The Role of Query Sessions in Extracting Instance Attributes from Web Search Queries

Marius Paşca, Enrique Alfonseca, Enrique Robledo-Arnuncio,
Ricardo Martin-Brualla*, and Keith Hall

Google Inc.
{mars,ealfonseca,era,rmbrualla,kbhall}@google.com

Abstract. Per-instance attributes are acquired using a weakly supervised extraction method which exploits anonymized Web-search query sessions, as an alternative to isolated, individual queries. Examples of these attributes are *top speed* for *chevrolet corvette*, or *population density* for *brazil*). Inherent challenges associated with using sessions for attribute extraction, such as a large majority of within-session queries not being related to attributes, are overcome by using attributes globally extracted from isolated queries as an unsupervised filtering mechanism. In a head-to-head qualitative comparison, the ranked lists of attributes generated by merging attributes extracted from query sessions, on one hand, and from isolated queries, on another hand, are about 12% more accurate on average, than the attributes extracted from isolated queries by a previous method.

1 Introduction

Motivation: Early work on information extraction studies how to train supervised systems on small to medium-sized document collections, requiring relatively expensive, manual annotations of data [1]. More recently, some authors investigate the possibility of obtaining annotated corpora more easily, through the creation of semi-automatic annotations [2]. But as larger amounts of textual data sources have become available at lower computational costs, either directly as document collections or indirectly through the search interfaces of the larger Web search engines, information extraction has seen a shift towards large-scale acquisition of open-domain information [3]. In this framework, information at mainly three levels of granularity is extracted from text, with weak or no supervision: class instances (e.g., *vicodin*, *oxycontin*); associated class labels (e.g., *painkillers*), and relations. These last may hold between instances (e.g., *france-capital-paris*) or classes (e.g., *countries-capital-cities*) [4, 5].

One type of relation that can be learned for classes and instances are their attributes (e.g., *side effects* and *maximum dose*), which capture quantifiable properties of their respective classes (e.g., *painkillers*) or instances (e.g., *oxycontin*), and thus serve as building blocks in the knowledge bases constructed around open-domain classes or instances. Consequently, a variety of attribute extraction methods mine textual data sources ranging from unstructured [6] or structured [7,8] text within Web documents, to human-compiled

* Contributions made during an internship at Google.

C. Gurrin et al. (Eds.): ECIR 2010, LNCS 5993, pp. 62–74, 2010.

encyclopedia [9], in an attempt to extract, for a given class, a ranked list of attributes that is as comprehensive and accurate as possible.

Using Query Session: Although Web search query logs have already been used for automatically extracting instance attributes [10, 11], as far as we know query session information (indicating which queries are issued by the same user within a limited amount of time) has not been explored for finding instance attributes. Session data is richer than simple sets of individual queries, because sessions contain queries issued in sequence, and may thus be related to one another. This paper explores the use of search queries for automatically extracting instance attributes, and shows that simple algorithms can produce results that are competitive with the current state of the art. Furthermore, by combining the results of this new approach with previous work [10, 11] we are able to produce ranked lists of attributes with much higher precision. This is an interesting result, indicating that the kind of information contained in session logs is complementary to the one that can be obtained from single-query logs and web documents.

Applications: The special role played by attributes, among other types of relations, is documented in earlier work on language and knowledge representation [12, 13]. It inspired the subsequent development of text mining methods aiming at constructing knowledge bases automatically [14]. In Web search, the availability of instance attributes is useful for applications such as search result ranking and suggestion of related queries [15], and has also been identified to be a useful resource in generating product recommendations [16].

2 Previous Work

Query Sessions: A query session is a series of queries submitted by a single user within a small range of time [17, 18, 19]. Information stored in the session logs may include the text of the queries, together with some metadata: the time, the type of query (e.g. using the normal or the advance form), and some user settings, such as the Web browser used [17].

One of the primary uses of query sessions is the identification of related queries [20, 21]. In turn, the related queries can be used, for example, as query suggestions to help users refine their queries, or query substitutions for increasing recall in sponsored search.

Typical intra-session association metrics are the chi-square test and the correlation coefficient [17], the Mutual Information and Pointwise Mutual Information metrics [22], or the log-likelihood ratio (LLR) [23, 24]. High LLR values indicate that two queries are substitutable, i.e., they are close in meaning and for most practical purposes one could be replaced with the other, as with *baby trolley* and *baby cart*. It was shown [24] that, if one removes all substitutable query pairs from sessions, the remaining pairs that still have high LLR are associated queries, which refer to closely related, but different concepts, e.g., *ski* and *snowboard*. Other metrics take into account user clicks to relate queries that lead to clicks on the same results [25, 26]. There is also increasing interest on classifying the relationships between consecutive queries in sessions, in order to identify the user intent [27, 28, 29] when issuing two consecutive queries.

Query logs have been used in the past for obtaining semantic information [30, 31, 32, 33, 34]. The most similar work that we have found is [35], which learn *query aspects*. The main differences are that (a) [35] does not make the distinction between class labels and attributes, as both can be considered aspects of queries; and (b) it is focused on clustering the attributes in very few (one to three) maximally informative aspects, whereas this paper focuses on maximising precision for large sets of attributes.

Learning Instance Attributes: Previous work on attribute extraction uses a variety of types of textual data as sources for mining attributes. Taking advantage of structured and semi-structured text available within Web documents, the method introduced in [7] assembles and submits list-seeking queries to general-purpose Web search engines, and analyzes the retrieved documents to identify common structural (HTML) patterns around class labels given as input, and potential attributes. Similarly, layout (e.g., font color and size) and other HTML tags serve as clues to acquire attributes from either domain-specific documents such as those from product and auction Web sites [36], or from arbitrary documents [37]. As an alternative to Web documents, articles within online encyclopedia can also be exploited as sources of structured text for attribute extraction, as illustrated by previous work using infoboxes and category labels [38, 39, 40] associated with articles within Wikipedia.

Working with unstructured text within Web documents, the method described in [6] applies manually-created lexico-syntactic patterns to document sentences in order to extract candidate attributes, given various class labels as input. The candidate attributes are ranked using several frequency statistics. If the documents are domain-specific, such as documents containing product reviews, additional heuristically-motivated filters and scoring metrics can be used to extract and rank the attributes [41]. In [15], the extraction is guided by a small set of seed instances and attributes rather than manually-created patterns, with the purpose of generating training data and extract new pairs of instances and attributes from text.

Web search queries have also been considered as a textual data source for attribute extraction, using lexico-syntactic patterns [10] or seed attributes [11] to guide the extraction, and leading to attributes of higher accuracy than those extracted with equivalent techniques from Web documents [42].

3 Extraction Method

Extraction from Query Sessions: Intuitively, some of the search engine users interested in information about an instance $\mathcal{I}$ may attempt to search for different characteristics of $\mathcal{I}$ during the same search session, in order to collect more complete information. For example, someone looking for information about the president of the United States may start with a query containing just his name, *barack obama*, and proceed with other queries to get more results containing some of his most relevant attributes, such as his biography, early life, quotes, opinions, poll results, etc. Although clearly not every user is expected to behave this way, as long as at least some users do display this demeanour, it is possible to learn relevant attributes about various instances.

Web search queries are typically very short, containing 2.8 words on average [43], although there is recent evidence that the average length of the queries has grown over

For i from 1 to $n-1$
1. For j from $i+1$ to n
   (a) If q_i is a prefix of q_j, strip the prefix from q_j and add the remainder as a candidate attribute for q_i.
   (b) Otherwise, stop the inner loop.

Fig. 1. Algorithm for collecting candidate attributes from a query session (i.e., a sequence of consecutive queries from the same user) of the form $\mathcal{S} = [q_1, q_2, ..., q_n]$

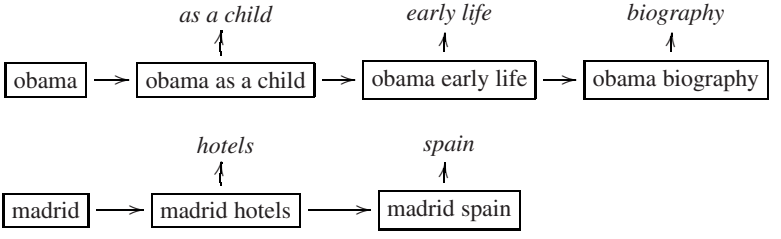

Fig. 2. Example fragments from query sessions, and candidate attributes extracted from them

time [44]. Queries may exhibit some linguistic structure, but this is typically simple: 71% of all query terms are proper or common nouns, with only 3% being prepositions, and almost 70% of the full queries are noun phrases [45]. Given their simple structure, queries seeking for information about an instance and an attribute in particular are most likely to occur in the simplest possible form, that is, the concatenation of the instance and the attribute, as in *barack obama biography*. Other forms, such as *biography of barack obama*, are equally valid but less likely to be submitted as queries.

The algorithm to collect candidate attributes from query sessions is shown in Figure 1. Figure 2 shows two examples of query sessions from which candidate attributes would be extracted: [as a child, early life, biography] for *obama*, and [hotels, spain] for *madrid*. After this step has been performed for all queries, candidate attributes within each instance are ranked among one another by their frequency.

Note that, if the session contains the two consecutive queries *[hotel]*, *[hotel madrid]*, then *madrid* will be extracted as a candidate attribute for the instance *hotel*. These cases will be taken care of in the filtering step.

Attribute Filtering: As can be seen in the previous example, the simple co-occurrence in consecutive queries far from guarantees that the extracted phrase is an attribute of the original query. In the example, *as a child* is a temporal restriction from a user interested in biographies, whereas *spain* is a different instance related to the original query through the country-capital relation.

In order to identify good attributes among inherently noisy phrases extracted from sessions, we have created automatically a whitelist that contains every phrase that appears as an attribute of at least one instance in the dataset associated with the attribute extraction method described in [10]. Any attribute from the ranked candidate lists that does not appear in the whitelist is removed. Table 1 shows the effect of this filtering for

Table 1. Example candidate attributes extracted for *shakespeare*

Rank	Before Filtering	After Filtering	Rank	Before Filtering	After Filtering
1	quotes	quotes	11	globe theater	costumes
2	plays	plays	12	romeo and juliet	family
3	globe theatre	biography	13	cartoon	characters
4	biography	poems	14	life	house
5	in love	sonnets	15	globe	death
6	in the park	facts	16	as you like it	bio
7	poems	life	17	midsummer night's dream	works
8	fishing	books	18	books	history
9	sonnets	pictures	19	macbeth	clothing
10	facts	timeline	20	hamlet	poetry

the query *shakespeare*. As can be seen, the second list is much more precise than the raw list of extracted phrases, as names of plays and other common phrases co-occurring with shakespeare are removed.

Unsupervised Merging of Attributes: To take advantage of the potentially different strengths of isolated queries vs. query sessions, as resources for attribute extraction, the ranked lists of attributes generated with the above method, on one hand, and as introduced in [10], on the other hand, are merged. More precisely, for each instance, a merged, ranked list of attributes is created from all attributes extracted by at least one of the two methods. The rank of an attribute in the merged list is determined by a merged score, assigned to an attribute $\mathcal{A}$ based on its ranks in the two input lists of attributes $\mathcal{L}$ as follows:

$$MergedScore(\mathcal{A}) = \frac{|\{\mathcal{L}\}|}{sum_{\mathcal{L}} Rank(\mathcal{A}, \mathcal{L})}$$

where $|\{\mathcal{L}\}|$ is the number of input lists of attributes (i.e., 2 in this case), and $Rank$ $(\mathcal{A}, \mathcal{L})$ is the rank of $\mathcal{A}$ in the input list $\mathcal{L}$ (or 1000, if $\mathcal{A}$ is not present in the input list $\mathcal{L}$). For each instance, the merged list of attributes is obtained by ranking the attributes in decreasing order of their merged scores.

4 Experimental Setting

Previous Approach: The method described in [10] is applied over a fully-anonymized set of English queries submitted to the Google search engine. The set contains about 50 million unique isolated, individual queries that are independent from one another. Each query is accompanied by its frequency of occurrence in the query logs. The sum of frequencies of all queries in the dataset is around 1 billion.

The extraction method introduced in [10] applies a few patterns (e.g., *the* $\mathcal{A}$ *of* $\mathcal{I}$, or $\mathcal{I}$*'s* $\mathcal{A}$, or $\mathcal{A}$ *of* $\mathcal{I}$) to queries within query logs, where an instance $\mathcal{I}$ is one of the most frequent 5 million queries from the repository of isolated queries, and $\mathcal{A}$ is a candidate attribute. For each instance, the method extracts ranked lists containing zero, one or more attributes, along with frequency-based scores.

Session-Based Approach: The method to extract attributes from sessions is applied over a repository containing roughly 5 billion anonymized query sessions from 2009. The attribute whitelist was created as the union of the top 50 attributes extracted across all instances from the previous approach. The total number of queries in all sessions is roughly 10 billion. Each session contains consecutive queries from a single user, such that every two consecutive queries were issued with no more than a few minutes between them.

Experimental Runs: The experiments evaluate three different runs: $\mathrm{Q}_y\mathrm{S}_n$, $\mathrm{Q}_n\mathrm{S}_y$ and $\mathrm{Q}_y\mathrm{S}_y$, where Q and S stand for extraction from individual queries or from sessions respectively, and y/n indicate whether the respective extraction method is enabled (y) or not (n). Thus, $\mathrm{Q}_y\mathrm{S}_n$ extracts attributes from individual queries; $\mathrm{Q}_n\mathrm{S}_y$ from query sessions; and $\mathrm{Q}_y\mathrm{S}_y$ is the unsupervised, rank-based merging of the ranked lists of attributes extracted by $\mathrm{Q}_y\mathrm{S}_n$ and $\mathrm{Q}_n\mathrm{S}_y$.

Target Instances: The runs $\mathrm{Q}_y\mathrm{S}_n$, $\mathrm{Q}_n\mathrm{S}_y$ and $\mathrm{Q}_y\mathrm{S}_y$ may naturally acquire ranked lists of attributes of different lengths for the same instance, due to the distinct characteristics of the underlying repositories (individual queries vs. query sessions vs. a combination). In order to avoid any uncontrolled effects of variable-length lists of attributes on the outcome of the evaluation, a random sample of 200 instances is drawn from the set of all instances, such that at least 50 attributes (i.e., the same as the number of attributes whose accuracy is evaluated per instance, as described later) are extracted for each instance in all runs. The sample is further inspected manually, in order to eliminate instances for which human annotators would likely need a long time to become familiar with the instance and its meanings, before they can assign correctness labels to the attributes extracted for the instance. The only purpose of the manual selection step is to keep the costs associated with the subsequent, manual evaluation of attributes within reasonable limits. To remove any possible bias towards instances with more or better attributes, the extracted attributes, if any, remain invisible during the manual selection of instances. For example, the instance *allan* (which may be one of several cities, a movie or one of many people) is discarded due to extreme ambiguity. Conversely, *attention deficit disorder* is retained, since it is relatively less difficult to notice that it refers to a particular disease. The manual selection of instances, from the random sample of 200 instances, results in an evaluation set containing 100 target instances, as shown in Table 2.

Evaluation Procedure: The measurement of recall requires knowledge of the complete set of items (in our case, attributes) to be extracted. Unfortunately, the manual enumeration of all attributes of each target instance, to measure recall, is unfeasible. Therefore, the evaluation focuses on the assessment of attribute accuracy.

To remove any bias towards higher-ranked attributes during the assessment of instance attributes, the top 50 attributes within the ranked lists of attributes produced by each run to be evaluated are sorted alphabetically into a common list. Each attribute of the common list is manually assigned a correctness label within its respective instance. In accordance with previously introduced methodology, an attribute is *vital* if it must be present in an ideal list of attributes of the instance (e.g., *side effects* for *digoxin*); *okay* if

Table 2. Set of 100 target instances, used in the evaluation of instance attribute extraction

Instances
17th century, accounting, alton towers, ancient greece, artificial intelligence, attention deficit disorder, beans, biodiesel, body language, brampton, brazil, cadmium, capri, cardboard, chhattisgarh, civil engineering, clay, cobol, communication skills, constantine, contemporary art, corporate governance, cortex, cricket, crisps, data warehousing, death penalty, decimals, delhi, dentist, digoxin, dns, electronic commerce, ferns, finland, forensics, fredericton, glycine, guinea pig, guitars, gurgaon, halogens, high blood pressure, hilary duff, instructional design, irrigation, jessica simpson, johnny depp, kidney stones, library science, lil romeo, majorca, manisha koirala, maya angelou, medicaid, medical records, methanol, mexico city, moon phases, nematodes, oil, pancho villa, pensacola, phosphorus, photography, physician assistant, podiatry, police brutality, prednisone, prose, qualitative research, railroads, reese witherspoon, refrigerator, reggaeton, resistors, richard branson, ritalin, robotics, rock n roll, san francisco, sheep, sickle cell disease, sindh, sir isaac newton, standard deviation, tata young, thyroid gland, titration, treason, tundra, utilitarianism, vida guerra, volcanos, warwick, wastewater treatment, wellbutrin, western canada, wlan, yoghurt

Table 3. Correctness labels for the manual assessment of attributes

Label	Value	Examples of Attributes
vital	1.0	beans: calories, digoxin: side effects, maya angelou: age
okay	0.5	fredericton: heritage, library science: current events, robotics: three laws
wrong	0.0	alton towers: park view, kidney stones: pain, contemporary art: urban institute

it provides useful but non-essential information; and *wrong* if it is incorrect [11]. Thus, a correctness label is manually assigned to a total of 9,137 attributes extracted for the 100 target instances, in a process that confirms that evaluation of information extraction methods can be quite time consuming.

An analysis of the correctness labels assigned by two human judges to 500 extracted attributes indicates an inter-annotator agreement of 88.79%, resulting in a Kappa score of 0.85, indicating substantial agreement in this task.

To compute the precision score over a ranked list of attributes, the correctness labels are converted to numeric values (*vital* to 1, *okay* to 0.5 and *wrong* to 0), 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 attributes, divided by N.

5 Evaluation Results

Extracted Instance Attributes: The first row in Table 4 contains the total number of instances for which at least one attribute was extracted and the total number of <instance,attribute> pairs extracted for $\mathrm{Q}_y\mathrm{S}_n$. For those 483,344 instances, the second row shows how many have at least one attribute extracted by $\mathrm{Q}_n\mathrm{S}_y$, and the total number of <instance,attribute> pairs extracted for them. Note that the numbers are not comparable, given that the original data sources: a) are different in size (the input

Table 4. For Q_yS_n: number of instances with at least one attribute and total number of <instance,attribute> pairs extracted. For Q_nS_y, among the previous instances, how many have attributes extracted and total number of attributes.

Method	Instances	<Instance,Attribute> Pairs
Q_yS_n	483,344	8,974,433
Q_nS_y	462,701	14,823,701

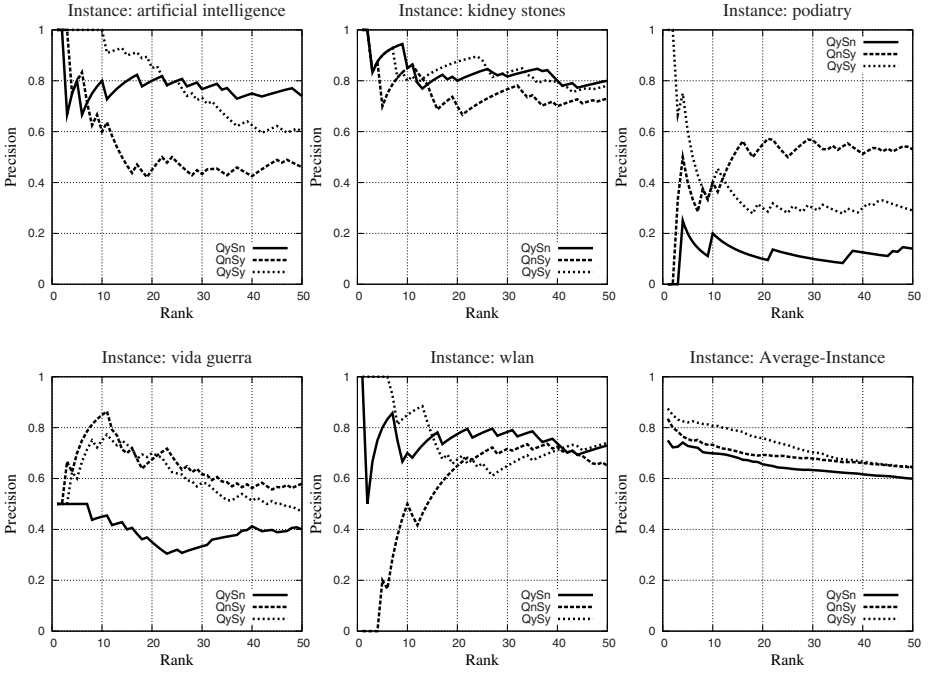

Fig. 3. Accuracy of the ranked lists of attributes extracted by various runs, for a few target instances and as an average over all target instances

for Q_yS_n is ten times smaller) and b) are different in nature (the input for Q_yS_n contains isolated queries, not sessions), and c) are affected by different restrictions (e.g., the vocabulary of instances in Q_yS_n is limited to the most frequent 5 million queries). However, as will be shown, precision improves when using session logs, without a drop in recall. Indeed, almost 96% of the instances for which Q_yS_n extracted some attributes also have attributes extracted by Q_nS_y, as illustrated in Table 4.

Session information may be more useful for extracting information about rare entities: in fact, Q_nS_y extracts attributes for many new instances whose attributes are not found with Q_yS_n. A possible reason is the fact that a user issuing two consecutive queries $\mathcal{I}$ and $\mathcal{I}\mathcal{A}$, where $\mathcal{A}$ is a common attribute name and $\mathcal{I}$ is an instance, may be a strong indicator that $\mathcal{I}$ has $\mathcal{A}$ as one of its attributes, and Q_nS_y is able to extract this

Table 5. Comparative accuracy of the attributes extracted in various experimental runs, for a few target instances and as an average over the entire set of target instances. Scores are expressed as *Abs*olute scores, *Rel*ative boosts (over Q_yS_n), and *Err*or reduction rates (over Q_yS_n). Scores are marked with * and † if they are indistinguishable at 99% and 95% confidence respectively.

Instance	Precision											
	@5			@10			@20			@50		
	Q_yS_n	Q_nS_y	Q_yS_y	Q_yS_n	Q_nS_y	Q_yS_y	Q_yS_n	Q_nS_y	Q_yS_y	Q_yS_n	Q_nS_y	Q_yS_y
17th century (*Abs*)	0.50	0.80	0.70	0.40	0.70	0.60	0.35	0.68	0.53	0.40	0.76	0.54
artificial intelligence (*Abs*)	0.80	0.80	1.00	0.80	0.60	1.00	0.80	0.45	0.85	0.74	0.46	0.60
brazil (*Abs*)	0.90	1.00	0.90	0.85	0.95	0.95	0.78	0.97	0.95	0.79	0.92	0.87
communication skills (*Abs*)	0.70	0.90	1.00	0.85	0.80	1.00	0.68	0.62	0.82	0.67	0.56	0.52
electronic commerce (*Abs*)	1.00	1.00	1.00	0.60	0.90	0.90	0.75	0.75	0.95	0.64	0.60	0.62
gurgaon (*Abs*)	0.70	0.90	0.80	0.85	0.75	0.85	0.65	0.72	0.78	0.60	0.76	0.70
kidney stones (*Abs*)	0.90	0.70	0.90	0.85	0.85	0.80	0.80	0.70	0.88	0.80	0.73	0.78
medicaid (*Abs*)	0.60	1.00	0.60	0.60	0.80	0.70	0.62	0.70	0.75	0.49	0.64	0.61
podiatry (*Abs*)	0.20	0.40	0.60	0.20	0.40	0.40	0.10	0.55	0.30	0.14	0.53	0.29
robotics (*Abs*)	0.50	0.40	0.80	0.75	0.30	0.55	0.57	0.30	0.53	0.58	0.31	0.41
sickle cell disease (*Abs*)	0.90	0.90	0.90	0.90	0.85	0.95	0.78	0.82	0.88	0.73	0.73	0.76
vida guerra (*Abs*)	0.50	0.70	0.60	0.45	0.85	0.75	0.35	0.68	0.70	0.40	0.58	0.47
wlan (*Abs*)	0.80	0.20	1.00	0.70	0.50	0.85	0.78	0.65	0.68	0.73	0.65	0.74
Avg-Inst (*Abs*)	0.73*	0.76*†	0.82	0.70	0.73*	0.81	0.66	0.69	0.76	0.60	0.64*†	0.65
Avg-Inst (*Rel*)	-	+4%	+12%	-	+4%	+16%	-	+5%	+15%	-	+7%	+8%
Avg-Inst (*Err*)	-	-11%	-33%	-	-10%	-37%	-	-9%	-29%	-	-10%	-13%

pair. On the contrary, if $\mathcal{I}$ does not appear any more in the logs, just those two individual queries do not provide enough support for Q_yS_n to extract that information.

Accuracy of Instance Attributes: Figure 3 plots precision values for ranks 1 through 50, for each of the experimental runs. The first five graphs in the figure show the precision over individual target instances. Several conclusions can be drawn after inspecting the results. First, the quality of the attributes extracted by a given run varies among instances. For example, the attributes extracted for the instance *kidney stones* are better than for *vida guerra*. Second, the experimental runs have variable levels of accuracy. The last (i.e., lower right) graph in Figure 3 shows the precision as an average over all target instances. Although none of the runs outperforms the others on each and every target instance, on average, Q_yS_y performs the best and Q_yS_n (i.e., the baseline) the worst, with Q_nS_y placed in-between. In other words, attributes are more accurate when extracted from sessions (Q_nS_y) rather than from individual queries (Q_yS_n) - although

Table 6. Ranked lists of attributes extracted for a sample of the target instances

Run	Top Extracted Attributes
Instance: 17th century:	
Q_yS_n	timeline, pictures, politics, fashion, anatomy, french classicism art, accessories, austria weapons, composers, era
Q_nS_y	fashion, clothing, art, costume, paintings, houses, weapons, names, timeline, music
Q_yS_y	fashion, timeline, composers, pictures, politics, authors, clothing, art, costume, anatomy
Instance: kidney stones:	
Q_yS_n	symptoms, causes, pictures, treatment, types, signs, prevention, signs and symptoms, removal, symptons
Q_nS_y	symptoms, treatment, pictures, causes, diet, symptoms in women, natural remedies, size, prevention, cure
Q_yS_y	symptoms, causes, pictures, treatment, prevention, types, size, symptons, signs, images
Instance: robotics:	
Q_yS_n	history, three laws, future, laws, 3 laws, basics, definition, applications, introduction, fundamentals
Q_nS_y	history, logo, basics, parts, career, competition, jobs, definition, technology, pictures
Q_yS_y	history, basics, definition, future, introduction, pictures, advantages, disadvantages, laws, types
Instance: sickle cell disease:	
Q_yS_n	symptoms, pictures, history, geographical distribution, causes, management, treatment, pathogenesis, pathology, new considerations in the treatment
Q_nS_y	symptoms, treatment, pictures, history, life expectancy, statistics, causes, cure, genetics, diagnosis
Q_yS_y	symptoms, pictures, history, treatment, causes, life expectancy, pathophysiology, effects, incidence, management

the attributes extracted in Q_yS_n across all instances do serve as a filtering mechanism for Q_nS_y, as explained earlier. The unsupervised merging of the extracted lists of attributes (Q_yS_y) gives an even larger improvement in accuracy relative to Q_yS_n.

For a more detailed analysis of qualitative performance, the upper part of Table 5 provides the precision scores for a sample of the target instances. For completeness, the scores in the table capture precision at the top of the extracted lists of attributes (rank 5) as well as over a wider range of those lists (ranks 10 and above). The table gives another view at how widely the quality of the extracted attributes may vary depending on the target instance. At the lower end, the precision of the attributes for the instances *podiatry* (with Q_yS_n) and *wlan* (with Q_nS_y) is as low as 0.20 at rank 5. At the higher end, the attributes for *sickle cell disease* are very good across all runs, with precision scores above 0.78 even at rank 20. The top attributes extracted for various instances are shown in Table 6.

When considering the comparative precision of the experimental runs in Table 5, run Q_yS_n extracts better attributes than Q_nS_y for some instances, e.g., at all ranks for *robotics* and especially for *wlan*. However, the opposite is true for most of the individual instances (e.g., *17th century*, *brazil*, *medicaid*, *podiatry*, *vida guerra*).

To better quantify the quality gap, the last rows of Table 5 show the precision computed as an average over all instances, rather than for each instance individually, and therefore they correspond to points on the curves from the last graph of Figure 3. Also shown in the table are the relative increases (*Rel*) and the reduction in the error rates (*Err*) at various ranks, for Q_nS_y and Q_yS_y, on one hand, relative to Q_yS_n, on the other hand. Consistently over all computed ranks, the precision is about 5% better on average when using sessions rather than individual queries, and about 12% better when merging attributes from sessions and individual queries in an unsupervised fashion. The results of P@20 shows that the combination is better than any of the standalone systems with 99%, and for P@50 the Q_yS_y system is better than the baseline Q_yS_n, also with 99% confidence. This is the most important result of the paper. It shows that query sessions represent a useful resource as a complement to individual queries, in instance attribute extraction.

6 Conclusions

This paper describes a procedure for extracting instance attributes from session query logs, by looking for pairs of consecutive queries such that the second one contains the first one as a prefix. A simple ranking function by query frequency produces results that improve (on aggregate) over a previous state-of-the-art system [10]. Compared to it, the main advantages are an increase in relative recall and precision scores, and the possibility of extracting attributes for less frequent queries.

More importantly, the combination of the two resources reduces the error rate between 13% and 37%, with respect to the precision of the extracted lists of attributes at various ranks. The improvement over the baseline system is statistically significant with 99% confidence for precision scores P@N with N higher or equal than 10.

Current work investigates ways of exploiting other information present in sessions, such as user clicks, to further improve the quality and coverage of the results.

References

1. Grishman, R., Sundheim, B.: Message Understanding Conference-6: a brief history. In: Proceedings of the 16th Conference on Computational Linguistics, vol. 1, pp. 466–471 (1996)
2. Chklovski, T., Gil, Y.: An analysis of knowledge collected from volunteer contributors. In: Proceedings of the National Conference on Artificial Intelligence, p. 564 (2005)
3. Etzioni, O., Banko, M., Soderland, S., Weld, S.: Open information extraction from the web. Communications of the ACM 51(12) (December 2008)
4. Sekine, S.: On-demand information extraction. In: Proceedings of the COLING/ACL on Main conference poster sessions, pp. 731–738 (2006)
5. Banko, M., Cafarella, M.J., Soderland, S., Broadhead, M., Etzioni, O.: Open information extraction from the Web. In: Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI 2007), pp. 2670–2676 (2007)
6. Tokunaga, K., Kazama, J., Torisawa, K.: Automatic discovery of attribute words from web documents. In: Proceedings of the 2nd International Joint Conference on Natural Language Processing (IJCNLP 2005), Jeju Island, Korea, pp. 106–118 (2005)

7. Yoshinaga, N., Torisawa, K.: Open-domain attribute-value acquisition from semi-structured texts. In: Proceedings of the Workshop on Ontolex, pp. 55–66 (2007)
8. Cafarella, M., Halevy, A., Wang, D., Zhang, Y.: Webtables: Exploring the power of tables on the web. Proceedings of the VLDB Endowment archive 1(1), 538–549 (2008)
9. Wu, F., Hoffmann, R., Weld, D.: Information extraction from Wikipedia: Moving down the long tail. In: Proceedings of the 14th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD 2008), pp. 731–739 (2008)
10. Paşca, M., Van Durme, B.: 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 2007), pp. 2832–2837 (2007)
11. Paşca, M.: Organizing and searching the World Wide Web of facts - step two: Harnessing the wisdom of the crowds. In: Proceedings of the 16th World Wide Web Conference (WWW 2007), Banff, Canada, pp. 101–110 (2007)
12. Pustejovsky, J.: The Generative Lexicon: a Theory of Computational Lexical Semantics. The MIT Press, Cambridge (1991)
13. Guarino, N.: Concepts, attributes and arbitrary relations. Data and Knowledge Engineering 8, 249–261 (1992)
14. Schubert, L.: Turing's dream and the knowledge challenge. In: Proceedings of the 21st National Conference on Artificial Intelligence (AAAI 2006), Boston, Massachusetts (2006)
15. Bellare, K., Talukdar, P., Kumaran, G., Pereira, F., Liberman, M., McCallum, A., Dredze, M.: Lightly-supervised attribute extraction. In: NIPS 2007 Workshop on Machine Learning for Web Search (2007)
16. Probst, K., Ghani, R., Krema, M., Fano, A., Liu, Y.: Semi-supervised learning of attribute-value pairs from product descriptions. In: Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI 2007), pp. 2838–2843 (2007)
17. Silverstein, C., Marais, H., Henzinger, M., Moricz, M.: Analysis of a very large Web search engine query log. In: ACM SIGIR Forum, pp. 6–12 (1999)
18. Jansen, B., Spink, A., Taksa, I.: Handbook of Research on Web Log Analysis. Information Science Reference (2008)
19. He, D., Goker, A.: Detecting session boundaries from web user logs. In: Proceedings of the BCS-IRSG 22nd Annual Colloquium on Information Retrieval Research, pp. 57–66 (2000)
20. Wen, J., Nie, J., Zhang, H.: Clustering user queries of a search engine. In: Proceedings of the International Conference on World Wide Web (2001)
21. Zhang, Z., Nasraoui, O.: Mining search engine query logs for query recommendation. In: Proceedings of the 15th International Conference on World Wide Web, pp. 1039–1040 (2006)
22. Church, K., Hanks, P.: Word association norms, mutual information, and lexicography. Computational Linguistics 16(1), 22–29 (1990)
23. Jones, R., Rey, B., Madani, O., Greiner, W.: Generating query substitutions. In: Proceedings of the 15th International Conference on World Wide Web, pp. 387–396 (2006)
24. Rey, B., Jhala, P.: Mining associations from Web query logs. In: Proceedings of the Web Mining Workshop, Berlin, Germany (2006)
25. Xue, G.R., Zeng, H.J., Chen, Z., Yu, Y., Ma, W.Y., Xi, W., Fan, W.: Optimizing Web search using Web click-through data. In: CIKM 2004: Proceedings of the 13th ACM International Conference on Information and Knowledge Management, pp. 118–126 (2004)
26. Ma, H., Yang, H., King, I., Lyu, M.R.: Learning latent semantic relations from clickthrough data for query suggestion. In: Proceeding of the 17th ACM Conference on Information and Knowledge Management, pp. 709–718 (2008)
27. Lau, T., Horvitz, E.: Patterns of search: Analyzing and modeling web query refinement. In: Proceedings of the International User Modelling Conference (1999)

28. Jones, R., Klinkner, K.L.: Beyond the session timeout: automatic hierarchical segmentation of search topics in query logs. In: CIKM 2008: Proceeding of the 17th ACM conference on Information and Knowledge Management, pp. 699–708 (2008)
29. Boldi, P., Bonchi, F., Castillo, C., Donato, D., Gionis, A., Vigna, S.: The query-flow graph: model and applications. In: CIKM 2008: Proceeding of the 17th ACM conference on Information and Knowledge Management, pp. 609–618 (2008)
30. Baeza-Yates, R., Tiberi, A.: Extracting semantic relations from query logs. In: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 76–85 (2007)
31. Shen, D., Qin, M., Chen, W., Yang, Q., Chen, Z.: Mining Web query hierarchies from clickthrough data. In: Proceedings of the National Conference on Artificial Intelligence (2007)
32. Paşca, M., Van Durme, B.: Weakly-supervised acquisition of open-domain classes and class attributes from web documents and query logs. In: Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics (ACL 2008), Columbus, Ohio, pp. 19–27 (2008)
33. Komachi, M., Makimoto, S., Uchiumi, K., Sassano, M.: Learning semantic categories from clickthrough logs. In: Proceedings of the ACL-IJCNLP 2009 Conference, Short Papers, pp. 189–192 (2009)
34. Pennacchiotti, M., Pantel, P.: Entity extraction via ensemble semantics. In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP 2009), Singapore, pp. 238–247 (2009)
35. Wang, X., Chakrabarti, D., Punera, K.: Mining broad latent query aspects from search sessions. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 867–876 (2009)
36. Wong, T., Lam, W.: An unsupervised method for joint information extraction and feature mining across different web sites. Data & Knowledge Engineering 68(1), 107–125 (2009)
37. Ravi, S., Paşca, M.: Using structured text for large-scale attribute extraction. In: Proceedings of the 17th ACM Conference on Information and Knowledge Management (CIKM 2008), pp. 1183–1192 (2008)
38. Suchanek, F., Kasneci, G., Weikum, G.: Yago: a core of semantic knowledge unifying WordNet and Wikipedia. In: Proceedings of the 16th World Wide Web Conference (WWW 2007), Banff, Canada, pp. 697–706 (2007)
39. Nastase, V., Strube, M.: Decoding Wikipedia categories for knowledge acquisition. In: Proceedings of the 23rd National Conference on Artificial Intelligence (AAAI 2008), Chicago, Illinois, pp. 1219–1224 (2008)
40. Wu, F., Weld, D.: Automatically refining the Wikipedia infobox ontology. In: Proceedings of the 17th World Wide Web Conference (WWW 2008), Beijing, China, pp. 635–644 (2008)
41. Raju, S., Pingali, P., Varma, V.: An unsupervised approach to product attribute extraction. In: Proceedings of the 31th European Conference on IR Research on Advances in Information Retrieval, pp. 796–800 (2009)
42. Paşca, M., Van Durme, B., Garera, N.: The role of documents vs. queries in extracting class attributes from text. In: Proceedings of the 16th International Conference on Information and Knowledge Management (CIKM 2007), Lisbon, Portugal, pp. 485–494 (2007)
43. Spink, A., Jansen, B., Wolfram, D., Saracevic, T.: From e-sex to e-commerce: Web search changes. IEEE Computer 35(3), 107–109 (2002)
44. Hogan, K.: Interpreting hitwise statistics on longer queries. Technical report, Ask.com (2009)
45. Barr, C., Jones, R., Regelson, M.: The linguistic structure of english web-search queries. In: Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing, pp. 1021–1030 (2008)