Information Processing and Management 53 (2017) 309–331

Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/infoproman

Sampling strategies for information extraction over the deep web

Pablo Barrio*, Luis Gravano

Columbia University, Computer Science Department, 500 West 120th Street, Room 405, MC0401, New York, NY 10027, USA

ARTICLE INFO

Article history:
Received 25 December 2015
Revised 21 November 2016
Accepted 23 November 2016
Available online 6 December 2016

Keywords:
Information extraction
Sampling
Deep web
Text mining
Scalability

ABSTRACT

Information extraction systems discover structured information in natural language text. Having information in structured form enables much richer querying and data mining than possible over the natural language text. However, information extraction is a computationally expensive task, and hence improving the efficiency of the extraction process over large text collections is of critical interest. In this paper, we focus on an especially valuable family of text collections, namely, the so-called deep-web text collections, whose contents are not crawlable and are only available via querying. Important steps for efficient information extraction over deep-web text collections (e.g., selecting the collections on which to focus the extraction effort, based on their contents; or learning which documents within these collections—and in which order—to process, based on their words and phrases) require having a representative document sample from each collection. These document samples have to be collected by querying the deep-web text collections, an expensive process that renders impractical the existing sampling approaches developed for other data scenarios. In this paper, we systematically study the space of query-based document sampling techniques for information extraction over the deep web. Specifically, we consider (i) alternative query execution schedules, which vary on how they account for the query effectiveness, and (ii) alternative document retrieval and processing schedules, which vary on how they distribute the extraction effort over documents. We report the results of the first large-scale experimental evaluation of sampling techniques for information extraction over the deep web. Our results show the merits and limitations of the alternative query execution and document retrieval and processing strategies, and provide a roadmap for addressing this critically important building block for efficient, scalable information extraction.

1. Introduction

Information extraction systems are complex software tools that discover structured information in natural language text. For example, an information extraction system trained to extract *Occurs − in*(*Natural Disaster, Location*) tuples would extract the tuple ⟨tornado, Adairsville⟩ from the text “the tornado caused significant damage in Adairsville.” Having information in structured form enables much richer querying and data mining than possible over the natural language text. Unfortunately, information extraction is a time-consuming task. Since text collections routinely contain millions of documents or more,

* Corresponding author.
E-mail addresses: pjbarrio@cs.columbia.edu (P. Barrio), gravano@cs.columbia.edu (L. Gravano).

http://dx.doi.org/10.1016/j.ipm.2016.11.006
0306-4573/© 2016 Elsevier Ltd. All rights reserved.

improving the efficiency and scalability of the information extraction process over these large text collections is critical. In this paper, we focus on an especially valuable family of text collections, namely, the so-called *deep-web text collections*, whose contents are not crawlable and are only available via querying (Bergman, 2001; Gupta & Bhatia, 2014; Raghavan & Garcia-Molina, 2001; Sherman & Price, 2003). Deep-web text collections many times exhibit a full-text search interface. (We rely on this interface to access the contents of the collection, as we discuss in Section 4.) Moreover, deep-web text collections cover a wide range of topics and are hence relevant to a broad spectrum of information extraction tasks. Efficiently processing the contents of these collections is thus of significant interest.

Important steps for efficient information extraction over deep-web text collections require having, for each collection, a representative document sample of documents that lead to the extraction of tuples for a relation of interest. We refer to the documents that lead to the extraction of tuples for a relation of interest as the *useful documents* for the information extraction task.[1] The document samples can be valuable, for instance, to decide on which collections to focus the extraction effort, based on their contents (Barrio, Gravano, & Develder, 2015a). For example, such document samples can reveal that the Federal Emergency Management Agency (FEMA) collection,[2] an up-to-date resource for natural disasters and other hazards in the United States, is a better collection for the extraction of the $Occurs-in$ relation than the PubMed collection,[3] a database for life sciences and biomedical research. Similarly, a document sample from a collection can be valuable to help select and rank the collection documents for the extraction task: for efficiency, we should attempt to process only useful documents, so techniques such as QXtract (Agichtein & Gravano, 2003), FactCrawl (Boden, Löser, Nagel, & Pieper, 2012), PRDualRank (Fang & Chang, 2011), and BAgg-IE and RSVM-IE (Barrio, Simões, Galhardas, & Gravano, 2015b) use these samples to learn words and phrases that separate useful documents for the information extraction task from the rest. The samples on which these techniques rely must be collected in a collection-specific way, because the focus and language of each collection generally differs from those of other collections.

Given an information extraction task, producing high-quality, representative document samples from a deep-web text collection is a challenging process, for two main reasons. *(1) Sampling efficiency:* the document sampling process has to be efficient and lightweight because, as discussed above, it is often used to make the overall information extraction execution over deep-web text collections efficient and scalable. This efficiency requirement is complicated by the fact that document samples can only be collected, by definition, by querying the (remote) deep-web text collections, which is expensive. Furthermore, as we will see, analyzing the documents as we retrieve them, to decide the composition of the samples, is also an expensive proposition because it often involves running the extraction system at hand on the documents. *(2) Sampling quality:* the document sampling process has to return documents that represent the relevant extraction-related document characteristics in each deep-web text collection. This quality requirement is complicated by the fact that the useful documents for the information extraction task are often a small minority of the collection documents. For example, under 2% of the 1.03 million documents in TREC 1–5 collections[4] are useful for $Occurs-in$ when processed with a state-of-the-art information extraction system. Furthermore, even within a relatively small number of documents, the sampling process should capture the large variations in language and general content in the documents.

Earlier efforts to address the efficiency and scalability of the extraction process have incorporated sampling in a relatively ad-hoc manner. Notably, QXtract (Agichtein & Gravano, 2003), FactCrawl (Boden et al., 2012), PRDualRank (Fang & Chang, 2011), and BAgg-IE and RSVM-IE (Barrio et al., 2015b) rely on document sampling to develop document retrieval or ranking strategies for an information extraction task at hand. Despite the important role of sampling in these techniques, the sampling approaches that they use are far from ideal, as we will see. Specifically, these techniques adopt flavors of sampling that rely on high-precision queries to target certain documents efficiently, but fail to capture the large variety of extraction-relevant document characteristics discussed above. Consequently, they miss important groups of documents during sampling, which other sampling strategies can effectively obtain, as we will show experimentally.

Query-based document sampling has also been studied beyond information extraction, for other text-centric tasks. As notable examples, Bar-Yossef and Gurevich (2008), Zhang, Zhang, and Das (2011), Wang, Liang, and Lu (2014a), and Wang, Liang, and Lu (2014b) developed document sampling techniques for the generation of unbiased descriptors of the collections. Unfortunately, these approaches are ineffective for our information extraction scenario, because they focus on obtaining random document samples. As we discussed above, our scenario requires that the document samples represent the often small minority of documents that lead to extraction output for a given information extraction task. To sufficiently characterize the documents in such small portions of the collections through random sampling, the above techniques would require issuing an exorbitant number of queries to the deep-web text collections.

[1] We do not consider the correctness of extracted tuples in our work. Instead, we trust the output of the information extraction system and focus on efficiently and effectively identifying useful documents for our extraction task of interest. For correctness, we could use the confidence score that the information extraction system often assigns to each extracted tuple. This approach has been adopted in Agichtein and Cucerzan (2005); Jain and Srivastava (2009) for the (related) task of identifying text collections with high-quality, or correct, tuples. Alternatively, to deem tuples as correct, we could adopt the statistical approach proposed in Jain, Doan, and Gravano (2008); Jain and Ipeirotis (2009); Jain, Ipeirotis, Doan, and Gravano (2009); Simões, Galhardas, and Gravano (2013) for the (related) task of building efficiency- and quality-aware execution plans to extract tuples from large text collections.

[2] http://www.fema.gov/.

[3] http://www.ncbi.nlm.nih.gov/pubmed.

[4] http://trec.nist.gov/data.html.