Knowledge Graphs for Social Good: An Entity-centric Search Engine for the Human Trafficking Domain

Mayank Kejriwal, Pedro Szekely

Abstract—Web advertising related to Human Trafficking (HT) activity has been on the rise in recent years. Answering entity-centric questions over crawled HT Web corpora to assist investigators in the real world is an important social problem, involving many technical challenges. This paper describes a recent entity-centric knowledge graph effort that resulted in a semantic search engine to assist analysts and investigative experts in the HT domain. The overall approach takes as input a large corpus of advertisements crawled from the Web, structures it into an indexed knowledge graph, and enables investigators to satisfy their information needs by posing investigative search queries to a special-purpose semantic execution engine. We evaluated the search engine on real-world data collected from over 90,000 webpages, a significant fraction of which correlates with HT activity. Performance on four relevant categories of questions on a mean average precision metric were found to be promising, outperforming a learning-to-rank approach on three of the four categories. The prototype uses open-source components and scales to terabyte-scale corpora. Principles of the prototype have also been independently replicated, with similarly successful results.

Index Terms—Knowledge graphs, Query-centric knowledge graph construction, Information retrieval for social good, Entity-centric search, Investigative questions, Knowledge discovery, Human trafficking, Semantic search

1 Introduction

DATA from various authoritative sources, including the National Human Trafficking Resource Center, show that human trafficking (HT) is not only on the rise in the United States, but is a problem of international proportions [1], [2]. The advent of the Web has made the problem worse [3]. Human trafficking victims are advertised both on the Open and Dark Web, with estimates of the number of (not necessarily unique) published advertisements numbering in the tens, if not hundreds, of millions [4].

In recent years, various agencies in the US have turned to technology to assist them in combating this problem through the suggestion of leads, evidence and HT indicators. An important goal is to answer *entity-centric questions* over noisy Web corpora crawled from a subset of Web domains known for HT-related activity. Entities are typically HT victims, such as *escorts*, but could also be latent entities such as *vendors*[1], who organize the activity.

Specifically, given a corpus of HT ads crawled over the Open and Dark Web, the broader problem of *investigative knowledge discovery* is to return, in near real-time, a ranked list of *structured* answers to entity-centric questions [5] e.g. *find the names and ads of Cuban escorts operating in Key Biscayne, Florida in May, 2009*. By structured, we mean that, in addition to entity identifiers, the investigator also wants to query on, and retrieve, *semantic attributes* such as name and phone number from a controlled domain vocabulary (denoted as an *investigative schema*). Investigators also want to query latent entities, which are not directly observed in the data, but have to be inferred using *clustering* methods. Finally, investigators want to perform selective aggregation operations on queried data e.g. *find the average price per hour of Mexican escorts operating in Long Beach, New York on Dec. 25, 2016*. We provide a motivating example describing the problem of locating latent cluster entities subsequently.

The motivation behind investigative knowledge discovery is to build a system that, with minimal technical training, can assist investigators (e.g. law enforcement officials on the ground) acquire key evidence and leads to both locate and prosecute vendors of human trafficking activity, and to take preventive action. The driving hypothesis is that accomplishing such goals is significantly aided by leveraging the vast amount of advertising data on the Web.

Two interrelated problems need to be solved to enable fine-grained entity-centric search. First, the classic IR problem of retrieving *relevant* documents (those that contain at least part of the information required to satisfy the user's information needs) needs to be solved. However, given the structured nature of the problem, as well as investigators' clustering and aggregation needs, document retrieval is not adequate by itself. A principled approach is required to impose domain-specific structure on the initial corpus of unstructured HTML pages.

Due to the specialized nature of the HT domain, numerous technical requirements must be fulfilled by such an approach. First, HT webpages tend to be extremely heterogeneous, and the collection of Web domains from which these webpages are obtained is diverse, exhibiting a *mesokurtic* (i.e. long-tail) distribution. Any system that is tailored to perform well on a few Web domains runs into the problem

• *M. Kejriwal and P. Szekely are affiliated with the Information Sciences Institute, USC Viterbi School of Engineering.*
E-mails: {kejriwal, pszekely}@isi.edu
Address: 4676 Admiralty Way, Ste. 1001, Marina Del Rey, CA 90292

1. HT rings, sometimes posing under the guise of spas and massage parlors.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TBDATA.2017.2763164, IEEE Transactions on Big Data

TABLE 1
Text fragments scraped from real-world human trafficking webpages, with relevant extractions (in bold). Section 4.1 describes the schema.

Italian 19 hello guys....My name is **charlotte** , New to town from **kansas**
[GORGOUS **BLONDE** beauty] ? FROM **Florida** ? (Petite) ? [CURVy]?
NO DISAPPOINTMENTS. 34C..**Brazilian,ITALIAN** beauty....
Hey gentleman im **Newyork** and i'm looking for generous
Hi guy's this is sexy **newyork** . & ready to party.
AVAILABLE NOW! ?? - **(1 two 1) six 5 six - 0 9 one 2 - 21**

of generalizing properly to other Web domains, even for relatively routine tasks such as text scraping. Second, the content is obfuscated so that key elements (e.g. phone numbers, age and price) cannot be searched for, or extracted, using naive keyword-based matching techniques. Technically, this is because the *language model* employed in the text is different from traditional natural language models. For example, the text fragment *Italian 19 hello guys...My name is charlotte, New to town from kansas* contains four key pieces of information (original location, name, nationality/ethnicity and age), which are difficult to automatically extract in an error-free manner for a large set of attributes. Some other examples are illustrated in Table 1.

Given Web domain heterogeneity, different webpages tend to employ different styles of writing, symbols and obfuscation (usually highly dependent on the individual being exploited), which hinders the acquisition of representative training data. Finally, extractions and inferences are necessarily fuzzy in that attributes are extracted with varying confidence levels.

Motivating Example: As a motivating example, suppose an investigator has the overarching goal of locating a human trafficking ring using only a phone number lead. For the sake of discussion, suppose the lead is the phone number[2] (121-656-0912) in the last row in Table 1. Given the phone number, represented in the normalized form above, a search engine on a *crawled Web corpus* potentially containing hundreds of millions of pages must not only locate the page(s) containing such phone numbers (such as the fragment in the last row in Table 1), but also locate webpages that are *indirectly connected* (in a well-defined manner described in Section 4.2.1) to the pages directly containing that phone number. Furthermore, to infer useful information, the investigator also wants to *aggregate* information e.g., the union of all *physical addresses* and *email addresses* contained in the ads that are part of such a 'cluster'. More advanced aggregations include averaging over ages and prices in the ads in the cluster e.g., to detect underage and/or high-end activity.

Our described system is able to directly address both problems, although the focus in this article is on search and representation, not aggregations. First, by constructing a relatively normalized knowledge graph from the raw corpus using *information extraction* tools, we allow the user to search for the phone number with high recall. We also model phone clusters through algorithms like random walk and connected components clustering. Furthermore, using the knowledge graph, we can also retrieve other attributes such as physical addresses and display them to a user on a GUI. This makes both search and exploration intuitive. Our system is already being used for such investigative purposes by over 200 law enforcement agencies in the US.

In contrast, for various technical reasons, such searches cannot always be carried out live on the Web (Section 3) using search engines like Google. To robustly solve problems such as the one outlined above, a good system must not only scale to large crawled corpora, possibly containing both irrelevant pages as well as pages describing the same underlying entity, but be robust to numerous sources of error, including information extraction and normalization problems. To the best of our knowledge, this is the first entity-centric search system to have addressed all of these challenges in a heretofore *computationally under-studied* domain (Section 2) while still yielding demonstrably useful results to investigative and domain experts (Section 5).

As further motivation, we note that, while the focus in this article is on human trafficking, the techniques and lessons learned are generic enough to be applicable to any domain that exhibits similar domain characteristics. The broad set of challenges that are approach is able to robustly address is described in Section 3.

Overview of Approach: We present a knowledge discovery approach that uses *query-centric knowledge graph construction and search* to robustly handle questions that are of particular interest to human trafficking investigators (Section 4). Because the needs of investigators, and human trafficking content, quickly evolve, our approach is designed to accommodate flexibility in the schema and the extraction technology used to construct the knowledge graph.

Figure 1 illustrates the overall architecture, with two high-level components. The first *offline* component is *knowledge graph construction* (KGC) [6], [7], which takes a crawled Web corpus as input and structures it into a semi-structured knowledge graph that is stored and indexed in a NoSQL database [8]. We outline the core elements of KGC in Section 4.2. The second *online* component, which constitutes the primary technical innovations in this paper, implements real-time *entity-centric* information retrieval [5]. This component allows users to express their (originally, natural language) needs in an intuitive but unambiguous query language (Section 4.3), and processes those requests using *semantic* query conversion and execution strategies (Section 4.4).

We implement our approach using open-source technology, and demonstrate promising empirical performance on real-world human trafficking datasets that were crawled from the Web (Section 5). We also provide posthoc error analyses, indicating interesting avenues for future research in the HT domain.

Contributions: We describe our major contributions as follows: (1) We present a novel approach for structured entity-centric knowledge discovery on irregular domains such as human trafficking (HT) that have massive potential for social good. Components of the approach reflect real-world investigative needs of flexibility and efficiency. (2) We present a real-time entity-centric query prototype that accepts as input a constructed, extremely noisy knowledge graph (KG), and uses *semantic execution plans* to formulate

2. We have changed the particulars of the actual phone number so that it is non-identifying. Note that the '21' at the end of the fragment is an age, not part of the phone number itself.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TBDATA.2017.2763164, IEEE Transactions on Big Data

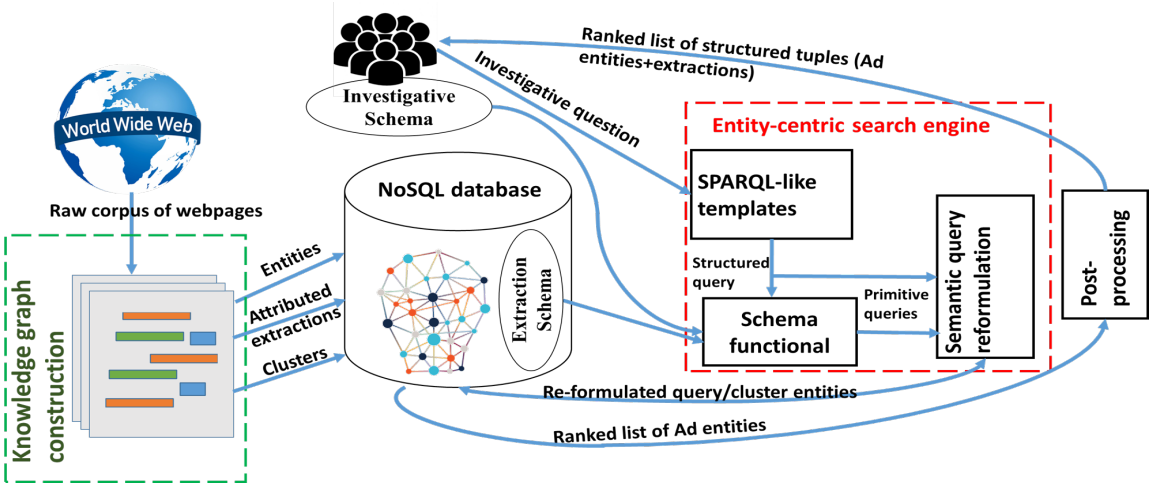

Fig. 1. A knowledge graph-based approach to entity-centric search in specialized domains like human trafficking. The dashed green box (knowledge graph construction) is executed offline, while the dashed red box (entity-centric search) is an online component that encapsulates the technical innovations in this work.

robust queries on the KG. (3) We implement and evaluate the query prototype using open-source software on a real-world HT corpus containing more than 90,000 recently crawled webpages manually annotated by experts, on a set of (externally formulated) investigative questions. The key principles of the prototype, and also the overall approach in Figure 1, have been independently successfully replicated by another team in this space. We verify scalability by executing the prototype on a multi-TB corpus.

2 Related Work

Investigative search has witnessed considerable research since the advent of the Web, and the Big Data movement, sometimes under a slightly different terminology (e.g., *exploratory* search) [9]. The problem we consider in this paper is similar, in that a certain class of domain experts (investigators) are interested in *structured* knowledge discovery over a noisy Web corpus, and is inspired by advances in a variety of fields, broadly classed into *knowledge graph construction* (KGC) [6], [7], and *entity-centric search* [10], [11].

2.1 Knowledge Graph Construction (KGC)

KGC draws on advances from a number of different research areas, including information *extraction* [12], information *integration* [13], and inferential tasks such as entity resolution [14]. Good examples of *architectures* that implement KGC principles are Domain-specific Insight Graphs (DIG) and DeepDive [4], [15]. Rather than attempt comprehensive coverage of either information integration or extraction, we provide coverage on the works that are most closely related to the work herein below.

Information integration (II). Information integration, also commonly known as data integration, is generally defined as the problem of providing a unified query interface over multiple data sources [16]. Information integration has been a research subject for at least 40 years [17]; recent developments and foundational principles are synthesized in the book by Doan et al. [18]. The problem has gained importance with the advent of Big Data [19], in domains ranging from enterprise to computational biology and Web search [20], [21], [22], [23]. Despite the enormous progress in information integration over the last few decades, there are some key challenges in the HT domain that preclude a direct adaptation of many existing techniques (Section 3).

Information extraction (IE). Information extraction is a core component of any information integration pipeline over Web corpora, as the unstructured webpages must first be structured in order for fine-grained queries to be executed over them. With the initial advent of the Web, *wrapper induction* systems had proved successful for several IE domains [24]. State-of-the-art work in the early 2000s (e.g. STALKER [25]) used machine learning methods for the wrapper induction problem [26]. Such methods were inherently data-driven, and were less brittle than rule-based wrapper architectures. IE systems have continued to evolve since then; Chang et al. provide a comparative survey of many of the leading IE techniques along three dimensions (*task domain*, *degree of automation* and the actual *techniques used*) [27]. A key finding of the survey is the dependence of techniques on the actual input format. For example, while unsupervised and semi-supervised methods are well-suited for template pages, regular expressions and supervised approaches tend to be more robust for non-template pages. A consequent problem arising from such diverse methodologies is evaluating precision and recall in a consistent way [27]. Keeping in line with these findings, we rely on a hybrid battery of extractors to accommodate the challenges (see Section 3) of our problem domain.

More recently, *OpenIE* has become a popular topic of research, owing to the need for IE techniques that do not rely on pre-specified vocabularies [28], [29]. In a preliminary version of the system, we tried state-of-the-art versions of OpenIE, including both old and new versions of the system proposed by [28]. Even when relevant extractions were obtained from the corpus of webpages, the precision and recall was judged to be too low to be useful. This largely motivated our earlier research on *focused* knowledge graph construction for illicit domains, albeit only for *keyword* queries that were easily amenable to GUI integration [4].

2.2 Entity-centric Search (ECS)

Entity-centric search is a broad area of research and was defined by Dalvi et al. as creating a 'semantically rich aggregate view' of concept instances on the Web [5]. Entity-centric search has led to novel insights about the search process itself, two examples being search as an action broker [11], knowledge base *acceleration* and filtering [30], interactive search and visualization [31], and search tailored for the Semantic Web [10]. The last area is particularly relevant to the present work, as we describe below.

An early entity-centric search prototype that is similar to our own effort is SWSE, which first crawls Web data and converts it to the RDF (Resource Description Framework) data model [10]. The overarching principles of SWSE are similar to our own system, in that the engine is designed to be domain-specific and both database and IR technology are leveraged for representation and querying; however, SWSE is designed to be schema-independent and to support keyword-style queries. In contrast, the system herein accommodates *precise, information-rich* queries that cannot be expressed using keywords, and that are designed to support both factoid and analytical (i.e. involving aggregations and clustering) needs at scale. This also distinguishes our approach from other similar research that fuses Semantic Web research with IR research on tasks such as *ad-hoc object retrieval* (AOR) [32], [33]. Despite the differences, important elements of our query prototype are inspired by the success demonstrated by these systems using *hybrid* (instead of purely structured or unstructured) search techniques for entity-centric search tasks.

In more recent years, entity-centric search has been used in commercial search engines like Google. For example, keyword searches like 'Albert Einstein' are now treated by Google both as keyword and as entity-centric search. For facilitating the latter, Google uses its underlying *Google Knowledge Graph* technology, based on a proprietary version of the (previously open-source) Freebase knowledge base [34], [35]. Our approach is similar in that it is also knowledge graph-centric. However, there are some important differences. First, we directly process Web corpora to construct a knowledge graph from both text and raw HTML rather than assume *a priori* availability of a sufficiently curated knowledge graph. Manual curation in a domain like human trafficking is almost impossible, meaning that our knowledge base is much noisier than even Freebase (and by reasonable extension, the Google Knowledge Graph) [34]. Second, we handle complex queries like clustering that current public-facing entity-centric search facilities in Google cannot handle. Most importantly, our domain of interest (human trafficking) exhibits a *long-tail* distribution that is uncharacteristic of the Google Knowledge Graph, which largely contains world knowledge from a handful of sources such as Wikipedia and Wordnet [36], [37].

Another related branch of research is *question answering*; however, unlike *question answering* systems [38], our approach is specifically optimized for investigative needs which allows us to restrain the scope of the questions and express them as *controlled-schema* queries in a language amenable to NoSQL executions. By controlled-schema, we mean that the attributes that can be queried are defined upfront, but the actual values retrieved by a search system are largely open-world i.e. do not obey strong normalization or format constraints. As is common in IR, the utility of such an answer is defined in terms of *relevance*, rather than 'correctness' in terms of database semantics. The queries also enable users to express aggregations, and retrieve clusters. In that sense, our system is more similar to structured query (e.g. SQL) prototypes on unstructured data [39]; however, the knowledge graph constructed by our system is largely semi-structured, containing a mixture of textual, numerical, pseudo-numerical and structured fields.

2.3 Human Trafficking

One of the most important aspects that separate this work from prior work is its focus on a *non-traditional* domain that has an outsize presence on the Web, and by some estimates is a multi-billion dollar industry, but due to technical and social reasons, has largely been ignored by the computational, knowledge management and IR research communities till quite recently [40], [41]. A notable exception in the knowledge graph construction domain is the *keyword DIG* (Domain-specific Insight Graphs) system [4]. Similar to DeepDive, keyword DIG implements KGC components, in addition to a GUI, and was evaluated on human trafficking data. The system described herein is implemented within the DIG architectural framework for repository continuity[3], but draws on a different set of components and supports much more than keyword queries. In particular, it allows users to intelligently query a constructed knowledge graph in expressive ways that cannot be handled (even in principle) by the keyword DIG. Finally, although the research described herein is specifically designed to investigate and combat human trafficking, we believe that the core elements of the overall problem and solution can be extended to other domains (e.g. from the Dark Web [42]) that are highly heterogeneous, dynamic and that deliberately obfuscate key information. Very recently, for example, the described system was extended to answering expressive queries in the securities fraud domain.

3 Challenges

Fulfilling investigators' entity-centric needs in domains such as human trafficking is tantamount to addressing a number of challenges that are very prevalent in such domains [3]:

Non-traditional domain: HT, and several domains like it, are largely characterized by illicit, organized activity, and have not been as extensively researched as traditional domains (e.g. enterprise). Directly adapting existing techniques from these domains is problematic, along with using external knowledge bases like Wikipedia [36], since the entities of interest (escorts and human trafficking victims) are not described in such knowledge bases. For example, we could not directly use tools like stemmers and tokenizers from standard NLP packages like NLTK [43] because the corpus contains many non-dictionary words and employs advanced obfuscation techniques (Table 1). The problem

3. For this reason, we will equivalently refer to the current system as *DIG* in the rest of the paper, using phrases like *the current DIG* and *keyword DIG* only where ambiguity arises.

is made much worse by the long-tail nature of the HT domain (see Appendix), since one cannot tune an algorithm for webpages from a small number of root URLs such as backpage.com.

Scale and irrelevance: The scale of the task, and the size of the corpus precluded us from using many serial algorithms that have a high memory imprint and long running times. Many of our most expensive tasks had to be run on Apache Spark [44]; furthermore, because we were not given an annotated ground-truth, and the corpora had many irrelevant webpages, we had to execute our core algorithms several times. Scale and irrelevance both proved to be key engineering challenges to be overcome.

Missing values and noise: In many cases, each page is typically missing information (e.g. hair color) that we would like to extract and use in our queries. However, it was unknown *a priori* which pages and Web domains were missing values for which attributes. It was often the case that extractors would get confused, and extract noisy values for attributes that were either missing or well-obfuscated. These observations strongly motivated the design of both extraction and query execution technology (Section 4).

Information obfuscation: A recurring challenge is *information obfuscation*, which includes obscure language models, excessive use of punctuations and special characters, presence of extraneous, hard-to-filter data (e.g. advertisements) in Web pages, irrelevant pages, lack of representative examples for (supervised) extractors, data skew and heterogeneity. Many pages exhibit more than one problem; a sampling of some pages in the corpus revealed that obfuscation was the norm rather than the exception. A concrete obfuscation example is an individual stating her phone number (+1-217-453-0004) as '2**1–7**4-5-3*-**-*oh-oh-oh***4'. A successful system would not only recover the original number from this text, but also infer (based on other information in the page) that it is a number from the United States (+1).

In the research literature, obfuscation is dealt with only in limited *syntactic* settings (e.g. privacy-preserving data mining or malicious code detection [45], [46]); we are not aware of any work that has addressed human-centric *semantic* obfuscation of the kind that predominates in HT.

Complex query types: In the HT domain, investigators are interested in several kinds of (subsequently formalized) queries. The simplest queries, in principle, are *point fact* (or factoid) queries that can be handled by key-value data stores like Elasticsearch assuming robust extractions, indexing and similarity computations. More complex aggregation and cluster queries over noisy data are far less straightforward. Even point fact queries turn out to be difficult when we consider both the noise and the variability in the knowledge graph construction.

Preclusion of live Web search: One could very well question if there is not some way to pose the question as a Google search query, at least to solve the initial problem of locating relevant Web pages, followed by online execution of extraction technology. There are two problems with such a thesis, even assuming efficiency of online extractions. First, investigators are often interested, not just in what escort ads are *presently* on the Web, but also in escort ads published in the past (and that may have been taken down subsequently). Many ads are published for a few days only. Building cases against trafficking requires establishing a pattern of behavior over time, so it is important to retain pages that may not be available at the moment of search. In summary, such searches are especially vital for the purposes of *evidence gathering*, which is directly relevant to the motivation of using the system for evidence-based social good. The second problem is that it is not obvious how keyword-based search engines can solve complex query types such as clustering and aggregation in a purely online fashion, using only a keyword index. Finally, traditional Web search principles, which rely on hyperlinks for the robust functioning of ranking algorithms like PageRank, do not hold in the HT domain, where relevant hyperlinks present in the HTML of an escort ad tend to be sparse. Most links are inserted by publishers to promote other content and cause traditional search crawlers to behave undesirably.

4 Approach

We attempt a solution to the investigative search problem by first constructing a *knowledge graph* from the raw corpus of HTML pages, using a variety of information extraction and clustering modules based on an underlying *investigative schema* (Section 4.1) [12]. We define a knowledge graph (KG) as a *directed, labeled, multi-relational* graph where nodes represent entities, attribute values or entity clusters (*latent* entities), and edges represent relationships.

Because of the imperfection of extraction and clustering modules, the constructed KG is typically very noisy (a non-trivial subset of extracted attribute values is incorrect), large-scale (containing tens of millions of nodes and edges), and is semi-structured (many attribute values are missing, multi-valued and even textual). Given such a knowledge graph, one solution to robustly answering entity-centric questions over the original corpus is to first pose the question in terms of an intuitive but unambiguous query language (Section 4.3), using terms from the investigative schema, and then query the constructed knowledge graph in a semantically well-defined manner (Section 4.4). For performance and quality reasons, some aspects of the system involve high degrees of automation, while others are manually driven.

4.1 Investigative Schema

To facilitate precise search capabilities and preclude errors in solutions to difficult problems such as word sense disambiguation [47], we develop a controlled vocabulary of terms denoted herein as an *investigative schema* $\mathcal{I}$, represented as a labeled, directed, *acyclic* graph, with *subclassOf*, *attributeOf*, *hasValue* and *memberOf* edges. Figure 2 illustrates the investigative schema used in the current prototype. This schema was defined collaboratively with actual investigators and domain experts, and contains four *primary* classes[4], in addition to a set of primitively typed *semantic attributes*[5].

We permit a semantic attribute to be multi-valued (a bag), but each value must have the primitive type of the attribute. Among the four classes, *EscortAd* and *MassageParlorAd* are referred to as *base classes*, themselves sub-classes

4. Instances of these classes are considered either *entities* or *latent* entities (for *Vendor*) in our domain of discourse.

5. That is, the type is either *number*, *date* or *string*.

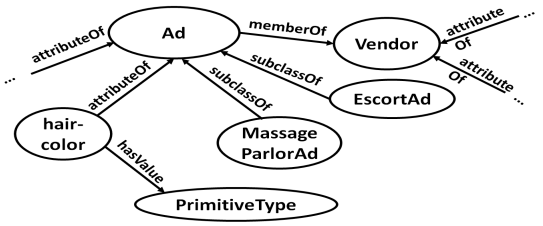

Fig. 2. A fragment of the current investigative schema $\mathcal{I}$.

of the superclass *Ad*; *Vendor* is denoted as a *cluster class*, of which an instance will always be a set of *Ad* instances (hence, the *memberOf* relation). Currently, *EscortAd* and *MassageParlorAd* share semantic attributes and we do not distinguish between them in our data, although in the post-processing module in Figure 1, an implemented supervised machine learning classifier can be used to offer a probabilistic classification.

We define a single set of semantic attributes for the *Ad* subclasses, namely *age, text-content, email, eye-color, hair-color, height, location, name, nationality/ethnicity, phone, posting-date, price, review-id, service, social-media-id, street-address, title, weight* and *multiple-providers*[6]. The *Vendor* cluster class has two attributes: *seed-phone* and *seed-email*, which are the respective *unions* of the phones and emails corresponding to the ads that are members of a *Vendor* cluster instance. The rationale for these attributes is that (1) phones and emails extracted from ads were used for inferring the clusters, as subsequently described, and (2) investigators prefer querying for cluster instances using phone and email facets.

More technically, the investigative schema is a constrained version of a *shallow ontology*, used often in the Semantic Web for schema-rich knowledge graphs such as DBpedia and GeoNames [48], [49]. The ontology defines a domain of discourse, and is critical for precisely capturing domain assumptions. More importantly, the schema defined above is not only simple but designed to be extensible: usually, real-world investigators want flexibility in the available set of semantic attributes e.g. we are looking to add semantic attributes such as *tattoo* and *drug use* to the schema.

4.2 Knowledge Graph Construction

An important first step in achieving the goals illustrated in Figure 1 is *knowledge graph construction* (KGC). KGC is an area of research in its own right [6], [7]; we only describe the important elements herein. The *online* component of the system (the entity-centric query engine) that is the technical focus of this work, is *agnostic* to the constructed knowledge graph (KG), although the overall quality of the answers will (in the general case) depend on the KG. To abstract implementation-specific details, we assume the availability of a battery of *attributed information extractors* for each semantic attribute in $\mathcal{I}$:

Definition 4.1. An *attributed information extractor* (a-IE) X is a 2-element tuple $< f, \mathcal{M} >$, where f, denoted as an information extractor, is a (many-many) mapping from an input string to a set of primitively-typed values, and $\mathcal{M}$ is a non-empty *extraction metadata dictionary* of key-value pairs.

In the literature, implementations of f include wrappers, text scrapers (for extracting text content from raw HTML pages), regular expressions, dictionaries, entity sets and external knowledge bases like GeoNames [49], [12]. While wrappers and text scrapers take the raw HTML content of a webpage (represented as a single long string) as input, other IEs take the output(s) of text scrapers as their inputs. We provide an overview of the extraction technology in Table 2.

The metadata of an a-IE typically includes a succinct representation of expert confidence in f. One compulsory attribute of an a-IE is a *semantic type*, which enables us to *type* each output of f in terms of the attributes in the investigative schema $\mathcal{I}$. A second compulsory attribute that is extremely important for non-technical domain experts navigating the knowledge graph is *provenance*. The simplest example of provenance is the original URL (and in many cases, the *cached* webpage, which may not be online at the time of search i.e. see Section 3)

Definition 4.2. Given an a-IE $X =< f, \mathcal{M} >$ and an investigative schema $\mathcal{I}$, a *semantic type* is a function mapping the output of f to a semantic attribute in $\mathcal{I}$. The output of f is referred to as an *extracted semantic type*.

Example 4.1. Consider a Conditional Random Field (CRF)-based f for the *hair-color* semantic attribute. f would take text as input and output a (possibly empty) set of hair colors (*extracted semantic types*). Suppose also that, during training, the CRF parameters can be optimized to deliver either high expected precision or high expected recall. Rather than choosing between the two, the KGC could have two a-IEs e.g. *hair_color_high_precision*=$< f, \mathcal{M} = \{[isHighPrecision, True], [semanticType, hair\text{-}color]\} >$; similarly, *hair_color_high_recall* is defined, except with a differently trained f, and with *isHighRecall* rather than *isHighPrecision*. Intuitively, the metadata allow us to use the a-IEs in a variety of ways in the query reformulation algorithm (Section 4.4). Note that the semantic types permit us to define KGC as a black box process precisely because they explicitly connect elements between the *actual* schema of the KG (the implementation-specific a-IEs) and $\mathcal{I}$.

The *Inferlink extractor* is a *wrapper-based* semi-supervised tool that takes a collection of HTML pages from a top-level Web domain as input [50], [26]. Wrappers are common in Web-based information extraction systems. Specific details on Inferlink are provided in an appendix.

The other algorithms described in Table 2 are widely used in the *natural language* IE community; we provide relevant references that guided the design of those algorithms. We note that, except for the Inferlink tool, the semantic types of extractions output by the other algorithms are pre-determined. The readability text extractor (RTE) is an off-the-shelf text scraper that takes HTML as input and outputs a text sequence. Its hyperparameters can be tuned

6. A Boolean flag indicating whether multiple people will be offering the service.

TABLE 2
Overview of the knowledge graph construction (KGC) extraction technology ontologically guided by the investigative schema in Section 4.1.

Extraction Technology	Investigative Attributes	Level of Effort/Engineering	Relevant References
Inferlink (Template clustering+wrapper-induced rules)	All 'structured' attributes	10-20 minutes trained expertise per Web domain (e.g. backpage.com)	[26], [24], [50]
Readability Text Extractor	Text	Hyperparameter tuning	[51]
Conditional Random Fields	Eye-color, Hair-color	Labeled data, feature engineering, hyperparameter tuning	[52], [53]
Dictionaries and Entity Sets+contextual classification using word embedding features	Name, Location (City, State, Country), Nationality, Ethnicity, Service	Procuring dictionaries, large text corpus, small set of labeled annotations/attribute	[54], [55], [56], [57]
Regular Expressions and Custom Programs	Email, Height, Phone, Posting-date, Price, Review-id, Social-media-id, Title, Weight	Programming regular expressions	[58], [59]
NLP rule	Street-address	Crafting the rule	[60], [43]

to yield either high recall or high precision[7]. The remaining extractors individually process the high-precision and high-recall text output by RTE and yield a set of extractions each with the corresponding metadata. Each extractor requires a different level of manual effort. For example, the NLP rule-based extractor is a hand-crafted rule that uses NLP features (POS tags etc.) to extract street addresses from the text. Ideally, if large quantities of training data had been available, a Conditional Random Field (CRF) could have been used, such as for eye-color and hair-color attributes. Unfortunately, street addresses are extremely sparse and irregular in human trafficking data. Off-the-shelf street address extractors also yielded extremely noisy performance.

In the current prototype, we treat KGC as an independent 'black box' that takes as input a raw corpus and yields a semi-structured knowledge graph; in the evaluations, we quantitatively compare the performance of two independently constructed knowledge graphs using both DIG, and also the DeepDive system [4], [15]. At the time of writing, the extractors listed in Table 2 are continuously being maintained and updated [61]. It is important to note that the subsequently described search engine directly uses the IE metadata to reformulate queries and make ranking decisions, allowing specific implementation details to be abstracted in that step. For instance, the metadata abstraction allowed the search to be conducted over knowledge graphs (constructed using DIG and DeepDive respectively) with two completely different extraction schemas.

4.2.1 Cluster Inference

Cluster inference is the process of deriving the latent entities (instances of the *Vendor* class in Figure 2) in the knowledge graph. The set of real-world cues that can be used to cluster human trafficking entities is a highly contentious issue, and includes text similarity, use of Unicode motifs, and phone and email *co-occurrence* data. While we use the last cue, any clustering algorithm can potentially be used to identify vendor instances. This also explains the rationale behind the cluster attributes (*seed-phone* and *seed-email*); given a 'seed' phone or email, we define the *query-centric* problem of retrieving the 'vendor' instance associated with that seed as the set of ad entities satisfying the following conditions: the seed was explicitly extracted from at least one of the ads in the set, and there is a *co-occurrence path* of phones and emails between any pair of ads in the set[8]. In other words, every vendor instance represents a *connected component* of ad instances, with an edge defined between two ads if they share phones or emails.

Naively using connected components as clusters is problematic for a number of reasons. First, there are some extremely common but irrelevant phone numbers that show up in a non-trivial number of pages. An example would be a customer service number from Verizon (or the ISP hosting the site), extracted from the bottom of the HTML page. Second, the noisy extractions output by the regular expression-based phone extractor are not independent. For example, a sequence of digits (such as a zipcode followed by a set of prices) is more likely to be mistaken for a phone (by the extractor) than an isolated age. Similar to the first problem, the second problem also leads to nodes in the constructed phone network that result in extremely large connected components (by serving as a 'bridge' between two connected components).

In a previous prototypical solution, we handled this problem by manually blacklisting nodes that had abnormally high degrees. In the current prototype, this step has been successfully automated by using a *random walk-based connected components* algorithm. Connections mediated by 'faulty' nodes are typically extremely weak owing to their high edge degrees, allowing us to control the formation of large connected components.

The result of information extraction and cluster inference is a *multi-relational directed, labeled graph*, denoted henceforth as the *knowledge graph* (KG), with two types of nodes: *Vendor* nodes and *Ad* (equivalently, *entity*) nodes. Every document that is processed by the extraction system is assigned a unique identifier and becomes an *entity* node in the KG.

4.2.2 Knowledge Graph Indexing

For real-time execution of complex queries, the knowledge graph must be properly *indexed* in an appropriate NoSQL (i.e. key-value) document store [8]. Specifically, each ad is considered a document with multiple fields, with each

7. For the latter, the text may be much 'cleaner' but could potentially be missing useful sentences and paragraphs.

8. An example of such a set for the seed phone p_1 is a set containing three ads (say ad_1, ad_2, and ad_3) that respectively contain phones $\{p_1, p_2\}, \{p_2, p_3\}$, and $\{p_3, p_4\}$

TABLE 3
Four investigative query categories, with a natural language example/category. Identifying information has been replaced.

Category	Example
Point Fact	List all ads, with social-media-id, containing phone 123-456, optionally with location Key Biscayne, Florida occurring somewhere in the ad text or page.
Cluster Identification	List all ads connected via a shared phone number or email to the phone number 987654.
Cluster Facet	List all ads connected via a shared phone number or email to the phone number 987654 that feature a cuban escort, with location filtered on Florida.
Cluster Aggregate	Find average height of escorts associated directly or via shared phone/email links with phone number 123-456-7890

field bijectively mapped to an a-IE. Since not every ad is guaranteed to have an extracted semantic type (or may have multiple such extractions), the documents are semi-structured i.e. will not have values for many fields in the general case. This is because, both for purposes of achieving high recall during search as well as robust handling of unexpected extraction outcomes, values are not necessarily from a controlled vocabulary or closed set (e.g., a set of pre-determined hair colors, or a phone number guaranteed to have an eligible US phone format). Given specific 'analyzer' instructions (the most important of which is the tokenizer used for chunking strings into bags of tokens), suitable inverted indices can be constructed for each such field, with the document id (the entity node) serves as the link between multiple field indices. Given a query in its *Domain-specific Query Language* (DQL), a good NoSQL engine like Elasticsearch or MongoDB makes judicious use of these inverted indices for fast retrieval and ranking. Specific implementation details are discussed in Section 5.

4.3 Query Language

Although investigative questions are *originally* posed in natural language, their restricted scope enables them to be manually expressible in an intuitive structured syntax that *resembles* SPARQL, a graph-pattern matching language popular in the Semantic Web [62]. The goal of such a language is to enable domain experts to *unambiguously* express their investigative needs by limiting the vocabulary of the queries to terms from $\mathcal{I}$. An important desiderata of the language, in addition to being *expressive* enough, is *simplicity*: it must enable users to frame investigative queries *by example*. That is, using some *templates*, domain experts should be able to express questions in a variety of interesting categories in the syntax of the language. Four such query categories are defined below. We provide a natural language example question per category in Table 3. The machine readable versions of the questions in the SPARQL-like language is subsequently explored.

4.3.1 Point Fact Queries

A point fact query is the structured equivalent of a *factoid* question in traditional question answering systems [38]. The answer to a point fact question is an *Ad* entity ID (representing an underlying document), and the corresponding

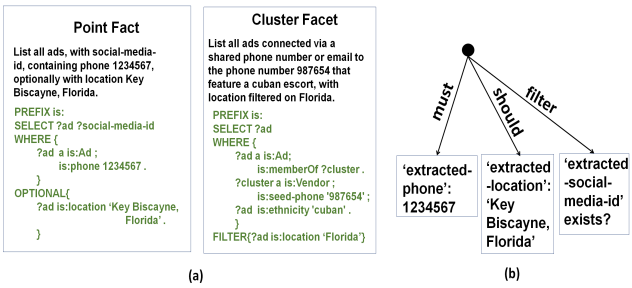

Fig. 3. (a) contains *point fact* and *cluster facet* query examples, with (temporary and request) *variables* pre-pended by '?'; (b) illustrates a *soft Boolean tree* (SBT) query.

values of any semantic attributes requested by the user. A point fact query may be thought of as an *entity-centric* query *faceted* on certain semantic attribute values, while requesting the values of certain other semantic attributes. Concept definitions and brief semantics are provided below, with a visual example (using the actual SPARQL-like syntax of the language [62]) in Figure 3 (a).

Definition 4.3. A *point fact facet* ρ is a 4-element tuple $< S, a_S, v, op >$ comprising a semantic class $S \in \mathcal{I}$, a semantic attribute a_S of S (or a subclass of S), a primitively typed literal value v and a Boolean 2-argument operator op.

Semantics. Given an entity e in the KG that is an instance of class S, ρ evaluates to True (or a non-zero score, for real-valued semantics) iff $e[a_S]$ exists and $op(e[a_S], v)$ is True, and evaluates to False (or a 0 score) otherwise.

Example 4.2. An example of a point fact facet is $< Ad, hair_color, `brunette', => $, represented more succinctly as *Ad[hair_color='brunette']*. Any entity that has the semantic attribute *hair_color* with corresponding value (i.e. extracted semantic type) *brunette*, would be retrieved with a non-zero score. The manner in which this score is computed e.g. using cosine similarity on tf-idf vectors obviously depends on the retrieval algorithms in the actual NoSQL database.

Definition 4.4. A point fact query $P =< C, C_O, C_F, R >$ is a 4-element tuple, where C, C_O and C_F (denoted as *mandatory*, *optional* and *filter* facet sets resp.) are mutually exclusive sets of point fact facets, and R is a *request* set of 2-element tuples of the form $< var, a_S >$, where var is denoted as a *request variable* and a_S (a semantic attribute in $\mathcal{I}$) is denoted as a *request attribute*.

Semantics. Considering a real-valued formulation, P will return 0 score for an entity e iff it returns 0 score for *any* facet in $C_F \cup C$, or there exists a request attribute in P that does not exist for e. For all other entities, the query will return a non-zero score, the specific computation of which will typically depend on the underlying retrieval system. For retrieved entity e, a request variable var will be *bound* to the values $e[a_S]$.

The reason for distinguishing between C and C_F above is that they are processed differently during query execution. Usually, scores of filter facets do not affect the final query score for a given entity e, as long as they have

non-zero scores. Furthermore, a good retrieval system that assigns real-valued scores to entities should be *monotonic* in C_O (i.e. the more optional conditions are satisfied, the higher the score, all else being the same).

4.3.2 Cluster Identification, Facet and Aggregate Queries

A cluster identification query requests a list of ad entities that belong to a cluster, identified by a cluster attribute (currently, *seed-phone* and *seed-email*). A cluster *facet* query is a generalization of both a point fact query and a cluster identification query, and may be expressed as a sequential combination[9] of both. In essence, it is a point fact query that is also allowed to facet on cluster attributes. In practice, this amounts to executing a point fact query on only a subset (i.e. the identified cluster) of ads in the KG.

The last query category is a *cluster aggregate* query, which further generalizes a cluster facet query by permitting a more expressive request component R. Given a finite set of aggregation functions, typically just *concat* (for text values), *min*, *max*, *mode* and *avg*, a cluster aggregate query is similar to a SQL query with a group-by clause. An involved definition of an aggregate query, omitted herein, is a pair $< P_{CF}, g >$, where P_{CF} is a cluster facet query, g is a semantic 'group-by' attribute, and request variables in $P_{CF}[R]$ are restrained to either g or an aggregation function applied to some other semantic attribute. Groupings and aggregations are handled by the post-processing module in Figure 1.

4.4 Query Execution

4.4.1 Query Reformulation

Although the extracted semantic types in the constructed KG have been semantically typed in terms of the semantic attributes in $\mathcal{I}$, directly executing a query posed in terms $\mathcal{I}$ is challenging. Not only is the semantic typing between $\mathcal{I}$ and the knowledge graph a-IEs a *many-one* mapping (see Ex. 4.1), but each a-IE offers its own tradeoff in terms of expected precision and recall, which complicates query reformulation.

Example 4.3. Consider a simple point fact query, with only one mandatory facet *Ad[hair_color='brunette']*, empty optional and filter facet sets and a single request variable (the *id* of the entity). Intuitively, when executing the query on a constructed KG containing both high-precision and high-recall a-IEs for *hair-color* (Ex. 4.1), we would want the score of an entity e to be more influenced by a match on its high-precision, rather than high-recall, extracted semantic types. If we had two mandatory facets on different semantic attributes, the retrieval system should systematically deal with multiple possibilities (e.g. when one facet matches on a high-recall extraction but the other matches on a high-precision extraction). More complex heuristics and external information sources like dictionaries are required for 'indirect' attribute values e.g. an ad may have extracted hair color 'brown' instead of 'brunette'.

9. In Figure 1, multiple query executions on the KG in response to a cluster facet (and also, cluster aggregate) query is represented by a bidirectional arrow; the 'final' output is always routed to the post-processing module.

Results in Section 5 provide some evidence that the effectiveness of approaches that do not take the noise and variability in the constructed KG into account are severely compromised in their ability to achieve high recall. For good performance, we approach query reformulation from a *hybrid* perspective, with the assumption that text, structured extractions and expert-suggested dictionaries and *external* knowledge sources are all necessary, depending both on the quality and semantic types of a-IEs. For example, in addition to the color dictionary suggested by the previous example, we also use the GeoNames knowledge base for obtaining and normalizing *location* extractions [49].

One must also take the quality and metadata of the a-IEs into account, suggested by the dictionary $\mathcal{M}$ (Definition 4.1) For example, we have prior evidence that our phone extractions are of high quality (*isHighPrecision* and *isHighRecall* are both *True* in the metadata of the phone a-IE), but our social media id extractions are not, mainly due to creative obfuscations of social media ids in the data. We also know that phones serve the role of *pseudo-identifiers* in the HT domain, and are highly discriminative of an entity when correctly extracted. Intuitively, we would *trust* a match on the phone much more than a match on the social media id attribute for ranking purposes; good query reformulation needs to *quantify* the value of this trust in a reformulated query.

Note that the reformulated query must be executable in the NoSQL database employed for storage, ranking and retrieval. With the KG indexing scheme outlined earlier, executing a simple point fact facet is straightforward in a key-value database with an inverted index on the key: one would use the a-IE(s) with semantic type a_S as the key(s), and the primitively typed literal v as the value. Common binary operators like $=, <$ etc. are also supported in addition to optional metadata like *boost* values that can be used to quantify trust or expected importance of the field for retrieval purposes.

To avoid implementation-specific details in the treatment below, we assume that each point fact facet can be mapped as one or more *parameterized* key-value queries, denoted as a *primitive query*, with the parameter vector depending on the specific NoSQL implementation, the metadata of the a-IE, as well as any other expert-suggested knowledge pertaining to a specific semantic type. Given our trust in phone extractions, for example, the boost parameter could be set to a high value when mapping a facet involving the phone semantic attribute to a primitive query. Similarly, one could include a link in the primitive query to a dictionary or API for hair color keyword expansion. Given a primitive query, the NoSQL database executes the query by assigning (whether implicitly[10] or explicitly) a score to every entity e in the indexed KG.

With the notion of a primitive query, we propose the concept of a *schema functional*. A common-use mathematical definition of a functional is a mapping from the space of functions to the space of real numbers. The schema functional defined below is similar, if we interpret point fact facets and primitive queries as *functions* from KG entities to

10. Efficient IR often assigns a 0 (or near-0) score to many entities for a query, using only inverted index information.

the real-line subset $[0, 1]$, and is the first step in reformulating a query posed in the abstract query language definitions in Section 4.3, using terms from $\mathcal{I}$. We assume point fact queries in the discussion below, and subsequently extend the treatment to the other query categories.

Definition 4.5. A *schema functional* $\mathcal{S}$ is a functional that takes as input a point fact facet and returns a set of parameterized primitive queries.

Example 4.4. Consider again the point fact facet *Ad[hair_color='brunette']* and the two a-IEs in Ex. 4.1, namely *hair_color_high_precision* and *hair_color_high_recall*, with semantic type *hair-color*. $\mathcal{S}$ could map the facet to the following set of (informally expressed) primitive queries: {<*hair_color_high_precision:'brunette', boost=3.0*>, <*hair_color_high_recall:'brunette', boost=0.5*>}, with *boost* (for rewarding an exact match on the specified query word by 'boosting' the score in an implementation-dependent way) comprising a user-defined parameter in $\mathcal{S}$. A third primitive query could also include an API for keyword-expanding 'brunette'.

At present, the schema functional and primitive query parameters like boost values are manually defined. It is theoretically possible to learn parameter values using a *learning-to-rank* mechanism; however, training data (comprising queries and answers) is extremely difficult to acquire in a poorly understood domain like HT. When evaluating the prototype, we experimented with multiple knowledge graph extraction schemas; in general, our implementation permits modifying and prototyping competing schema functionals within the space of only a few minutes.

While the schema functional defines appropriate facet mappings between the extracted semantic types in the constructed KG, and the semantic attributes in $\mathcal{I}$, as primitive query functions, it does not provide a solution to the problem of *composing* these primitive queries into a single query. Such compositions are necessary both when a functional returns a set with more than one primitive query, as in the example above, and also with multiple facets, possibly in different facet sets (e.g. a mandatory vs. optional point fact facet).

A *recursive* representation for combining primitive queries is a *soft Boolean tree* (SBT) query with labeled branches, where the label must be from the set {*must*, *must not*, *should*, *filter*}. Figure 3(b) illustrates an example. Each leaf node represents a primitive query, while a non-leaf node is an SBT. Given an entity e and an SBT query q, the score of e when executing q may be computed using Algorithm 1. The algorithm computes scores in a bottom-up fashion by using *soft Boolean* rules (see below), and returns the final score as the score of the root.

The rationale behind the SBT representation, as well as relevance score computation in Algorithm 1, is that it enables representing a *soft* version of arbitrary *Boolean* conditions, including negations. Intuitively, a must branch encodes a *conjunction*, a should branch encodes a *disjunction*, and *must not* encodes a *conjunction of negations*. Because the score of a 'negation' is not well-defined, *must not* nodes have semantics that resemble filters. The SBT also has direct support for representing an arbitrary filter facet set using the *filter* branch.

Algorithm 1 Scoring an entity given a soft Boolean query

Input: Entity e and soft Boolean query P'
Output: Relevance score $score(e, P')$ of e given P'

1) Assign weight to each *leaf* node by scoring e (e.g., using tf-idf) for each primitive query in P'
2) Iteratively (going bottom-up) assign $weight$ to each *non-leaf* node using the first rule that is satisfied:
   a) **IF** a *must not* child node **EXISTS** and has a non-zero score **OR** a *filter* child node **EXISTS** and has a 0 score **OR** a *must* child node **EXISTS** and has a 0 score **OR** (there are no *must* children **AND** at least one *should* child node **EXISTS AND ALL** *should* children have 0 scores) **THEN** $weight := 0.0$
   b) **IF** there are no *must* children **AND** there are no *should* children **THEN** $weight := 1.0$
   c) $weight :=$ the average over the weights of all *must* and *should* children.
3) **return** $score(e, P')$ as score assigned to the root node

In practice, an SBT has an equivalent representation in the Domain-specific Query Language (DQL) of the open-source Elasticsearch database (including the scoring function), which also efficiently formulates and executes a query plan using a given set of inverted indices, and Lucene at its backend. Formulating a reasonable *physical* query execution plan can be offloaded onto an Elasticsearch server without having to engineer it from scratch.

The treatment above describes the representation of the SBT query, but the question still remains on how one could derive such a query from a point fact query defined earlier.

Definition 4.6. Given a schema functional $\mathcal{S}$ and a point fact query P, a *semantic strategy* is a mapping from P to an SBT query P', with the primitive queries in P' generated using $\mathcal{S}$.

We denote these strategies as semantic, not syntactic, because the semantics of P' may be designed to approximate, not equate, the semantics of P. The intuition is that, because of the noise in the KG, the approximate semantics may be better than the original semantics in satisfying user needs (the *intended* semantics). We describe three currently deployed semantic strategies below:

(1) **Semantics-preserving strategy**: Although we do not prove it here, it is possible to express a point fact query P as an SBT query P' such that, for a given entity e, $score(e, P') > 0 \Leftrightarrow score(e, P) > 0$. In other words, the strategy is *semantics-preserving* (in the sense of query syntax); in particular, the truth of filter and mandatory facets, as well as existence of request attributes, are strictly interpreted by the strategy. Intuitively, C, C_O and C_F facets are mapped to *must*, *should* and *filter* nodes respectively, while a nested *must not* node is employed if the *op* argument in a facet is an inequality. The SBT in Figure 3 (b) is obtained by applying

this strategy to the point fact query in (a).
(2) **Optional-semantics strategy:** As mentioned earlier, the knowledge graph is typically noisy; many attributes could be missing, are incorrectly extracted, or are multi-valued, with only some of the values being correct. For such graphs, semantics-preserving strategies tend to be brittle. A robust alternate strategy is to *first* form a *new* point fact query with different semantics by setting $C^{new} = C_F^{new} = \{\}$, $C_O^{new} = C \cup C_F \cup C_O$, and *then* apply the semantics-preserving query to this *less constrained* new query.
(3) **Keyword strategy:** This strategy treats each entity as a text-only document, and collates the literal values in the facets (in all the facet sets) as a set of keywords. We search these keywords against the union of entity text attributes.

Given a point fact query P, we apply the semantic strategies to P and combine the three SBT trees into a composite *soft disjunctive query* (SDQ) that can be executed by the database. The score assigned to an entity by an SDQ query is simply the *maximum* (i.e. the Lukasiewicz OR) of the scores assigned by the three SBT queries. Taking the maximum enables us to gracefully combine the benefits of the three strategies above, since in practice, a more constrained query strategy yields fewer non-zero scores, but the non-zero scores are *higher* than the scores assigned by corresponding less constrained strategies.

For a cluster facet query, we compose and execute two SDQ queries in sequence. First, we use the cluster attribute to retrieve sets of ads (cluster instances); if the retrieval yields non-empty results, we use the retrieved ads as a filter and re-execute the cluster query as a point fact query[11]. If the retrieval yields empty results, we issue the re-interpreted query without an ad filter. We found this strategy to be extremely robust to the presence of missing phone and email extractions that were also rare in the dataset (and thus had no corresponding cluster).

4.4.2 Post-processing retrieved results

Recall that each query has a request component R. If R only contains an id attribute, post-processing is usually unnecessary and the ranked list of retrieved entity identifiers can be displayed to the user as the final answer. Given the many-one mapping between semantic types and a-IEs, a *specific* KGC-dependent protocol is required for binding non-id request variables in R to KG attribute values. DIG KGC, for example, supports both high_precision and high_recall versions of different information extractors, as we described earlier in Example 4.1. One protocol is to impose a *total order* on different a-IEs with the same semantic type e.g. values yielded by high-precision a-IEs (if they exist for a retrieved entity e) could be favored over high-recall a-IEs.

For aggregations, an additional protocol is needed to compute *soft* equivalents of aggregation functions like *min*, *max* and *mode*, by first executing the underlying cluster facet query, and then utilizing the relevance scores of the corresponding entities as weights. We leave a complete empirical analysis of this issue for future work as the issue of post-processing protocols is largely orthogonal to entity-centric *search*, namely the retrieval of a ranked list of relevant entities in response to a query. For aggregate queries, the goal is to retrieve a ranked list of entities that are relevant for computing the aggregation. We do not evaluate post-processing in this paper; the prototype permits the user to specify, or choose between implemented, protocols.

11. With the minor alteration that *seed_phone/email* are replaced with *phone/email*.

5 Evaluations

5.1 Methodology

Phases. Because of the nature of human trafficking data, as well as early uncertainty about KGC and query execution, evaluations were conducted in two phases as core components of the DARPA MEMEX program[12] under which the queries and corpora were released and manually annotated. With minor differences, both phases involved the same investigative schema and the same query categories, but different numbers of queries and different corpora. The first phase, denoted as the *dry run*, occurred in July 2016, while the second phase, denoted as the *final run* concluded in November 2016.

Questions and Corpora. Dry run evaluations were conducted on an exhaustively annotated corpus containing 4000 webpages, almost all of which were relevant to human trafficking, while the final run corpus contained 90,000 webpages (about 8 GB of raw uncompressed data, before KGC), a significant fraction of which were annotated, but which also contained non-human trafficking pages. These annotations were used for running external evaluation scripts; at no point during system development were they released to any of the contracted teams. Evaluations during the dry/final run involved 10/102, 2/50, 14/50 and 14/100 point fact, cluster id, cluster facet and cluster aggregate questions respectively. All questions were devised by a customer-facing company that was in direct contact with law enforcement and other potential users of the system at the time. System scalability was evaluated on larger corpora, described later.

Infrastructure and Implementation. We consider two KGC systems in this paper: *Lattice*, and the *DIG* KGC described earlier in Section 4.2. Although the Lattice[13] extractions are proprietary and specific to the dataset, it is known that the technology is underpinned by DeepDive [15]. DIG extraction technology is open-source and freely available [61], and was executed using Apache Spark [44]. We used Elasticsearch[14] v2.4.1 as our NoSQL database. For infrastructure, we used a server on Amazon Web Services for the dry run, whereas for the final run we used a proprietary cluster (with a total of 173 GB memory and 30 virtual cores), made available as part of the MEMEX program under which this work was funded. Although the corpora used in the experiments are sensitive and cannot be made public, the code, software and major sub-systems, including both the search engine as well as extraction technology implemented for knowledge graph construction [61], are open-source and accessible via the project page[15].

12. http://www.darpa.mil/program/memex
13. https://lattice.io/
14. https://www.elastic.co/downloads/past-releases/elasticsearch-2-4-1
15. http://usc-isi-i2.github.io/dig/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TBDATA.2017.2763164, IEEE Transactions on Big Data

Evaluation Protocol and Metrics. Three teams were contracted by the program for independently developing their own entity-centric search systems. We use Team/System 1 as the identifier for our team. While Teams 1 and 2 used a knowledge-graph centric approach, with Team 2 replicating the main components of Team 1 for the *final run*, Team 3 used a learning-to-rank approach with domain-specific features derived directly from the HTML pages, along with limited annotations from an off-the-shelf Named Entity Recognition system. Because the evaluation was organized as a two-phase challenge, complete annotations have not been released, and all three teams were subject to the same constraints and datasets, we use the other two systems as baselines and case studies for comparative analyses. We also report on posthoc error analyses conducted on site in the aftermath of the November phase.

KGC is evaluated using a *coverage* measure, since it is most pertinent to entity-centric search and can be evaluated without bias. Specifically, for each semantic attribute in the investigative schema, we record and report the number of entities (Web documents) per semantic attribute for which each KGC system extracted at least one value. We also report these coverage numbers at the level of *top level domains* (TLDs) to illustrate the robustness of the KGC system coverage to mesokurtic domains like HT[16].

Mean Average Precision (MAP) was used for evaluating the entity-centric search engine for all four query categories. For each question, each system was permitted a finite 'payload', namely a reasonable limit of 500 to the length of the ranked list of document IDs submitted per question. For point fact queries evaluated during the *dry run*, MAP is equivalent to MRR (Mean Reciprocal Rank) since exactly one relevant entity was annotated per point fact question during that phase. During the final run, more than one relevant entity was possible per point fact query.

5.2 Results and Analysis

KGC. Figure 4 illustrates the coverage of two KGC systems (DIG and Lattice) on the *final run* corpus. Figure 4 (a) measures coverage counts at the webpage level, while (b) measures coverage at TLD level. We report separate coverage counts of DIG for the sets of high-precision and high-recall a-IEs. The results show that, in general, the Lattice KG has lower coverage on many attributes than the DIG KG: the Lattice KGC process requires TLD-specific tuning for each new TLD, and hence only provides coverage for a small fraction of TLDs (the 'short tail'). Although the precision of extractions cannot be computed without webpage annotations, we performed our own precision-based study by independently and randomly sampling a small set of 100 pages from the final corpus for each of the *phone*, *social-media-id*, *price* and *name* semantic attributes, and manually annotating the correctness of both Lattice and DIG extractions for the 100 entities in each of the 4 sets. The precision of Lattice extractions was about 3% higher than that of DIG only on the phone attribute; on the other three attributes, the precision of DIG extractions equaled those of

16. An example of a TLD is backpage.com, which could yield many webpages, all with the same TLD. There is a one-many mapping between TLDs and webpages.

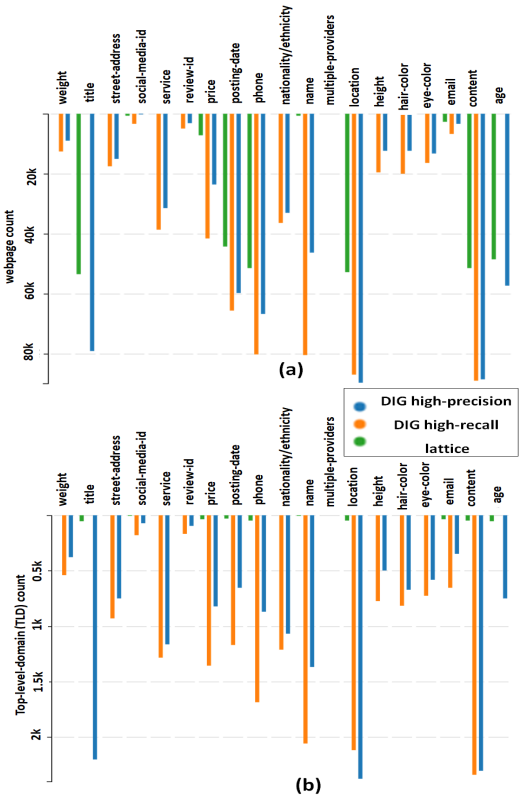

Fig. 4. KG coverage counts for two KGC systems, DIG and Lattice, measured at the level of webpages (a) and TLDs (b).

TABLE 4
Final run query results of three independent systems on Mean Average Precision (MAP), for four query categories.

System	Point Fact	Cluster Id.	Cluster Facet	Cluster Agg.
1 (DIG)	0.67	0.66	0.59	0.35
2	0.75	0.53	0.57	0.53
3	0.71	0.21	0.37	0.06

Lattice on the entities where extractions for both existed; as illustrated in Figure 4, the Lattice KGC is only defined for a small set of manually specified Web domains. Also, for the social-media-id attribute, Lattice coverage was 0 even in those specified domains.

Query execution (dry run). During the dry run, one of the teams (Team 2) implemented a conservative querying approach that obeyed the original semantics of the query, regardless of the noise in the knowledge graph. This is similar to using only the semantics-preserving strategy, which is a reasonable baseline. We implemented the three semantic strategies, as earlier outlined, with a relatively simple schema functional. Due to technical difficulties, the third team was not able to submit results in time for the official dry run, but submitted results for the official final run. Multiple submissions were encouraged for the dry run.

Overall, our dry run prototype outperformed the baseline on all four query categories, for all runs, and was the only prototype to achieve non-zero scores on all query categories. Thus, one lesson indicated by this early exercise was that only using the conservative approach is not robust,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TBDATA.2017.2763164, IEEE Transactions on Big Data

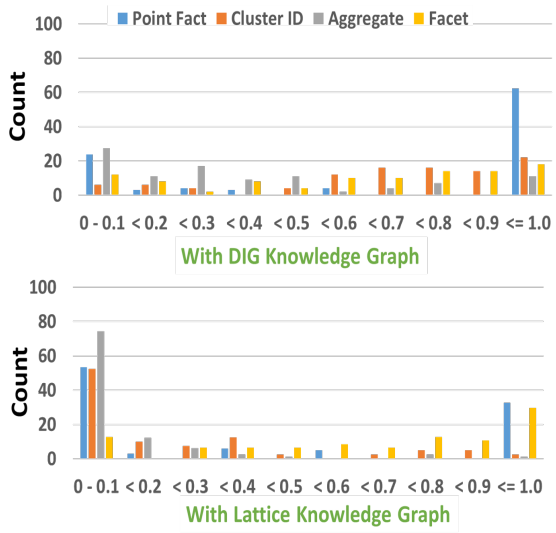

Fig. 5. A *non-cumulative* Mean Average Precision (MAP) histogram plot of our query prototype on two different KGs (DIG and Lattice).

compared to using a combination of constrained and less-constrained, query strategies on real-world KGs. A second lesson learned from an analysis of our own runs was that the keyword strategy is indispensable for queries where the right answer is a difficult-to-structure webpage from which only portions of the text could be reliably extracted. Specifically, we found that, for the same KG, *not* including the keyword strategy led to a decline of more than 60% on point fact recall. Since the effectiveness of the query prototype was so strongly dependent on the actual data in the knowledge graph, all teams refined their query prototypes for the final run (described below).

Query execution (final run). Table 4 tabulates the results of all three systems. Each team was only permitted one submission for a KG of their choice and construction in the final run. Given the coverage of DIG compared to Lattice in Figure 4, we (System 1) chose DIG for our KGC module, executed all queries using the prototype described in this paper and submitted results. System 2 used the Lattice KG in conjunction with extracted semantic types from internally developed a-IEs. For their query prototype, they independently replicated the principles described in this paper (also using Elasticsearch as their NoSQL database) with a sequence of conservative and keyword-based text-only semantic strategies. As earlier described, System 3 employed learning-to-rank.

The results show that, although the learning-to-rank approach is competitive with the two KG-based systems on point fact queries, it does not do as well on the other categories. The performances of the KG-based systems are consistent and heavily correlated: out of all categories, both systems perform the best on point fact questions and the worst on cluster aggregate questions. We believe that this indicates, although more evidence is needed, that the knowledge graph-centric approach is amenable to *replication* even with so many moving parts. Most importantly, the results show that the systems are viable in the real-world; on average, the MAP is over 0.5, meaning that (on point fact queries) the correct entity is usually retrieved within the top 1 or 2, even in response to a complex combination of multiple facets.

Scalability. We assessed the scaling capabilities of entity-centric search by duplicating the final run evaluation process using the same set of questions, but knowledge graphs constructed on two different corpora containing 2.3 million and 53.7 million webpages respectively[17]. On both corpora the 302 questions were found to execute in less than 60 minutes, on the same cluster infrastructure described earlier. This scalability was achievable because of the excellent scaling properties of both Apache Spark and Elasticsearch. Similar properties for System 2 have been verified.

6 Discussion

To investigate the effect of the constructed knowledge graph on query effectiveness, we performed a controlled experiment wherein we executed our query prototype (on all final run questions) on the two KGs constructed by Lattice and DIG respectively. A breakdown of the results is illustrated in Figure 5. While there is a high correlation between the results, the query prototype is more robust on the DIG KG and yields more top 1 answers, especially for point fact queries. Two phenomena may explain these results. First, as shown in Figure 4, Lattice extractions seem to be optimized for the short tail, which may have led to considerably lower performance on queries where the correct answer could only be found in the long tail. A second explanation is that our query prototype may be biased in favor of DIG, since we used the DIG KG during the dry run phase to refine our strategies. Future work will quantify the contributions of both factors to the differences noted in Figure 5.

For point fact questions, a complete error analyses of the cases where the correct entity was *not* retrieved as the *top* result is ongoing (see Appendix). Several interesting results have emerged from this exercise, as some errors are specific to human trafficking while others are not (e.g., ambiguity). Error analyses are important because they illustrate the various ways in which system results can prove to be invalid e.g., a non-response from the system in response to a query does not imply that relevant responses do not exist. In other words, being aware of the errors guides not just developers, but also users, of the system, since queries can be re-formulated to circumvent some of the validity issues.

7 Impact

The DIG system described in this article has been exposed to over 200 law enforcement agencies in the US through a GUI that provides access to over 100 million webpages processed using the knowledge graph-centric extraction and search technology described herein. Since adoption, the MEMEX outputs, including DIG, have had considerable real-world impact. The most striking example is a recent report by the District Attorney of New York that showed that the use of DIG and other MEMEX tools have resulted in an increase in human trafficking investigations conducted in prostitution related arrest cases (in New York City) from less than 1% (pre-MEMEX) to more than 62% (post-MEMEX).

17. Uncompressed, the two corpora have disk size 185 GB and 4 TB respectively.

In a recently concluded case in San Francisco, a man was recently sentenced to 97 years to life for human trafficking[18], and his four accomplices were apprehended as well. More than 25 victims were rescued. The office of the San Francisco DA has directly acknowledged MEMEX in these cases.

While it is unwise to extrapolate the above to the potential long-term impact of MEMEX, the evidence shows that the tools, both in design and efficacy, are already proving useful and that the results in the previous section are reflective of real-world potential. Using a case study example, more details on how DIG is used in practice by law enforcement to glean actionable intelligence are included in the appendix, including screenshots of the DIG GUI.

8 Future Work and Conclusion

Our most important future goal is *direct* processing of queries in their natural language form. In practice, this would entail developing a conversational agent that interfaces with the user more naturally. A user would directly issue natural language questions, such as the examples in Table 3, and the underlying system would first translate them, possibly probabilistically, into structured queries before executing them in the semantic manner outlined herein. The interaction does not have to be unconstrained or make open-world assumptions. For example, we may preclude *attribute synonymy* by limiting the user to specifying attributes only by their names in the investigative schema, and also request the user to specify one of the four query categories in Table 3 before posing a question. While such constraints can simplify the task of developing a conversational agent, the underlying machine-readable problem still needs to be adequately solved. If the machine reading is indeed probabilistic, we also need a good framework for combining the (probabilistic) lists of answers that are output when executing such queries.

Conclusion: This paper presented a knowledge graph-centric approach that leverages the strengths of both structured and unstructured data to robustly execute investigative entity-centric search queries. The approach has been evaluated on real-world human trafficking data, and independently replicated; its integration into a GUI widely used by US law enforcement agencies is being actively explored at the time of writing.

Acknowledgements. We gratefully acknowledge our collaborators and all (former and current) members of our team who contributed their efforts and expertise to DIG, particularly during the dry and final evaluation runs: Amandeep Singh, Linhong Zhu, Lingzhe Teng, Nimesh Jain, Rahul Kapoor, Muthu Rajendran R. Gurumoorthy, Sanjay Singh, Majid Ghasemi Gol, Brian Amanatullah, Craig Knoblock and Steve Minton. This research is supported by the Defense Advanced Research Projects Agency (DARPA) and the Air Force Research Laboratory (AFRL) under contract number FA8750- 14-C-0240. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DARPA, AFRL, or the U.S. Government.

18. http://www.sfgate.com/crime/article/Man-sentenced-to-97-years-in-human-trafficking-7294727.php

References

[1] S. Harrendorf, M. Heiskanen, and S. Malby, *International statistics on crime and justice*. European Institute for Crime Prevention and Control, affiliated with the United Nations (HEUNI, 2010.

[2] E. U. Savona and S. Stefanizzi, *Measuring human trafficking*. Springer, 2007.

[3] V. Greiman and C. Bain, "The emergence of cyber activity as a gateway to human trafficking," in *Proceedings of the 8th International Conference on Information Warfare and Security: ICIW 2013*. Academic Conferences Limited, 2013, p. 90.

[4] P. Szekely, C. A. Knoblock, J. Slepicka, A. Philpot, A. Singh, C. Yin, D. Kapoor, P. Natarajan, D. Marcu, K. Knight *et al.*, "Building and using a knowledge graph to combat human trafficking," in *International Semantic Web Conference*. Springer, 2015, pp. 205–221.

[5] N. Dalvi, R. Kumar, B. Pang, R. Ramakrishnan, A. Tomkins, P. Bohannon, S. Keerthi, and S. Merugu, "A web of concepts," in *Proceedings of the twenty-eighth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*. ACM, 2009, pp. 1–12.

[6] J. Pujara, H. Miao, L. Getoor, and W. W. Cohen, "Knowledge graph identification," 2013.

[7] X. Dong, E. Gabrilovich, G. Heitz, W. Horn, N. Lao, K. Murphy, T. Strohmann, S. Sun, and W. Zhang, "Knowledge vault: A web-scale approach to probabilistic knowledge fusion," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2014, pp. 601–610.

[8] J. Han, E. Haihong, G. Le, and J. Du, "Survey on nosql database," in *Pervasive computing and applications (ICPCA), 2011 6th international conference on*. IEEE, 2011, pp. 363–366.

[9] G. Marchionini, "Exploratory search: from finding to understanding," *Communications of the ACM*, vol. 49, no. 4, pp. 41–46, 2006.

[10] A. Hogan, A. Harth, J. Umrich, and S. Decker, "Towards a scalable search and query engine for the web," in *Proceedings of the 16th international conference on World Wide Web*. ACM, 2007, pp. 1301–1302.

[11] T. Lin, P. Pantel, M. Gamon, A. Kannan, and A. Fuxman, "Active objects: Actions for entity-centric search," in *Proceedings of the 21st international conference on World Wide Web*. ACM, 2012, pp. 589–598.

[12] C.-H. Chang, M. Kayed, M. R. Girgis, and K. F. Shaalan, "A survey of web information extraction systems," *IEEE transactions on knowledge and data engineering*, vol. 18, no. 10, pp. 1411–1428, 2006.

[13] A. Doan, A. Halevy, and Z. Ives, *Principles of data integration*. Elsevier, 2012.

[14] A. K. Elmagarmid, P. G. Ipeirotis, and V. S. Verykios, "Duplicate record detection: A survey," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 19, no. 1, pp. 1–16, 2007.

[15] F. Niu, C. Zhang, C. Ré, and J. W. Shavlik, "Deepdive: Web-scale knowledge-base construction using statistical learning and inference." *VLDS*, vol. 12, pp. 25–28, 2012.

[16] M. Lenzerini, "Data integration: A theoretical perspective," in *Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*. ACM, 2002, pp. 233–246.

[17] R. El-Masri and G. Wiederhold, "Data model integration using the structural model," in *Proceedings of the 1979 ACM SIGMOD international conference on Management of data*. ACM, 1979, pp. 191–202.

[18] A. Doan, A. Halevy, and Z. Ives, *Principles of data integration*. Elsevier, 2012.

[19] X. L. Dong and D. Srivastava, "Big data integration," in *Data Engineering (ICDE), 2013 IEEE 29th International Conference on*. IEEE, 2013, pp. 1245–1248.

[20] N. Gupta, A. Y. Halevy, B. Harb, H. Lam, H. Lee, J. Madhavan, F. Wu, and C. Yu, "Recent progress towards an ecosystem of structured data on the web." in *ICDE*. Citeseer, 2013, pp. 5–8.

[21] M. Brambilla, S. Ceri, and A. Y. Halevy, "Special issue on structured and crowd-sourced data on the web." *VLDB J.*, vol. 22, no. 5, pp. 587–588, 2013.

[22] D. Gomez-Cabrero, I. Abugessaisa, D. Maier, A. Teschendorff, M. Merkenschlager, A. Gisel, E. Ballestar, E. Bongcam-Rudloff, A. Conesa, and J. Tegnér, "Data integration in the era of omics: current and future challenges," *BMC systems biology*, vol. 8, no. 2, p. 1, 2014.

[23] D. Loshin, *Big data analytics: from strategic planning to enterprise integration with tools, techniques, NoSQL, and graph*. Elsevier, 2013.

[24] N. Kushmerick, "Wrapper induction for information extraction," Ph.D. dissertation, University of Washington, 1997.

[25] I. Muslea, S. Minton, and C. Knoblock, "Stalker: Learning extraction rules for semistructured, web-based information sources," in *Proceedings of AAAI-98 Workshop on AI and Information Integration*. AAAI Press Menlo Park, CA, 1998, pp. 74–81.

[26] K. Lerman, S. Minton, and C. A. Knoblock, "Wrapper maintenance: A machine learning approach," *J. Artif. Intell. Res.(JAIR)*, vol. 18, pp. 149–181, 2003.

[27] C.-H. Chang, M. Kayed, M. R. Girgis, and K. F. Shaalan, "A survey of web information extraction systems," *IEEE transactions on knowledge and data engineering*, vol. 18, no. 10, pp. 1411–1428, 2006.

[28] O. Etzioni, M. Banko, S. Soderland, and D. S. Weld, "Open information extraction from the web," *Communications of the ACM*, vol. 51, no. 12, pp. 68–74, 2008.

[29] M. Banko, M. J. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni, "Open information extraction from the web." in *IJCAI*, vol. 7, 2007, pp. 2670–2676.

[30] J. R. Frank, M. Kleiman-Weiner, D. A. Roberts, F. Niu, C. Zhang, C. Ré, and I. Soboroff, "Building an entity-centric stream filtering test collection for trec 2012," DTIC Document, Tech. Rep., 2012.

[31] P. Saleiro, J. Teixeira, C. Soares, and E. Oliveira, "Timemachine: Entity-centric search and visualization of news archives," in *European Conference on Information Retrieval*. Springer, 2016, pp. 845–848.

[32] A. Freitas, E. Curry, J. G. Oliveira, and S. O'Riain, "Querying heterogeneous datasets on the linked data web: challenges, approaches, and trends," *IEEE Internet Computing*, vol. 16, no. 1, pp. 24–33, 2012.

[33] A. Tonon, G. Demartini, and P. Cudré-Mauroux, "Combining inverted indices and structured search for ad-hoc object retrieval," in *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2012, pp. 125–134.

[34] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, "Freebase: a collaboratively created graph database for structuring human knowledge," in *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*. AcM, 2008, pp. 1247–1250.

[35] A. Singhal, "Introducing the knowledge graph: things, not strings," *Official google blog*, 2012.

[36] E. Gabrilovich and S. Markovitch, "Computing semantic relatedness using wikipedia-based explicit semantic analysis." in *IJcAI*, vol. 7, 2007, pp. 1606–1611.

[37] G. A. Miller, "Wordnet: a lexical database for english," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[38] L. Hirschman and R. Gaizauskas, "Natural language question answering: the view from here," *natural language engineering*, vol. 7, no. 04, pp. 275–300, 2001.

[39] A. Jain, A. Doan, and L. Gravano, "Sql queries over unstructured text databases," in *2007 IEEE 23rd International Conference on Data Engineering*. IEEE, 2007, pp. 1255–1257.

[40] M. Hultgren, M. E. Jennex, J. Persano, and C. Ornatowski, "Using knowledge management to assist in identifying human sex trafficking," in *System Sciences (HICSS), 2016 49th Hawaii International Conference on*. IEEE, 2016, pp. 4344–4353.

[41] H. Alvari, P. Shakarian, and J. K. Snyder, "A non-parametric learning approach to identify online human trafficking," in *Intelligence and Security Informatics (ISI), 2016 IEEE Conference on*. IEEE, 2016, pp. 133–138.

[42] H. Chen, *Dark web: Exploring and data mining the dark side of the web*. Springer Science & Business Media, 2011, vol. 30.

[43] S. Bird, "Nltk: the natural language toolkit," in *Proceedings of the COLING/ACL on Interactive presentation sessions*. Association for Computational Linguistics, 2006, pp. 69–72.

[44] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica, "Spark: cluster computing with working sets." *HotCloud*, vol. 10, pp. 10–10, 2010.

[45] M. Duckham and L. Kulik, "A formal model of obfuscation and negotiation for location privacy," in *International Conference on Pervasive Computing*. Springer, 2005, pp. 152–170.

[46] A. Moser, C. Kruegel, and E. Kirda, "Limits of static analysis for malware detection," in *Computer security applications conference, 2007. ACSAC 2007. Twenty-third annual*. IEEE, 2007, pp. 421–430.

[47] M. Stevenson and Y. Wilks, "Word sense disambiguation," *The Oxford Handbook of Comp. Linguistics*, pp. 249–265, 2003.

[48] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives, "Dbpedia: A nucleus for a web of open data," in *The semantic web*. Springer, 2007, pp. 722–735.

[49] M. Wick and B. Vatant, "The geonames geographical database," *Available from World Wide Web: http://geonames. org*, 2012.

[50] "Inferlink r&d capabilities," http://www.inferlink.com/our-work#research-capabilities-section, accessed: 2017-04-28.

[51] "Readability text extractor," https://www.readability.com/, accessed: 2017-04-28.

[52] S. Sarawagi and W. W. Cohen, "Semi-markov conditional random fields for information extraction," in *Advances in neural information processing systems*, 2004, pp. 1185–1192.

[53] F. Peng and A. McCallum, "Information extraction from research papers using conditional random fields," *Information processing & management*, vol. 42, no. 4, pp. 963–979, 2006.

[54] P. Pantel, E. Crestan, A. Borkovsky, A.-M. Popescu, and V. Vyas, "Web-scale distributional similarity and entity set expansion," in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2-Volume 2*. Association for Computational Linguistics, 2009, pp. 938–947.

[55] B. B. Dalvi, W. W. Cohen, and J. Callan, "Websets: Extracting sets of entities from the web using unsupervised information extraction," in *Proceedings of the fifth ACM international conference on Web search and data mining*. ACM, 2012, pp. 243–252.

[56] R. Kapoor, M. Kejriwal, and P. Szekely, "Using contexts and constraints for improved geotagging of human trafficking webpages," *arXiv preprint arXiv:1704.05569*, 2017.

[57] M. Kejriwal and P. Szekely, "Information extraction in illicit web domains," in *Proceedings of the 26th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 2017, pp. 997–1006.

[58] Y. Li, R. Krishnamurthy, S. Raghavan, S. Vaithyanathan, and H. Jagadish, "Regular expression learning for information extraction," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2008, pp. 21–30.

[59] R. Krishnamurthy, Y. Li, S. Raghavan, F. Reiss, S. Vaithyanathan, and H. Zhu, "Systemt: a system for declarative information extraction," *ACM SIGMOD Record*, vol. 37, no. 4, pp. 7–13, 2009.

[60] A. Doan, R. Ramakrishnan, and S. Vaithyanathan, "Managing information extraction: state of the art and research directions," in *Proceedings of the 2006 ACM SIGMOD international conference on Management of data*. ACM, 2006, pp. 799–800.

[61] "Isi extraction toolkit repository," https://github.com/usc-isi-i2/etk, accessed: 2017-04-29.

[62] E. PrudHommeaux, A. Seaborne *et al.*, "Sparql query language for rdf," *W3C recommendation*, vol. 15, 2008.

[63] A. Ferraram, A. Nikolov, and F. Scharffe, "Data linking for the semantic web," *Semantic Web: Ontology and Knowledge Base Enabled Tools, Services, and Applications*, p. 169, 2013.

Mayank Kejriwal is a research scientist at Information Sciences Institute, University of Southern California Viterbi School of Engineering. He graduated in 2016 with his PhD from the University of Texas at Austin. He has been actively involved in researching, testing and integrating various data mining techniques in the Domain-specific Insight Graph (DIG) architecture, most notably entity resolution, information extraction, and entity-centric information retrieval. DIG is widely used by US law enforcement agencies to combat human trafficking; its extensibility to other domains with the potential for social good such as securities fraud and weapons trafficking is also being investigated. Along with Pedro Szekely and Craig A. Knoblock, he is a co-author on a textbook on *Knowledge Graphs* that will be published by MIT Press.

Pedro Szekely is a Research Associate Professor at ISI. He received his PhD in 1987 from Carnegie Mellon University and is the principal investigator and project leader on the DIG project, currently funded under the DARPA MEMEX program. His research expertise is in rapid and robust construction of domain-specific knowledge graphs. He has served on numerous program committees and his work on DIG has won multiple best paper awards.