An Investigative Search Engine for the Human Trafficking Domain

Mayank Kejriwal, Pedro Szekely

Information Sciences Institute
{kejriwal,pszekely}@isi.edu

Abstract. Enabling intelligent search systems that can navigate and facet on entities, classes and relationships, rather than plain text, to answer questions in complex domains is a longstanding aspect of the Semantic Web vision. This paper presents an investigative search engine that meets some of these challenges, at scale, for a variety of complex queries in the human trafficking domain. The engine provides a real-world case study of synergy between technology derived from research communities as diverse as Semantic Web (investigative ontologies, SPARQL-inspired querying, Linked Data), Natural Language Processing (knowledge graph construction, word embeddings) and Information Retrieval (fast, user-driven relevance querying). The search engine has been rigorously prototyped as part of the DARPA MEMEX program and has been integrated into the latest version of the Domain-specific Insight Graph (DIG) architecture, currently used by hundreds of US law enforcement agencies for investigating human trafficking. Over a hundred millions ads have been indexed. The engine is also being extended to other challenging illicit domains, such as securities and penny stock fraud, illegal firearm sales, and patent trolling, with promising results.

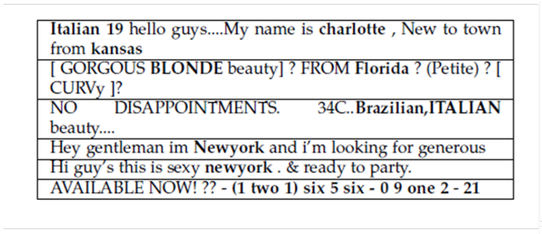
Keywords: Knowledge graphs, Investigative search, Human trafficking, Illicit domains, Knowledge graph construction

1 Introduction

Recent studies confirm a formidable reach of illicit players both online and offline. For example, data from the National Human Trafficking Resource Center shows that human trafficking (HT) is not only on the rise in the United States, but is a problem of international proportions [12], [21]. The advent of the Web has made the problem worse [10]. Human trafficking victims are advertised both on the Open and Dark Web, with estimates of the number of (not necessarily unique) published advertisements being in the hundreds of millions [22].

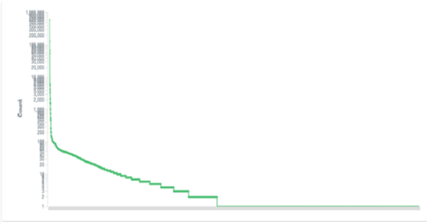
In recent years, various agencies in the US have turned to technology to assist them in combating this problem through the suggestion of leads, evidence and HT indicators. An important goal is to answer *entity-centric questions* over noisy Web corpora crawled from a subset of Web domains known for HT-related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

As a running example, consider the following real-world investigative request[1] (in natural language): *Find the average price per hour charged by escorts associated directly, or via shared phone/email links, with phone number 123-456-7890.* In the context of investigative querying, we refer to such a query as a *cluster aggregate query,* as correctly executing the query representation of such a question on a hypothetically *perfect* (sound and complete) database of instances can be achieved by the following two-step approach: first, *cluster* escort ads based on the seed phone (123-456-7890) in a well-defined manner that is described in Section 4.1, and then *aggregate* (in this case, average) all prices per hour in the clustered ads.



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**

(a) Text fragments scraped from real-world human trafficking webpages, with relevant extractions in bold.



(b) An interpolated semi-log curve illustrating the long tail distribution of the number of pages (y-axis) against each of the 4,354 Web domains (e.g., backpage.com), ordered by frequency on the x-axis

Fig. 1. Illustration of obfuscations (a) and the long-tail effect (b) in illicit domains like human trafficking.

Even assuming such a machine-understandable query representation (e.g., using a SPARQL subset), achieving good performance on real-world data is challenging. First, *information extraction* (IE) is not a solved problem, even in well-studied domains like social media, news corpora or even Wikipedia [3]. The cluster aggregate query above requires both phone numbers and prices per hour to be correctly extracted from scraped websites. The quality of extractions, even coarse extractions like the main content text from the HTML of escort ads, is much worse in illicit domains, especially when the source is a messy webpage that has been downloaded by semi-automatic crawlers. We also note that the *language model* in such illicit domains is highly irregular, impeding the use of standard tools from the NLP literature (Figure 1a). Second, illicit domains have been empirically observed to exhibit the *long-tail* effect (Figure 1b), meaning that optimizing for, or expending manual effort on, a few of the biggest Web domains has limited utility. In many cases, investigators are explicitly interested in the long tail as a source of leads and evidence. Third, from a systems-level standpoint, integration of the different components is a difficult issue, as is the very real problem of scale, since the corpora often comprise millions of web-

[1] Identifying information has been replaced in all examples used in this paper.

pages residing in a distributed file system (DFS). Challenges and motivations are further detailed in Section 3.

In this paper, we present an in-use investigative search engine that utilizes interdisciplinary research from both within and beyond the Semantic Web to satisfy a motivated set of investigative information needs (Section 3). A high-level overview of both the architecture as well as the dataflow is provided in Section 4. Our search engine has been integrated into the Domain-specific Insight Graph (DIG) architecture, which is currently used by hundreds of law enforcement agencies to combat human trafficking (Section 5), and is accessible via a GUI. DIG provides analysis and search capabilities over more than a hundred million webpages collected by crawlers over two years. In addition to the search engine, the DIG GUI also supports advanced facilities such as entity-centric page views over unusual entities such as phones, aggregations, temporal and geolocation analytics, and image similarity search. The search engine has been evaluated by DARPA in a competitive setting, and the engine is currently being extended to other illicit domains of vital investigative importance, notable examples being securities fraud and illegal firearm sales.

2 Related Work

The contributions in this work rely on several different research areas. Rather than attempt a comprehensive survey, we only provide pointers to overviews of these areas. *Knowledge Graph Construction* (KGC) is an important component of the DIG architecture. KGC draws on advances from a number of different research areas, including information *extraction* [3], information *integration* [6], and inferential tasks such as entity resolution [7]. In addition to DIG, another example of a principled KGC architecture is DeepDive [18].

Entity-centric Search (ECS) is another broad area of research that has influenced DIG, 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 [17], knowledge base *acceleration* and filtering [8], interactive search and visualization [19], and search tailored for the Semantic Web [14]. An early entity-centric search prototype that is similar to our own effort is SWSE, which first crawls Web data and converts it to RDF [14]. 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) [9], [23]. 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.

Another related branch of research is *question answering*; however, unlike *question answering* systems [13], 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' as in 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 [16]; however, the knowledge graph constructed by our system is more unorthodox, containing a mix of textual, numerical, pseudo-numerical and structured fields.

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 [15], [2]. We also note that, although the research described herein is specifically designed to investigate and combat human trafficking, the core elements of the overall problem and solution can be extended to other domains (e.g. from the Dark Web [4]) 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.

The technology described in this paper is implemented within the Domain-specific Insight Graph (DIG) architecture. The basic *knowledge graph-centric principles* of the DIG architecture (DIGv1.0) were published in an earlier work [22]. The earlier work did not include semantic search technology (only basic keyword search) or the advanced GUI capabilities that are the primary contributions of the architecture described herein and that are currently being used by law enforcement.

3 Motivation

To illustrate the motivation behind an investigative search engine, we start with a set of *information needs* that investigative experts are interested in exploring further. For example, given a seed phone number, investigators may want to find the (1) pages directly containing this phone number, (2) the average of all prices mentioned in those pages, (3) the time-series profile of the phone number, (4) the locations mentioned in the ads, plotted on a map, and (5) phone number recommendations, with corresponding pages, relevant to the current inquiry (i.e. the provided phone number). More generally, there may be multiple constraints and attributes[2] (e.g., instead of a query phone number, we are given a date range and a city), and the request may require us to retrieve tuples of attributes, ranked in order of relevance, or conduct different kinds of aggregations (e.g., max, average) on an attribute of the retrieved pages.

The rationale for the requests above are strongly influenced by real-world considerations. In illicit domains, investigative officials must commit limited resources to field investigations of leads. Usually, the initial lead comes from an on-the-ground source like a tip-off. If a system can provide useful (even if partial) solutions to (1)-(5), and a means of verifying the solutions, resource allocation can be significantly optimized. The consequence is that many more cases can be initiated and prosecuted, and the search for evidence becomes more efficient. A long-term consequence is that the *barrier-for-entry* to such illicit activities is significantly strengthened.

For present purposes, we make the closed-world assumption that the request must be facilitated over a pre-crawled Web corpus. The main reason why such an assumption is not only reasonable, but necessary, is that 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 offline 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.

Another problem that precludes live search (e.g. on Google) is obfuscation. Some examples were provided earlier in Figure 1a. 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. We observed that most links are inserted by publishers to promote other content and may, in fact, cause traditional search crawlers to behave unexpectedly. The investigative search engine presented herein is designed to address these challenges, and be especially useful for the related purposes of *gathering evidence* and *minimizing investigative effort*, which is directly relevant to the motivation of using the system for evidence-based social good.

4 Architectural Overview

Figure 2 illustrates the architecture of the investigative search engine in DIGv3.0 (henceforth referred to as DIG). Prior to invoking DIG, a *domain-discovery system* is used to crawl large corpora from the Web based on a *relevance model* that depends, among other things, on a few seed links to relevant websites and a set of descriptive keywords. For example, in the human trafficking domain, the keywords could be 'escort', 'incall' etc. and links could include escort ads on Web

[2] As described subsequently, attributes are not open-world, but bound by a shallow ontology called an *investigative schema* (Section 4.1).

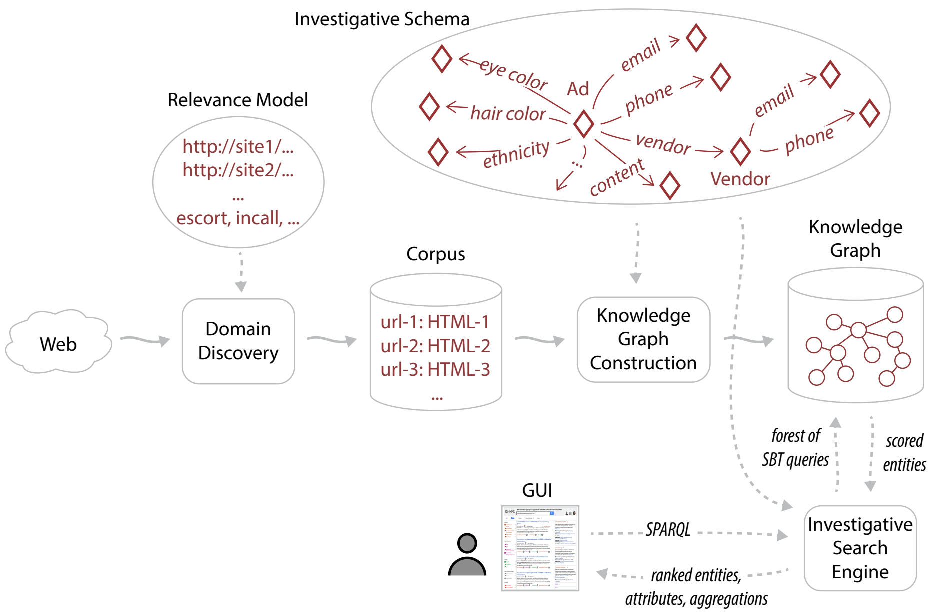


Fig. 2. An architectural overview of Domain-specific Insight Graph (DIG).

domains like `backpage.com`. The output of domain discovery typically results in corpora that contain at least a few million pages, and when fine-tuned for good recall on a particular illicit domain like human trafficking (that becomes better understood over time) can eventually yield hundreds of millions of webpages.

Once a Web corpus has been acquired, knowledge graph construction (KGC) is used in an offline phase to populate the DIG knowledge graph (KG) according to an *investigative schema* (IS) that was collaboratively developed over multiple weeks. The IS is a shallow ontology (inspired by schema.org) that contains important domain-specific classes and attributes either derived directly from schema.org (if it exists e.g., *date*) or added to the schema upon request (e.g., *hair-color*). While we expect the IS to be periodically refined based on user studies and feedback , the KGC process assumes a fixed IS in terms of which the knowledge graph is *semantically typed* using tools like Karma [11]. We also note that the DIG KGC is designed to be necessarily *query-centric* i.e. the KG is constructed with the explicit goal of supporting investigative search and not with optimizing KG quality (in terms of precision of facts in the KG) *per se*.

Once a KG is semi-automatically constructed from a raw Web corpus, it is stored and indexed in a NoSQL database (Elasticsearch). The indexed KG, and the basic key-value query engine provided by Elasticsearch, interface with the investigative search engine that is used to power the GUI. In subsequent sections, we detail all of these steps further. In the interest of space, we focus on the relevant design principles, rather than technical or algorithmic details. Core Semantic Web technologies that have prominent employment in DIG include the use of SPARQL for internal query representation, investigative schemas largely inspired by shallow ontologies like schema.org, and the use of Linked Data sources like Geonames for facilitating knowledge graph construction.

4.1 Knowledge Graph Construction (KGC)

For the purposes of this paper, we define KGC as a process that takes as input a raw corpus of Web documents, and constructs a (potentially noisy) knowledge graph (KG) of labeled nodes and labeled edges. Nodes and edge labels are semantically typed in terms of classes and properties in an investigative schema (IS), as mentioned earlier. Semantic typing is necessary because the set of extractions in the knowledge graph may not coincide with the IS, and in some cases, the type of an extraction may not even be known in advance. For example, a wrapper-based tool could extract structured fields from a webpage, but may not be able to automatically determine the type of each field. In other cases, the problem can be trivial; e.g., a regular expression program designed only to extract phone numbers will always have its extractions semantically typed as class *Phone Number* in the IS. We used the Karma tool for semantic typing [11]. Note that, from the search engine's (and hence user's) perspective, the KG can only be accessed and faceted on classes and properties from the IS.

Information Extraction (IE). To accommodate the requirements of quality, recall and robustness, we designed a *query-centric IE architecture* that considers a diverse *battery* of IE modules inspired by research in multiple research communities [3]. Included among the modules are NLP algorithms like Conditional Random Field (CRF)-based Named Entity Recognizers (for schema attributes like hair color, eye color and ethnicity), rule-based algorithms like regular expressions (for attributes like phone numbers and email addresses), and entity sets, dictionaries, lexicons and external knowledge bases (a good example being Geonames) for relatively closed-world attributes like services, names and locations. In a few cases, the extractors are *hybrid*: for example, we combine rule-based and NLP-based techniques for extracting street addresses. Finally, we also use a state-of-the-art semi-automatic wrapper-based tool, Inferlink [1], which can robustly extract structured fields on a per-domain basis with only 10 minutes of manual effort per Web domain. Because of the long tail distribution of human trafficking Web domains, we restricted Inferlink to a few of the largest Web domains, a notable example being `backpage.com`.

Each IE algorithm requires different types and levels of manual supervision. In the case of regular expressions and lexicons, for example, the extractor runs automatically once constructed, but is *non-adaptive*. On the other hand, CRFs achieve higher performance and can adapt more easily to complex domains but require high quantities of manually annotated data. For this reason, we only use CRFs for a few attributes like *hair color*, *eye color* and *ethnicity*.

An important aspect about our IE modules is that, while some are (either by design, or through parameter configuration) constructed to deliver high expected precision, others are designed to deliver high expected recall. This is one of the key innovations of DIG, as we subsequently describe. Even when the knowledge graph is extremely noisy, queries can still be answered reliably by implementing *investigative search strategies* that jointly consider high-precision and high-recall extractions in the ranking algorithm.

Phone-email Clustering. In traditional KGC architectures, all extracted entities tend to be explicitly present in the text[3]. However, investigators are also often interested in *latent* entities that are not mentioned explicitly in the text, but need to be inferred from multiple entity extractions. Such inference relies heavily on sociological and domain-specific factors, and cannot be obviously automated without knowledge of these factors[4].

For example, in the human trafficking domain, it is well known that phones and emails serve as *pseudo-identifiers* for clustering escorts together and thereby discovering *vendors* that traffic these individuals. To facilitate phone-email clustering for the purpose of vendor discovery, we construct a weighted network where the nodes are entity (i.e. ad) IDs, and two nodes are connected by an edge iff they share a phone number or email extraction. Next, we use a clustering algorithm based on random walks to discover latent cluster entities (of a special type *Vendor* in our investigative schema). We found random walk-based clustering to be extremely scalable, executable on millions of nodes and edges without parallel programming, and to successfully account for the issue of data skew. Skew issues are very common with respect to phone extractions, even when the extraction is correct. For example, webpages often contain customer service phone numbers that can lead to significant distortions if a classic (i.e. non-stochastic) connected components algorithm is used for discovering vendors.

It is important to note that our clusters are *explainable*, and make intuitive sense. This is in contrast to, for example, deep learning and latent space algorithms, which yield higher performance but at significant cost to human-centric explainability. As we show in Section 5.2, domain experts cite explainability as a significant barrier to trusting advanced AI and semantic technology for expensive field investigations.

4.2 Storage, Representation and Indexing

Once the domain-specific knowledge graph has been constructed, it needs to be successfully queried by investigators. Since both efficiency and robustness are important, the knowledge graph needs to be stored, represented and indexed in a way that supports fast, expressive query execution. Furthermore, since we rely on open-source components, and can only achieve scale in cloud infrastructure, we only availed of technology that is both well-documented, released under a permissive license, and actively maintained by a technical community.

Detailed empirical and systems-level comparisons between the viability of triplestores, and those of NoSQL key-value databases like MongoDB and Elasticsearch, illustrated considerable advantages enjoyed by the latter. In recent

[3] In more advanced knowledge graph *completion* systems, additional links can be inferred, such as in entity resolution. However, nodes are still explicitly extracted.

[4] An additional problem with using powerful machine learning tools here is the lack of training data, and the uniqueness of each case.

years, MongoDB and Elasticsearch have both continued to enjoy widespread success in domains well beyond traditional databases and IR: they have been used to power expressive GUI applications, process terabytes of data, and are amenable to easy (i.e. by default, without expert programming) horizontal scalability in machines provisioned by cloud service providers like Amazon Web Services and Microsoft Azure. REST APIs for interfacing with, and retrieving data from, Elasticsearch in established data exchange formats like JSON are available in all major programming languages and paradigms (including Apache Spark) with full documentation and error handling facilities. Triplestores were not found to enjoy the same level of support and are not natively offered or uniformly supported by the major cloud service providers.

4.3 Search and Querying

Soft Boolean Tree Queries. The investigative search engine is a customized ranking and retrieval engine that receives input directly from the DIG GUI as queries with multiple constraints. We use SPARQL for expressing the constraints (OPTIONAL and FILTER clauses, as well as aggregations) supported by DIG. Keyword queries are also supported given that the IS includes attributes such as *title* and *content*, which can be used as constraints. The search engine in Figure 2 takes the SPARQL query as input and uses a set of semantic reformulation strategies to convert the SPARQL query into a forest of *soft boolean tree* (SBT) queries that can be executed on the Elasticsearch database.

Detailing the formal syntax of SBT queries is beyond the scope of this paper. Intuitively, an SBT query is defined recursively as a labeled tree, where a leaf node is always a qualified key-value pair (denoted as a *leaf query*, with the key denoting an IS attribute, and the value denoting a literal value. An edge label must always be selected from the set {*must, must not, should, filter*}. *Must* and *must not* have AND and NOT semantics, *should* has OR semantics and *filter* is a hard constraint, similar to FILTER clauses in SPARQL. In this respect, an SBT query is like a propositional formula with an additional FILTER operator. However, because the tree is soft, each leaf query is first scored by Elasticsearch using measures like tf-idf. Scores are propagated upwards to the root node using intuitive combination rules. For example, if a FILTER clause is violated, the parent node along that branch automatically receives a score of 0. For more details on the Elasticsearch query language, we refer the reader to the documentation[5].

There are two reasons why SBT queries were found to be extremely expedient in the search architecture. First, Elasticsearch can execute even large, complicated tree queries (and more generally forest queries, which are sets of weighted trees) extremely fast owing to judicious use of inverted indices that are compiled during the knowledge graph indexing phase. Second, SBT queries are

[5] `https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl.html`

robust to noise in the knowledge graph *if* the SPARQL query is *appropriately reformulated.*

For example, consider the (intuitively expressed) query: find all the escorts that have phone number *123-456-7890* and that have hair color *brunette.* This is a difficult query for several reasons. First, due to imperfections in extraction technology, either the phone number or the hair color did not get extracted, even though they are present in the text. Second, the extraction may not even have been present explicitly in the text. For example, there may be an escort ad that has the phone number 123-456-7890 and that has hair color *brown* in the text. Unless the system knows that *brunette* is the same as *brown* in this context, an *exact* interpretation of the original SPARQL query (which does not encase either of the two constraints in OPTIONAL) will be brittle.

To make query execution more robust without sacrificing the quality of the ranking (e.g., completely ignoring the original query semantics by naively encasing all constraints in OPTIONAL) we make use of three equally weighted *semantic reformulation strategies.* Each strategy separately processes a SPARQL query to yield an SBT query. In the current system, a combination of three strategies was found to yield good results:

Semantics-preserving strategy: This strategy is designed to preserve the semantics of the original SPARQL query in the following sense: an entity that should not get retrieved under strict SPARQL semantics (assuming execution against a triplestore) will get a score of 0 and not get retrieved by the search engine. Thus, this strategy respects the original user preferences, as stated.

Optional-semantics strategy: This strategy encodes the naive strategy mentioned earlier. All constraints are now qualified as OPTIONAL, even if they were not explicitly declared as such. This 'weaker' SPARQL query is now reformulated using the semantics-preserving strategy.

Keyword strategy: This strategy treats each entity as a text-only document, and collates the literal values in the facets (the constraints in the SPARQL query) as a set of keywords. We search these keywords against the union of the text attributes (e.g. title, description, body etc.) of the entity. This strategy is similar to those in classic Information Retrieval search systems.

The result reformulation is a forest query comprising three equally weighted SBT trees, which is subsequently executed against the indexed KG. In addition to the strategies mentioned above, we also include domain-specific functions for (1) value expansion using entity-sets to improve recall e.g., expanding emerald to green, or japanese to asian; (2) weighting pseudo-identifier constraints like phone and email higher so that they get boosted in the final ranking, if found. All of these functions are easily customized, as they are self-contained and static.

Given the forest query, Elasticsearch executes it against the KG and, in real time, returns a set of scored entities, which we rank and display to the user on the GUI in descending order of scores. Using the GUI, the user can further explore each entity, or use facilities like facets to narrow search even further in an exploratory fashion.

Implementation. To ease integration, almost all the code in DIG is implemented in Python, including the investigative search engine. The toolkit encapsulating the battery of information extractors is publicly available as a Github project[6]. DIG also makes use of the parallelism in Apache Spark to efficiently process millions of webpages crawled by the domain discovery systems. The indexed knowledge graph is stored in the Elasticsearch NoSQL database, which is used to power the interface.

5 Real-world Usage

DIG is currently being used by over 200 law enforcement agencies, and its permanent transition to at least one state agency (the District Attorney of New York) is currently underway. We illustrate real-world usage and impact of DIG from three different viewpoints, including a case study involving a potential real-world victim soliciting as an escort online, and user studies involving actual domain experts and investigators in the human trafficking domain.

We also note that the investigative search engine has been evaluated on at least two different manually annotated, real-world human trafficking corpora, and four categories of complex queries, prepared by DARPA under the MEMEX program[7]. Specifically, the search engine was last evaluated by DARPA on 102 point fact (PF) questions, and 200 compound HT questions involving clustering and/or aggregations, against two competitive baselines (one from academia and one from industry) participating in the MEMEX program. The KG on which system retrieval performance was tested was constructed from 90,000 HT webpages collected and annotated by DARPA. Scalability was tested by executing the system for all 302 questions on multiple domain discovery corpora, with the largest being a multi-terabyte dataset containing 53.7 million webpages. On average, our system could reformulate and answer each query in less than 12 seconds on the largest corpus, using a regular Internet connection.

The evaluations were blind, and conducted by both NIST and DARPA: the teams building their respective search engines were not exposed to any ground-truth before final submission of results. In the first controlled evaluation (conducted in the summer of 2016), our system performed the best out of all submitted systems, achieving non-zero scores on all query categories. In the second evaluation (conducted in November, 2016), DIG achieved the best results on two of the four categories, and was competitive on the other categories.

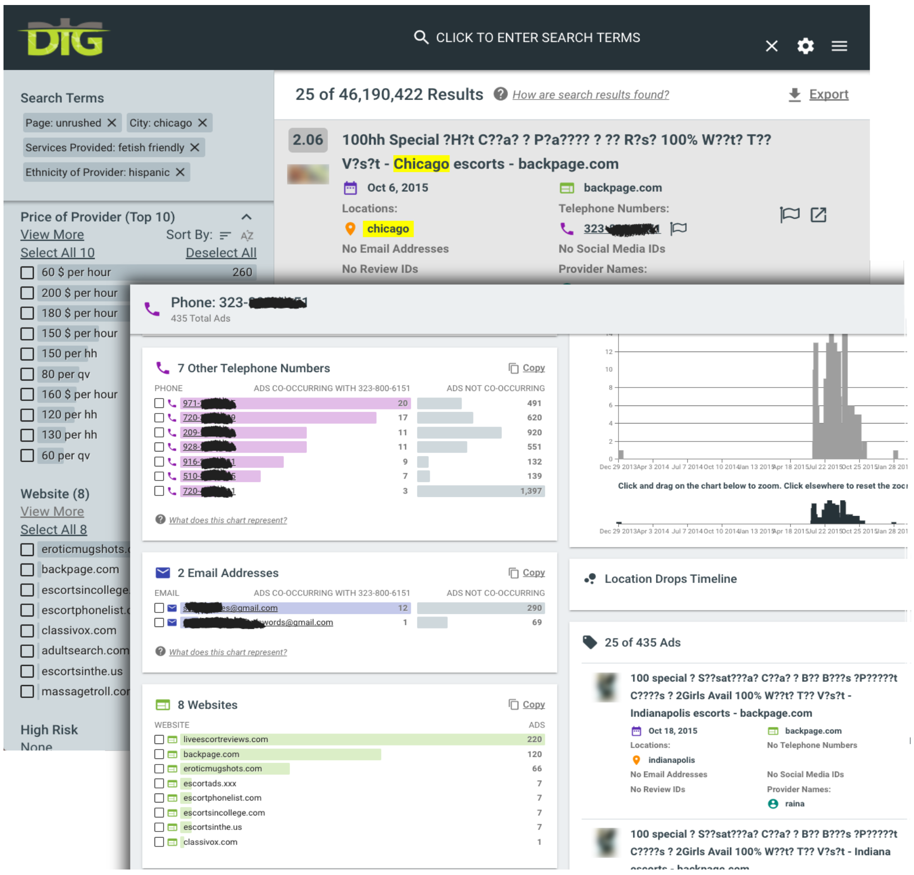
More importantly, the absolute performance of DIG suggested that the system was ready to be deployed for detailed investigative search to law enforcement. For example, on the *point fact* query category, which resembles *factoid*

[6] `https://github.com/usc-isi-i2/etk`

[7] An example of a category (cluster aggregate) was provided earlier in the introduction.

questions in the question answering literature, our performance was close to 70% on a key *mean average precision* metric[8].

5.1 Case Study



CLICK TO ENTER SEARCH TERMS

Search Terms

Page: unrushed ✕ City: chicago ✕ Services Provided: fetish friendly ✕ Ethnicity of Provider: hispanic ✕

Price of Provider (Top 10)
View More Sort By:
Select All 10 Deselect All

60 $ per hour	260
200 $ per hour	
180 $ per hour	
150 $ per hour	
150 per hh	
80 per qv	
160 $ per hour	
120 per hh	
130 per hh	
60 per qv	

Website (8)
View More
Select All 8

eroticmugshots.
backpage.com
escortsincollege.
escortphonelist.
classivox.com
adultsearch.com
escortsinthe.us
massagetroll.co

High Risk
None

25 of 46,190,422 Results How are search results found? Export

2.06 100hh Special ?H?t C??a? ? P?a???? ? ?? R?s? 100% W??t? T?? V?s?t - Chicago escorts - backpage.com
Oct 6, 2015 backpage.com
Locations: chicago
Telephone Numbers: 323[redacted]
No Email Addresses
No Social Media IDs
No Review IDs
Provider Names:

Phone: 323-[redacted]
435 Total Ads

7 Other Telephone Numbers Copy

PHONE	ADS CO-OCCURRING WITH 323-800-6151	ADS NOT CO-OCCURRING
971-[redacted]	20	491
720-[redacted]	17	620
209-[redacted]	11	920
928-[redacted]	11	551
916-[redacted]	9	132
510-[redacted]	7	139
720-[redacted]	3	1,397

What does this chart represent?

2 Email Addresses Copy

EMAIL	ADS CO-OCCURRING WITH 323-800-6151	ADS NOT CO-OCCURRING
[redacted]es@gmail.com	12	290
[redacted]words@gmail.com	1	69

What does this chart represent?

8 Websites Copy

WEBSITE	ADS
liveescortreviews.com	220
backpage.com	120
eroticmugshots.com	66
escortads.xxx	7
escortphonelist.com	7
escortsincollege.com	7
escortsinthe.us	7
classivox.com	1

12
10
8
6
4
2
0
Dec 29 2013 Apr 3 2014 Jul 7 2014 Oct 10 2014 Jan 13 2015 Apr 18 2015 Jul 22 2015 Oct 25 2015 Jan 28 201
Dec 29 2013 Apr 3 2014 Jul 7 2014 Oct 10 2014 Jan 13 2015 Apr 18 2015 Jul 22 2015 Oct 25 2015 Jan 28 201

Click and drag on the chart below to zoom. Click elsewhere to reset the zoo

Location Drops Timeline

25 of 435 Ads

100 special ? S??sat???a? C??a? ? B?? B???s ?P?????t C????s ? 2Girls Avail 100% W??t? T?? V?s?t - Indianapolis escorts - backpage.com
Oct 18, 2015 backpage.com
Locations: indianapolis
No Telephone Numbers
No Email Addresses
No Social Media IDs
No Review IDs
Provider Names: raina

100 special ? S??sat???a? C??a? ? B?? B???s ?P?????t C????s ? 2Girls Avail 100% W??t? T?? V?s?t - Indiana escorts - backpage.com

Fig. 3. A case-study illustrating complex investigative search in DIG. Details are provided in the text.

In Figure 3, we illustrate (using a specific case study) how the DIG interface, search engine and knowledge graph can be used to answer an actual investigative

[8] This is also why it was possible to conduct user studies on actual investigative domain experts in the first place, since the technology has to meet a minimum standard before it can be presented to real-world users.

question that requires profiling and identifying a potential *vendor*[9], also referred to as a *ring* or a *stable* in the online sex trafficking literature.

First, a hypothetical investigator searches on the leads that she has, namely that an escort who is potentially in the ring is in Chicago, is of hispanic ethnicity and often uses words like *fetish friendly* and *unrushed* in her ads to describe the services she offers. In the top left corner of the underlaid image in Figure 3, these search terms, as they are input to the search engine, are displayed. DIG returns a diverse set of results with *facets*; among the facets shown on the interface, for example, are *price* facets and *Web domain* facets.

Because the search is under-defined (many escorts meet the search conditions posed by the investigator), as it often is in investigative search scenarios, an investigator browses through the GUI, possibly exploring a few ads, and finds an ad that looks promising for eliciting further investigative leads. On this ad page (not shown in the figure), which contains such details as the extracted information from the ad, as well as activity timelines and images, the investigator clicks on a *phone number* associated with the escort in question. The search engine now displays an entity-centric page, where the entity is a phone number. This is a facility that is uniquely associated with investigative domains, and to the best of our knowledge, DIG is the only known system that has achieved it at scales of millions of phone numbers, and hundreds of millions of pages, in the human trafficking domain.

On the phone-centric page, the investigator finds information that would potentially take her months to discover, assuming the relevant pages were still online while the investigations were being conducted. The most important leads are a list of phone numbers that are *connected*, through a strong co-occurrence chain, to the phone under investigation. There are also email addresses, as well as a characterization of the levels of activity. Finally, to facilitate detailed investigations, we also list ads where these phone numbers occur. All of these ads are permanently retained in our back-end DFS. Ultimately, these ads and phone numbers can be used both to acquire evidence (possibly in retrospect) to support an ongoing prosecution, as well as to make the case for initiating a prosecution in the first place. We are aware of real-world cases in recent times that have used DIG for both purposes.

5.2 User Studies: Protocol and Observations

The GUI was evaluated by conducting controlled usability studies on 8 subject matter experts (SMEs) from 4 different US states. Due to confidentiality reasons, and because the results of the usability study are still being analyzed by NIST at the time of writing, we cannot release the affiliations of any of these experts other than that they are real-world users, and are affiliated with offices in the US that prosecute or investigate offenses like human trafficking in some

[9] Recall that vendors cannot be 'extracted' (since they do not explicitly advertise) and must be inferred through phone-email extractions and clustering.

capacity. Although we cannot release actual quantitative data concerning the GUI evaluations, we summarize the protocol and some observations.

Each SME was tasked with answering a set of 8 *lead generation* and 8 *lead investigation* questions, with 30 and 15 minutes allocated per question respectively. Each study (involving a single participant) was conducted in blocks of 2 hours, over a period of a week. Both DIG, and a second search system developed by a private enterprise, were evaluated in controlled settings. Both teams were allowed to hold a 45 minute training session to demonstrate the features of the GUI to the SMEs. No team had access to any of the test questions.

Lead generation questions are important for locating potential vendors (the latent entities) through careful scrutiny of published entities (especially locations and pseudo-identifiers, like phones) and their corresponding activity, while lead investigation questions have more explicit information needs. Each question requires significant exploration of the information space in the GUI.

The System Usability Scale (SUS) metric was used by NIST for assessing usability [20]. The DIG SUS mean score was higher than that[10] the second system, and within the margin of error, the DIG SUS score crossed the threshold of 70, widely believed to match real-world usability criteria.

We close this section with two important observations that emerged from the study, but that we believe are underestimated in the AI literature. First, *source retention* (in this case, the HTML page) was considered vital; even if the precision and the recall are above 90%, investigators still want an option to verify directly with the source to foster system trust. DIG offers this option. Second, it is important to *explain* why an algorithm produces an output that it does. In this sense, deep neural networks should be treated with caution, as their outputs are not obviously explainable. In current work, we are actively exploring methods for automatically annotating our extractions and outputs with feasible explanations, using techniques developed in recent research on explainable AI.

6 Future Work and Conclusion

The search engine was last evaluated by DARPA on 102 point fact (PF) questions, and 200 compound HT questions involving clustering and/or aggregations, against two competitive baselines participating in the MEMEX program. The KG on which system retrieval performance was tested was constructed from 90,000 HT webpages collected and annotated by DARPA. Scalability was tested by executing the system for all 302 questions on multiple domain discovery corpora, with the largest being a multi-terabyte dataset containing 53.7 million webpages. On average, our system could reformulate and answer each query in less than 12 seconds on the largest corpus, using a regular internet connection.

The success of the Web has had the unfortunate consequence of lowering the barrier-to-entry for players in illicit domains. With judicious use of semantic technology, this trend can not only be reversed, but vendors in illicit domains can

[10] Significance testing of this claim is currently under way.

be located, investigated and prosecuted in a timely manner by resource-strapped agencies. In this paper, we presented and described an investigative search system that uses knowledge graph construction and multi-interface querying to support these goals. Semantic Web components that played an important role include (1) the SPARQL language for internal query representation, (2) shallow ontologies inspired by the likes of schema.org for representing investigative domains, and (3) external Linked Data knowledge bases like Geonames for information extraction. The system is currently in use by hundreds of law enforcement agencies to combat human trafficking, has undergone rigorous usability studies that employed actual domain experts, and is extensible.

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.

References

1. Inferlink r&d capabilities. `http://www.inferlink.com/our-work#research-capabilities-section`. Accessed: 2017-04-28.
2. 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*, pages 133–138. IEEE, 2016.
3. 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*, 18(10):1411–1428, 2006.
4. H. Chen. *Dark web: Exploring and data mining the dark side of the web*, volume 30. Springer Science & Business Media, 2011.
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*, pages 1–12. ACM, 2009.
6. A. Doan, A. Halevy, and Z. Ives. *Principles of data integration.* Elsevier, 2012.
7. A. K. Elmagarmid, P. G. Ipeirotis, and V. S. Verykios. Duplicate record detection: A survey. *Knowledge and Data Engineering, IEEE Transactions on*, 19(1):1–16, 2007.
8. 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. Technical report, DTIC Document, 2012.
9. 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*, 16(1):24–33, 2012.

10. 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*, page 90. Academic Conferences Limited, 2013.
11. S. Gupta, P. Szekely, C. A. Knoblock, A. Goel, M. Taheriyan, and M. Muslea. Karma: A system for mapping structured sources into the semantic web. In *Extended Semantic Web Conference*, pages 430–434. Springer, 2012.
12. 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.
13. L. Hirschman and R. Gaizauskas. Natural language question answering: the view from here. *natural language engineering*, 7(04):275–300, 2001.
14. 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*, pages 1301–1302. ACM, 2007.
15. 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*, pages 4344–4353. IEEE, 2016.
16. A. Jain, A. Doan, and L. Gravano. Sql queries over unstructured text databases. In *2007 IEEE 23rd International Conference on Data Engineering*, pages 1255–1257. IEEE, 2007.
17. 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*, pages 589–598. ACM, 2012.
18. F. Niu, C. Zhang, C. Ré, and J. W. Shavlik. Deepdive: Web-scale knowledge-base construction using statistical learning and inference. *VLDS*, 12:25–28, 2012.
19. P. Saleiro, J. Teixeira, C. Soares, and E. Oliveira. Timemachine: Entity-centric search and visualization of news archives. In *European Conference on Information Retrieval*, pages 845–848. Springer, 2016.
20. J. Sauro. Measuring usability with the system usability scale (sus), 2011.
21. E. U. Savona and S. Stefanizzi. *Measuring human trafficking*. Springer, 2007.
22. 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*, pages 205–221. Springer, 2015.
23. 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*, pages 125–134. ACM, 2012.