Ranking Class Labels Using Query Sessions

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

Abstract

The role of search queries, as available within query sessions or in isolation from one another, in examined in the context of ranking the class labels (e.g., *brazilian cities*, *business centers*, *hilly sites*) extracted from Web documents for various instances (e.g., *rio de janeiro*). The co-occurrence of a class label and an instance, in the same query or within the same query session, is used to reinforce the estimated relevance of the class label for the instance. Experiments over evaluation sets of instances associated with Web search queries illustrate the higher quality of the query-based, re-ranked class labels, relative to ranking baselines using document-based counts.

1 Introduction

Motivation: The offline acquisition of instances (*rio de janeiro*, *porsche cayman*) and their corresponding class labels (*brazilian cities, locations, vehicles, sports cars*) from text has been an active area of research. In order to extract fine-grained classes of instances, existing methods often apply manually-created (Banko et al., 2007; Talukdar et al., 2008) or automatically-learned (Snow et al., 2006) extraction patterns to text within large document collections.

In Web search, the relative ranking of documents returned in response to a query directly affects the outcome of the search. Similarly, the quality of the relative ranking among class labels extracted for a given instance influences any applications (e.g., query refinements or structured extraction) using the extracted data. But due to noise in Web data and limitations of extraction techniques, class labels acquired for a given instance (e.g., *oil shale*) may fail to properly capture the semantic classes to which the instance may belong (Kozareva et al., 2008). Inevitably, some of the extracted class labels will be less useful (e.g., *sources*, *mutual concerns*) or incorrect (e.g., *plants* for the instance *oil shale*). In previous work, the relative ranking of class labels for an instance is determined mostly based on features derived from the source Web documents from which the data has been extracted, such as variations of the frequency of co-occurrence or diversity of extraction patterns producing a given pair (Etzioni et al., 2005).

Contributions: This paper explores the role of Web search queries, rather than Web documents, in inducing superior ranking among class labels extracted automatically from documents for various instances. It compares two sources of indirect ranking evidence available within anonymized query logs: a) co-occurrence of an instance and its class label in the same query; and b) co-occurrence of an instance and its class label, as separate queries within the same query session. The former source is a noisy attempt to capture queries that narrow the search results to a particular class of the instance (e.g., *jaguar car maker*). In comparison, the latter source noisily identifies searches that specialize from a class (e.g., *car maker*) to an instance (e.g., *jaguar*) or, conversely, generalize from an instance to a class. To our knowledge, this is the first study comparing inherently-noisy queries and query sessions for the purpose of ranking of open-domain, labeled class instances.

Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics, pages 1607–1615,
Portland, Oregon, June 19-24, 2011. ©2011 Association for Computational Linguistics

The remainder of the paper is organized as follows. Section 2 introduces intuitions behind an approach using queries for ranking class labels of various instances, and describes associated ranking functions. Sections 3 and 4 describe the experimental setting and evaluation results over evaluation sets of instances associated with Web search queries. The results illustrate the higher quality of the query-based, re-ranked lists of class labels, relative to alternative ranking methods using only document-based counts.

2 Instance Class Ranking via Query Logs

Ranking Hypotheses: We take advantage of anonymized query logs, to induce superior ranking among the class labels associated with various class instances within an IsA repository acquired from Web documents. Given a class instance $\mathcal{I}$, the functions used for the ranking of its class labels are chosen following several observations.

- *Hypothesis H_1*: If $\mathcal{C}$ is a prominent class of an instance $\mathcal{I}$, then $\mathcal{C}$ and $\mathcal{I}$ are likely to occur in text in contexts that are indicative of an IsA relation.
- *Hypothesis H_2*: If $\mathcal{C}$ is a prominent class of an instance $\mathcal{I}$, and $\mathcal{I}$ is ambiguous, then a fraction of the queries about $\mathcal{I}$ may also refer to and contain $\mathcal{C}$.
- *Hypothesis H_3*: If $\mathcal{C}$ is a prominent class of an instance $\mathcal{I}$, then a fraction of the queries about $\mathcal{I}$ may be followed by queries about $\mathcal{C}$, and vice-versa.

Ranking Functions: The ranking functions follow directly from the above hypotheses.

- *Ranking based on H_1* (using documents): The first hypothesis H_1 is a reformulation of findings from previous work (Etzioni et al., 2005). In practice, a class label is deemed more relevant for an instance if the pair is extracted more frequently and by multiple patterns, with the scoring formula:

$$Score_{H1}(\mathcal{C},\mathcal{I}) = Freq(\mathcal{C},\mathcal{I}) \times Size(\{Pattern(\mathcal{C})\})^2 \quad (1)$$

where $Freq(\mathcal{C},\mathcal{I})$ is the frequency of extraction of $\mathcal{C}$ for the instance $\mathcal{I}$, and $Size(\{Pattern(\mathcal{C})\})$ is the number of unique patterns extracting the class label $\mathcal{C}$ for the instance $\mathcal{I}$. The patterns are hand-written, following (Hearst, 1992):

⟨[..] $\mathcal{C}$ [such as|including] $\mathcal{I}$ [and|,|.]⟩,
where $\mathcal{I}$ is a potential instance (e.g., *diderot*) and $\mathcal{C}$ is a potential class label (e.g., *writers*). The boundaries are approximated from the part-of-speech tags of the sentence words, for potential class labels $\mathcal{C}$; and identified by checking that $\mathcal{I}$ occurs as an entire query in query logs, for instances $\mathcal{I}$ (Van Durme and Paşca, 2008).

The application of the scoring formula (1) to candidates extracted from the Web produces a ranked list of class labels $\mathrm{L}_{H1}(\mathcal{I})$.

- *Ranking based on H_2* (using queries): Intuitively, Web users searching for information about $\mathcal{I}$ sometimes add some or all terms of $\mathcal{C}$ to a search query already containing $\mathcal{I}$, either to further specify their query, or in response to being presented with sets of search results spanning several meanings of an ambiguous instance. Examples of such queries are *happiness emotion* and *diderot philosopher*. Moreover, queries like *happiness positive psychology* and *diderot enlightenment* may be considered to weakly and partially reinforce the relevance of the class labels *positive emotions* and *enlightenment writers* of the instances *happiness* and *diderot* respectively. In practice, a class label is deemed more relevant if its individual terms occur in popular queries containing the instance. More precisely, for each term within any class label from $\mathrm{L}_{H1}(\mathcal{I})$, we compute a score $TermQueryScore$. The score is the frequency sum of the term within anonymized queries containing the instance $\mathcal{I}$ as a prefix, and the term anywhere else in the queries. Terms are stemmed before the computation.

Each class label $\mathcal{C}$ is assigned the geometric mean of the scores of its N terms $\mathcal{T}_i$, after ignoring stop words:

$$Score_{H2}(\mathcal{C},\mathcal{I}) = (\prod_{i=1}^{N} TermQueryScore(\mathcal{T}_i))^{1/N} \quad (2)$$

The geometric mean is preferred to the arithmetic mean, because the latter is more strongly affected by outlier values. The class labels are ranked according to the means, resulting in a ranked list $\mathrm{L}_{H2}(\mathcal{I})$. In case of ties, $\mathrm{L}_{H2}(\mathcal{I})$ keeps the relative ranking from $\mathrm{L}_{H1}(\mathcal{I})$.

- *Ranking based on H_3* (using query sessions): Given the third hypothesis H_3, Web users searching for information about $\mathcal{I}$ may subsequently search for more general information about one of its classes $\mathcal{C}$. Conversely, users may specialize their search from a class $\mathcal{C}$ to one of its instances $\mathcal{I}$. Examples of such queries are *happiness* followed later by *emotions*, or *diderot* followed by *philosophers*; or *emo-*

tions followed later by *happiness*, or *philosophers* followed by *diderot*. In practice, a class label is deemed more relevant if its individual terms occur as part of queries that are in the same query session as a query containing only the instance. More precisely, for each term within any class label from $\mathrm{L}_{H1}(\mathcal{I})$, we compute a score $TermSessionScore$, equal to the frequency sum of the anonymized queries from the query sessions that contain the term and are: a) either the initial query of the session, with the instance $\mathcal{I}$ being one of the subsequent queries from the same session; or b) one of the subsequent queries of the session, with the instance $\mathcal{I}$ being the initial query of the same session. Before computing the frequencies, the class label terms are stemmed.

Each class label $\mathcal{C}$ is assigned the geometric mean of the scores of its terms, after ignoring stop words:

$$Score_{H3}(\mathcal{C},\mathcal{I}) = (\prod_{i=1}^{N} TermSessionScore(\mathcal{T}_i))^{1/N} \quad (3)$$

The class labels are ranked according to the geometric means, resulting in a ranked list $\mathrm{L}_{H3}(\mathcal{I})$. In case of ties, $\mathrm{L}_{H3}(\mathcal{I})$ preserves the relative ranking from $\mathrm{L}_{H1}(\mathcal{I})$.

Unsupervised Ranking: Given an instance $\mathcal{I}$, the ranking hypotheses and corresponding functions $\mathrm{L}_{H1}(\mathcal{I})$, $\mathrm{L}_{H2}(\mathcal{I})$ and $\mathrm{L}_{H3}(\mathcal{I})$ (or any combination of them) can be used together to generate a merged, ranked list of class labels per instance $\mathcal{I}$. The score of a class label in the merged list is determined by the inverse of the average rank in the lists $\mathrm{L}_{H1}(\mathcal{I})$ and $\mathrm{L}_{H2}(\mathcal{I})$ and $\mathrm{L}_{H3}(\mathcal{I})$, computed with the following formula:

$$Score_{H1+H2+H3}(\mathcal{C},\mathcal{I}) = \frac{N}{\sum_i^N Rank(\mathcal{C}, L_{Hi})} \quad (4)$$

where N is the number of input lists of class labels (in this case, 3), and Rank($\mathcal{C}$, L_{Hi}) is the rank of $\mathcal{C}$ in the input list of class labels L_{Hi} (L_{H1}, L_{H2} or L_{H3}). The rank is set to 1000, if $\mathcal{C}$ is not present in the list L_{Hi}. By using only the relative ranks and not the absolute scores of the class labels within the input lists, the outcome of the merging is less sensitive to how class labels of a given instance are numerically scored within the input lists. In case of ties, the scores of the class labels from $\mathrm{L}_{H1}(\mathcal{I})$ serve as a secondary ranking criterion. Thus, every instance $\mathcal{I}$ from the IsA repository is associated with a ranked list of class labels computed according to this ranking formula. Conversely, each class label $\mathcal{C}$ from the IsA repository is associated with a ranked list of class instances computed with the earlier scoring formula (1) used to generate lists $\mathrm{L}_{H1}(\mathcal{I})$.

Note that the ranking formula can also consider only a subset of the available input lists. For instance, $Score_{H1+H2}$ would use only $\mathrm{L}_{H1}(\mathcal{I})$ and $\mathrm{L}_{H2}(\mathcal{I})$ as input lists; $Score_{H1+H3}$ would use only $\mathrm{L}_{H1}(\mathcal{I})$ and $\mathrm{L}_{H3}(\mathcal{I})$ as input lists; etc.

3 Experimental Setting

Textual Data Sources: The acquisition of the IsA repository relies on unstructured text available within Web documents and search queries. The queries are fully-anonymized queries in English submitted to Google by Web users in 2009, and are available in two collections. The first collection is a random sample of 50 million unique queries that are independent from one another. The second collection is a random sample of 5 million query sessions. Each session has an initial query and a series of subsequent queries. A subsequent query is a query that has been submitted by the same Web user within no longer than a few minutes after the initial query. Each subsequent query is accompanied by its frequency of occurrence in the session, with the corresponding initial query. The document collection consists of a sample of 100 million documents in English.

Experimental Runs: The experimental runs correspond to different methods for extracting and ranking pairs of an instance and a class:

- from the repository extracted here, with class labels of an instance ranked based on the frequency and the number of extraction patterns ($Score_{H1}$ from Equation (1) in Section 2), in run $\underline{\mathbf{R}_d}$;
- from the repository extracted here, with class labels of an instance ranked via the rank-based merging of: $Score_{H1+H2}$ from Section 2, in run $\underline{\mathbf{R}_p}$, which corresponds to re-ranking using co-occurrence of an instance and its class label in the same query; $Score_{H1+H3}$ from Section 2, in run $\underline{\mathbf{R}_s}$, which corresponds to re-ranking using co-occurrence of an instance and its class label, as separate queries within the same query session; and $Score_{H1+H2+H3}$ from Section 2, in run $\underline{\mathbf{R}_u}$, which corresponds to re-ranking using both types of co-occurrences in queries.

Evaluation Procedure: The manual evaluation of open-domain information extraction output is time consuming (Banko et al., 2007). A more practical alternative is an automatic evaluation procedure for ranked lists of class labels, based on existing resources and systems.

Assume that there is a gold standard, containing gold class labels that are each associated with a gold set of their instances. The creation of such gold standards is discussed later. Based on the gold standard, the ranked lists of class labels available within an IsA repository can be automatically evaluated as follows. First, for each gold label, the ranked lists of class labels of individual gold instances are retrieved from the IsA repository. Second, the individual retrieved lists are merged into a ranked list of class labels, associated with the gold label. The merged list can be computed, e.g., using an extension of the $Score_{H1+H2+H3}$ formula (Equation (4)) described earlier in Section 2. Third, the merged list is compared against the gold label, to estimate the accuracy of the merged list. Intuitively, a ranked list of class labels is a better approximation of a gold label, if class labels situated at better ranks in the list are closer in meaning to the gold label.

Evaluation Metric: Given a gold label and a list of class labels, if any, derived from the IsA repository, the rank of the highest class label that matches the gold label determines the score assigned to the gold label, in the form of the reciprocal rank of the match. Thus, if the gold label matches a class label at rank 1, 2 or 3 in the computed list, the gold label receives a score of 1, 0.5 or 0.33 respectively. The score is 0 if the gold label does not match any of the top 20 class labels. The overall score over the entire set of gold labels is the mean reciprocal rank (MRR) score over all gold labels from the set. Two types of MRR scores are automatically computed:

• **MRR**$_f$ considers a gold label and a class label to match, if they are identical;

• **MRR**$_p$ considers a gold label and a class label to match, if one or more of their tokens that are not stop words are identical.

During matching, all string comparisons are case-insensitive, and all tokens are first converted to their singular form (e.g., *european countries* to *european country*) using WordNet (Fellbaum, 1998). Thus, *insurance carriers* and *insurance companies* are considered to not match in MRR$_f$ scores, but match in MRR$_p$ scores. On the other hand, MRR$_p$ scores may give credit to less relevant class labels, such as *insurance policies* for the gold label *insurance carriers*. Therefore, MRR$_p$ is an optimistic, and MRR$_f$ is a pessimistic estimate of the actual usefulness of the computed ranked lists of class labels as approximations of the gold labels.

Query Set: Sample of Queries
Q$_e$ (807 queries): 2009 movies, amino acids, asian countries, bank, board games, buildings, capitals, chemical functional groups, clothes, computer language, dairy farms near modesto ca, disease, egyptian pharaohs, eu countries, fetishes, french presidents, german islands, hawaiian islands, illegal drugs, irc clients, lakes, macintosh models, mobile operator india, nba players, nobel prize winners, orchids, photo editors, programming languages, renaissance artists, roller costers, science fiction tv series, slr cameras, soul singers, states of india, taliban members, thomas edison inventions, u.s. presidents, us president, water slides
Q$_m$ (40 queries): actors, actresses, airlines, american presidents, antibiotics, birds, cars, celebrities, colors, computer languages, digital camera, dog breeds, dogs, drugs, elements, endangered animals, european countries, flowers, fruits, greek gods, horror movies, idioms, ipods, movies, names, netbooks, operating systems, park slope restaurants, planets, presidents, ps3 games, religions, renaissance artists, rock bands, romantic movies, states, universities, university, us cities, vitamins

Table 1: Size and composition of evaluation sets of queries associated with non-filtered (Q$_e$) or manually-filtered (Q$_m$) instances

4 Evaluation

IsA Repository: The IsA repository, extracted from the document collection, covers a total of 4.04 million instances associated with 7.65 million class labels. The number of class labels available per instance and vice-versa follows a long-tail distribution, indicating that 2.12 million of the instances each have two or more class labels (with an average of 19.72 class labels per instance).

Evaluation Sets of Queries: Table 1 shows samples of two query sets, introduced in (Paşca, 2010) and used in the evaluation. The first set, denoted **Q**$_e$,

Query Set	Min	Max	Avg	Median
Number of Gold Instances:				
Q_e	10	100	70.4	81
Q_m	8	33	16.9	17
Number of Query Tokens:				
Q_e	1	8	2.0	2
Q_m	1	3	1.4	1

Table 2: Number of gold instances (upper part) and number of query tokens (lower part) available per query, over the evaluation sets of queries associated with non-filtered gold instances (Q_e) or manually-filtered gold instances (Q_m)

is obtained from a random sample of anonymized, class-seeking queries submitted by Web users to Google Squared. The set contains 807 queries, each associated with a ranked list of between 10 and 100 gold instances automatically extracted by Google Squared.

Since the gold instances available as input for each query as part of Q_e are automatically extracted, they may or may not be true instances of the respective queries. As described in (Paşca, 2010), the second evaluation set $\mathbf{Q}_m$ is a subset of 40 queries from Q_e, such that the gold instances available for each query in Q_m are found to be correct after manual inspection. The 40 queries from Q_m are associated with between 8 and 33 human-validated instances.

As shown in the upper part of Table 2, the queries from Q_e are up to 8 tokens in length, with an average of 2 tokens per query. Queries from Q_m are comparatively shorter, both in maximum (3 tokens) and average (1.4 tokens) length. The lower part of Table 2 shows the number of gold instances available as input, which average around 70 and 17 per query, for queries from Q_e and Q_m respectively. To provide another view on the distribution of the queries from evaluation sets, Table 3 lists tokens that are not stop words, which occur in most queries from Q_e. Comparatively, few query tokens occur in more than one query in Q_m.

Evaluation Procedure: Following the general evaluation procedure, each query from the sets $\mathbf{Q}_e$ and $\mathbf{Q}_m$ acts as a gold class label associated with the corresponding set of instances. Given a query and its instances $\mathcal{I}$ from the evaluation sets Q_e or Q_m, a merged, ranked lists of class labels is computed out of the ranked lists of class labels available in the underlying IsA repository for each instance $\mathcal{I}$. The evaluation compares the merged lists of class labels, with the corresponding queries from Q_e or Q_m.

Query Token	Cnt.	Examples of Queries Containing the Token
countries	22	african countries, eu countries, poor countries
cities	21	australian cities, cities in california, greek cities
presidents	18	american presidents, korean presidents, presidents of the south korea
restaurants	15	atlanta restaurants, nova scotia restaurants, restaurants 10024
companies	14	agriculture companies, gas utility companies, retail companies
states	14	american states, states of india, united states national parks
prime	11	australian prime ministers, indian prime ministers, prime ministers
cameras	10	cameras, digital cameras olympus, nikon cameras
movies	10	2009 movies, movies, romantic movies
american	9	american authors, american president, american revolution battles
ministers	9	australian prime ministers, indian prime ministers, prime ministers

Table 3: Query tokens occurring most frequently in queries from the Q_e evaluation set, along with the number (Cnt) and examples of queries containing the tokens

Accuracy of Lists of Class Labels: Table 4 summarizes results from comparative experiments, quantifying a) horizontally, the impact of alternative parameter settings on the computed lists of class labels; and b) vertically, the comparative accuracy of the experimental runs over the query sets. The experimental parameters are the number of input instances from the evaluation sets that are used for retrieving class labels, I-per-Q, set to 3, 5, 10; and the number of class labels retrieved per input instance, C-per-I, set to 5, 10, 20.

Four conclusions can be derived from the results. First, the scores over Q_m are higher than those over Q_e, confirming the intuition that the higher-quality

	Accuracy								
I-per-Q	3			5			10		
C-per-I	5	10	20	5	10	20	5	10	20
MRR$_f$ computed over Q$_e$:									
R$_d$	0.186	0.195	0.198	0.198	0.207	0.210	0.204	0.214	0.218
R$_p$	0.202	0.211	0.216	0.232	0.238	0.244	0.245	0.255	0.257
R$_s$	**0.258**	**0.260**	**0.261**	**0.278**	**0.277**	**0.276**	**0.279**	**0.280**	**0.282**
R$_u$	0.234	0.241	0.244	0.260	0.263	0.270	0.274	0.275	0.278
MRR$_p$ computed over Q$_e$:									
R$_d$	0.489	0.495	0.495	0.517	0.528	0.529	0.541	0.553	0.557
R$_p$	0.520	0.531	0.533	0.564	0.573	0.578	0.590	0.601	0.602
R$_s$	**0.576**	**0.584**	**0.583**	**0.612**	**0.616**	0.614	**0.641**	0.636	0.628
R$_u$	0.561	0.570	0.571	0.606	0.614	**0.617**	0.640	**0.641**	**0.636**
MRR$_f$ computed over Q$_m$:									
R$_d$	0.406	0.436	0.442	0.431	0.447	0.466	0.467	0.470	0.501
R$_p$	0.423	0.426	0.429	0.436	0.483	0.508	0.500	0.526	0.530
R$_s$	**0.590**	**0.601**	**0.594**	**0.578**	**0.604**	**0.595**	**0.624**	**0.612**	**0.624**
R$_u$	0.481	0.502	0.508	0.531	0.539	0.545	0.572	0.588	0.575
MRR$_p$ computed over Q$_m$:									
R$_d$	0.667	0.662	0.660	0.675	0.677	0.699	0.702	0.695	0.716
R$_p$	0.711	0.703	0.680	0.734	0.731	0.748	0.733	0.797	0.782
R$_s$	**0.841**	**0.822**	**0.820**	**0.835**	**0.828**	**0.823**	**0.850**	**0.856**	**0.844**
R$_u$	0.800	0.810	0.781	0.795	0.794	0.779	0.806	0.827	0.816

Table 4: Accuracy of instance set labeling, as full-match (MRR$_f$) or partial-match (MRR$_p$) scores over the evaluation sets of queries associated with non-filtered instances (Q$_e$) or manually-filtered instances (Q$_m$), for various experimental runs (I-per-Q=number of gold instances available in the input evaluation sets that are used for retrieving class labels; C-per-I=number of class labels retrieved from IsA repository per input instance)

input set of instances available in Q$_m$ relative to Q$_e$ should lead to higher-quality class labels for the corresponding queries. Second, when I-per-Q is fixed, increasing C-per-I leads to small, if any, score improvements. Third, when C-per-I is fixed, even small values of I-per-Q, such as 3 (that is, very small sets of instances provided as input) produce scores that are competitive with those obtained with a higher value like 10. This suggests that useful class labels can be generated even in extreme scenarios, where the number of instances available as input is as small as 3 or 5. Fourth and most importantly, for most combinations of parameter settings and on both query sets, the runs that take advantage of query logs (R$_p$, R$_s$, R$_u$) produce the highest scores. In particular, when I-per-Q is set to 10 and C-per-I to 20, run R$_u$ identifies the original query as an exact match among the top three to four class labels returned (score 0.278); and as a partial match among the top one to two class labels returned (score 0.636), as an average over the Q$_e$ set. The corresponding MRR$_f$ score of 0.278 over the Q$_e$ set obtained with run R$_u$ is 27% higher than with run R$_d$.

In all experiments, the higher scores of R$_p$, R$_s$ and R$_u$ can be attributed to higher-quality lists of class labels, relative to R$_d$. Among combinations of parameter settings described in Table 4, values around 10 for I-per-Q and 20 for C-per-I give the highest scores over both Q$_e$ and Q$_m$.

Among the query-based runs R$_p$, R$_s$ and R$_u$, the highest scores in Table 4 are obtained mostly for run R$_s$. Thus, between the presence of a class label and an instance either in the same query, or as separate queries within the same query session, it is the latter that provides a more useful signal during the reranking of class labels of each instance.

Table 5 illustrates the top class labels from the ranked lists generated in run R$_s$ for various queries from both Q$_e$ and Q$_m$. The table suggests that the computed class labels are relatively resistant to noise and variation within the input set of gold instances. For example, the top elements of the lists of class la-

Query	Query	Gold Instances		Top Labels Generated Using Top 10 Gold Instances
	Set	Cnt.	Sample from Top Gold Instances	
actors	Q_e	100	abe vigoda, ben kingsley, bill hickman	actors, stars, favorite actors, celebrities, movie stars
	Q_m	28	al pacino, christopher walken, danny devito	actors, celebrities, favorite actors, movie stars, stars
computer languages	Q_e	59	acm transactions on mathematical software, applescript, c	languages, programming languages, programs, standard programming languages, computer programming languages
	Q_m	17	applescript, eiffel, haskell	languages, programming languages, computer languages, modern programming languages, high-level languages
european countries	Q_e	60	abkhazia, armenia, bosnia & herzegovina	countries, european countries, eu countries, foreign countries, western countries
	Q_m	19	belgium, finland, greece	countries, european countries, eu countries, foreign countries, western countries
endangered animals	Q_e	98	arkive, arabian oryx, bagheera	species, animals, endangered species, animal species, endangered animals
	Q_m	21	arabian oryx, blue whale, giant hispaniolan galliwasp	animals, endangered species, species, endangered animals, rare animals
park slope restaurants	Q_e	100	12th street bar & grill, aji bar lounge, anthony's	businesses, departments
	Q_m	18	200 fifth restaurant bar, applewood restaurant, beet thai restaurant	(none)
renaissance artists	Q_e	95	michele da verona, andrea sansovino, andrea del sarto	artists, famous artists, great artists, renaissance artists, italian artists
	Q_m	11	botticelli, filippo lippi, giorgione	artists, famous artists, renaissance artists, great artists, italian artists
rock bands	Q_e	65	blood doll, nightmare, rockaway beach	songs, hits, films, novels, famous songs
	Q_m	15	arcade fire, faith no more, indigo girls	bands, rock bands, favorite bands, great bands, groups

Table 5: Examples of gold instances available in the input, and actual ranked lists of class labels produced by run R_s for various queries from the evaluation sets of queries associated with non-filtered gold instances (Q_e) or manually-filtered gold instances (Q_m)

bels generated for *computer languages* are relevant and also quite similar for Q_e vs. Q_m, although the list of gold instances in Q_e may contain incorrect items (e.g., *acm transactions on mathematical software*). Similarly, the class labels computed for *european countries* are almost the same for Q_e vs. Q_m, although the overlap of the respective lists of 10 gold instances used as input is not large. The table shows at least one query (*park slope restaurants*) for which the output is less than optimal, either because the class labels (e.g., *businesses*) are quite distant semantically from the query (for Q_e), or because no output is produced at all, due to no class labels being found in the IsA repository for any of the 10 input gold instances (for Q_m). For many queries, however, the computed class labels arguably capture the meaning of the original query, although not necessarily in the exact same lexical form, and sometimes only partially. For example, for the query *endangered animals*, only the fourth class label from Q_m identifies the query exactly. However, class labels preceding *endangered animals* already capture the notion of *animals* or *species* (first and third labels), or that they are *endangered* (second label).

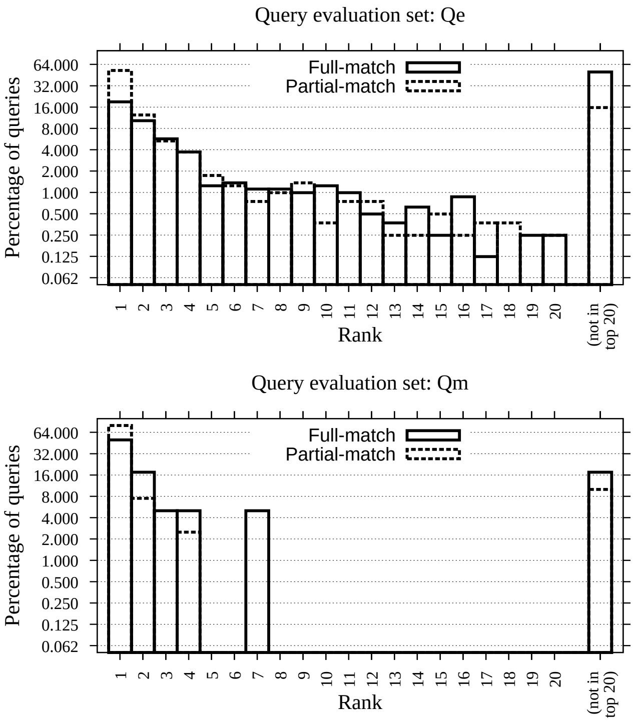

Figure 1: Percentage of queries from the evaluation sets, for which the earliest class labels from the computed ranked lists of class labels, which match the queries, occur at various ranks in the ranked lists returned by run R_s

Figure 1 provides a detailed view on the distribution of queries from the Q_e and Q_m evaluation sets, for which the class label that matches the query occurs at a particular rank in the computed list of class labels. In the first graph of Figure 1, for Q_e, the query matches the automatically-generated class label at ranks 1, 2, 3, 4 and 5 for 18.9%, 10.3%, 5.7%, 3.7% and 1.2% of the queries respectively, with full string matching, i.e., corresponding to MRR_f; and for 52.6%, 12.4%, 5.3%, 3.7% and 1.7% respectively, with partial string matching, corresponding to MRR_p. The second graph confirms that higher MRR scores are obtained for Q_m than for Q_e. In particular, the query matches the class label at rank 1 and 2 for 50.0% and 17.5% (or a combined 67.5%) of the queries from Q_m, with full string matching; and for 52.6% and 12.4% (or a combined 67%), with partial string matching.

Discussion: The quality of lists of items extracted from documents can benefit from query-driven ranking, particularly for the task of ranking class labels of instances within IsA repositories. The use of queries for ranking is generally applicable: it can be seen as a post-processing stage that enhances the ranking of the class labels extracted for various instances by any method into any IsA repository.

Open-domain class labels extracted from text and re-ranked as described in this paper are useful in a variety of applications. Search tools such as Google Squared return a set of instances, in response to class-seeking queries (e.g., *insurance companies*). The labeling of the returned set of instances, using the re-ranked class labels available per instances, allows for the generation of query refinements (e.g., *insurers*). In search over semi-structured data (Cafarella et al., 2008), the labeling of column cells is useful to infer the semantics of a table column, when the subject row of the table in which the column appears is either absent or difficult to detect.

5 Related Work

The role of anonymized query logs in Web-based information extraction has been explored in tasks such as class attribute extraction (Paşca and Van Durme, 2007), instance set expansion (Pennacchiotti and Pantel, 2009) and extraction of sets of similar entities (Jain and Pennacchiotti, 2010). Our work compares the usefulness of queries and query sessions for ranking class labels in extracted IsA repositories. It shows that query sessions produce better-ranked class labels than isolated queries do. A task complementary to class label ranking is entity ranking (Billerbeck et al., 2010), also referred to as ranking for typed search (Demartini et al., 2009).

The choice of search queries and query substitutions is often influenced by, and indicative of, various semantic relations holding among full queries or query terms (Jones et al., 2006). Semantic relations may be loosely defined, e.g., by exploring the acquisition of untyped, similarity-based relations from query logs (Baeza-Yates and Tiberi, 2007). In comparison, queries are used here to re-rank class labels capturing a well-defined type of open-domain relations, namely IsA relations.

6 Conclusion

In an attempt to bridge the gap between information stated in documents and information requested in search queries, this study shows that inherently-noisy queries are useful in re-ranking class labels extracted from Web documents for various instances, with query sessions leading to higher quality than isolated queries. Current work investigates the impact of ambiguous input instances (Vyas and Pantel, 2009) on the quality of the generated class labels.

References

R. Baeza-Yates and A. Tiberi. 2007. Extracting semantic relations from query logs. In *Proceedings of the 13th ACM Conference on Knowledge Discovery and Data Mining (KDD-07)*, pages 76–85, San Jose, California.

M. Banko, Michael J Cafarella, S. Soderland, M. Broadhead, and O. Etzioni. 2007. Open information extraction from the Web. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI-07)*, pages 2670–2676, Hyderabad, India.

B. Billerbeck, G. Demartini, C. Firan, T. Iofciu, and R. Krestel. 2010. Ranking entities using Web search query logs. In *Proceedings of the 14th European Conference on Research and Advanced Technology for Digital Libraries (ECDL-10)*, pages 273–281, Glasgow, Scotland.

M. Cafarella, A. Halevy, D. Wang, E. Wu, and Y. Zhang. 2008. WebTables: Exploring the power of tables on the Web. In *Proceedings of the 34th Conference on Very Large Data Bases (VLDB-08)*, pages 538–549, Auckland, New Zealand.

G. Demartini, T. Iofciu, and A. de Vries. 2009. Overview of the INEX 2009 Entity Ranking track. In *INitiative for the Evaluation of XML Retrieval Workshop*, pages 254–264, Brisbane, Australia.

O. Etzioni, M. Cafarella, D. Downey, A. Popescu, T. Shaked, S. Soderland, D. Weld, and A. Yates. 2005. Unsupervised named-entity extraction from the Web: an experimental study. *Artificial Intelligence*, 165(1):91–134.

C. Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database and Some of its Applications*. MIT Press.

M. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th International Conference on Computational Linguistics (COLING-92)*, pages 539–545, Nantes, France.

A. Jain and M. Pennacchiotti. 2010. Open entity extraction from Web search query logs. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING-10)*, pages 510–518, Beijing, China.

R. Jones, B. Rey, O. Madani, and W. Greiner. 2006. Generating query substitutions. In *Proceedings of the 15h World Wide Web Conference (WWW-06)*, pages 387–396, Edinburgh, Scotland.

Z. Kozareva, E. Riloff, and E. Hovy. 2008. Semantic class learning from the Web with hyponym pattern linkage graphs. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics (ACL-08)*, pages 1048–1056, Columbus, Ohio.

M. Paşca and B. Van Durme. 2007. 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.

M. Paşca. 2010. The role of queries in ranking labeled instances extracted from text. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING-10)*, pages 955–962, Beijing, China.

M. Pennacchiotti and P. Pantel. 2009. Entity extraction via ensemble semantics. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP-09)*, pages 238–247, Singapore.

R. Snow, D. Jurafsky, and A. Ng. 2006. Semantic taxonomy induction from heterogenous evidence. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (COLING-ACL-06)*, pages 801–808, Sydney, Australia.

P. Talukdar, J. Reisinger, M. Paşca, D. Ravichandran, R. Bhagat, and F. Pereira. 2008. Weakly-supervised acquisition of labeled class instances using graph random walks. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing (EMNLP-08)*, pages 582–590, Honolulu, Hawaii.

B. Van Durme and M. Paşca. 2008. Finding cars, goddesses and enzymes: Parametrizable acquisition of labeled instances for open-domain information extraction. In *Proceedings of the 23rd National Conference on Artificial Intelligence (AAAI-08)*, pages 1243–1248, Chicago, Illinois.

V. Vyas and P. Pantel. 2009. Semi-automatic entity set refinement. In *Proceedings of the 2009 Conference of the North American Association for Computational Linguistics (NAACL-HLT-09)*, pages 290–298, Boulder, Colorado.