Large-scale Semantic Profile Extraction

Michael Gubanov
MIT CSAIL
The Stata Center
Cambridge, MA, 02139
michaelgubanov@csail.mit.edu

Michael Stonebraker
MIT CSAIL
The Stata Center
Cambridge, MA, 02139
stonebraker@csail.mit.edu

1. INTRODUCTION

Web-search engines usually can be outperformed by specialized systems optimized for a specific domain or type of data. Halevy et al in [1] demonstrate a use case for a specialized spatial search of Google Fusion Tables, whereby the user searches for bike trails in the San Francisco Bay Area and can see the result on a Google map. The same query submitted to the general-purpose Google Web-search engine returns many irrelevant search results.

Relevance of returned search results is a key property for any search-engine and hence an important and appreciated problem in Databases, Information Retrieval and Web-search [2, 3, 4, 5, 6, 7, 8, 9]. Users of any search-engine strongly prefer to get the most relevant search results first; otherwise, they have to spend time curating search-results.

Content providers on the Web and in other settings usually exhibit a specific focus for their postings. For example, information at `http://www.nasdaq.com` is usually in the financial domain, *Britney Spears* is mostly tweeting about the music, and *Business Wire* often publishes about acquisitions. It is rare for a source[1] to cover a wide variety of topics.

In this paper, we introduce and demonstrate a data structure designed to capture a semantic sketch of a data source, along with algorithms and similarity measures that can be used to extract, populate, and match similar profiles. For example, a newspaper *Business Wire* often publishes about *acquisitions* and therefore has this Named Entity type highly ranked in its profile (see Table 1).

We run our experiments on a corpus of 45 million Web pages - `Web45M`, provided by the Web aggregator Recorded Future [10], extract $\approx$ 1.4 million profiles, and leverage them to outperform general-purpose Web-search on certain types of queries.

The paper is organized as follows. Section 2 defines the *semantic profile* and describes the algorithms to extract semantic profiles. Section 3 describes a large-scale storage engines used to run the experiments. Section 4 introduces the similarity measures useful to match and find information sources that are alike. Section 5 demonstrates how the profiles can be used to improve general-purpose Web-search engines as well as for expert mining. We finish in Section 6 by discussing related and future work.

[1]except general purpose newspapers and aggregators

(c) 2014, Copyright is with the authors. Published in Proceeding of the 17th International Conference on Extending Database Technology (EDBT 2014) on OpenProceedings.org. Distribution of this paper is permitted under the terms of the Creative Commons license CC-by-nc-nd 4.0

Name	Type	Weight
Business Wire	BusinessTransaction	962.2
Business Wire	Acquisition	941.59
Business Wire	Continent	838.95
Business Wire	MedicalTreatment	644.54
Business Wire	CompanyTechnology	608.71
...	...	...
Business Wire	TVShow	506.86
Business Wire	DiseaseOutbreak	479.34

Table 1: Business Wire profile

Categories and Subject Descriptors

H.2 [**Database Management**]: Heterogeneous Databases; H.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

Keywords

Databases, Large-scale Data Integration; Web-search

2. SEMANTIC PROFILE

Def. A *semantic profile* P is a data structure encapsulating the main *types of named entities*[2] and *terms* from a datasource. It could also be used to describe multi-source content, such as the set of documents by a specific author. It can be represented as a triple $P = (S, \Theta, T)$, where:

- S - a unique name for the datasource
- Θ - n main weighted Named Entity types in S: $\{(\theta_i, w_i)\}$, $i = 1..n$, $\theta_i - i_{th}$ *type*, $w_i - i_{th}$ *weight*
- T - m main weighted document terms of S: $\{(t_j, w_j)\}$, $j = 1..m$, $t_j - j_{th}$ *term*, $w_j - j_{th}$ *weight*

We apply a modified version of TF/IDF algorithm [11] to calculate weights for `Web45M`:

$$w_{\theta_i j} = \theta f_{ij} \times \log(N/sf_{\theta_i}), \quad w_{t_i j} = tf_{ij} \times \log(N/sf_{t_i}),$$

- θf_{ij} - number of occurrences of *type* θ_i in source j
- sf_{θ_i} - number of sources containing θ_i
- N - total number of sources in `Web45M`

[2]further referred to as *types*

10.5441/002/edbt.2014.64

- tf_{ij} - number of occurrences of term t_i in source j
- sf_{t_i} - number of sources containing t_i

Informally, a profile is intended to capture a *semantic sketch* of an information source. It accumulates the main types of named entities and terms that appear in the source. For example, the semantic sketch for *Business Wire* is shown in Table 1. Tables 2, 5 illustrate parts of the profiles extracted extracted from `Web45M` for *The Kansas City Star*, and *Chicago Tribune*.

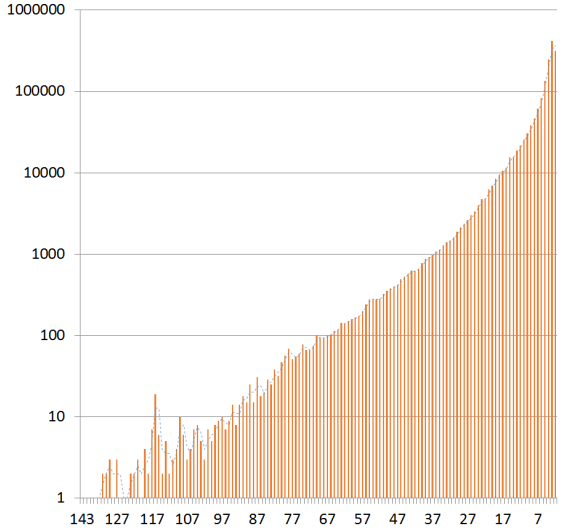

Figure 1: Web45M Semantic Sketches Statistics

We know that these newspapers discuss different topics relevant to their reading community. We can see that their profiles closely reflect these differences in topics. The profile for *Business Wire* in Table 1 is strongly business-oriented - we can see *BusinessTransaction* and *Acquisition* among its main types. Less dominant, but also present - *TVShow* and *DiseaseOutbreak*. By contrast, the profile of *The Kansas City Star* is mostly family-oriented. We can see *Movie, TVShow, MusicGroup*, and *TVStation* among its main types. Less dominant, but still present - *Currency*, and *MedicalCondition*. The profile for *Chicago Tribune* clearly indicates a newspaper of a big city; however, it is very different from *Business Wire*. Here dominating types are *Arrest*, *PersonLocation*, *Conviction*, and *MedicalCondition*. Also present are *SportsEvent*, *Technology*, *NaturalDisaster*. Finally, Table 4 has the profile for finance-related tweets. As in *Business Wire* we can see the types *BusinessTransaction*, *Acquisition*, and main terms such as *merger*, *servers*, and *Exxon*.

Figure 1 illustrates the distribution of the source semantic sketches and their respective types in the `Web45M` corpus. We can see it is close to an exponential distribution. There are just a few sources publishing about more than a hundred different types and an abundance of sources focused on publishing just about a few types. On the Y-axis (logarithmic scale) we plot the number of sources, and on X-axis - the number of types.

3. STORAGE

We used both a large-scale semi-structured sharded storage engine for the `Web45M` corpus as well as a parallel large-scale relational engine for structured analytics running in a distributed environment. The relational table with the extracted summarized profiles has 900 million rows, the semi-structured distributed dataset takes ≈ 1TB space without indexes. We can see in Table 3 `Web45M` consists of 242 distributed 2GB extents and has more than 17 million entries.

Name	Type	Weight
The Kansas City Star	Movie	876.57
The Kansas City Star	Country	765.55
The Kansas City Star	PersonCareer	731.32
The Kansas City Star	Currency	655.93
The Kansas City Star	Region	519.17
The Kansas City Star	PublishedMedium	495.23
The Kansas City Star	Quotation	391.63
The Kansas City Star	TVShow	371.22
The Kansas City Star	NaturalFeature	345.59
The Kansas City Star	MedicalCondition	328.59
The Kansas City Star	MusicGroup	320.92
The Kansas City Star	PersonLocation	296.12
The Kansas City Star	TVStation	271.3
The Kansas City Star	MusicAlbum	230.31

Table 2: Kansas City Star profile

```
> db.instance.stats();
{
        "ns" : "Web45M",
        "count" : 17731744,
        "numExtents" : 242,
        "nindexes" : 1,
        "lastExtentSize" : 1903786752,
        "totalIndexSize" : 733651904,
        ...
}
```

Table 3: Sharded Web45M collection statistics

4. SIMILARITY MEASURES

Def: *F-similarity* between the data sources s_i and s_j is defined as a summation of all their common types' weights. θ in the formula below is the intersection of types from the profiles of s_i and s_j, w_i^θ is the weight of such a common type from the profile of s_i, w_j^θ - the weight of the same type, but from the profile of s_j.

$$f(s_i, s_j) = \sum_{\theta \in s_i \times s_j} (w_i^\theta + w_j^\theta)$$

Def: *G-similarity* between the data sources s_i and s_j is defined as a summation of all their common types' and terms' weights (for each type). θ in the formula below is the intersection of types from the profiles of s_i and s_j, w_i^θ is the weight of such a common type from the profile of s_i, w_j^θ - the weight of the same type, but from the profile of s_j. t in the second summation is the intersection of terms from the

Name	Type	Weight(Type)	Term	Weight(Term)
Financial (@finance_outlook)	BusinessTransaction	18.85	merger	24.24
Financial (@finance_outlook)	BusinessTransaction	18.85	Dish-Sprint	24.01
Financial (@finance_outlook)	BusinessTransaction	18.85	Softbank	18.2
Financial (@finance_outlook)	Acquisition	16.63	Lenovo	13.2
Financial (@finance_outlook)	Acquisition	16.63	Servers	13.2
Financial (@finance_outlook)	Acquisition	16.63	merger	12.12
Financial (@finance_outlook)	Event	16.63	profit	17.83
Financial (@finance_outlook)	Event	12.18	Exxon	14.36
Financial (@finance_outlook)	Event	12.18	quarterly	13.99

Table 4: Financial @finance_outlook Profile

Name	Type	Weight
Chicago Tribune	Region	994.26
Chicago Tribune	PersonLocation	843.67
Chicago Tribune	Arrest	797.63
Chicago Tribune	PersonCommunication	749.33
Chicago Tribune	Announcement	702.56
Chicago Tribune	Conviction	702.06
Chicago Tribune	MedicalCondition	638.13
Chicago Tribune	PublishedMedium	630.29
Chicago Tribune	NaturalDisaster	544.1
Chicago Tribune	SportsEvent	513.53
Chicago Tribune	Technology	505.64

Table 5: Chicago Tribune profile

profiles of s_i and s_j for a type τ from θ, w_i^t is the weight of a common term from the profile of s_i for the same τ, w_j^t - the weight of the same term, but from the profile of s_j for the same τ.

$$g(s_i, s_j) = \sum_{\theta \in s_i \times s_j} (w_i^\theta + w_j^\theta + \sum_{t \in t_i \times t_j} (w_i^t + w_j^t))$$

Informally, F-similarity is a similarity score for two data sources that increases with the number of overlapping types from their profiles. In addition to that G-similarity also takes into account the number of overlapping terms for each common type.

Matching: Now we will discuss matching of the data sources by their profiles using the similarity measures introduced above. One of the similar sources to the *New York Times* (w.r.t. F-similarity) turns out to be `thefinance.sg` - an online Singapore financial newspaper. We can see the types that these sources share and the similarity score by type in Table 6. One of the similar authors to *Louis Charbobbeau* of the *Japan Herald* turns out to be *David Sine* w.r.t. to G-similarity by the types and terms used in the published materials. We can see these types and terms both authors are using in Table 7.

Source 1	Source 2	Type	Weight
New York Times	thefinance.sg	Person	167.75
New York Times	thefinance.sg	Company	165.18
New York Times	thefinance.sg	Event	158.04
New York Times	thefinance.sg	Position	129.82
New York Times	thefinance.sg	Currency	105.12
New York Times	thefinance.sg	Product	85.40
New York Times	thefinance.sg	City	77.87

Table 6: Matching NY Times and `thefinance.sg`

5. APPLICATIONS

First, we demonstrate the *large-scale extraction* of several semantic profiles from `Web45M` using the definitions from Section 2. We can see that the extracted profiles (Tables 1-2, 5) provide a short summary of main types that are usually discussed by a datasource or an author. Hence, the profile extraction algorithm can be very useful to quickly gain insight into a new large-scale datasource, corpus, or even the background of a new person by having access to her documents.

Second, we demonstrate the *profile matching* algorithms from Section 4 that can be useful to quickly find similar datasources or people from a large pool.

Finally, we demonstrate that taking information by topic from the specialized sources yields more relevant search re-

Source 1	Source 2	Type	Weight(Type)	Term	Weight(Term)
Louis Charbonneau	David Sine	Country	1224.89	Korea	82.2
Louis Charbonneau	David Sine	Country	1224.89	Lebanon	75.38
Louis Charbonneau	David Sine	Country	1224.89	place	73.7
Louis Charbonneau	David Sine	Country	1224.89	attack	59.88
Louis Charbonneau	David Sine	Country	1224.89	April	47.93
Louis Charbonneau	David Sine	Country	1224.89	against	30.7
Louis Charbonneau	David Sine	Event	380.56	rebel	81.78
Louis Charbonneau	David Sine	Event	380.56	action	78.63

Table 7: Common types&terms of Louis Charbobbeau of Japan Herald and David Sine

Name	Type	Weight
Gaming & Hacking Elite	Product	96.55
Gaming & Hacking Elite	OrgEntity	14.95
Gaming & Hacking Elite	Event	11.09
Gaming & Hacking Elite	OpSystem	7.21
Gaming & Hacking Elite	Person	6.93
Gaming & Hacking Elite	City	5.99
Gaming & Hacking Elite	Company	4.39

Table 8: Gaming & Hacking Elite Profile

sults compared to using the general purpose Web-search engines for certain kinds of queries. For example, an analyst might query Google Web-search to find out what was the position of the State Department regarding a meeting about missiles in Iran in 2008. She would get many irrelevant results mentioning this event, but miss the needed details about the Department of State. However, search over documents with profiles on such topics would return less noise or, which is the same thing - *more relevant* search results.

Also, all queries addressing a topic irrelevant to the datasource, would return less noise. For example, from the profile for *Gaming&Hacking Elite* - a Facebook Community about Video Games, we can see which types it usually does cover and which it does *not*. Hence, the search-engine equipped with profiles would return no results for the query: `"Gaming&Hacking Elite" TvShow`, because this source does *not* address any TV shows as it is evident from its profile. Google and any other profile-oblivious search-engine, however, would return irrelevant search results, because it did not take the profile into account and just ran the query through its index containing all kinds of Web pages. Hence, a search over the `Web45M` corpus using all profiling information demonstrates superior precision to a general-purpose Web-search engine on such queries.

To prove more general relevance guarantees, a more robust search relevance evaluation on a larger set of queries is needed [12], which is a subject of ongoing work.

6. RELATED AND FUTURE WORK

Online media and the social Web is the most dynamic part of the Internet that has very stringent requirements for search, relevance and information freshness. Those challenges suggest research topics for the real-time social Web. Specifically, [8] discusses one of the first systems for social analytics on news, where a user can "explore public reaction on articles relevant to a topic". Another challenge related to social networks and the Web in general is a huge volume of data, and the need for technologies to handle this volume [4, 5, 6], as well as the new ways to explore and visualize data and the result of analyses [7, 8]. Future work includes performing a robust relevance evaluation on a larger sets of queries, and calculating the specific relevance gains [12].

7. REFERENCES

[1] A. Halevy. Data publishing and sharing using fusion tables. In *CIDR*, 2013.

[2] M. Gubanov, L. Popa, H. Ho, H. Pirahesh, P. Chang, and L. Chen. Ibm ufo repository. In *VLDB*, 2009.

[3] M. Gubanov, A. Pyayt, and L. Shapiro. Readfast: Browsing large documents through ufo. In *IRI*, 2011.

[4] Stratos Idreos, Martin L. Kersten, and Stefan Manegold. Self-organizing tuple reconstruction in column-stores. In *SIGMOD*. ACM, 2009.

[5] Ioannis Alagiannis Ryan Johnson Idreos, Stratos and Anastasia Ailamaki. Here are my data files. here are my queries. where are my results? In *CIDR*, 2011.

[6] S. Idreos. Cracking big data. In *ERCIM News*, 2012.

[7] Stratos Idreos and Erietta Liarou. dbtouch: Analytics at your fingertips. In *CIDR*, 2013.

[8] S. Amer-Yahia, S. Anjum, A. Ghenai, A. Siddique, S. Abbar, S. Madden, A. Marcus, and M. El-Haddad. Maqsa: a system for social analytics on news. In *SIGMOD*, 2012.

[9] Sihem Amer-Yahia. Crowd sourcing literature review in sunflower. In *WWW CrowdSearch*, 2012.

[10] online: http://www.recordedfuture.com.

[11] Karen Sparck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28:11–21, 1972.

[12] M. Gubanov and A. Pyayt. Readfast: High-relevance search-engine for big text. In *ACM CIKM*, 2013.