Open 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

In this paper we propose a completely unsupervised method for open-domain entity extraction and clustering over query logs. The underlying hypothesis is that classes defined by mining search user activity may significantly differ from those typically considered over web documents, in that they better model the *user space*, i.e. users' perception and interests. We show that our method outperforms state of the art (semi-)supervised systems based either on web documents or on query logs (16% gain on the clustering task). We also report evidence that our method successfully supports a real world application, namely keyword generation for sponsored search.

1 Introduction

Search engines are increasingly moving beyond the traditional keyword-in document-out paradigm, and are improving user experience by focusing on user-oriented tasks such as query suggestions and search personalization. A fundamental building block of these applications is recognizing structured information, such as, *entities* (e.g., mentions of people, organizations, or locations) or relations among entities (Cao et al., 2008; Hu et al., 2009). For this, search engines typically rely on large collections of entities and relations built using information extraction (IE) techniques (Chaudhuri et al., 2009).

Commonly used IE techniques follow two main assumptions: (1) IE focuses on extracting information from syntactically and semantically "well-formed" pieces of texts, such as, news corpora and web documents (Pennacchiotti and Pantel, 2009); (2) extraction processes are bootstrapped with some pre-existing knowledge of the target domain (e.g entities are typically extracted for pre-defined categories, such as Actors, Manufacturers, Persons, Locations (Grishman and Sundheim, 1996)). Prior work (Banko et al., 2007), has looked into relaxing the second assumption and proposed open information extraction (OIE), a domain-independent and scalable extraction paradigm, which however focuses mostly on web corpora.

In this paper, we argue that for user-oriented applications discussed earlier, IE techniques should go beyond the traditional approach of using "well-formed" text documents. With this in mind, we explore the utility of search query logs, a rich source of user behaviors and perception, and build techniques for open entity extraction and clustering over query logs. We hypothesize that web documents and query logs model two different spaces: web documents model the *web space*, i.e. general knowledge about entities and concepts in an objective and generic way; search query logs model the *user space*, i.e. the users' view and perception of the world in a more specific fashion, where available information directly expresses users' needs and intents. For example, in a web space, 'britney spears' will tend to be similar and be clustered with other singers, such as 'celine dion' and 'bruce springsteen'. On the contrary, in the users' space, she is highly similar and clustered with other gossiped celebrities like 'paris hilton' and 'serena williams': the users' space better models the users' perception of that person; such a space is then highly valuable for all those applications where users' perceptions matters (Section 4).

To computationally model our hypothesis for OIE over search query logs, we present a two phase approach to OIE for search query logs. The first phase (*entity extraction*) extracts entities from the search query logs using an unsupervised approach, by applying pattern-based heuristics and statistical measures. The second phase (*entity clustering*) induces classes over these entities by applying clustering techniques. In summary, our main contribu-

tions are: (1) We propose and instantiate a novel model for open information extraction over web search query logs; and we apply it to the task of entity extraction and clustering. (2) We show how we characterize each extracted entity to capture the 'user space', and induce classes over the entities. (3) We present an extensive evaluation over real-life datasets showing that query logs is a rich source for domain-independent user-oriented extraction tasks (Section 3). We also show the practicality of our approach by incorporating it into a real-world application, namely keyword suggestions for sponsored search (Section 4).

2 Open Entity Extraction on Query Log

In this section, we present our method for open entity extracion from query logs. We first describe our heuristic method for extracting entities (Section 2.1), and then three different feature 'user spaces' to cluster the entities (Section 2.2).

2.1 Entity Extraction

In our setting, entities correspond to Named Entities. i.e. they are defined using the standard named entity types described in (Sekine et al., 2002)[1]. In this paper, we use a set of entities extracted from query log, obtained by applying a simple algorithm (any other query log entity extraction method would apply here, e.g. (Pasca, 2007b)). The algorithm is based on the observation that oftentimes users construct their search query by copy-pasting phrases from existing texts. Due to this phenomenon, user queries often carry over surface-level properties such as capitalization and tokenization information. Our approach realizes this observation by identifying contiguous capitalized words from a user query. (In our experiments, we observed that 42% of the queries had at least one upper-case character.) Specifically, given a query $Q = q_1\ q_2\ q_3 \cdots\ q_n$, 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. The set of candidate entities is then cleaned by applying a set of heuristics, thus producing the final set of entities. In particular, for each extracted entity, we assign two confidence scores: a Web-based *representation score* and a query-log-based *standalone score*. The representation score checks if the case-sensitive representation observed for E in Q, is the most likely representation for E, as observed on a Web corpus (e.g., 'DOor HANGing TIps' is assigned a low representation score). The standalone score is based on the observation that a candidate E should often occur in a standalone form among the search query logs, in order to get the status of a proper named entity as defined in (Sekine et al., 2002; Grishman and Sundheim, 1996). In practice, among the query logs we must find queries of the form $Q == E$, capturing the fact that users are looking to learn more about the given entity[2].

2.2 Entity Clustering

The clustering phase takes as input any of the feature spaces presented in the rest of this section, and groups the entities according to the similarity of their vectors in the space. The desiderata for a clustering algorithm for the task of open-domain information extraction are the following: (1) The algorithm must be highly scalable, efficient, and able to handle high dimensionality, since the number of queries and the size of the feature vectors can be large; (2) We do not know in advance the number of clusters; therefore, the algorithm needs not to require a pre-defined number of clusters.

Any clustering algorithm fulfilling the above requirements would fit here. In our experiments, we adopt a highly scalable Map-Reduce implementation of the hard-clustering version of Clustering by Committee (CBC), a state-of-the-art clustering algorithm presented in (Pantel and Lin, 2002).

Context Feature Space. The basic hypothesis for the *context feature space*, is that an entity can be effectively represented by the set of contexts in which it appears in queries. This allows to capture the users' view of the entity, i.e. what people query, and want to know about the entity. This is similar to that proposed by Pasca (2007b; 2007a), i.e. that queries provide good semantics cues for modeling named entities.

Our query log feature space may significantly differ from a classical contextual feature space com-

[1] We exclude 'Time' and 'Numerical Expressions', which are out of the scope of our study.

[2] We refer the readers to (Pennacchiotti and Jain, 2010) for details on the entity extraction algorithms.

puted over a Web corpus, since the same entity can be differently perceived and described in the two corpora (query log and Web). Consider for example the entity 'galapagos islands'. Typical contexts on the Web and query log for this entity are:

web:	endemic birds
web:	big turtles
web:	charles darwin foundation
web:	sensitive water
qlog:	trip to
qlog:	diving
qlog:	where are the
qlog:	travel package

The difference between the two representations implies that entities that are similar on the Web, are not necessarily similar on query logs. For example, on the Web 'galapagos islands' is very similar to other countries such as 'tasmania', 'guinea' and 'luxemburg'; while on query log is similar to other sea-side travel destination and related concepts, such as 'greek isle', 'kauai snorkeling' and 'south america cruise'. Our new similarity computed over query log, is potentially useful for those applications in which is more important to represent users' intents, than an objective description of entities (e.g. in query suggestion and intent modeling).

To obtain our contextual representation we proceed as follows. For each entity e, we identify all queries in the query log, in which e appears. Then, we collect the set of all suffixes and postfixes of the entity in those queries. For example, given the entity 'galapagos islands' and the query 'summer 2008 galapagos islands tour', the contexts are: 'summer 2008' and 'tour'.

Once the set of all contexts of all entities has been collected, we discard contexts appearing less than τ-times in the query log, so to avoid statistical biases due to data sparseness (in the reported experiments we set $\tau = 200$). We then compute the corrected pointwise mutual information ($cpmi$) (Pantel and Ravichandran, 2004) between each instance and each context c as:

$$cpmi(e,c) = log_2 \frac{f(e,c) \cdot f(*,*)}{f(e) \cdot f(c)} \cdot M \quad (1)$$

where $f(e,c)$ is the number of times e and c occur in the same query; $f(e)$ and $f(c)$ is the count of the entity and the context in the query log; $f(*,*)$ the overall count of all co-occurrences between contexts and entities; and M is the correction factor presented in (Pantel and Ravichandran, 2004), that eases the pmi's bias towards infrequent entities/features. Each instance is then represented in the feature space of all contexts, by the computed pmi values. Note that our method does not use any NLP parsing, since queries rarely present syntactic structure. This guarantees the method to be computationally inexpensive and easily adaptable to languages other than English.

Clickthrough Feature Space. During a search session, users issue a search query for which the search engine presents a list of result urls. Of the search results, users choose those urls that are representative of their intent. This interaction is captured by means of a click, which is logged by most search engines as *click-through data*. For instance, a search log may contain the following clicked urls for a query 'flv converter', for different users:

$user_1$:	www.flv-converter.com
$user_2$:	www.videoconverterdownload.com/flv/
$user_3$:	www.ripzor.com/flv.html

Our main motivation behind clustering entities based on past user click behavior is that non-identical queries that generate clicks on the same urls capture similar user intent. Thus, grouping entities that were issued as a query and generated user clicks on the same url may be considered similar. For instance, the query 'convert flv' may also generate clicks on one of the above urls, thus hinting that the two entities are similar. We observed that websites tend to dedicate a url per entity. Therefore, grouping by click urls can lead to clusters with synonyms (i.e., different ways of representing the same entity) or variants (e.g., spelling errors). To get more relevant clusters, instead of grouping entities by the click urls, we use the base urls. For instance, the url `www.ripzor.com/flv.html` is generalized to `www.ripzor.com`.

With the advent of encyclopedic websites such as, `www.wikipedia.org` and `wwww.youtube.com`, naively clustering entities by the clickthrough data can led to non-similar entities to be placed in the same cluster. For instance, we observed the most frequently clicked base url for both 'gold retriever' and 'abraham lincoln' is `www.wikipedia.org`. To address this issue, in our experiments we employed a

stop-list by eliminating top-5 urls based on their inverse document frequency, where an entity is intended as the 'document'.

In practice, each extracted entity e is represented by a feature vector of size equal to the number of distinct base urls in the click-through data, across all users. Each dimension in the vector represents a url in the click-through information. The value f of an entity e for the dimension associated with url j is computed as:

$$f(e,j) = \begin{cases} \frac{w(e,j)}{\sqrt{\sum_i^{|\mathcal{U}|} w(e,i)^2}} & \text{if url } j \text{ clicked for query } e; \\ 0 & \text{otherwise.} \end{cases}$$

where $\mathcal{U}$ is the set of base urls found in click-through data when entity e was issued as a query; and $w(e,i)$ is the number of time the base url i was clicked when e was a query.

Hybrid Feature Space. We also experiment a hybrid feature space, which is composed by the normalized union of the two feature spaces above (i.e. context and clickthrough). Though more complex hybrid models could be applied, such as one based on ensemble clustering, we here opt for a simple solution which allows to better read and compare to other methods.

3 Experimental Evaluation

In this section, we report experiments on our clustering method. The goal of the experiment is two-fold: (1) evaluate the intrinsic quality of the clustering methods, i.e. if two entities in the same cluster are similar or related from a web user's perspective; (2) verify if our initial hypothesis holds, i.e. if query log based features spaces capture different properties than Web based feature spaces (i.e. the 'user space'). In Section 3.1 we describe our experimental setup; and, in 3.2 we provide the results. We couple this intrinsic evaluation with an extrinsic application-driven one in Section 4.

3.1 Experimental Settings

In the experiments we use the following datasets:

Query log: A random sample of 100 million, fully anonymized queries collected by the Yahoo! search engine in the first 3 months of 2009, along with their frequency. This dataset is used to generate both the context and the clickthrough feature spaces for the clustering step.

Web documents A collection of 500 million web pages crawled by a Yahoo! search engine crawl. This data set is used to implement a web-based feature space that we will compare to in Section 3.2.

Entity set A collection of 2,067,385 entities, extracted with the method described in 2.1, which shows a precision of 0.705 ± 0.044. Details on the evaluation of such method are available in citetechrep, where a full comparison with state-of-the-art systems such as (Pasca, 2007b) and (Banko et al., 2007) are also reported.

Evaluation methodology: Many clustering evaluation metrics have been proposed, ranging from Purity to Rand-statistics and F-Measure. We first select from the original 2M entity set, a random set of n entities biased by their frequency in query logs, so to keep the experiment more realistic (more frequent entities have more chances to be picked in the sample). For each entity e in the sample set, we derived a random list of k entities that are clustered with e. In our experiments, we set $n = 10$ and $k = 20$. We then present to a pool of paid editors, each entity e along with the list of co-clustered entities. Editors are requested to classify each co-clustered entity e_i as *correct* or *incorrect*. An entity e_i is deemed as correct, if it is similar or related to e from a web user's perspective: to capture this intuition, the editor is asked the question: 'If you were interested in e, would you be also interested in e_i in any intent?'.[3] Annotators' agreement over a random set of 30 entities is $kappa = 0.64$ (Marques De Sá, 2003), corresponding to substantial agreement. Additionally, we ask editors to indicate the *relation type* between e and e_i (synonyms, siblings, parent-child, topically related).

Compared methods:

`CL-CTX`: A CBC run, based on the query log context feature space (Section 2.2).

`CL-CLK`: A CBC run, based on the clickthrough feature space (Section 2.2).

[3] For example, if someone is interested in 'hasbro', he could be probably also be interested in 'lego', when the intent is buying a toy. The complete set of annotation guidelines is reported in (Pennacchiotti and Jain, 2010).

method	# cluster	avg cluster size
CL-Web	1,601	240
CL-CTX	875	1,182
CL-CLK	4,385	173
CL-HYB	1,580	478

Table 1: Statistics on the clustering results.

CL-HYB: A CBC run, based on the hybrid space that combines CL-CTXand CL-CLK(Section 2.2).
CL-Web: A state-of-the-art open domain method based on features extracted from the Web documents data set (Pantel et al., 2009). This method runs CBC over a space where features are the contexts in which an entity appears (noun chunks preceding and following the target entity); and feature value is the *pmi* between the entity and the chunks.

Evaluation metrics: We evaluate each method using **accuracy**, intended as the percentage of correct judgments.

3.2 Experimental Results

Table 3 reports accuracy results. CL-HYB is the best performing method, achieving 0.85 accuracy, respectively +4% and +11% above CL-CLK and CL-Web. CL-CTX shows the lowest performance. Our results suggest that query log spaces are more suitable to model the 'user space' wrt web features. Specifically, clickthrough information are most useful confirming our hypothesis that queries that generate clicks on the same urls capture similar user intents.

To have an anedoctical and practical intuition on the results, in Table 2 we report some entities and examples of other entities from the same clusters, as obtained from the CL-HYB and CL-Web methods. The examples show that CL-HYB builds clusters according to a variety of relations, while CL-Web mostly capture sibling-like relations.

One relevant of such relations is topicality. For example, for 'aaa insurance' the CL-HYB cluster mostly contains entities that are topically related to the American Automobile Association, while the CL-Web cluster contains generic business companies. In this case, the CL-HYB approach simply chose to group together entities having clicks to 'aaa.com' and appearing in contexts as 'auto club'. On the contrary, CL-Web grouped according to contexts such as 'selling' and 'company'. The entity 'hip osteoarthritis' shows a similar behavior: CL-HYB groups entities topically related to orthopedic issues, since most of the entities are sharing contexts such as 'treatment' and 'recovery' and, at the same time, clicks to urls such as 'orthoinfo.aaos.org'.

entity	CL-HYB	CL-Web
aaa insurance	roadside assistance personal liability insurance international driving permits aaa minnesota	loanmax pilot car service localnet fibermark
paris hilton	brenda costa adriana sklenarikova kelly clarkson anja rubik federica ridolfi	julia roberts brad pitt nicole kidman al pacino tom hanks
goldie hawn	bonnie hunt brad pitt tony curtis nicole kidman	julia roberts brad pitt nicole kidman al pacino
hip osteoarthritis	atherosclerosis pneumonia hip fracture breast cancer	wrist arthritis disc replacement rotator cuff tears shoulder replacement
acer america	acer aspire accessories aspireone acer monitors acer customer service	microsoft casio computer borland software sony

Table 2: Sample of the generated entity clusters.

Another interesting observation regards entities referring to people. The 'paris hilton' and 'goldie hawn' examples show that the CL-Web approach groups famous people according to their category – i.e. profession in most cases. On the contrary, query log approaches tend to group people according to their social attitude, when this prevails over the profession. In the example, CL-HYB clusters the actress 'goldie hawn' with other actors, while 'paris hilton' is grouped with an heterogeneous set of celebrities that web users tend to query and click in a same manner: In this case, the social persona of 'paris hilton' prevails over its profession (actress/singer). This aspect is important in many applications, e.g. in query suggestion, where one wants to propose to the user entities that have been similarly queried and clicked.

In order to check if the above observations are not anedoctical, we studied the *relation type* annotation provided by the editors (Table 4). Table shows that query log based methods are more varied in the type of clusters they build. Table 5 shows the difference between the clustering obtained using the different methods and the overlap between the produced clusters.

As regard a more attentive analysis of the different query log based methods, CL-CTX has the

method	Precision
CL-Web	0.735
CL-CTX	0.460
CL-CLK	0.815 †
CL-HYB	**0.850** †

Table 3: Precision of various clustering methods († indicates statistical-significant better than the CL-Web method, using t-test).

class	method			
	CL-Web	CL-CTX	CL-CLK	CL-HYB
topic	0.27	0.46	0.46	0.40
sibling	0.72	0.43	0.29	0.32
parent	-	0.09	0.13	0.09
child	0.01	-	0.01	0.02
synonym	0.01	0.03	0.12	0.16

Table 4: Fraction of entities that have been classified by editors in the different relation types.

method	labelled clusters			
	CL-CTX	CL-CLK	CL-HYB	CL-Web
CL-CTX	-	0.2	0.53	0.29
CL-CLK	0.21	-	0.54	0.34
CL-HYB	0.53	0.51	-	0.31
CL-Web	0.33	0.35	0.41	-

Table 5: Purity of clusters for each method using clusters from other methods as "labelled" data.

lowest performance. This is mainly due to the fact that contextual data are sometimes too sparse and generic. For example 'mozilla firefox' is clustered with 'movie program' and 'astro reading' because they share only some very generic contexts such as 'free downloads'. In order to get more data, one option is to relax the τ threshold (see Section 2) so to include more contexts in the semantic space. Unfortunately, this would have a strong drawback, in that low-frequency context tend to be idiosyncratic and spurious. A typical case regards recurring queries submitted by robots for research purposes, such as 'who is X', 'biography of X', or 'how to X'. These queries tend to build too generic clusters containing people or objects. Another relevant problem of the CL-CTX method is that even when using a high τ cut, clusters still tend to be too big and generic, as statistics in Table 4 shows.

CL-CTX, despite the low performance, is very useful when combined with CL-CLK. Indeed the CL-HYB system improves +4% over the CL-CLK system alone. This is because the CL-HYB method is able to recover some misleading or incomplete evidence coming from the CL-CLK using features provided by CL-CLK. For example, editors judged as incorrect 11 out of 20 entities co-clustered with the entity 'goldie hawn' by CL-CLK. Most of these errors are movies (e.g. 'beverly hills cops') soap operas (e.g. 'sortilegio') and directors, because all have clicks to 'imdb.com' and 'movies.yahoo.com'. CL-HYB recovers these errors by including features coming from CL-CTX such as 'actress'.

In summary, query log spaces group together entities that are similar by web users (this being topical similarity or social attitude), thus constituting a practical model of the 'user space'.

4 Keywords for Sponsored Search

In this section we explore the use of our methods for keyword generation for sponsored search. In sponsored search, a search company opens an auction, where on-line advertisers bid on specific keywords (called *bidterms*). The winner is allowed to put its ad and link on the search result page of the search company, when the bidterm is queried. Bidterm suggestion tools (adWords, 2009; yahooTool, 2009) are used to help advertiser in selecting bidterms: the advertiers enters a seed keyword (*seed*) expressing the intent of its ad, and the tool returns a list of suggested keywords (*suggestions*) that it can use for bidding – e.g for the seed 'mp3 player', a suggestion could be 'ipod nano'. The task of generating bid suggestions (i.e. *keyword generation*) is typically automatic, and has received a growing attention in the search community. The main problem of existing methods for suggestion (adWords, 2009; yahooTool, 2009; wordTracker, 2009) is that they produce only suggestions that contain the initial seed (e.g. 'belkin mp3 player' for the seed 'mp3 player'), while nonobvious (and potentially less expensive) suggestions not containing the seed are neglected (e.g. 'ipod nano' for 'mp3 player'). For example for 'galapagos islands', a typical production system suggests 'galapagos islands tour' which cost almost 5$ per click; while the less obivous 'isla santa cruz' would cost only 0.35$. Below we show our method to discover such nonobvious suggestions, by retrieving entities in the same cluster of a given seed.

4.1 Experimental Setting

We evaluate the quality of the suggestions proposed by different methods for a set of seed bidterms., adopting the evaluation schema in (Joshi and Motwani, 2006)

Dataset Creation. To create the set of seeds, we use Google skTool[4]. The tool provides a list of popular bid terms, organized in a taxonomy of advertisement topics. We select 3 common topics: tourism, vehicles and consumer-electronics. For each topic, we randomly pick 5 seeds among the 800 most popular bid terms, which also appear in our entity set described in Section 3.1.[5]. We evaluate a system by collecting all its suggestions for the 15 seeds, and then extracting a random sample of 20 suggestions per seed.

Evaluation and Metrics. We use precision and Nonobviousness. **Precision** is computed by asking two experienced human experts to classify each suggestion of a given seed, as *relevant* or *irrelevant*. A suggestion is deemed as relevant if any advertiser would likely choose to bid for the suggestion, having as intent the seed. Annotator agreement, evaluated on a subset of 120 suggestions is $kappa = 0.72$ (substantial agreement). Precision is computed as the percentage of suggestions judged as relevant. **Nonobviousness** is a metric introduced in (Joshi and Motwani, 2006), capturing how nonobvious the suggestions are. It simply counts how many suggestions for a given seed do not contain the seed itself (or any of its variants): this metric is computed automatically using string matching and a simple stemmer.

Comparisons. We compare the suggestions proposed by `CL-CTX`, `CL-CLK`, and `CL-HYB`, against `Web` and two reference state-of-the-art production systems: Google AdWords (`GOO`) and Yahoo Search Marketing Tool (`YAH`). As concerns our methods, we extract as suggestions the entities that occur in the same cluster of a given seed.

[4]http://www.google.com/sktool

[5]The final set of 15 bid terms is: *tourism:*galapagos islands,holiday insurance,hotel booking,obertauern,wagrain; *vehicles:*audi q7,bmw z4,bmw dealers,suzuki grand vitara,yamaha banshee; *consumer electr:*canon rebel xti,divx converter,gtalk,pdf reader,flv converter.

4.2 Experimental Results

Precision results are reported in the second column of Table 6. Both `CL-CLK` and `CL-HYB` outperform `Web` in precision, `CL-HYB` being close to the upper-bound of the two production systems. As expected, production systems show a very high precision but their suggestions are very obvious. Our results are fairly in line with those obtained on a similar dataset, by Joshi and Motwani (2006).

A closer look at the results shows that most of the errors for `CL-CTX` are caused by the same problem outlined in Section 3.2: Some entities are wrongly assigned to a cluster, because they have some high $cpmi$ context feature which is shared with the cluster centroid, but which is not very characteristic for the entity itself. This is particularly evident for some of the low frequency entities, where $cpmi$ values could not reflect the actual semantics of the entity. For example the entity 'nickelodeon' (a kids tv channel in UK) is assigned to the cluster of 'galapagos islands', because of the feature 'cruise': indeed, some people query about 'nickelodeon cruise' because the tv channel organizes some kids cruises. Other mistakes are due to feature ambiguity. For example, the entity 'centurion boats' is assigned to the cluster of 'obertauern' (a ski resort in Austria), because they share the ambiguous feature 'ski' (meaning either winter-ski or water-ski). As for the `CL-CLK` system, some of the errors are caused by the fact that some base url can refer to very different types of entities. The `CL-HYB` system achieves a higher precision wrt `CL-CTX` and `CL-CLK`: the combination of the two spaces decreases the impact of misleading features –e.g. for 'yamaha bunshee', all `CL-HYB` 's suggestions are correct, while almost all `CL-CLK` 's suggestions are incorrect: the hybrid system recovered the negative effect of the misleading feature `ebay.com`, by backing up on features from the contextual subspace (e.g. 'custom', 'specs', 'used parts').

Nonobviousness results are reported in column three of Table 6. All our systems return a high number of nonobvious suggestions (all above 50%).[6] On the contrary, `GOO` and `YAH` show low perfor-

[6]Note that very high values for `CL-CTX` may be misleading, as many of the suggestions proposed by this system are incorrect (see precision results) and hence non-obvious (e.g., 'derek lewis' for 'galapagos islands').

method	Precision	Nonobviousness
GOO	0.982	0.174
YAH	0.966	0.195
Web	0.814	0.827
CL-CTX	0.547	0.963
CL-CLK	0.827	0.630
CL-HYB	0.946	0.567

Table 6: Results for keyword generation.

mance, as both systems are heavily based on the substring matching technique. This strongly motivates the use of semantic approaches as those we propose, that guarantee at the same time both a higher linguistic variety and an equally high precision wrt the production systems. For example, for the seeds 'galapagos islands', GOO returns simple suggestions such as 'galapagos islands vacations' and 'galapagos islands map'; while CL-HYB returns 'caribbean mexico' and 'pacific dawn', two terms that are semantically related but dissimilar from the seed. Remember that these letter terms are related to the seed because they are similar in the user space, i.e. users looking at 'galapagos islands' tend to similarly look for 'caribbean mexico' and 'pacific dawn'. These suggestions would then be very valuable for tourism advertisers willing to improve their visibility through a non-trivial and possibly less expensive set of bid terms.

5 Related Work

While literature abounds with works on entity extraction from web documents (e.g. (Banko et al., 2007; Chaudhuri et al., 2009; Pennacchiotti and Pantel, 2009)), the extraction of classes of entities over query logs is a pretty new task, recently introduced in (Pasca, 2007b). Pasca's system extracts entities of pre-defined classes in a semi-supervised fashion, starting with an input class represented by a set of seeds, which are used to induce typical query-contexts for the class. Contexts are then used to extract and select new candidate instances for the class. A similar approach is also adopted in (Sekine and Suzuki, 2007). Pasca shows an improvement of about 20% accuracy, compared to existing Web-based systems. Our extraction algorithm differs from Pasca's work in that it is completely unsupervised. Also, Pasca's cannot be applied to OIE, i.e. it only works for pre-defined classes. Our clustering approach is related to Lin and Wu's work (Lin and Wu, 2009). Authors propose a semi-supervised algorithm for query classification. First, they extract a large set of 20M phrases from a query log, as those unique queries appearing more than 100 times in a Web corpus. Then, they cluster the phrases using the K-means algorithm, where features are the phrases' bag-of-words contexts computed over a web corpus. Finally, they classify queries using a logistic regression algorithm. Our work differs from Lin and Wu, as we focus on entities instead of phrases. Also, the features we use for clustering are from query logs and click data, not web contexts.

6 Conclusions

We presented an open entity extraction approach over query logs that goes beyond the traditional web corpus, with the goal of modeling a 'user-space' as opposed to an established 'web-space'. We showed that the clusters generated by query logs substantially differ from those by a Web corpus; and that our method is able to induce state-of-the-art quality classes on a user-oriented evaluation on the real world task of keyword generation for sponsored search. in future work we plan to: (i) experiment different clustering algorithms and feature models, e.g. soft-clustering for handling ambiguous entities; (ii) integrate the Web space and the query log spaces; (iii) embed our methods in tools for intent modeling and query suggestion.

References

adWords. 2009. Google adwords. adwords.google.com/select/keywordtoolexternal.

Banko, Michele, Michael Cafarella, Stephen Soderland, Matthew Broadhead, and Oren Etzioni. 2007. Open information extraction from the web. In *Proceedings of IJCAI*.

Cao, Huanhuan, Daxin Jiang, Jian Pei, Qi He, Zhen Liao, Enhong Chen, and Hang Li. 2008. Context-aware query suggestion by mining click-through and session data. In *Proceedings of KDD-08*.

Chaudhuri, Surajit, Venkatesh Ganti, and Dong Xin. 2009. Exploiting web search to generate synonyms for entities. In *Proceedings of WWW-09*.

Grishman, R. and B. Sundheim. 1996. Message understanding conference- 6: A brief history. In *Proceedings of COLING*.

Hu, Jian, Gang Wang, Fred Lochovsky, Jian tao Sun, and Zheng Chen. 2009. Understanding user's query intent with Wikipedia. In *Proceedings of WWW-09*, pages 471–480.

Joshi, Amruta and Rajeev Motwani. 2006. Keyword generation for search engine advertising. In *Proceedings of Sixth IEEE-ICDM*.

Lin, Dekang and Xiaoyun Wu. 2009. Phrase clustering for discriminative learning. In *Proceedings of ACL-IJCNLP-2009*.

Marques De Sá, Joaquim P. 2003. *Applied Statistics*. Springer Verlag.

Pantel, Patrick and Dekang Lin. 2002. Discovering word senses from text. In *Proceedings KDD-02*, Edmonton, Canada.

Pantel, Patrick and Deepak Ravichandran. 2004. Automatically labeling semantic classes. In *Proceeding of HLT-NAACL-2004*.

Pantel, Patrick, Eric Crestan, Arkady Borkovsky, Ana-Maria Popescu, and Vishnu Vyas. 2009. Web-scale distributional similarity and entity set expansion. In *Proceedings of EMNLP-09*, Singapore.

Pasca, Marius. 2007a. Organizing and searching the world wide web of facts - step two: Harnessing the wisdom of the crowds. In *Proceedings of the WWW-2007*, Banff, Canada.

Pasca, Marius. 2007b. Weakly-supervised discovery of named entities using web search queries. In *Proceedings of CIKM-2007*.

Pennacchiotti, Marco and Alpa Jain. 2010. Open Information Extraction from Web Search Query Logs. Technical Report YL-2010-003, Yahoo! Labs, Sunnyvale, USA.

Pennacchiotti, Marco and Patrick Pantel. 2009. Entity extraction via ensemble semantics. In *Proceedings of EMNLP-2009*, pages 238–247, Singapore. Association for Computational Linguistics.

Sekine, Satoshi and Hisami Suzuki. 2007. Acquiring ontological knowledge from query logs. In *Proceedings of WWW-07*.

Sekine, Satoshi, Kiyoshi Sudo, and Chikashi Nobata. 2002. Extended named entity hierarchy. In *Proceedings of LREC-2002*.

wordTracker. 2009. Word tracker. www.wordtracker.com.

yahooTool. 2009. Yahoo search marketing. searchmarketing.yahoo.com.