Bringing Head Closer to the Tail with Entity Linking

Manisha Verma
University College London
m.verma@cs.ucl.ac.uk

Diego Ceccarelli
IMT Lucca
ISTI CNR, Pisa
diego.ceccarelli@isti.cnr.it

ABSTRACT

With the creation and rapid development of knowledge bases, it has become easier to understand the underlying semantics of unstructured text (short or long) on the web. In this work we especially look at the impact of entity linking on search logs. Search queries follow a Zipfian distribution wherein other than few popular queries (*head queries*), a significant percentage of queries (*tail queries*) occur rarely. Given a search log, there is sufficient data to analyze head queries but insufficient data (low frequency, limited clicks) to draw any conclusions about tail queries. In this work we focus on quantifying the extent of overlap between long tail and head queries by means of entity linking. We specifically analyze the frequency distribution of entities in head and tail queries. Our analysis shows that by means of entity linking, we can indeed bridge the gap between the head and tail.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Information Filtering*

Keywords

Entity linking; Annotations; Web Usage Mining.

1. INTRODUCTION

At present, a text query is the most prominent means to find information with a search engine. This makes search log mining or analysis unavoidable as understanding user dynamics, demographics or interests is instrumental in enriching user's search experience. *Web Usage Mining* [7] studies user needs in order to improve both user satisfaction and engagement on the website.

Search queries tend to follow a heavy-tailed Zipf Distribution [1], wherein a large fraction of queries occur too infrequently. This large set is called the *long tail*. While popular queries (*head queries*) are easy to analyze given their tremendous search volume, the small frequencies, low clicks and reformulations of tail queries makes it difficult to conclusively analyze them. Analysis of tail queries is not new, however, existing work [3, 4] looks at user behavior associated with rare queries whereas we postulate that the content of such queries can also provide important cues to both understand and associate them with popular queries.

Understanding rare queries only with bag-of-words approaches is hindered by multiple interpretations or incomplete information in query text. However, with the creation and constant development of knowledge bases, it is now possible to recognize underlying semantics of a query. Particularly, in this work, we propose to exploit **Entity Linking** (EL) for analyzing queries both at the head and the tail. Entity Linking helps enrich raw text with entities from a knowledge base; recently many approached for applying EL to queries have been proposed and evaluated during the Entity Recognition and Disambiguation Challenge[1].

Hollink *et al.* [5] exploited EL for web usage mining: in their work they consider a sample of queries related to entertainment from Yahoo! Logs. Their focus is to study the *types* of the queries (e.g., trailer, movies, dvd), and on finding type patterns among sessions and queries. However, in this work, we are interested in studying the relationship between the head and the tail queries through the entities they contain. Our primary research questions are:

- Are tail queries a different means to inquire about entities mentioned in the head queries?
- Can we find tail queries about entities that are not searched in the head (we will call them *tail entities*)?
- Can we find a relationship between tail entities and *head entities*?

The following sections cover the technique used for annotating a large query log with entities and our preliminary findings on the enriched log.

2. ENRICHING AOL QUERY LOG DATA

We perform our analysis on the AOL query log, since it is publicly available[2]. AOL log consists of approximately 20 million queries submitted by 650,000 users from March to May 2006. Queries are normalized (text lowercased, non ascii characters removed) and there are in total 10,154,742 distinct queries. We extract 2 distinct sets from these queries:

[1] `http://web-ngram.research.microsoft.com/ERD2014/`
[2] `http://www.gregsadetsky.com/aol-data/`

ESAIR'14, November 7, 2014, Shanghai, China.

http://dx.doi.org/10.1145/2663712.2666196

Q_{tail} The set of queries in the long tail, i.e. queries that appear in the log with a frequency *lower than or equal* to 2. The set contains $7,746,607$ distinct queries, i.e. 76% of distinct queries, but it is 26% of the total volume of the queries.

Q_{head} The set of queries in the head. It contains queries that appear with a frequency *greater than* 99. The set contains $19,953$ distinct queries, i.e. 0.002% if we look at the distinct queries, but still these queries represent 26% of total query volume.

Although, the two sets differ in number of queries ($\sim$ 19K versus $\sim$ 7M), they cover the same fraction of total queries issued to the search engine. All our analysis are performed on these two sets.

Enriching the Queries.

The first step of the analysis is to associate search queries with entities in a knowledge base. The Entity Linking task consists of identifying small fragments of text (called *spots*), which may refer to an entity (represented by a URI) within a knowledge base (KB). For example, 'NY', 'NYC' will link to New York City in a KB. Usually EL task consists of two steps: i) **Spot Detection**: given the input document (in our case a query), the spots are detected and for each spot a list of candidate entities is returned; and ii) **Disambiguation**: for each ambiguous spot (e.g., `Brazil` could refer to the country or the football team), a single entity is selected to be linked to the spot.

We performed only the first step of the Entity Linking process: spot detection. We do not perform disambiguation for two reasons: i) Since we consider individual queries and not sessions, the context is not sufficient and probably not useful to correctly disambiguate the query, and ii) Disambiguation is usually computationally more expensive since it involves the pairwise comparison of the candidate entities of the detected spots to compute *relatedness*[6] distance between them.

We identify spots in both the head and the tail queries using Dexter [2]. The linker exploits a dictionary of more than 10 million spots extracted from titles and anchor texts of a recent English Wikipedia dump. Given a query, all possible n-grams (with n between one and six, and considering only n-grams longer than 2 characters) are generated and matched against the dictionary. The system identifies at least one spot in $13,977$ (70%) and $4,901,987$ (63%) Q_{head} and Q_{tail} respectively. For each spot we also collect: i) the **position in the query**, the start and end position in the query ii) the **link probability**, the probability of the spot linking to an entity, computed by the number of occurrences of the spot as anchor text in Wikipedia divided by the number of occurrences as plain text and iii) **the candidate entities**, a list of possible candidate entities for the spot; for each candidate we also retrieve its **commonness**, the probability $p(e|s)$ that the spot s refers to the entity e.

3. ANALYSIS

The sets containing the spots detected in the Q_{head} and Q_{tail} are referred to as S_{head} and S_{tail} respectively. For this work, we decided to map each spot to the entity with the *highest commonness*, i.e., the highest probability $p(e|s)$ to be associated with the spot s in question.

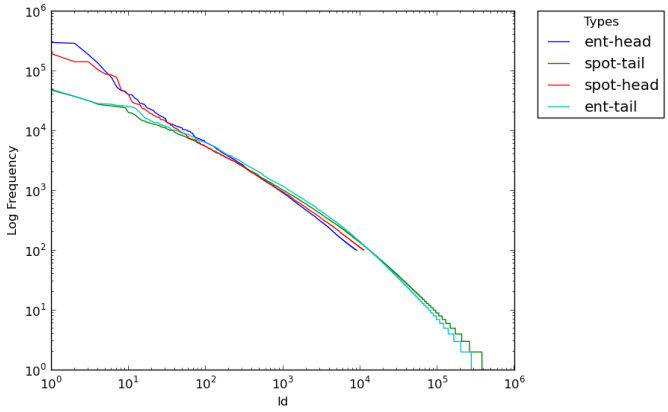

Figure 1: Frequency distributions for spots and entities in Q_{head} (spot-head, ent-head) and Q_{tail} (spot-tail, ent-tail)

Some real query examples with annotations (spots are highlighted in bold) are provided below.

- **first citizens bank** and trust **Raleigh NC** is annotated with the entities `Raleigh,_North_Carolina` and `First_Citizens_Bank`;
- **Tennessee walking horses** in **Missouri** is annotated with the entities `Missouri` and `Tennessee_Walking_Horse`;
- **i love you remix** featuring **Jim Jones** and **black rob** is annotated with the entities `I_Love_You,_Man`, `Remix`, `Jim_Jones_(rapper)`, `Black_Rob`.

We denote with E_{head} and E_{tail}, the sets containing the most probable entities for the spots. To reemphasize, while E_{head} are entities found in Q_{head}, E_{tail} are found in Q_{tail}. It is worth noting that this mapping could collapse different spots on the same entity (e.g., *united states* and *Usa* will both refer to United States). Since we do not complete the disambiguation phase, we understand that selection of highest scored entity each time for a spot induces some bias in this analysis. We shall, in future, also take into account remaining spots and terms in the query for entity disambiguation.

There are $11,152$ ($|S_{head}|$) distinct spots and $8,949$ ($|E_{head}|$) distinct entities in the head. In the tail, the system identifies $550,953$ ($|S_{tail}|$) distinct spots and $379,342$ ($|E_{tail}|$) distinct entities. It is also interesting to note the size of the intersections: $|S_{head} \cap S_{tail}| = 11,027$ $|E_{head} \cap E_{tail}| = 8,888$, this means that almost all the spots and entities in the head appear in the tail.

Figure 1 shows the frequency distribution of both head and tail spots as well as head and tail entities: it is interesting to note that while queries in head and tail follow totally different distributions, when we look at their spots/entities, the distributions are similar. In particular, as expected, head spots and head entities follow a similar distribution, as well as the spots and entities in the tail.

Do the most frequent entities in E_{head} and E_{tail} overlap? Not so much: we sorted the entities based on their total frequency within all the queries of Q_{head} and Q_{tail} to compare the ranked lists at different cutoffs (50, 100, 500,

Q_{head}				Q_{tail}			
S_{head}		E_{head}		S_{tail}		E_{tail}	
google	342,602	Google	349,337	florida	47,718	Florida	49,366
myspace	194,093	Yahoo	299,718	texas	37,388	Texas	37,526
yahoo	142,361	Myspace	289,353	ohio	31,861	Ohio	31,905
ebay	142,257	EBay	187,633	edu	26,641	New_York	28,396
yahoo.com	104,696	MapQuest	135,179	state	26,066	.edu	26,642
mapquest	88,617	Google_Search	98,112	california	25,233	U.S._state	26,392
google com	85,670	Hotmail	53,925	new york	24,865	California	25,859
my space	48,401	Bank_of_America	46,922	hotel	20,018	Real_estate	25,232
www.yahoo.com	44,198	Craigslist	45,586	real estate	19,702	Myspace	24,998
internet	39,865	Ask.com	39,873	myspace	18,533	Floruit	24,207
ebay com	30,652	Internet	39,865	restaurant	17,065	Restaurant	21,996
hotmail.com	28,492	Pornography	35,089	michigan	15,635	Hotel	20,289
map quest	27,949	Tattoo	33,113	new jersey	14,813	Nudity	18,245
craigslist	27,222	American_Idol	28,890	georgia	14,525	United_States	16,680
american idol	23,665	Yahoo!_Mail	28,238	black	13,921	Michigan	15,763

Figure 2: Most 15 frequent spots and entities detected in the head (S_{head}, E_{head}) and in the tail (S_{tail}, E_{tail})

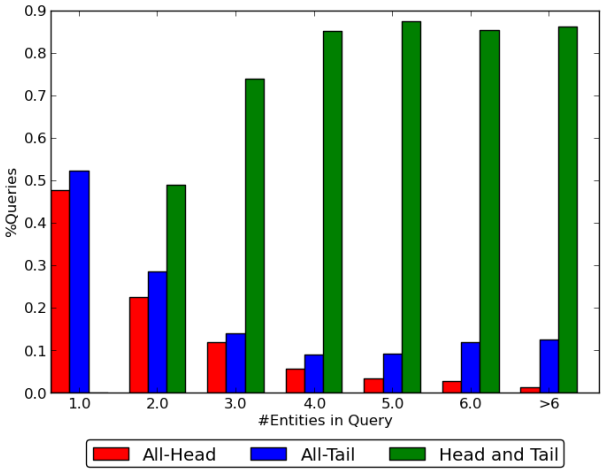

Figure 3: Percentage of tail queries with only head, only tail or both entities

5000). We then computed the Jaccard distance between the ranked lists, obtaining 0.25 when considering the most 5000 most frequent entities, and 0.21 considering the most frequent spots. At smaller cutoffs the Jaccard was considerably small, when we consider the top 50 entities in head and tail, Jaccard equals to 0.05. Table 3 lists the most frequent entities and spots in both head and the tail. It is interesting to observe that, while the head contains several navigational entities (google, yahoo etc.), the tail contains several US countries.

There are several queries in the log with multiple entities. On average tail queries tend to have more entities than head. To investigate whether tail queries inquire about entities in the head, we analyze percentage of tail queries containing only head entities, only tail entities ($\in E_{tail} \setminus E_{head}$) or both. Figure 3 depicts the percentage of queries with respect to number of entities in the query. One can clearly see that a large percentage of tail queries contains both head and tail entities. Moreover, this percentage gradually rises with the number of entities in the query. This indicates that in longer queries people look for connections among popular entities and rare entities.

4. CONCLUSION AND FUTURE WORK

We presented our preliminary results on using entity linking in order to find hidden connections between popular and rare queries. In this work we observed that: i) spots and entities in the head and in the tail of the AOL query log follow similar trends but ii) the most frequent entities are different iii) the more spots a query contains (i.e., the longer it is) the higher is the probability that head entities and tail entities co-occur. For future work, we would like to improve the quality of entity linking, implementing state of the art methods on queries and exploiting other informations in the logs (sessions, clicks etc). Finally, it would be interesting to study the relationships among the entities identified in the query log (e.g., they occur in the same query, in same session or share same clicks) and compare them with the relations that we have in the knowledge base.

Acknowledgements This work was partially supported by the EU project E-CLOUD (no. 325091), the Regional (Tuscany) project SECURE! (POR CReO FESR 2007/2011), and the Regional (Tuscany) project MAPaC (POR CReO FESR 2007/2013).

5. REFERENCES

[1] R. Baeza-Yates, A. Gionis, F. Junqueira, V. Murdock, V. Plachouras, and F. Silvestri. The impact of caching on search engines. In *SIGIR*. ACM, 2007.

[2] D. Ceccarelli, C. Lucchese, S. Orlando, R. Perego, and S. Trani. Dexter: an open source framework for entity linking. In *ESAIR*. ACM, 2013.

[3] D. Downey, S. Dumais, and E. Horvitz. Heads and tails: Studies of web search with common and rare queries. In *SIGIR*. ACM, 2007.

[4] S. Goel, A. Broder, E. Gabrilovich, and B. Pang. Anatomy of the long tail: Ordinary people with extraordinary tastes. In *WSDM*. ACM, 2010.

[5] L. Hollink, P. Mika, and R. Blanco. Web usage mining with semantic analysis. In *WWW*, 2013.

[6] D. Milne and I. H. Witten. Learning to link with wikipedia. In *CIKM*. ACM, 2008.

[7] F. Silvestri. Mining query logs: Turning search usage data into knowledge. *Foundations and Trends in Information Retrieval*, 4, 2010.