What You Seek is What You Get: Extraction of Class Attributes from Query Logs

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

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

Abstract

Within the larger area of automatic acquisition of knowledge from the Web, we introduce a method for extracting relevant attributes, or quantifiable properties, for various classes of objects. The method extracts attributes such as *capital city* and *President* for the class *Country*, or *cost*, *manufacturer* and *side effects* for the class *Drug*, without relying on any expensive language resources or complex processing tools. In a departure from previous approaches to large-scale information extraction, we explore the role of Web query logs, rather than Web documents, as an alternative source of class attributes. The quality of the extracted attributes recommends query logs as a valuable, albeit little explored, resource for information extraction.

1 Introduction

Despite differences in the types of targeted information, as well as underlying algorithms and tools, a common theme shared across recent approaches to information extraction is an aggressive push towards *large-scale* extraction. Documents spanning various genres, news corpora, or the Web are readily available, providing significant amounts of textual content towards the acquisition of relations such as InstanceOf [Pantel and Ravichandran, 2004], Country-CapitalOf-City [Cafarella *et al.*, 2005], Company-HeadquartersIn-Location [Agichtein and Gravano, 2000], Person-AuthorOf-Invention [Lita and Carbonell, 2004] or Person-BornIn-Year [Paşca *et al.*, 2006]. As a next step beyond extracting instances of a target relation that is specified in advance, this paper aims at acquiring sets of relations that constitute prominent attributes, or quantifiable properties, of a given class of instances. As an illustration, desirable class attributes targeted here include, among others:

• *manufacturer*, *cost*, *side effects*, *pharmacokinetics* and *dosage* (for *Drug*);

• *capital city*, *population*, *gross domestic product* and *prime minister* (for *Country*);

• *paintings*, *artistic style*, *influences* and *birth place* (for *Painter*).

*Contributions made during an internship at Google.

From a pure Artificial Intelligence perspective, class attributes recommend themselves as building blocks towards the appealing, and yet elusive goal of constructing large-scale knowledge bases automatically. Yet our work is originally motivated by an activity that permeates modern societies and is undertaken by millions of people every day, namely Web search. Named entities constitute a large fraction of the queries submitted to the largest Web search engines. In response to a query that refers to a named entity, Web search engines could augment their results with a compilation of specific facts, based on the set of attributes extracted in advance for the class to which the named entity belongs.

Another, more immediate application of class attributes in Web search is to process a query stream to identify and flag queries that request factual information. This is particularly useful in the more frequent, but difficult case when likely-factual queries are keyword-based rather than expressed in natural language. For example, if the search engine has specialized functionality for answering simple natural-language questions such as *"What is the altitude of Guadalajara?"*, the knowledge that *altitude* is a prominent attribute of the class *City* enables the search engine to handle keyword-based queries such as *"altitude guadalajara"* and *"guadalajara altitude"* similarly to their cleaner, natural-language counterpart.

In this spirit, and in a significant departure from previous approaches to large-scale information extraction, the target information (in this case, class attributes) is not mined from document collections. Instead, we explore the role of Web query logs, rather than documents, as an alternative source of class attributes. To our knowledge this corresponds to the first endeavor in large-scale knowledge acquisition from query logs. Sections 2 and 3 explain why we believe that query logs represent a valuable, albeit unexplored, resource for information extraction. Section 4 introduces a robust and scalable method for extracting quantifiable attributes of arbitrary classes. The method initially relies on a small set of linguistically motivated extraction patterns applied to each entry from the query logs, then employs a series of Web-based precision-enhancement filters to refine and rank the candidate attributes. Section 5 presents evaluation results when applying the method to a sample of approximately 50 million queries selected from a larger set of queries collectively submitted by Web users to the Google search engine.

2 Strategies for Extraction

2.1 Class-Driven Extraction

If the target class is specified by mentioning only the class name (e.g., *country* or *drug/medicine*), a possible extraction strategy is to identify authoritative documents that explicitly enumerate the attributes of the class. For the class *Car*, an ideal enumeration could be *"The most important properties of a car are its class, quality of handling, engine displacement, top speed, fuel economy [..]"*. Unfortunately, the likelihood of such enumerations occurring in natural language for arbitrary classes is quite low, even within Web-scale collections. Moreover, the scale and noisy character of the Web rule out the use of any expensive tools that would be required for the necessary levels of text understanding. For these reasons, class-driven extraction is both unlikely to produce useful attributes, and impractical to apply to large amounts of text.

2.2 Instance-Driven Extraction

Since a class (concept) is traditionally a placeholder for a set of instances (objects) that share similar attributes (properties) [Dowty *et al.*, 1980], the attributes of a given class can be derived by extracting and inspecting the attributes of individual instances from that class. With this strategy, the attributes of the class *Car* are extracted by inspecting attributes extracted for *Chevrolet Corvette*, *Toyota Prius*, *Volkswagen Passat*, etc. This is particularly appealing with large sources of open-domain text (e.g., the Web) for two reasons. First, 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 [Agichtein and Gravano, 2000]; 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 [Shinzato and Torisawa, 2004]; or d) selecting prominent clusters of instances from distributionally similar phrases acquired from a large text collection [Lin and Pantel, 2002]. In other words, specifying the target class as a set of instances rather than the class name is simply a matter of formatting (or converting) the input class, rather than a limiting assumption. Second, named entities are well represented on the Web (whether documents or queries), whereas it is relatively much harder to find common-sense knowledge such as the fact that *"all countries have a capital city"*, especially in an explicit natural-language form that lends itself to robust extraction.

3 Sources of Data

3.1 Extraction from Document Collections

Previous studies in textual information extraction consistently use some form of a document collection as their preferred data source. In particular, many recent studies take advantage of the increasing amount of textual content available in Web documents. Similarly, for the task of extracting attributes for arbitrary classes, Web documents represent an obvious choice based on a simple observation:

Hypothesis 1: Let $\mathcal{C}$ be a class, and $\mathcal{I} \in \mathcal{C}$ be instances of that class. If $\mathcal{A}$ is a prominent attribute of $\mathcal{C}$, then some of the documents that contain information about $\mathcal{I}$ are also likely to contain mentions of $\mathcal{A}$ in the context of $\mathcal{I}$.

3.2 Extraction from Query Logs

At first sight, choosing queries over documents as the source data may seem counterintuitive due to disadvantages in several key aspects: ability to convey rather than inquire about information; sheer size of available data; and availability of contextual clues. Indeed, common wisdom suggests that textual documents tend to assert information (statements or facts) about the world in the form of expository text. Comparatively, search queries can be thought of as being nothing more than noisy, keyword-based approximations of often-underspecified user information needs (interrogations). In fact, even with a large document collection there is no guarantee that any relevant documents exist that provide an answer, confirm, or refute the interrogation to which a random query corresponds. From a strictly quantitative standpoint, the amount of text within query logs is at a clear disadvantage against the much larger textual content available within the Web documents. Finally, additional contextual information is usually available in Web documents both internally, i.e., the context in which an instance occurs in text, and externally, e.g., through anchor text associated with incoming links. In contrast, little, if any additional context is available in Web queries since their median length is only two words. Despite these disadvantages, our choice of exploring query logs is motivated by the following general hypothesis about availability and use of knowledge.

Assume an abstract search scenario, in which a pool of users has access to a large body of common-sense knowledge - so large that no user can possibly memorize each individual assertion from the knowledge base. In order to find new knowledge that they need, users can search the existing knowledge base. Each query that they submit is an inquiry for new information. However, *users formulate their queries based on the common-sense 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:

Hypothesis 2: If knowledge is generally prominent or relevant, people will (eventually) ask about it.

The hypothesis becomes increasingly useful as the number of users and the quantity and breadth of the available knowledge increase. Indeed, to switch from an abstract to a more concrete scenario, if it is common-sense knowledge that something (e.g., *top speed*) is a prominent attribute of a given class (e.g., *Car*), then a fraction of the set of Web search queries submitted by people from around the globe are likely to capture that knowledge:

Hypothesis 3: Let $\mathcal{C}$ be a class, and $\mathcal{I} \in \mathcal{C}$ be instances of that class. If $\mathcal{A}$ is a prominent attribute of $\mathcal{C}$, then a fraction of the queries about $\mathcal{I}$ are likely to ask about $\mathcal{A}$ in the context of various instances $\mathcal{I}$.

Among the possible choices in data sources (Web documents vs. query logs) and format of the information about the target class (class-driven vs. instance-driven), we opt for an instance-driven extraction method from query logs, which is presented in the next section.

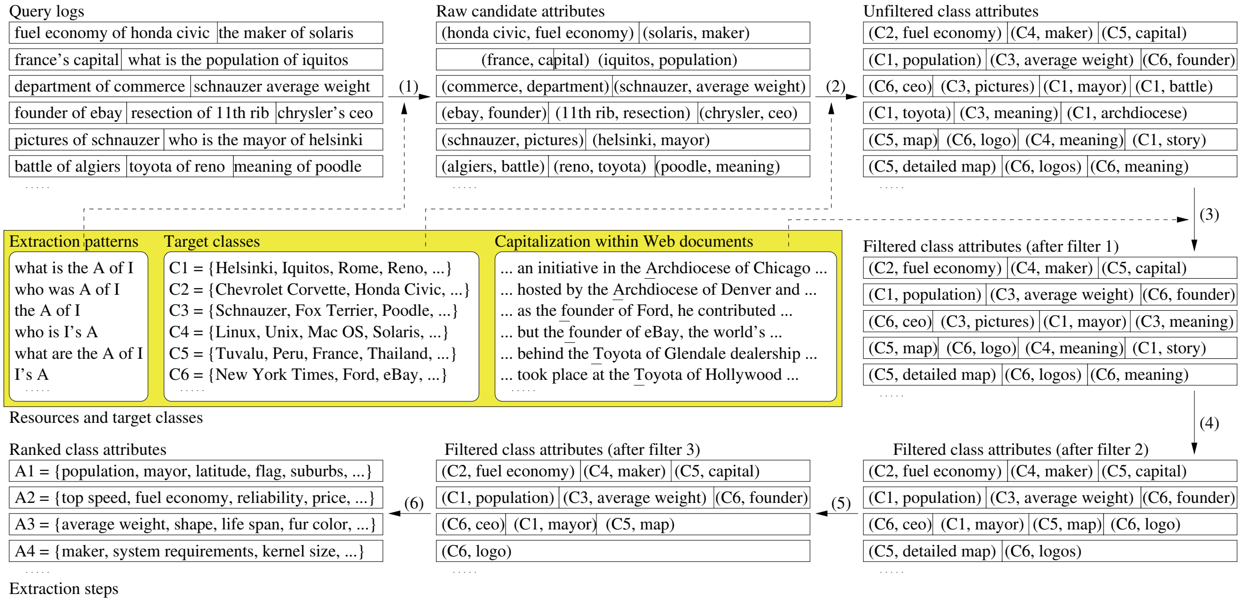

Figure 1: Overview of data flow during class attribute extraction

4 Extraction Method

4.1 Selection of Candidate Attributes

Our extraction method consists of three stages: selection of candidate attributes for the given set of target classes (Steps 1 and 2 from Figure 1); filtering of the attributes for higher quality (Steps 3 through 5); and ranking of the attributes that pass all filters (Step 6).

For robustness and scalability, a small set of linguistically-motivated patterns extract potential pairs of a class label (or instance) and an attribute from query logs. This corresponds to Step (1) from Figure 1. For each pair, a weighted frequency is computed as a weighted sum of the frequencies of the input queries within the query logs. The weights are set such that full-fledged natural-language queries (e.g., *"what is the population of iquitos"*) have a higher contribution towards the weighted frequencies. In a useful extension that better captures frequency data available within query logs, the logs are scanned again for any keyword-based queries that contain the elements from the collected pairs (in any order) and nothing else (e.g., *"population iquitos"* or *"iquitos population"*). The motivation of this step is that keyword-based queries are used much more often by Web users as compared to natural-language queries. The frequencies of those matching queries are added to the counts computed initially for the pairs.

Figure 1 illustrates how Step (2) converts the collected pairs into a smaller set of yet-unfiltered attributes that apply only to instances of the target classes. For example, *Honda Civic* is an instance of the target class C_2. It also appears in the first pair shown as collected after Step (1). Therefore the attribute *"fuel economy"* is extracted for C_2. The weighted frequency of each pair output by Step (2) (e.g., (C_5, *capital*)) aggregates the weighted frequencies of the source pairs (e.g., (*france*, *capital*), (*peru*, *capital*), (*thailand*, *capital*), etc.).

4.2 Filtering of Candidate Attributes

Due to the simplicity of the extraction patterns applied during pre-processing, and the relative low quality of the average Web query, the candidate attributes selected after Step (2) are quite noisy. A series of filters successively improve the quality of the sets of class attributes.

The first filter - Step (3) from Figure 1 - identifies and discards attributes that are proper names or are part of longer proper names. Since many users of search engines make no upper vs. lower-case distinction when they type their queries, it is not feasible to detect the true case of the candidate attributes based on query logs alone. However, it is relatively straightforward (although computationally expensive) to recover case information heuristically, by scanning Web documents for mentions of the candidate attributes. More specifically, let $\mathcal{A}$ be an attribute to be checked (e.g., *toyota*) and $\mathcal{I}$ be instances (e.g., *Helsinki*, *Glendale*, *Hollywood*) of the class. If the pattern "the upper-case-$\mathcal{A}$ of $\mathcal{I}$" (e.g., *"the Toyota of Glendale"*) occurs more frequently in Web documents than the pattern "the lower-case$\mathcal{A}$ of $\mathcal{I}$" (e.g., *"the founder of Ford"*) does, then the attribute is discarded as upper-case (unless it is a known person title such as *Queen*); otherwise it is retained. Although simple, this technique proves quite effective in discarding spurious attributes such as the above-mentioned *toyota*, as well as *archdiocese*, *battle*, etc.

The second filter corresponds to Step (4) from Figure 1. It discards candidate attributes that are deemed to be generic (e.g., *meaning*, *story*, *picture*, *pictures* and *summary*), in that they are simultaneously associated with many target classes.

The third filter (Step (5) from Figure 1) aims at reducing the number of attributes that are semantically close to one another within a class, thus increasing the diversity and usefulness of the overall set. For the sake of simplicity, we prefer a fast heuristic that flags attributes as potentially redundant if

Class	Size	Examples of Instances
Drug	346	*Ibuprofen, Tobradex, Prilosec*
Company	738	*Ford, Xerox, Longs Drug Stores*
Painter	1011	*Georgia O'Keeffe, Ossip Zadkine*
City	591	*Hyderabad, Albuquerque, Tokyo*
Country	265	*Yemen, India, Paraguay, Egypt*

Table 1: Size and example of instances in each target class

they have a low edit distance to, or share the same head word with, another attribute already encountered in the list.

4.3 Ranking of Candidate Attributes

In summary, the weighted frequencies of the pairs of a class and an attribute that pass all filters are derived successively from the original frequencies in query logs. The computation takes the frequencies from query logs, weights them based on which pattern they match, and then adds frequencies together as different entries are collapsed into identical pairs during pre-processing and then selection of attributes.

Using the notation $W_f(\mathcal{C}, \mathcal{A})$ for the weighted frequency of the attribute $\mathcal{A}$ within the class $\mathcal{C}$, the first formula (*frequency*) for scoring the attributes is:

$$Score(\mathcal{C}, \mathcal{A}) = W_f(\mathcal{C}, \mathcal{A})$$

In other words, the first formula exploits the weighted frequencies directly to determine the relative order of the returned attributes within each class. To better capture attributes that are specific to a particular class, a second formula (*reranked*) computes PMI-inspired [Turney, 2001] scores:

$$Score(\mathcal{C}, \mathcal{A}) = W_f(\mathcal{C}, \mathcal{A}) \times S_f(\mathcal{C}, \mathcal{A}) \times \log \frac{W_f(\mathcal{C}, \mathcal{A}) \times N}{W_f(\mathcal{C}) \times W_f(\mathcal{A})}$$

where N is the total frequency over all pairs, and $S_f(\mathcal{C}, \mathcal{A})$ is a smoothing factor to avoid an over-emphasis of rare attributes. With either formula, the top attributes in each class are returned in Step (6) from Figure 1.

5 Evaluation

5.1 Experimental Setting

Query Logs: The data source for attribute extraction experiments is a random sample of around 50 million unique, fully-anonymized queries from larger query logs collected by the Google search engine in the first few months of 2006. All queries are in English, and are accompanied by their total frequency of submission within the logs.

Target Classes: Due to the time intensive nature of manual accuracy judgments often required in the evaluation of information extraction systems [Agichtein and Gravano, 2000; Cafarella *et al.*, 2005], we restricted ourselves to a conservative number of classes for testing. The test set comprises five target classes, namely *Drug*, *Company*, *Painter*, *City*, and *Country*, which respectively correspond to artifacts, organizations, roles, and two geographic entities. Each of these classes is specified as a set of representative instances, details on which are given in Table 1. [1] As explained earlier in Section 2.2, our apparent reliance on pre-specifying the target class through a set of instances is not a limiting factor.

[1]The set of instances for *Country* contains a number of popularly used abbreviations, as well as geopolitically distinct areas such as Taiwan and the West Bank.

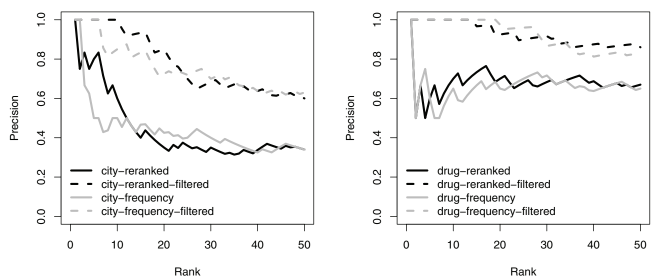

Figure 2: Precision as a function of rank. Dotted lines represent lists that have been through heuristic filtering (Section 4.2), black denotes lists *reranked* based on the PMI-inspired formula, and gray represents *frequency*-ranked lists.

Label	Value	Examples of attributes
vital	1.0	(*Country*, *president*), (*Drug*, *cost*)
okay	0.5	(*City*, *restaurant*), (*Company*, *strengths*)
wrong	0.0	(*Painter*, *diary*), (*Drug*, *users*)

Table 2: Correctness labels for the manual assessment

5.2 Results

Precision: Four distinct lists of attributes are evaluated for each class, corresponding to the combination of the use of one of the two ranking methods (*frequency* or PMI-*reranked*) with either applying all filters (e.g., *City-frequency-filtered*) or none of them (e.g., *City-frequency*). Each of the first 100 elements of the extracted lists of attributes is assigned a correctness label. Similarly to the methodology previously proposed to evaluate answers to Definition questions [Voorhees, 2003], 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. To compute the overall precision score, the correctness labels are converted to numeric values as shown in Table 2. Precision at rank n in a given list is thus measured as the sum of the assigned values of the first n attribute candidates, divided by n. With *frequency* ranking and with all filters enabled, Table 3 illustrates the top attributes returned for the five target classes. A few of these attributes that are deemed to be *wrong* include *long term use* for *Drug*, and *best* for *City*. Table 4 summarizes the resulting precision values at various ranks.

Figure 2 gives a detailed illustration of the effects of filtering (described earlier in Section 4.2) and reranking when applied to the classes *City* and *Drug*. The accuracy varies among different classes. Somewhat surprisingly, using the *reranked* scoring formula does not bring significant benefits over the simpler *frequency* ranking, as seen in the lack of separation between gray versus black lines. On the other hand, the heuristic filtering increases accuracy significantly, as illustrated by the difference between dotted and bold lines, es-

Country: *capital, population, president, map, capital city, currency, climate, flag, culture, leader*
Drug: *side effects, cost, structure, benefits, mechanism of action, overdose, long term use, price, synthesis, pharmacology*
Company: *ceo, future, president, competitors, mission statement, owner, website, organizational structure, logo, market share*
City: *population, map, mayor, climate, location, geography, best, culture, capital, latitude*
Painter: *paintings, works, portrait, death, style, artwork, bibliography, bio, autobiography, childhood*

Table 3: Top ten attributes acquired from query logs

Class	Precision				
	@10	@20	@30	@40	@50
Country	0.95	0.93	0.90	0.89	0.82
Drug	0.90	0.93	0.83	0.79	0.79
Company	0.90	0.75	0.63	0.56	0.50
City	0.85	0.73	0.72	0.66	0.65
Painter	0.95	0.93	0.85	0.84	0.75

Table 4: Precision at various ranks n

pecially within the crucial [2] top ten entries.

Recall: In general, the measurement of recall requires knowledge of the complete set of items (in our case, attributes) to be extracted. Unfortunately, this number is often unavailable in information extraction tasks [Hasegawa *et al.*, 2004; Cafarella *et al.*, 2005]. Following a scenario that matches well the intended use of our extracted attributes, an unbiased gold standard collects attributes of the class *Country* from the first 1,893 main-task questions used in past editions of the Question Answering track of the Text REtrieval Conference [Voorhees and Tice, 2000]. Table 5 gives these attributes sorted by frequency of appearance among the questions, along with their respective ranks as exact strings within our own extracted attribute list. For example, the gold-standard attribute *capital city*, collected from questions such as *"What is the capital city of Algeria?"*, is present among the extracted attributes at rank 5.

Alternate Sources of Evaluation Data: In a more holistic evaluation of recall, starting from a few provided sample attributes for the classes *Dog* and *Planet*, survey participants were asked to provide a set of what they considered to be important attributes for each of our five target classes. Each of our extracted lists of attributes are searched by hand in order to find candidates that are deemed semantically equivalent to what participants provided. [3] A sample of these results are provided in Table 6. Note that (*City*, *quality of living*) is not an attribute that we would consider to be *vital*, as compared to some equivalent version of (*Drug*, *is it addictive*) or (*Company*, *is it nonprofit*). From the survey results it was primarily these sort of binary valued attributes that we failed to extract, leading us to consider if in the future alternate patterns may potentially improve our performance.

Specifically for the class *Country*, Table 7 contains examples of attributes mentioned by textual descriptions found in the CIA Factbook. [4] The table also shows where the attributes occur in our own extracted list. Unsurprisingly, "common" attributes such as *flag* and *map* are more attainable via query logs versus specific, yet equally correct, gold-standard attributes such as *household income consumption by percentage share* or *manpower reaching military service age annually*. In fact, we feel that the automatic extraction of such detailed, "summarizing" attributes from natural-language text is beyond current state of the art in information extraction.

[2] Web search engines also tend to display ten results by default.

[3] For example, the survey response (*Country*, *income*) was deemed equivalent to the extracted (*Country*, *per capita income*).

Attribute	Rank	Attribute	Rank
capital	1	*emperor*	-
population	2	*date of independence*	50
Prime Minister	19	*currency*	6
leader	10	*area*	23
capital city	5	*Queen*	76
President	3	*GDP*	32
size	20		

Table 5: Gold-standard attributes for *Country* from TREC questions, with their rank in the filtered, *frequency*-ranked list

Attribute	R.	Attribute	R.
(*Painter*, *nationality*)	30	(*Painter*, *influences*)	11
(*Drug*, *is it addictive*)	-	(*Painter*, *awards*)	-
(*Drug*, *side effects*)	1	(*Country*, *income*)	88
(*Company*,*is it nonprofit*)	-	(*Country*, *neighbors*)	-
(*Company*, *competitors*)	4	(*City*, *mayor*)	3
(*City*, *quality of living*)	-	(*City*, *taxes*)	30

Table 6: Sample of attributes given by survey participants and their ranks in the filtered, *frequency*-ranked list (R. = Rank)

6 Comparison to Previous Work

Although our work naturally fits into the larger goal of building knowledge bases automatically from text [Craven *et al.*, 2000; Schubert, 2002], to our knowledge we are the first to explore the use of query logs for the purpose of attribute extraction. Similar goals motivated a few other recent studies, although the scale, underlying methods and data sources differ significantly. In [Chklovski, 2003], the acquisition of attributes relies on Web users who *explicitly* specify class attributes given a class name. In contrast, we may think of our approach as Web users *implicitly* giving us the same type of information. The method proposed in [Tokunaga *et al.*, 2005] 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*

[4] http://www.cia.gov/cia/publications/factbook

Attribute	Rank	Attribute	Rank
natural resources	21	*irrigated land*	-
terrain	132	*administrative divisions*	-
climate	7	*infant mortality rate*	-

Table 7: Sample of *Country* attributes from the CIA Factbook and their rank within the filtered, *frequency*-ranked list

of Prilosec?") used in [Tokunaga *et al.*, 2005].

We might view at least a fraction of the Web query logs as a collection of interrogations corresponding to pre-generated, natural-sounding questions, except that they have been stripped of most of their syntax, and have been mixed in with queries that were never questions to begin with. From this perspective, our task is related to classifying queries into questions vs. non-questions.

7 Conclusion

Recent studies in large-scale knowledge acquisition from text are fueled by the belief that the Web as a whole represents a huge repository of human knowledge. Taking this idea further, this paper presented a method for extracting a particular type of knowledge, namely class attributes, based on the hypothesis that Web search queries as a whole also mirror a significant amount of knowledge. The knowledge is implicitly encoded in obfuscated queries that are usually ambiguous, keyword-based approximations of often-underspecified user information needs. In one of the first attempts to systematically decode and exploit a very small part of the information that Web queries wear on their sleeves, we introduced a robust model for extracting class attributes from query logs, producing average precision levels of 91% and 78%, over the top ten and top thirty extracted attributes respectively.

While the approach relies heavily on query logs, we have begun exploring how to incorporate additional evidence from more traditional sources, such as natural language text and semi-structured text (e.g., tables), which may help further improve the quality of the output.

Building upon recent contributions towards large-scale knowledge acquisition, we see our efforts as part of a larger program aimed at constructing large fact databases. In our vision, successive text mining stages start from unlabeled natural language text and 1) identify clusters of instances grouped into semantic classes (cf. [Lin and Pantel, 2002]); 2) assign labels to the corresponding classes (cf. [Pantel and Ravichandran, 2004]); 3) acquire the most prominent attributes of each class (e.g., through our method); and 4) identify the values of the attributes for various instances of the class (based on current state of the art in answer extraction).

References

[Agichtein and Gravano, 2000] 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.

[Cafarella *et al.*, 2005] 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.

[Chklovski, 2003] T. Chklovski. *Using Analogy to Acquire Commonsense Knowledge from Human Contributors.* PhD thesis, MIT Artificial Intelligence Laboratory, 2003.

[Craven *et al.*, 2000] M. Craven, D. DiPasquo, D. Freitag, A. McCallum, T. Mitchell, K. Nigam, and S. Slattery. Learning to construct knowledge bases from the world wide web. *Artificial Intelligence*, 118:69–113, 2000.

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

[Hasegawa *et al.*, 2004] T. Hasegawa, S. Sekine, and R. Grishman. Discovering relations among named entities from large corpora. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 415–422, Barcelona, Spain, 2004.

[Lin and Pantel, 2002] 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.

[Lita and Carbonell, 2004] 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.

[Paşca *et al.*, 2006] M. Paşca, D. Lin, J. Bigham, A. Lifchits, and A. Jain. Organizing and searching the World Wide Web of facts - step one: the one-million fact extraction challenge. In *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI-06)*, Boston, Massachusetts, 2006.

[Pantel and Ravichandran, 2004] 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.

[Schubert, 2002] L. Schubert. Can we derive general world knowledge from texts? In *Proceedings of the Human Language Technology Conference (HLT-02)*, pages 94–97, San Diego, California, 2002.

[Shinzato and Torisawa, 2004] 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.

[Tokunaga *et al.*, 2005] 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*, pages 106–118, Jeju Island, Korea, 2005.

[Turney, 2001] P. Turney. Mining the Web for synonyms: PMI-IR vs. LSA on TOEFL. In *Proceedings of the 12th European Conference on Machine Learning (ECML-01)*, pages 491–502, Freiburg, Germany, 2001.

[Voorhees and Tice, 2000] E.M. Voorhees and D.M. Tice. Building a question-answering test collection. In *Proceedings of the 23rd International Conference on Research and Development in Information Retrieval (SIGIR-00)*, pages 200–207, Athens, Greece, 2000.

[Voorhees, 2003] 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.