CrowdGather: Budgeted Entity Extraction over Structured Data Domains

ABSTRACT

Crowd entity extraction has become a popular means of acquiring data for many applications, including recommendation systems, listing aggregation and knowledge base compilation. Most of the current solutions focus on entity extraction for specific queries and do not consider entity extraction over broader entity domains. Due to the time and cost of human labor, considering each query in isolation may incur large costs when applied to broader domains, thus, limiting the applicability of current approaches.

In this paper, we explore the problem of *budgeted entity extraction* over *structured entity domains*. We consider domains that can be fully described by a collection of predicates, each characterized by a hierarchical structure. We develop new statistical tools that enable users to reason about the gain of issuing *further queries* in the presence of little information and show how to exploit the dependencies across different points of the data domain to obtain more accurate estimates. We also demonstrate how budgeted entity extraction over large domains can be cast as an adaptive optimization problem that seeks to maximize the number of extracted entities while minimizing the overall extraction cost. We evaluate our techniques with experiments on both synthetic and real-world data.

1. INTRODUCTION

Combining human computation with traditional computer systems has been recently proven beneficial in extracting knowledge and acquiring data for many application domains, including recommendation systems [1], knowledge base completion [2, 5], entity extraction and structured data collection [4, 3]. In fact, extracting information from the crowd has been shown to provide access to more fine grained information that may belong to the long tail of the web or even be completely unavailable on the web.

However, due to the time and cost of human labor, crowd-based information extraction is limited to small data domains and raises several challenges when used in large data domains. Next, we use a real-world scenario to illustrate these challenges.

1.1 Challenges

We consider the scenario of building a crowdsourced event directory with detailed information about each event. Typically, event aggregators, such as Eventbrite [1] or Lanyrd [2] use crowdsourcing to collect information about different types of events over different locations. Typically, event aggregators are interested in collecting diverse events spanning from conferences and music festivals to political rallies for multiple locations across different countries. The goal of is to

1.2 Contributions

Give an example of entity extraction over a structured data domain.

First example is that of extracting events. Attributes: Location, Type,

Second example is that of e

Give examples of dependencies: (1) side information, (2) empty sub-domains.

Mention current work.

Combining human computation with traditional computer systems has been recently proven beneficial in extracting knowledge and acquiring data for many application domains, including recommendation systems [1], knowledge base completion [2, 5], entity extraction and structured data collection [4, 3].

Limitation of current work: (1) focus on specific domain points, ignoring the obtained information when issuing new queries at different points of the data domain, large cost, may not capture the tail of the items in the domain, (2) pattern mining approaches, do not consider the overall budget provided by a user.

Contributions, estimate the expected gain from further queries at different points in the hierarchy. Exploit indirect information to obtain more accurate estimates. Cast budgeted entity extraction as an adaptive optimization problem.

2. CROWDSOURCED ENTITY EXTRACTION

In this section we first review the problem of *crowdsourced entity extraction* and introduce a *budgeted* version of the basic problem. Then, we formally define the problem of *budgeted entity extraction over hierarchical domains*.

2.1 Entity Extraction Queries

[1] https://www.eventbrite.com
[2] http://lanyrd.com

2.2 Extracting Entities over Hierarchies

3. FRAMEWORK OVERVIEW

3.1

3.2

3.3

4. THE GAIN OF FURTHER QUERIES

5. PROFITABLE QUERYING POLICIES

6. RELATED WORK

Beth's work
Tova's work
CrowdFill work

7. CONCLUSIONS

8. REFERENCES

[1] Y. Amsterdamer, S. B. Davidson, T. Milo, S. Novgorodov, and A. Somech. OASSIS: query driven crowd mining. In *International Conference on Management of Data, SIGMOD 2014, Snowbird, UT, USA, June 22-27, 2014*, pages 589–600, 2014.

[2] S. K. Kondredi, P. Triantafillou, and G. Weikum. Combining information extraction and human computing for crowdsourced knowledge acquisition. In *30th IEEE International Conference on Data Engineering*, ICDE '14, 2014.

[3] H. Park and J. Widom. Crowdfill: A system for collecting structured data from the crowd. In *23rd International World Wide Web Conference (WWW)*, 2014.

[4] B. Trushkowsky, T. Kraska, M. J. Franklin, and P. Sarkar. Crowdsourced enumeration queries. In *Proceedings of the 2013 IEEE International Conference on Data Engineering (ICDE 2013)*, ICDE '13, pages 673–684, 2013.

[5] R. West, E. Gabrilovich, K. Murphy, S. Sun, R. Gupta, and D. Lin. Knowledge base completion via search-based question answering. In *Proceedings of the 23rd International Conference on World Wide Web*, WWW '14, pages 515–526, 2014.