ambien, Paxil, Zithromax, Wellbutrin, Norco, Lipitor, Amoxicillin, Alprazolam, Cipro, Omeprazole
Flower	59	Rose, Lotus, Maple, Iris, Lily, Violet, Daisy, Lavender, Tulip, Orchid, Daffodil, Sunflower, Dahlia, Columbine, Camellia, Hyacinth, Begonia, Poinsettia, Amaryllis
HeavenlyBody	97	Earth, The Sun, Mercury, Uranus, Jupiter, Mars, Venus, Antares, Alpha Centauri, Saturn, Canopus, Vega, Regulus, Sirius, Altair, Sargas, Rigel, Alhena
Mountain	245	K12, Everest, Mont Blanc, Table Mountain, Etna, Mount Shasta, Annapurna, Mount Rainier, Pikes Peak, Matterhorn, Monte Rosa, Mauna Loa, Aconcagua
Movie	626	The Office, Star Wars, Die Hard, The Rock, Back to the Future, Lost in Translation, Fight Club, A Beautiful Mind, Das Boot, Rain Man, Charlie and the Chocolate Factory
NationalPark	59	Joshua Tree National Park, Zion National Park, Great Sand Dunes National Park, Grand Teton National Park, Rocky Mountain National Park, Sequoia National Park
Painter	1011	Marcel Duchamp, Pablo Picasso, Diego Rivera, Titian, Salvador Dali, Claude Monet, Frida Kahlo, Vincent van Gogh, El Greco, Edgar Degas, Peter Paul Rubens
ProgLanguage	101	A++, C, C++, BASIC, JavaScript, Java, Perl, Ada, Python, Occam, Common Lisp, Forth, Fortran, Smalltalk, Visual Basic, AWK, Algol, Datalog, Mathematica
Religion	128	Christianity, Buddhism, Judaism, Islam, Hinduism, Taoism, Confucianism, Wicca, Baptism, Scientology, Pantheism, Tibetan Buddhism, Shamanism, Sikhism, Puritanism
SoccerTeam	116	Real Madrid, Manchester United, FC Barcelona, Werder Bremen, Anderlecht Brussels, Ajax Amsterdam, AC Milan, Atletico Madrid, Austria Wien, Deportivo La Coruna
University	501	Harvard, University of Oslo, Stanford, CMU, Yale, Tsing Hua University, University of Utah, Florida State University, Boston College, Dartmouth College
VideoGame	450	Half Life, Final Fantasy, Need for Speed, Quake, Gran Turismo, Age of Empires, Kingdom Hearts, Perfect Dark, Dragon Quest, Sim City, Twisted Metal, Spy Hunter
Wine	60	Port, Rose, Champagne, Bordeaux, Rioja, Chardonnay, Chianti, Syrah, Pinot Noir, Merlot, Cabernet Sauvignon, Sauvignon Blanc, Riesling, Zinfandel, Malbec

Table 2: Target classes with examples of instances

The second graph in Figure 2 shows the distribution of the input Web documents, according to the number of words after documents were cleaned of HTML tags. As expected, the distribution of documents is quite different from that of queries, as illustrated by the two graphs in Figure 2. First, the possible range of the number of words is much wider in the case of Web documents, since documents are significantly longer than queries. Consequently, the percentage of documents having any given length is quite small, regardless of the length. Second, longer documents tend to occur less frequently, throughout the entire range of the document length.

3.2 Target Classes

The target classes selected for experiments are each specified as an (incomplete) set of representative instances, details on which are given in Table 2. The number of given instances varies from 50 (for *CartoonCharacter*) to 1500 (for *Actor*), with a median of 197 instances per class. The classes also differ with respect to the domain of interest (e.g., Health for *Drug* vs. Entertainment for *Movie*), instance capitalization (e.g., instances in *BasicFood* usually occur in text in lower rather than upper case), and conceptual type (e.g., abstraction for *Religion* vs. group for *SoccerTeam* vs. activity for *VideoGame*).

Quantitatively, the selected target classes also exhibit great variation from the point of view of their popularity within query logs, measured by the sum of the frequencies of the input queries that fully match any of the instances of each class (e.g., the queries *san francisco* for *City*, or *harvard* for *University*). As shown in Figure 3, the corresponding frequency sums per target class vary considerably, ranging between 65,556 (for *Wine*) and 29,361,706 (for *Company*). Therefore, we choose what we feel to be a large enough number of classes (20) to properly ensure varied experimentation on several dimensions, while taking into account the time intensive nature of manual accuracy judgments often required in the evaluation of information extraction systems [2, 4].

4. RESULTS

4.1 Evaluation Procedure

Multiple lists of attributes are evaluated for each class, corresponding to the combination of the use of one of the two ranking functions (frequency-based or normalized) on either Web documents (e.g., *D-freq*) or query logs (e.g., *Q-*

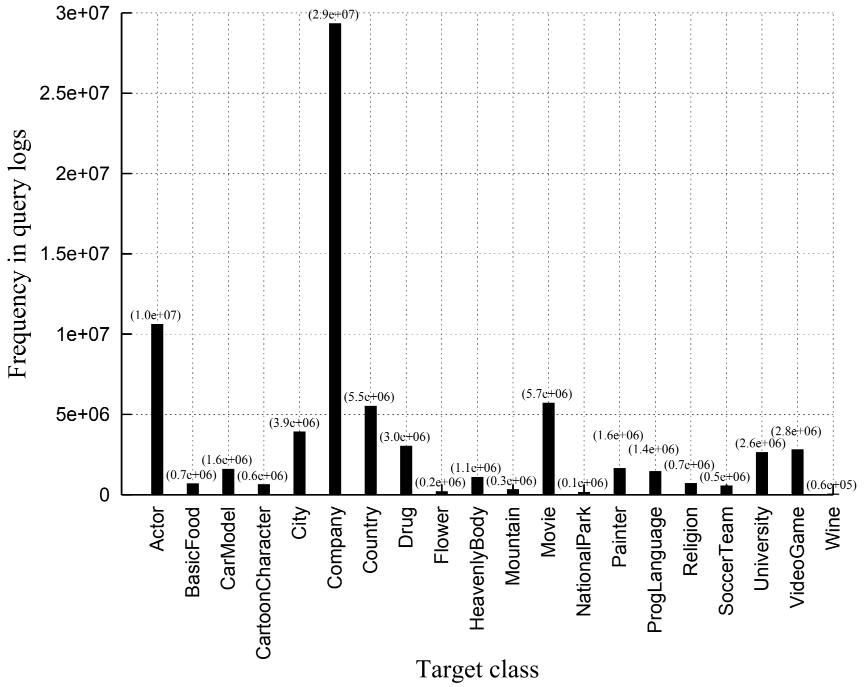

Figure 3: Popularity of the target classes, measured by the aggregated frequency of input queries that are full, case-insensitive matches of any of the instances in each target class

Label	Value	Examples of Attributes
vital	1.0	Actor: date of birth
		BasicFood: fat content
		CartoonCharacter: voice actor
		Flower: botanical name
		ProgLanguage: portability
		Wine: taste
okay	0.5	Actor: beauty secrets
		CarModel: marketing strategies
		Company: vision
		HeavenlyBody: relative size
		NationalPark: reptiles
		Religion: sacred animals
wrong	0.0	BasicFood: low carb
		City: edge
		CarModel: driver
		SoccerTeam: clash
		Movie: fiction
		Mountain: ash

Table 3: Correctness labels for the manual assessment of attributes

norm). To remove any undesirable psychological bias towards higher-ranked attributes during the assessment, the elements of each list to be evaluated are sorted alphabetically into a merged list.

A human judge manually assigns a correctness label to each attribute of the merged list within its respective class. Similarly to methodology previously proposed to evaluate answers to Definition questions [21], an attribute is *vital* if it must be present in an ideal list of attributes of the target class; *okay* if it provides useful but non-essential information; and *wrong* if it is incorrect. Thus, a correctness label is manually assigned to a total of 5,859 attributes extracted for the 20 target classes, in a process that confirms that evaluation of information extraction methods can be quite time consuming.

To compute the overall precision score over a given ranked list of extracted attributes, the correctness labels are converted to numeric values as shown in Table 3. Precision at some rank N in the list is thus measured as the sum of the assigned values of the first N candidate attributes, divided by N.

4.2 Precision

For a formal analysis of qualitative performance, Table 4 provides a detailed picture of precision scores for each of the twenty target classes. For completeness, the scores in the table capture precision at the very top of the extracted lists of attributes (rank 5) as well as over a wider range of those lists (ranks 10 through 50).

Two conclusions can be drawn after inspecting the results. First, the quality of the results varies among classes. At the lower end, the precision for the class *Wine* is below 0.40 at rank 5. At the higher end, the attributes for *Company* are very good, with precision scores above 0.90 even at rank 20. Second, documents and queries are not equally useful in class attribute extraction. The attributes extracted from documents are better at the very top of the list (rank 5) for the class *SoccerTeam* and at all ranks for *City*. However, the large majority of the classes have higher precision scores when the attributes are extracted from queries rather than documents. The differences in quality are particularly high for classes like *HeavenlyBody*, *CarModel*, *BasicFood*, *Flower* and *Mountain*. To better quantify the quality gap, the last rows of Table 4 show the precision computed as an average over all classes, rather than for each class individually. Consistently over all computed ranks, the precision is about 45% better on average when using queries rather than doc-

Class	Precision											
	@5		@10		@20		@30		@40		@50	
	D	Q	D	Q	D	Q	D	Q	D	Q	D	Q
Actor (*Abs*)	0.30	0.70	0.45	0.85	0.55	0.82	0.56	0.80	0.55	0.75	0.55	0.72
BasicFood (*Abs*)	0.40	1.00	0.25	1.00	0.27	0.77	0.20	0.68	0.15	0.67	0.16	0.69
CarModel (*Abs*)	0.30	1.00	0.50	0.95	0.55	0.77	0.46	0.80	0.51	0.76	0.49	0.74
CartoonCharacter (*Abs*)	0.50	0.60	0.50	0.55	0.45	0.45	0.41	0.43	0.43	0.47	0.42	0.45
City (*Abs*)	0.60	0.40	0.60	0.30	0.50	0.22	0.46	0.16	0.42	0.27	0.42	0.31
Company (*Abs*)	1.00	1.00	1.00	0.95	0.97	0.90	0.83	0.83	0.78	0.80	0.69	0.76
Country (*Abs*)	0.60	1.00	0.50	0.90	0.50	0.87	0.55	0.85	0.52	0.87	0.52	0.88
Drug (*Abs*)	0.80	1.00	0.80	1.00	0.75	1.00	0.55	0.90	0.56	0.86	0.55	0.87
Flower (*Abs*)	0.40	1.00	0.30	0.90	0.25	0.77	0.23	0.71	0.20	0.65	0.18	0.59
HeavenlyBody (*Abs*)	0.40	1.00	0.35	1.00	0.42	0.97	0.53	0.88	0.41	0.83	0.38	0.79
Mountain (*Abs*)	0.20	0.80	0.10	0.90	0.12	0.85	0.11	0.71	0.12	0.66	0.17	0.62
Movie (*Abs*)	0.60	0.90	0.55	0.95	0.70	0.90	0.60	0.80	0.56	0.76	0.51	0.74
NationalPark (*Abs*)	1.00	1.00	0.60	0.70	0.55	0.82	0.50	0.80	0.45	0.68	0.42	0.63
Painter (*Abs*)	0.80	1.00	0.80	1.00	0.80	0.97	0.76	0.95	0.73	0.93	0.72	0.88
ProgLanguage (*Abs*)	0.40	1.00	0.40	0.95	0.62	0.72	0.61	0.70	0.55	0.67	0.50	0.58
Religion (*Abs*)	0.80	0.90	0.70	0.95	0.67	0.95	0.51	0.86	0.48	0.86	0.47	0.82
SoccerTeam (*Abs*)	0.80	0.50	0.50	0.55	0.35	0.42	0.36	0.35	0.33	0.26	0.33	0.22
University (*Abs*)	0.60	0.80	0.65	0.90	0.47	0.82	0.51	0.81	0.48	0.72	0.46	0.65
VideoGame (*Abs*)	0.90	1.00	0.80	0.70	0.70	0.57	0.55	0.51	0.55	0.55	0.52	0.48
Wine (*Abs*)	0.40	0.20	0.30	0.40	0.20	0.42	0.26	0.38	0.25	0.33	0.28	0.27
Average-Class (*Abs*)	0.59	0.84	0.53	0.82	0.52	0.75	0.48	0.70	0.45	0.67	0.44	0.63
Average-Class (*Rel*)	-	+42%	-	+54%	-	+44%	-	+45%	-	+48%	-	+43%
Average-Class (*Err*)	-	-60%	-	-61%	-	-47%	-	-42%	-	-58%	-	-43%

Table 4: Precision of attributes extracted with normalized ranking from Web documents (D) versus query logs (Q), expressed as *Abs*olute scores, *Rel*ative boosts (Q over D), and *Err*or reduction rates (Q over D)

Class	Top Extracted Attributes
BasicFood	D: [species, pounds, cup, kinds, lbs, bowl..]
	Q: [nutritional value, health benefits, glycemic index, varieties, nutrition facts, calories..]
CarModel	D: [fuel usage, models, driver, appropriate derivative, assembly, reviews, sales, likes..]
	Q: [reliability, towing capacity, gas mileage, weight, price, pictures, top speed..]
Drug	D: [manufacturer, dose, mg, effectiveness, patient ratings, efficacy, dosage, tablets..]
	Q: [side effects, half life, mechanism of action, overdose, long term use, synthesis..]
Heavenly Body	D: [observations, spectrum, planet, spectra, conjunction, transit, temple, surface..]
	Q: [atmosphere, surface, gravity, diameter, mass, rotation, revolution, moons, radius..]
Religion	D: [teachings, practice, beliefs, religion spread, principles, emergence, doctrines..]
	Q: [basic beliefs, teachings, holy book, practices, rise, branches, spread, sects..]

Table 5: Top attributes extracted with normalized ranking for various classes from Web documents (D) vs. query logs (Q)

uments. This is the most important result of the paper. It shows that query logs represent a competitive resource against document collections in class attribute extraction.

As an alternative to Table 4, Table 5 illustrates the top attributes extracted from text for a few of the target classes. Documents produce more spurious items, as indicated by a more frequent presence of attributes that are deemed *wrong*, such as *bowl* for *BasicFood*, *mg* for *Drug*, or *temple* for *HeavenlyBody*. The highest-ranked attributes acquired from query logs are relatively more useful, particularly for the first three classes shown in Table 5.

Input Docs	Precision				
	@10	@20	@30	@40	@50
20%	0.53	0.49	0.43	0.40	0.38
50%	0.53	0.47	0.44	0.42	0.39
100%	0.53	0.52	0.48	0.45	0.43

Table 6: Impact of extraction with normalized ranking from a fifth vs. half vs. all of the Web documents

The precision results confirm and quantify the qualitative advantage of query logs over documents, in the task of attribute extraction. However, the experiments do not take into account the fact that it is more likely for an extraction pattern to match a portion of a document rather than a query, simply because a document contains more raw text. Other things being equal, although the percentage of spurious attributes among all extracted attributes is expected to be similar when extracted from the 100 million documents vs. 50 million query logs, the absolute number of such spurious attributes is expected to be higher from documents. Although it is not really intuitive that using *too many* input documents could result in lower precision due to an overwhelming number of spurious attributes, additional experiments verify whether that may be the case. Table 6 compares the precision at various ranks as an average over all classes, when attributes are extracted (*D-norm*) from 20%, 50% or 100% of the available input Web documents. The table shows that, in fact, using fewer documents does not improve precision, which instead degrades slightly.

Figure 4 provides a graphical comparison of precision from all Web documents vs. query logs, at all ranks from 1 through 50. Besides the head-to-head comparison of the two types of data sources, the graphs show the added benefit of *norm*alized (as opposed to *freq*uency-based) ranking, which is more ap-

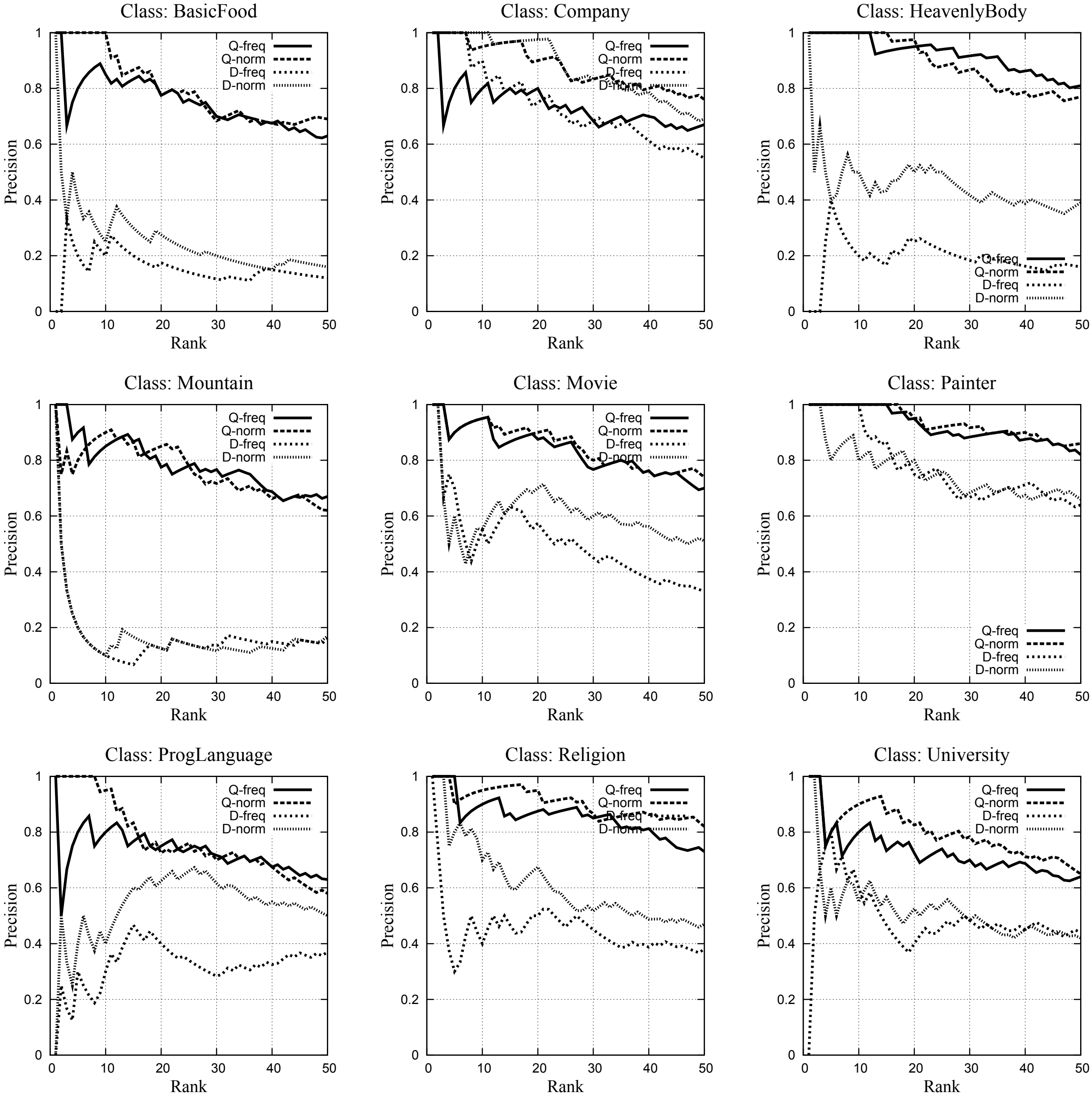

Figure 4: Impact of frequency-based (*freq*) vs. normalized (*norm*) ranking for attributes extracted for various classes from Web documents (D) vs. query logs (Q)

parent in the case of attributes extracted from documents (*D-norm* vs. *D-freq*).

4.3 Coverage

Since the ideal, complete set of items to be extracted is usually not available, most studies on Web information extraction are forced to forgo the evaluation of recall and focus instead on measuring precision [4]. Similarly, the manual enumeration of the complete set of attributes of each target class, to measure recall, is unfeasible. As a tractable alternative to evaluating recall, the attributes extracted from documents (with *D-freq* or *D-norm*) that were manually judged as *vital* during the evaluation of precision are temporarily considered as a reference set for measuring the relative recall. Given this reference set of attributes and a list of attributes acquired from query logs, the evaluation of the latter consists in automatically verifying whether each attribute from query logs is an exact, case-insensitive string match of one of the attributes in the reference set. Therefore, the scores computed as an average over all target classes in Table 7 represent lower bounds on relative recall rather than actual relative recall values, since extracted attributes that are se-

Class	Recall				
	@10	@20	@30	@40	@50
Actor	0.16	0.28	0.28	0.28	0.36
BasicFood	0.12	0.25	0.37	0.50	0.50
CarModel	0.08	0.12	0.12	0.12	0.12
CartoonCharacter	0.09	0.09	0.14	0.14	0.14
City	0.03	0.11	0.11	0.15	0.26
Company	0.07	0.14	0.17	0.17	0.19
Country	0.16	0.20	0.26	0.26	0.30
Drug	0.10	0.13	0.27	0.37	0.37
Flower	0.08	0.08	0.25	0.41	0.41
HeavenlyBody	0.25	0.35	0.40	0.40	0.40
Mountain	0.11	0.11	0.11	0.22	0.22
Movie	0.14	0.29	0.37	0.48	0.48
NationalPark	0.15	0.21	0.21	0.26	0.26
Painter	0.14	0.19	0.21	0.24	0.29
ProgLanguage	0.13	0.16	0.19	0.22	0.22
Religion	0.15	0.25	0.35	0.40	0.40
SoccerTeam	0.08	0.17	0.21	0.21	0.21
University	0.04	0.08	0.16	0.24	0.28
VideoGame	0.12	0.15	0.21	0.25	0.28
Wine	0.05	0.11	0.11	0.11	0.11
Average-Class	0.11	0.17	0.22	0.27	0.29

Table 7: Coverage of the list of attributes extracted with normalized ranking from query logs, relative to the set of vital attributes extracted from Web documents

mantically equivalent but lexically different to one of the attributes in the reference set (e.g., plural forms, different spelling, synonyms etc.) unfairly receive no credit. The performance varies by class, with relative recall values in the range from 0.03 (for *City*) to 0.25 (for *HeavenlyBody*) at rank 10. Similarly, recall values vary from 0.11 (for *Wine*) to 0.50 (for *BasicFood*) at rank 50.

5. COMPARISON TO PREVIOUS WORK

In terms of scale and general goals, our work fits into a broader trend towards large-scale information extraction. Previous studies rely exclusively on large document collections, for mining pre-specified types of relations such as InstanceOf [15], Person-AuthorOf-Invention [11], Company-HeadquartersIn-Location [2] or Country-CapitalOf-City [4] from text. In contrast, we explore the role of both document collections and query logs in extracting an open, rather than pre-specified type of information, namely class attributes. A related recent approach [18] pursues the goal of unrestricted relation discovery from text.

Our extracted attributes are relations among objects in the given class, and objects or values from other, “hidden” classes. Determining the type of the “hidden” argument of each attribute (e.g., *Person* and *Location* for the attributes *chief executive officer* and *headquarters* of the class *Company*) is beyond the scope of this paper. Nevertheless, the lists of extracted attributes have direct benefits in gauging existing methods for harvesting pre-specified semantic relations [4, 14], towards the acquisition of relations that are of real-world interest to a wide set of Web users, e.g., towards finding *mechanisms of action* for *Drugs* and *health benefits* for *BasicFood*.

Query logs have been a natural candidate in efforts to improve the quality of information retrieval, either directly through re-ranking of retrieved documents [23, 22, 1] and query expansion [6], or indirectly through the development of spelling correction models [8]. [13] were the first to explore query logs as a resource for acquiring explicit relations, but evaluated their approach on a very small set of target classes, without a comparison to traditional document-based methods. Such a comparative study is highly useful, if not necessary, before further explorations based on query logs.

In [5], the acquisition of attributes and other knowledge relies on Web users who explicitly specify it by hand. In contrast, we may think of our approach as Web users implicitly giving us the same type of information, outside of any systematic attempts to collect knowledge of general use from the users.

The method proposed in [20] applies lexico-syntactic patterns to text within a small collection of Web documents. The resulting attributes are evaluated through a notion of question answerability, wherein an attribute is judged to be valid if a question can be formulated about it. More precisely, evaluation consists in users manually assessing how natural the resulting candidate attributes are, when placed in a *wh-* question. Comparatively, our evaluation is stricter. Indeed, many attributes, such as *long term uses* and *users* for the class *Drugs*, are marked as wrong in our evaluation, although they would easily pass the question answerability test (e.g., *“What are the long term uses of Prilosec?”*) used in [20].

6. CONCLUSION

Confirming our intuition that Web query logs as a whole mirror a significant amount of knowledge present within Web documents, the experimental results of this paper introduce query logs as a valuable resource in textual information extraction. Somewhat surprisingly, a robust method for extracting class attributes produces significantly better results when applied to query logs rather than Web documents, thus holding the promise of a new path in research in information extraction. Ongoing work includes a model for combining the two types of data sources while accounting for the difference in their variability, a weakly supervised method based on seeds rather than patterns, and exploration of the role of query logs in other information extraction tasks.

7. REFERENCES

[1] E. Agichtein, E. Brill, and S. Dumais. Improving Web search ranking by incorporating user behavior information. In *Proceedings of the 29th ACM Conference on Research and Development in Information Retrieval (SIGIR-06)*, pages 19–26, Seattle, Washington, 2006.

[2] E. Agichtein and L. Gravano. Snowball: Extracting relations from large plaintext collections. In *Proceedings of the 5th ACM International Conference on Digital Libraries (DL-00)*, pages 85–94, San Antonio, Texas, 2000.

[3] T. Brants. TnT - a statistical part of speech tagger. In *Proceedings of the 6th Conference on Applied Natural Language Processing (ANLP-00)*, pages 224–231, Seattle, Washington, 2000.

[4] M. Cafarella, D. Downey, S. Soderland, and O. Etzioni. KnowItNow: Fast, scalable information extraction from the Web. In *Proceedings of the Human*

Language Technology Conference (HLT-EMNLP-05), pages 563–570, Vancouver, Canada, 2005.

[5] T. Chklovski and Y. Gil. An analysis of knowledge collected from volunteer contributors. In *Proceedings of the 20th National Conference on Artificial Intelligence (AAAI-05)*, pages 564–571, Pittsburgh, Pennsylvania, 2005.

[6] H. Cui, J. Wen, J. Nie, and W. Ma. Probabilistic query expansion using query logs. In *Proceedings of the 11th World Wide Web Conference (WWW-02)*, pages 325–332, Honolulu, Hawaii, 2002.

[7] D. Dowty, R. Wall, and S. Peters. *Introduction to Montague Semantics*. Springer, 1980.

[8] M. Li, M. Zhu, Y. Zhang, and M. Zhou. Exploring distributional similarity based models for query spelling correction. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (COLING-ACL-06)*, pages 1025–1032, Sydney, Australia, 2006.

[9] X. Li and D. Roth. Learning question classifiers. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING-02)*, pages 556–562, Taipei, Taiwan, 2002.

[10] D. Lin and P. Pantel. Concept discovery from text. In *Proceedings of the 19th International Conference on Computational linguistics (COLING-02)*, pages 1–7, 2002.

[11] L. Lita and J. Carbonell. Instance-based question answering: A data driven approach. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-04)*, pages 396–403, Barcelona, Spain, 2004.

[12] R. Mooney and R. Bunescu. Mining knowledge from text using information extraction. *SIGKDD Explorations*, 7(1):3–10, 2005.

[13] M. Paşca and B. Van Durme. What you seek is what you get: Extraction of class attributes from query logs. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI-07)*, pages 2832–2837, Hyderabad, India, 2007.

[14] P. Pantel and M. Pennacchiotti. Espresso: Leveraging generic patterns for automatically harvesting semantic relations. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (COLING-ACL-06)*, pages 113–120, Sydney, Australia, 2006.

[15] P. Pantel and D. Ravichandran. Automatically labeling semantic classes. In *Proceedings of the 2004 Human Language Technology Conference (HLT-NAACL-04)*, pages 321–328, Boston, Massachusetts, 2004.

[16] M. Remy. Wikipedia: The free encyclopedia. *Online Information Review*, 26(6):434, 2002.

[17] L. Schubert. Turing's dream and the knowledge challenge. In *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI-06)*, Boston, Massachusetts, 2006.

[18] Y. Shinyama and S. Sekine. Preemptive information extraction using unrestricted relation discovery. In *Proceedings of the 2006 Human Language Technology Conference (HLT-NAACL-06)*, pages 204–311, New York, New Tork, 2006.

[19] K. Shinzato and K. Torisawa. Acquiring hyponymy relations from web documents. In *Proceedings of the 2004 Human Language Technology Conference (HLT-NAACL-04)*, pages 73–80, Boston, Massachusetts, 2004.

[20] K. Tokunaga, J. Kazama, and K. Torisawa. Automatic discovery of attribute words from Web documents. In *Proceedings of the 2nd International Joint Conference on Natural Language Processing (IJCNLP-05)*, pages 106–118, Jeju Island, Korea, 2005.

[21] E. Voorhees. Evaluating answers to definition questions. In *Proceedings of the 2003 Human Language Technology Conference (HLT-NAACL-03)*, pages 109–111, Edmonton, Canada, 2003.

[22] G. Wang, T. Chua, and Y. Wang. Extracting key semantic terms from Chinese speech query for Web searches. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL-03)*, pages 248–255, Sapporo, Japan, 2003.

[23] Z. Zhuang and S. Cucerzan. Re-ranking search results using query logs. In *Proceedings of the 15th International Conference on Information and Knowledge Management (CIKM-06)*, pages 860–861, Arlington, Virginia, 2006.