Domain-Independent Entity Extraction from Web Search Query Logs

Alpa Jain
Yahoo! Labs
Sunnyvale, CA, 94089
alpa@yahoo-inc.com

Marco Pennacchiotti
Yahoo! Labs
Sunnyvale, CA, 94089
pennac@yahoo-inc.com

ABSTRACT

Query logs of a Web search engine have been increasingly used as a vital source for data mining. This paper presents a study on large-scale *domain-independent entity extraction from search query logs*. We present a completely unsupervised method to extract entities by applying pattern-based heuristics and statistical measures. We compare against existing techniques that use Web documents as well as search logs, and show that we improve over the state of the art. We also provide an in-depth qualitative analysis outlining differences and commonalities between these methods.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning—*knowledge acquisition*

General Terms

Algorithms

Keywords

entity extraction, query logs, data mining

1. QUERY LOGS ENTITY EXTRACTION

Entity extraction is an important part of many Web-based applications [3], defined as the task of extracting entities of pre-defined classes (e.g., 'Brad Pitt' for the class Actors). Typically, extraction tasks are run over "well-formed" documents such as news articles or web pages [1]. Recently, Pasca [6] proposed to extract entities from query-logs instead of a classical Web corpus, by using a semi-supervised approach that inputs a pre-defined list of classes represented by a small set of hand-made seeds. Extracting entities from query logs instead of web corpora has several advantages, in view of query-log-based application [4]. Along this direction, we present a completely unsupervised (i.e. we do not need seeds) and domain-independent (i.e., we do not need pre-defined classes) extraction method over query logs. Our study is inspired by the open domain information extraction (OIE) framework, that has been recently implemented to extract entities from Web-scale corpora, using simple unsupervised techniques [2]. In the same spirit, our method is also unsupervised, by applying heuristics that specifically address OIE over query-logs. To our knowledge, ours is the first attempt to devise an algorithm specifically designed to achieve the following two goals at the same time: (a) extract entities from a query log; (b) extract entities in an open-domain fashion (OIE).

WWW 2011, March 28–April 1, 2011, Hyderabad, India.
ACM 978-1-4503-0637-9/11/03.

Starting with raw search query logs, our approach performs three steps: (1) identify candidate entities from search logs, (2) select reliable entities from the candidates using confidence scores, (3) execute a subsumption filter to eliminate noise.

Generating candidate entities: User queries are short and lack syntactic structure, thus impairing traditional extraction approaches based on contextual evidence and syntactic interpretation. We therefore rely on a simple observation: oftentimes users construct their search query by copy-pasting phrases from existing texts in a web pages, thus carrying over surface-level properties such as capitalization. We generate candidate entities as contiguous capitalized words from a user query: given a query $Q = q_1\ q_2 \cdots\ q_n$, we define a candidate entity $E = e_1\ e_2\ \cdots\ e_m$ as the maximal sequence of words (i.e., alpha-numeric characters) in the query such that each word e_i in the entity begins with an uppercase character. This method can be far from perfect: for instance, a small fraction of user search queries are entered using only uppercase characters. We then discard spurious entities by employing text-based evidence described next.

Deriving confidence scores: We assign two confidence scores to a candidate entity E. The *representation score* captures the intuition that the case-sensitive representation observed for E in Q, should be a likely representation for E, as observed on a Web corpus. For example given the query 'Galapagos Island vacations', we assign a high score to the candidate 'Galapagos Island' because we can observe it oftentimes on the Web. On the contrary, given the query 'DOor HANGING tips', we assign a low score to the candidate 'DOor HANGING', as it is seldom observed in that representation. More formally, the score is computed as:

$$r_w(E) = \frac{|\gamma(E)|}{\sum_{i \in O(E)} |\gamma(i)|} \quad (1)$$

where $|x|$ is the number of occurrences of a string x in the corpus, $O(E)$ is the set of all occurrences of string E, $\gamma(i)$ is a case-sensitive representation of the string i.

The *standalone score* is based on the observation that a candidate E should often occur in a standalone form among the query logs, in order to get the status of proper entity:

$$s_q(E) = \frac{|Q == E|}{|\textit{queries that contain } E|} \quad (2)$$

We retain candidate entities for which $r_w(E) \geq \tau_r$ *and* $s_q(E) \geq \tau_s$ (we experimentally set $\tau_r = 0.1$ and $\tau_s = 0.2$).

Applying the Subsumption Filter: As a final step, we deal with boundary detection. Often, the score-based filters miss erroneous candidates that have substantial overlap with good candidates. For example 'Barack Obama' and 'Barack Obama Biography' are both often used in capitalized form and in standalone queries, but the

latter is not an entity. We then eliminate candidates that completely subsume another one (e.g. 'Barack Obama Biography').

2. EXPERIMENTAL EVALUATION

Compared methods: We experiment over a sample of 100M queries from the Yahoo! search engine in the first three months of 2009 (JN, FB, MR). We compare the following systems over MR (JN and FB are for training when needed):

QL-Base: A baseline query log system, applying only the candidate generation step described in Section 1. QL-Conf: A system, applying candidate generation and confidence score filtering. QL-Full: Our full query log extraction method. QL-Pasca: The state-of-the-art query logs Pasca's system [6], bootstrapped as in [6] with 5 seeds per category (pattern extraction performed on JN, FB and MR, instance extraction and ranking on MR); we computed coverage at rank 20,000 and at full rank. Web: The state-of-the-art Web-based open-domain entity extraction system described in [5]; as corpus we use 500 million web pages crawled by the Yahoo! search engine; this system allows us to compare Web-based and query-log-based extraction.

Evaluation method: We perform two experiments. In the **accuracy experiment** we draw a uniform random sample of 400 entities for each method, and ask two expert annotators if an entity is correct or not (Kappa agreeement over a shared set of 50, is $kappa = 0.75$). In the **coverage experiment**, we estimate a method's coverage over a gold set extracted from Wikipedia, for five entity classes [1]: actors (ACT), athletes (ATH), cities (CIT), diseases (DIS), and movies (MOV).

Evaluation metrics: We use **accuracy:** the fraction of correct entities returned by a method, computed as $\frac{|S_c|}{|S|}$, where S_c is the set of correct entities among the whole set S, and; **coverage:** the fraction of entities in the Wikipedia gold set G that are extracted by a method, as $\frac{|G \cap S|}{|G|}$.

2.1 Experimental Results

Accuracy Experiment: Table 1 reports the accuracy as well as the total number of entities extracted for all open-domain extraction techniques. Note that results for QL-Pasca cannot be reported, as it is not an open-domain system, hence numbers would not be comparable. As for the query log based systems, results show that applying filters to the raw entities, largely improves the accuracy at the cost of reducing the total number of entities. As we will see in the coverage experiment, the decrease in the total number of entities reduces the coverage by only a small margin, indicating that the filters correctly discard erroneous candidates. The application of the confidence scores is highly effective: QL-Conf improves +23% points over the baseline QL-Base. Candidates such as 'Buy' and 'News' are discarded for their low representation score, while 'HP Printer Software Free Download' and 'About Israel' are filtered using the standalone score. The subsumption filter further improves the accuracy (QL-Full improves +6.5% over QL-Conf) showing that the confidence scores alone cannot do the whole job. For example, 'Chicago White Sox Tickets' passes the confidence filters, but is discarded, since it subsumes 'Chicago White Sox'.

The Web system performs cosiderably worse than the query log systems. Typical errors include: candidates where the name of a person is augmented with his title (e.g. 'Professor Iam Wilut'); tokenization errors (e.g.'surely–') ; noun phrase commonly capitalized on the Web, which are not entities (e.g. 'Positive Test Results'). Overall, results suggest query logs as a promising source for extracting entities with high accuracy, in an open-domain fashion. Indeed, some effective techniques peculiar to query logs can not be used on the web, such as the *standalone score*.

[1] An exhaustive generic set of entities is not available in the literature and is impractical to build.

method	# instances	accuracy
QL-Base	15,077,998	0.359 $\pm$0.047
QL-Conf	2,391,201	0.640 $\pm$0.047
QL-Full	2,067,385	0.705 $\pm$0.044
Web	2,388,412	0.499 $\pm$0.049

Table 1: Number of instances extracted by each system and related accuracy.

method	ACT	ATH	CIT	DIS	MOV
QL-Base	**72.75**	52.48	**70.17**	**40.45**	**57.13**
QL-Conf	64.16	39.33	14.63	23.78	28.61
QL-Full	63.02	38.92	14.46	22.60	26.43
QL-Pasca	10.51	2.12	18.73	4.00	20.01
Web	65.37	**55.65**	68.09	35.20	37.46

Table 2: Coverage of systems over the set of Wikipedia classes.

Coverage Experiment: Table 2 reports coverage for the target classes. As expected, of the various extraction methods involving query logs, QL-Base, shows the highest coverage, but at the expense of hurting the accuracy as discussed above. In comparison with the Web, our methods that use filters, QL-Conf and QL-Full, show lower coverage due to the aggressive round of the applied filters, especially the standalone ratio: in case of instances from CIT, this may prove to be too restrictive as users tend to query for information about a city, (e.g., 'Rome attractions' or 'Shanghai map') as opposed to querying for the city (e.g., 'Rome' or 'Shanghai'). Extending our confidence score functions to incorporate other evidence available in query logs to relax this effect is part of future work. As a final remark, our methods outperform QL-Pasca for all but the CIT class for the reasons discussed above. It is noteworthy that none of the systems achieve full coverage over Wikipedia, which does not mean that Wikipedia is an exhaustive source of entities. Indeed, there are many correct instances extracted by the experimented systems that are not present in Wikipedia. To support this, we sampled and evaluated 100 entities extracted by QL-Full which are not in Wikipedia. We observed that 61% of these instances are correct, i.e. Wikipedia misses a large number of entities.

In conclusion, we presented a completely unsupervised method to extract entities from query logs, that improves over the state of the art. This paper takes an initial step towards building OIE over search query logs, opening ample space for promising future work on information extraction, and for building domain-independent applications based on query logs, such as tools for intent modeling and query suggestion.

3. REFERENCES

[1] S. Chaudhuri, V. Ganti, and D. Xin. Exploiting web search to generate synonyms for entities. In *WWW-09*, 2009.

[2] D. Downey, M. Broadhead, and O. Etzioni. Locating complex named entities in web text. In *IJCAI*, 2007.

[3] J. Hu, G. Wang, F. Lochovsky, J. tao Sun, and Z. Chen. Understanding user's query intent with Wikipedia. In *WWW-09*, 2009.

[4] A. Jain and M. Pennacchiotti. Open entity extraction fromweb search query logs. In *COLING-2010*, 2010.

[5] P. Pantel, E. Crestan, A. Borkovsky, A.-M. Popescu, and V. Vyas. Web-scale distributional similarity and entity set expansion. In *EMNLP-09*, 2009.

[6] M. Pasca. Weakly-supervised discovery of named entities using web search queries. In *CIKM-2007*, 2007.