"THEME ARTICLE", "FEATURE ARTICLE", or "COLUMN" goes here: The theme topic or column/department name goes after the colon.

Investigative Knowledge Discovery for Combating Illicit Activities

Mayank Kejriwal
USC Viterbi School of Engineering

Pedro Szekely
USC/ISI

Craig A. Knoblock
University of Southern California (USC)

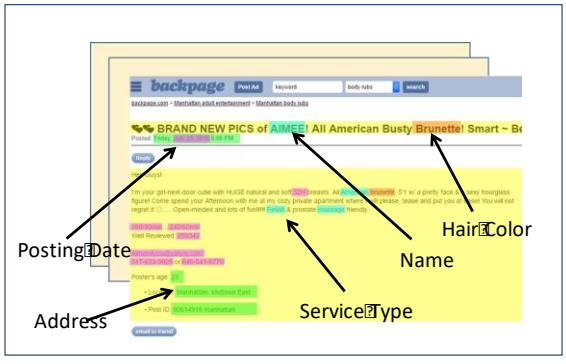
Developing scalable, semi-automatic approaches to derive insights from a domain-specific Web corpus is a longstanding research problem in the knowledge discovery community. The problem is particularly challenging in *illicit* fields, such as human trafficking, where traditional assumptions concerning information representation are frequently violated. In this article, we describe an end-to-end *investigative knowledge discovery* system for illicit Web domains. We built and evaluated a prototype, involving separate components for information extraction, semantic modeling and query execution, on a real-world human trafficking Web corpus containing 1.3 million pages, with promising results. The prototype includes a GUI currently used by US law enforcement agencies to combat illicit activity.

Knowledge discovery from raw corpora is a broad research area that involves diverse tasks such as ontology engineering[1], information extraction[2], information retrieval[3] and visualization[4]. In this article, we assume that a set of *domain experts* (typically, law enforcement agencies) is interested in knowledge discovery of an *investigative* nature in an *illicit* Web domain such as human trafficking (HT).

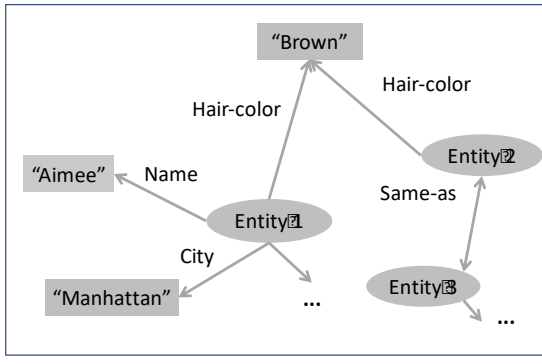
Many sub-problems listed above are already known to be difficult in traditional fields like news and social media. Illicit fields, being untraditional and relatively under-studied, are challenging in several different ways. First, such fields tend to be surprisingly *diverse*, with the distribution of page count across Web domains (e.g. *backpage.com* in human trafficking) exhibiting a *meso-kurtic* trend. That is, the 'tail' of an illicit field is long, as opposed to a distribution where analyzing just one or two Web domains is sufficient. A direct implication is that quality and coverage of system modules can vary widely across a domain-specific corpus; for example, it can be hard to acquire training data that generalizes easily across all Web domains.

Second, HT webpages often employ *information obfuscation* for key attributes like phone-number to deter automatic searches by law enforcement. Information obfuscation includes obscure language models, excessive use of punctuations and special characters, presence of extraneous, hard-to-filter data (e.g. *embedded* advertisements or artefacts) in Web pages, irrelevant pages, lack of representative examples for (supervised) extractors, data skew and heterogeneity. Many webpages exhibit more than one problem; a sampling of some pages in the human trafficking corpus available to us revealed that some form of obfuscation was almost always employed.

Due to their non-traditional content, direct adaptation of existing techniques from traditional fields is problematic. Even simple NLP tools like stemmers and tokenizers must be adapted before they can be deployed on illicit Web corpora. Consider, for example, the representative text fragment *'AVAILABLE NOW! ?? - (1 two 3) four 5 six - 7 8 nine 0 - 21'*. Both the phone number (123-456-7890) and age (21) can be difficult to extract for Named Entity Recognition (NER) tools trained on traditional text corpora.



(a) Domain-specific information extraction on a million-page human trafficking web corpus



(b) The Web corpus structured as an interconnected *knowledge graph*

Point Fact

What is the ad with the earliest post date containing number 7075610282?

```
PREFIX qpr:
SELECT ?ad
WHERE {
    ?ad a qpr:Ad ;
        qpr:phone '7075610282' ;
        qpr:posting_date ?posting_date .
    }
ORDER BY ?posting_date
LIMIT 5
```

Cluster

List all ads connected via a shared phone number, email, or street address to the phone number 9177362938 that feature a cuban escort

```
PREFIX qpr:
SELECT ?cluster ?ad
WHERE {{
    ?cluster a qpr:cluster ;
        qpr:seed '9177362938' ;
        qpr:ad ?ad }
    ?ad  qpr:ethnicity 'cuban'
    }
```

Aggregate

List all ads in Seattle, WA that include an ethnicity in the ad text. In the answer field, concatenate and list ethnicities

```
PREFIX qpr:
SELECT ?ethnicity (count(?ad) AS
?count)
(group_concat(?ad;separator=',') AS
?ads) WHERE {
    ?ad a qpr:Ad ;
        qpr:location 'Seattle, WA' ;
        qpr:ethnicity ?ethnicity }
GROUP BY ?ethnicity
ORDER BY DESC(?count) LIMIT 1
```

(c) Three categories of investigative questions of interest to investigators, with SPARQL-like templates (blue)

Figure 1: An illustration of important technical steps in the investigative knowledge discovery problem.

CONTRIBUTIONS

Given the challenges described above, and the social utility of using technology to combat illicit activities, we propose an investigative knowledge discovery approach to perform domain-specific search in dynamic, illicit fields. Our approach takes as input, raw webpages crawled over multiple Web domains, and uses a composite set of tools, including high-recall information extraction (Figure 1a) and semantic typing, to structure the multi-domain corpus into a semi-structured knowledge graph (Figure 1b). Being high-recall, our knowledge graph construction (KGC) approach (based on Domain Insight Graphs or DIG4) is designed to handle illicit-field challenges such as information obfuscation without trivially degrading precision.

A key contribution not included in the DIG KGC is a robust entity-centric search (ECS) engine that permits investigators to pose analytical questions to the system from three categories, denoted here as point fact, cluster and aggregate (Figure 1c). Together, the three categories are expressive enough to capture a wide class of investigative information needs, and also have a fairly regular syntax. Consequently, questions in each category can be composed by instantiating (and if necessary, supplementing with more constraints) a template that is itself written in a simple SPARQL5-like structured language, using terms from a specified domain ontology, as subsequently discussed. The ECS engine implements a number of query-reformulation strategies designed to handle various kinds of noise introduced during KGC.

RELATED WORK

The technical innovations in this work draw on two broad fields, namely *knowledge graph construction* and *structured information retrieval*.

Knowledge graph construction (KGC) is the process of structuring a raw corpus of unstructured data into a *knowledge graph*, defined as a directed, labeled multi-relational graph where nodes are (possibly multi-type) entities and attributes, and labeled edges are either entity-entity or entity-attribute relationships. Important KGC steps[4] include information extraction, entity resolution, semantic typing and clustering. We subsequently detail these steps, and the specific technology used in our approach. We note that, although individual steps have been well-explored in the literature, composite KGC systems are still rare. One exception is the Domain Insight Graph (DIG) system[4], developed in our own group, which was used as the KGC component in our approach. Another example is the DeepDive[5] architecture.

Structured information retrieval has traditionally been linked to entity-centric search, as well as search over RDF datasets, in the Semantic Web[3]. A popular line of work explores the issuing of keyword queries over structured datasets, and using the structure in the data to retrieve more relevant, entity-centric results[6]. In contrast, the queries explored in this paper are more expressive (Figure 1c), but executing such queries using traditional triplestores is not straightforward both due to scale, as well as the noisy, incomplete nature of the constructed knowledge graph. A robust query engine must account for the challenges in illicit fields that were earlier described, including values that are obfuscated or not properly extracted. The proposed query execution engine is designed to handle such challenges.

We also draw attention to the *non-investigative*[7–8] analytical potential of the knowledge discovery system in this paper. Although this aspect is not detailed herein, we note that the system can be used to support or refute certain hypotheses[9] especially prevalent in popular imagination e.g. whether human trafficking activity increases contemporaneously with events like Super Bowl[10]. Carefully designed studies can be used to also study the *socio-ethnic* impact of HT by collecting statistics e.g. on *ethnicity* extractions.

APPROACH

As a preliminary step, a *domain discovery* team, typically comprising both people and software-driven agents in a reinforcement learning paradigm, is initially engaged to crawl the Web and scrape an inclusive Web corpus that contains pages of interest from multiple Web domains. Sim-

ultaneously, the domain experts collaboratively model the field by constructing a *domain ontology* to support subsequent analysis in a structured manner. Taking HT as an illicit field example, some terms in the ontology are generic (e.g. date, name), but others tend to be domain-specific (e.g. hair-color).

Given these inputs (raw webpages from multiple Web domains, and a domain ontology), we developed a real-time approach that can answer questions in the templated format (Figure 1c), and that also powers a GUI for visual analytics.

Before detailing the approach, we briefly describe the ontology engineering process. While designing detailed ontologies for traditional fields like biomedicine is a well-studied problem[1], requiring lengthy collaboration between domain experts and knowledge engineers, experts in illicit fields prefer *broad, shallow ontologies* that can be easily constructed and visualized, and have high coverage of concepts without axiomatic or functional detail. In our implemented prototype, for example, the HT domain ontology was constructed by supplementing a relevant subset of generic terms from the widely-used *schema.org* vocabulary[11] with domain-specific terms. The ontology was defined and finalized in a short time-period, with mostly remote collaborations. Constructing other illicit-field ontologies is expected to follow a similar process.

Figure 2 illustrates the architecture of the approach. We assume that the original corpus has been placed into a distributed file system (e.g. HDFS). The corpus is first processed using a sequence of knowledge graph construction modules. Next, the constructed knowledge graph is indexed and loaded into a key-value database. Using a custom entity-centric query execution engine, this database is used to support both expressive query execution (using a command-line frontend) and visualization (using a graphical frontend).

Information Extraction

Since robust *information extraction*[2] is key to constructing knowledge graphs that can support fine-grained information retrieval, we accommodate a suite of extractors that are *diverse* along several dimensions, including performance, required supervision as well as mode of supervision (e.g. manually-crafted regular expressions vs. annotated training data). Specifically, we construct a *high-recall* knowledge graph by *mapping* each term in the domain ontology to an appropriate set of extractors. We use Conditional Random Field-based extractors for non-numerical, closed-category terms like *hair-color* and *eye-color* on which we found them to quickly generalize (even across Web domains) using only a few labeled annotations, semi-automatic wrapper-based extractors for *structured HTML elements* embedded in the webpage, regular expression-based extractors for highly obfuscated, but still constrained, terms like *phone-number*, *social-media-id* and *review-site-id* that tended to be specific to groups of Web domains, text scrapers for extracting useful *text*, *title* and *descriptions* from the webpage, and semantic lexicons for *non-obfuscated* terms like *location* that can be reliably extracted using open resources. A good example of an *exhaustive* semantic lexicon is GeoNames[12], which we used for extracting location attributes (e.g. city, state and country) from the text in the webpage. In current work, we are also exploring more advanced capabilities of GeoNames, including both entity *disambiguation* (e.g. Paris, Texas vs. Paris, France) and collective *inference* for location attributes not directly mentioned in the webpage.

In the general case, extractions are *multi-valued*; it is also not uncommon for a certain attribute (e.g. name) to not get extracted at all from a given webpage, despite being present in the underlying data. As noted above, the knowledge graph also contains freeform text attributes. In other words, the graph is both *semi-structured* and *noisy*. This is why one cannot *directly* query the graph by storing it in a triplestore (or database) and executing a SPARQL query *verbatim*. We subsequently outline more robust search specifically designed for noisy graphs.

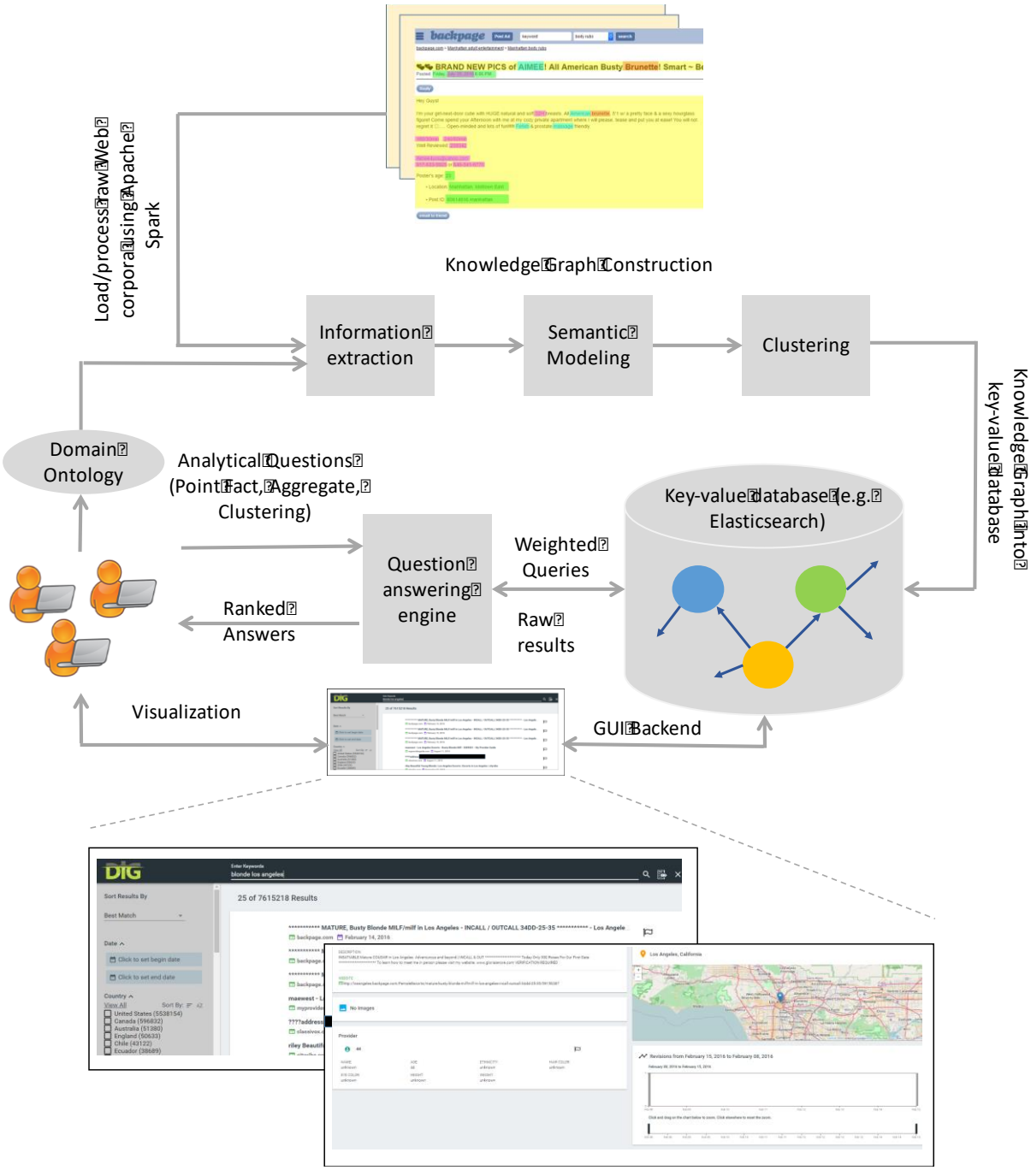


Figure 2: Architectural overview of the approach, including both the command-line front end (question answering by posing equivalent SPARQL-like queries; see Figure 1c) and the graphical frontend that is supported by the underlying Domain Insight Graph (DIG) infrastructure.

Semantic Modeling

Semantic modeling[13] follows the information extraction step, and is used both to map extractions to terms in the domain ontology, as well as to heuristically remove meaningless extractions (e.g. containing only HTML tags). In some cases, where there are only one-to-one mappings (e.g. between the CRF extractor for *hair-color*, and a similar *hair-color* term in the ontology), the mapping is trivial. Complications arise when extractor semantics are unknown for a given webpage,

or when there are multiple extractions for a single term in the ontology (e.g. multiple regular expression-derived date instances for the *date* ontology term).

We use the Karma semantic modeler[13], which was developed in our research group, and is already widely deployed in numerous contexts. Karma is useful both for performing semi-automatic semantic mapping, and for executing domain-specific scripts for operations such as data cleaning.

Vendor Resolution

Although information extraction and semantic modeling are sufficient for answering *point-fact* and *aggregate* questions, *cluster* questions demand an additional layer of abstraction in the constructed knowledge graph. In the real-world, players in illicit fields seldom act alone but offer their services through so-called *vendors*. In HT, vendors may represent intermediaries and even specific locations fronted by a legitimate business (e.g. a massage parlor). Automatically 'clustering' individuals into such latent vendors is a non-trivial, and in many cases, an ill-defined *entity resolution* problem that is highly dependent on domain-specific needs and is referred to herein as *vendor resolution*.

To perform vendor resolution reliably, we execute a *connected components* graph algorithm on manually specified *pseudo-identifiers* such as phone numbers and email addresses to discover vendor entities. We use a blacklist of identifiers, discovered through a graph analytics algorithm and manually verified, to circumvent *data skew* (usually a result of both noisy extractions and a few rare identifiers that bridge otherwise disconnected components). In the most recent prototype, we integrated scalable *random walk*-based techniques to fuzzily discover entities while demonstrating robustness to data skew.

Indexing and Loading

Once structured entities have been vendor-resolved, we load entities and vendors into a key-value Elasticsearch database, and index all attributes in a variety of ways. For instance, we support both *raw* look-ups where a string must exactly match against an attribute for the corresponding entity to be retrieved, as well as *token-based* lookups that are more common in search engine-style IR. Hybrid indexing strategies, that rely on both structured and unstructured data, are key to the functioning of query execution, described below. In the rest of this discussion, we refer to the data loaded into Elasticsearch as an *indexed knowledge graph*.

Command-line Frontend: Question Answering

Given that questions can be presented using structured instantiated templates (Figure 1c), a natural solution to the question answering problem is to write scripts that convert each instantiated SPARQL template into an *equivalent* Elasticsearch query, expressed as a weighted tree of key-value queries. In principle, although not proved here rigorously, such a *semantics-preserving* conversion is always possible for point-fact questions, while multiple queries are required for cluster questions, and a post-processing module is required for aggregate questions. Unfortunately, semantics-preserving query conversions can fail when the knowledge graph contains noisy and missing elements.

To achieve a more robust outcome, we approach the problem by designing a *fuzzy* query execution engine that converts an instantiated SPARQL template into an Elasticsearch query using 'conversion strategies', which may *not* be semantics-preserving. We implement three such strategies (described below) in our question answering engine, and empirically verify the effectiveness of the engine in the next section. Intuitively, each strategy yields one sub-query that is assigned a weight, indicating its importance. All weighted sub-queries are collected and integrated into a single tree query that is then executed over the Elasticsearch server.

The three conversion strategies that are currently implemented in our prototype are briefly described below:

1. Semantics-preserving strategy: This strategy is designed to rigidly interpret the original query. For a given entry in the Elasticsearch database to be retrieved by the server, all conditions in the query, as stated, must be fulfilled. Using the point-fact query in Figure 1c as an example, a semantics-preserving query requires at least one posting-date and phone-number attribute to have been extracted from a page, in addition to at least one extracted phone-number attribute value mapping exactly to the literal '7075610282', for that page to be retrieved with non-zero score.
2. Fuzzy strategy: This strategy interprets each condition in the original query like an OPTIONAL clause. The resulting Elasticsearch query assigns scores to entries in proportion to the number of fulfilled clauses, as well as the degree of similarity. Using the running example above, each condition (existence of attributes, as well as specific attribute value mappings) is now assigned optional semantics. As long as one condition is met, Elasticsearch semantics guarantee that the corresponding page will be returned with non-zero score.
3. Information Retrieval (IR) strategy: The IR strategy only uses the text attributes (e.g. description and title) for search, and ignores all structured attributes in the indexed semi-structured knowledge graph. It ignores all ontological terms (e.g. phone-number) in the query and converts all specified literal values (e.g. '7075610282') into 'keywords' that are searched for in the indexed text attributes. We note that this is typically the least restrictive strategy since an entity in the database receives a non-zero (albeit, low) score against an issued query if a single literal value in the query matches a token in a text attribute. Returning to the running example, as long as the literal '7075610282' is found inside an extracted text field, the page gets retrieved with non-zero score.

All sub-queries are currently assigned equal weight (1/3) by default for all three query categories, as this option was found to work well empirically. One possible reason for this empirical finding is that typically, strategy I tends to dominate both strategy II and strategy III when it yields non-zero score for a document, and similarly, strategy II dominates strategy III when it yields non-zero score. In essence, the final score 'favors' the highest score obtained by the most *constrained non-zero* scoring strategy. We also note that the engine is designed to be extensible and to permit easy modification, addition and removal of strategies and weights, if deemed necessary.

The engine operates in *real-time* by virtue of relying on the hybrid indexing strategies briefly described earlier in *Indexing and Loading*. The strategies are also designed to be *generic*; given a shallow ontology in *any* field, and a mapping from terms in the ontology to the various Elasticsearch indexes, the engine can convert any instantiated query to an Elasticsearch query (itself a weighted combination of sub-queries), execute the query over the Elasticsearch server, and return a ranked list of *fine-grained* answers. The *structured fields* obtained via information extraction are necessary both for Strategy I and II above (i.e. query reformulation and search), as well as fine-grained *answer retrieval* (e.g. a ranked list of not just relevant pages, but also *dates* extracted from those pages, as in Figure 1c)

An interesting implementation question is whether the knowledge graph can be stored in a triplestore like Virtuoso.[14] While it is technically possible to implement query strategies by strategically loosening constraints on the original SPARQL queries, both the scaling and setup capabilities proved to be important impediments. Elasticsearch, for example, is designed to be *horizontally scalable* on commodity infrastructure and is offered by major cloud vendors. On-demand scaling is important as the system is expected to supports corpora containing hundreds of millions of ads in the long run. Furthermore, it is not clear if the aforementioned query execution strategies are amenable to triplestore implementations. Finally, fast NoSQL databases like Elasticsearch also provide robust *visualization* support, such as in the current GUI.

Graphical frontend: DIG

The current GUI facilitates intuitive exploratory browsing using faceted search, map plugins, charts, links and support for both structured and unstructured display of attributes. The GUI is currently supported by the open-source *Domain Insight Graph* (DIG) architecture that was developed in our group in an earlier phase of the DARPA MEMEX program. We refer the reader to the cited work on DIG[4] for more details. Currently, we are also exploring a merging of both

the command-line and graphical frontends into a *unified* GUI, supported by a combination of DIG, Elasticsearch and the structured IR strategies.

PROTOTYPE EVALUATION

DARPA MEMEX Human Trafficking Challenge

We implemented a prototype of our approach for the 4-week Human Trafficking Challenge that was organized by the DARPA MEMEX program in the summer of 2016. Three teams developed analytical approaches for the question answering component of the challenge, of which our approach was the only one that was fully end-to-end (i.e. integrated *both* knowledge graph construction and query execution). Collaboration was encouraged between the participants for implementing sub-components; e.g. one of the teams utilized both our knowledge graph and that of another team, but independently developed their question answering engine. The evaluations were carried out in two test phases.

In the first test, each system was input a multi-domain Web corpus of 1.3 million pages, most of which are from HT (but a significant number comprises irrelevant pages e.g. job ads), and a set of 10 point-fact questions, 16 cluster questions and 14 aggregate questions respectively. We denote this corpus as the *exhaustive corpus*. All teams submitted their answers to the 40 questions, which were evaluated *externally* by the challenge organizers. At no point during the challenge and system fine-tuning was the ground-truth answer set released to the participants.

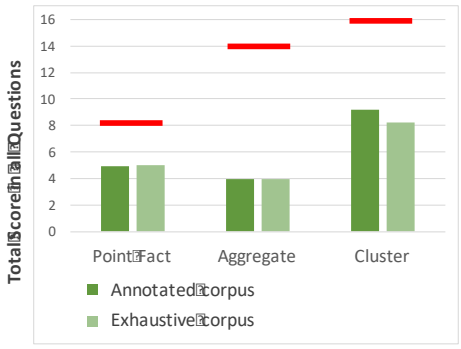
In the second test, a smaller corpus of about 4000 pages from multiple Web domains, all of which had been manually annotated with *ground-truth extractions* by domain experts in a prior phase, was released to all participants without any annotations. This so-called *annotated corpus* contained very few irrelevant pages, and pages containing the correct answers to the 40 questions (whether directly or derivatively) were guaranteed to be in this corpus. However, just like in the first test phase, the ground-truth extractions were withheld; teams executed queries on constructed knowledge graphs. Answers re-submitted by the participants were again externally evaluated. The research agenda in conducting the two phases, and comparing results, was to determine the *robustness* of each of the systems to noise, scale and irrelevance.

Evaluation Procedure

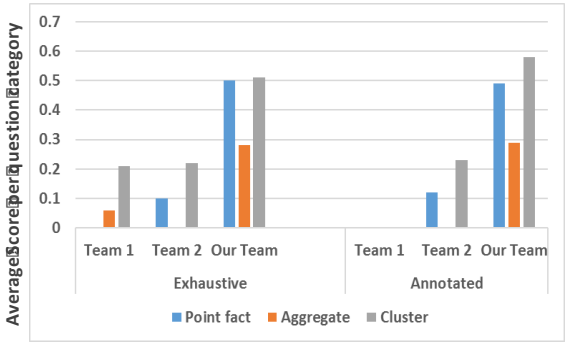
For *each* of the 40 questions, a *ranked, scored list* of *tuples* is returned. Each tuple intuitively corresponds to a webpage (representing an underlying *advertisement* featuring a human trafficking victim); hence an ID of the webpage is always included in the answer. Also included are the finer-grained answers (e.g. a post-date per returned ad in the point-fact question in Figure 1c).

To evaluate each tuples-list using the *annotated corpus*, the organizers first removed all tuples from the list corresponding to the few pages in the corpus that had *not* been manually annotated (and for which the relevance statuses were unknown). For each such *pruned* list, the *Normalized Discounted Cumulative Gain (NDCG)* score was computed. NDCG is a popular Information Retrieval (IR) metric that logarithmically dampens the relevance scores of documents the lower they are ranked in a retrieved list. Because it is normalized, the score of the overall list (which is the sum of individual scores) is guaranteed to be between 0 and 1.

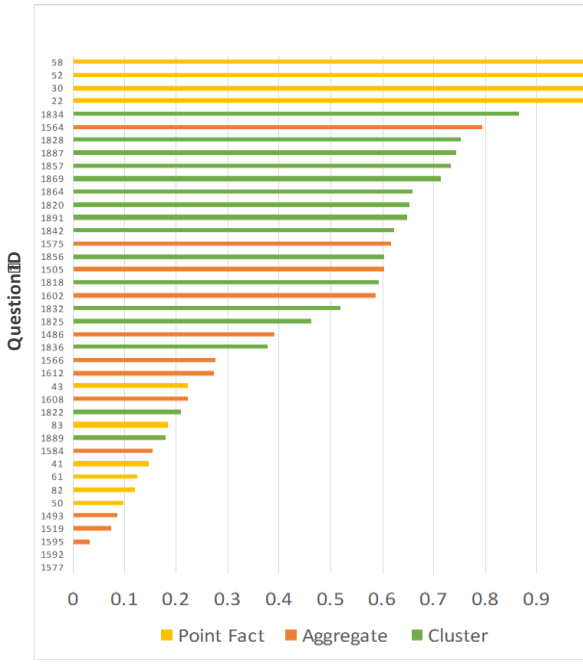
Since NDCG needs the relevance score of a retrieved item (in this case, a tuple) in the list, it is appropriate for the question answering task. The organizers computed the relevance scores in two different ways. The *automated NDCG* was computed in the classic IR fashion by ignoring all extraction fields in the answer tuples, and only assigning a tuple a 0-1 score based solely on whether it contained the correct page ID. To evaluate the extractions, the *manual NDCG* was computed by a challenge organizer judging each tuple and assigning it a 0-1 score if the information content of the overall tuple was determined satisfactory for investigative needs. We note that subjective relevance judgments are typically unavoidable in IR applications.



(a) Manual NDCG of our approach on both corpora. Red lines indicate perfect scores



(b) Manual NDCG of participating teams in the DARPA MEMEX Human Trafficking challenge on both corpora



(c) Automated NDCG score per question of our approach on annotated corpus

Figure 3: Three representative results from prototype evaluations in the DARPA MEMEX Human Trafficking challenge held in July-August 2016 in Washington D.C.

Results

We report three representative sets of results in Figure 3 to illustrate the performance of our approach, on both the annotated and exhaustive corpora, using the two NDCG metrics. Figure 3a shows that, on the manual NDCG metric, we retrieve relevant answers to questions from each category, with best relative performance in the point-fact category. The result also illustrates the robustness of our approach with respect to the *size* and *noise* in the two corpora. Specifically, we find that the approach does not get confused by the many irrelevant pages and extractions in the exhaustive corpora that are produced during knowledge graph construction and indexing.

Figure 3b shows that, of all participating prototypes in the challenge, our prototype performed the best, and was the only one with non-zero results on *all* question categories on *both* corpora. A per-query result breakdown in Figure 3c illustrates the stability of our results. In other words, our scores are not 'skewed' by a few well-performing queries; non-zero scores (on the automated NDCG) are achieved for almost all queries. We believe that both high-recall knowledge graph construction, as well as robust question answering, contributed to these scores.

Error Analysis

In a post-evaluation phase conducted by DARPA in November, we analyzed *point-fact* questions on which the system did not retrieve the correct answer in the top 1. Our observations illustrated a variety of underlying causes, both simple and complex. Simple causes included *misspellings* in fields like *name* (e.g. Asheera in the query vs. Asheerah in the document), and *heterogeneous formats* (e.g. height in inches vs. centimeters), which nevertheless require ongoing sophisticated engineering (e.g. phonetic indices to handle name misspellings) due to their ad-hoc nature.

Complex causes included *text homogeneity* (e.g. ads that are copies of one another, but have minor differences such as name and hair-color), which especially causes problems for the IR strategy, and *difficult extractions* (e.g. due to irregularly formatted HTML elements like tables). The latter problem is especially difficult to resolve because of the *long-tail* nature of a multi-Web domain human trafficking corpus. Cluster and aggregate question error analyses yielded similar findings, since technically, these questions are involved variants of point-fact questions. With more *field exploration* over time, we believe that performance on all three query categories will uniformly improve.

PRACTICAL IMPLICATIONS

NDCG results in Figure 3 show that the system is ready for use in real-world operational scenarios involving *point-fact* and *cluster* queries. We note that, while point-fact queries can be answered using the current keyword-style GUI, answering cluster queries requires much manual effort, including deep analysis by domain experts. Results in Figure 3b show that, with 50-60% average NDCG per cluster question, this task can largely be automated, allowing law enforcement to quickly query for, and uncover, *latent* 'vendors' providing human-trafficking services through advertisement of victims. The results have encouraged us to start merging the command-line prototype into the current GUI. Simultaneously, we are seeking to improve aggregate query performance.

FUTURE WORK

Due to its overwhelming success, the Web has attracted many players from illicit enterprises. Combating such activity requires interdisciplinary research from the AI community. A broad goal, towards which we are making steady progress, is to enable *non-technical* domain experts to deploy the system with minimal effort, both serially and in Big Data ecosystems like Spark, in *new* investigative fields such as illegal online weapons sales. We are also making our information extractors relatively modular, so that they operate independently of each other and can be updated, replaced or otherwise modified by a domain expert as deemed fit. Lastly, we are incorporating feedback from our users (mainly law enforcement agencies and DARPA) into the new GUI to better facilitate search.

ACKNOWLEDGEMENTS

The authors thank Lingzhe Teng and Amandeep Singh for their valuable contributions, and the anonymous reviewers for their helpful feedback. 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. Kotis, K., & Vouros, G. A. (2006). Human-centered ontology engineering: The HCOME methodology. Knowledge and Information Systems, 10(1), 109-131.
2. Chang, C. H., Kayed, M., Girgis, M. R., & Shaalan, K. F. (2006). A survey of web information extraction systems. IEEE transactions on knowledge and data engineering, 18(10), 1411-1428.
3. Hogan, A., Mellotte, M., Powell, G., & Stampouli, D. (2012, May). Towards fuzzy query-relaxation for RDF. In Extended Semantic Web Conference (pp. 687-702). Springer Berlin Heidelberg.

4. Szekely, P., Knoblock, C. A., Slepicka, J., Philpot, A., Singh, A., Yin, C., ... & Stallard, D. (2015, October). Building and using a knowledge graph to combat human trafficking. In International Semantic Web Conference (pp. 205-221). Springer International Publishing.
5. Niu, F., Zhang, C., Ré, C., & Shavlik, J. W. (2012). DeepDive: Web-scale Knowledge-base Construction using Statistical Learning and Inference. VLDS, 12, 25-28.
6. Hogan, A., Harth, A., Umbrich, J., Kinsella, S., Polleres, A., & Decker, S. (2011). Searching and browsing linked data with swse: The semantic web search engine. Web semantics: science, services and agents on the world wide web, 9(4), 365-401.
7. Dubrawski, A., Miller, K., Barnes, M., Boecking, B., & Kennedy, E. (2015). Leveraging publicly available data to discern patterns of human-trafficking activity. Journal of Human Trafficking, 1(1), 65-85.
8. Alvari, H., Shakarian, P., & Snyder, J. K. (2016, September). A non-parametric learning approach to identify online human trafficking. In Intelligence and Security Informatics (ISI), 2016 IEEE Conference on (pp. 133-138). IEEE.
9. Miller, K., Kennedy, E., & Dubrawski, A. (2016). Do Public Events Affect Sex Trafficking Activity?. arXiv preprint arXiv:1602.05048.
10. Mogulescu, K. (2014). The super bowl and sex trafficking. New York Times.
11. Ronallo, J. (2012). HTML5 Microdata and Schema. org. Code4Lib Journal, 16.
12. Wick, M. (2011). GeoNames. GeoNames Geographical Database.
13. Taheriyan, M., Knoblock, C. A., Szekely, P., & Ambite, J. L. (2016). Learning the semantics of structured data sources. Web Semantics: Science, Services and Agents on the World Wide Web, 37, 152-169.
14. Erling, O., & Mikhailov, I. (2009). RDF Support in the Virtuoso DBMS. In Networked Knowledge-Networked Media (pp. 7-24). Springer Berlin Heidelberg.
15. Prud, E., & Seaborne, A. (2006). SPARQL query language for RDF.

ABOUT THE AUTHORS

Mayank Kejriwal (Research Scientist): Mayank Kejriwal is a Computer Scientist in the Information Integration group in the Information Sciences Institute (ISI) at the USC Viterbi School of Engineering. Prior to joining ISI, he graduated in 2016 with his Masters and Ph.D. in Computer Science from the University of Texas at Austin under the supervision of Daniel P. Miranker. His dissertation has been published as a book in the Studies in the Semantic Web series. At ISI, he works extensively on problems of knowledge graph construction and information extraction. Along with Pedro Szekely and Craig Knoblock, he is co-authoring a textbook on knowledge graphs. He has teaching experience in databases, information integration, and Artificial Intelligence. kejriwal@isi.edu

Pedro Szekely (Research Associate Professor): Pedro Szekely is a research team leader at USC/ISI and a research associate professor in computer science. He received his Ph.D. from Carnegie Mellon University. His research focuses on the rapid construction of domain-specific knowledge graphs and integration of open source data from the Web and corporate databases. The systems developed in his group have been used to construct knowledge graphs on weapons trafficking, patent trolls, counterfeit electronics and human

trafficking. The knowledge graph for human trafficking is used by law enforcement agencies to identify victims and build legal cases against traffickers. He has decades of teaching experience in information integration. pszekely@isi.edu

Craig Knoblock (Research Professor): Craig Knoblock is a Research Professor of both Computer Science and Spatial Sciences at the University of Southern California (USC), Research Director of Information Integration at the Information Sciences Institute, and Associate Director of the Informatics Program at USC. He received his Ph.D. from Carnegie Mellon University in computer science. His research focuses on techniques for describing, acquiring, and exploiting the semantics of data. He has worked extensively on source modeling, schema and ontology alignment, entity and record linkage, data cleaning and normalization, extracting data from the Web, and combining all of these techniques to build knowledge graphs. He has published more than 250 journal articles, book chapters, and conference papers on these topics. Dr. Knoblock is a Fellow of the Association for the Advancement of Artificial Intelligence (AAAI), a Distinguished Scientist of the Association of Computing Machinery (ACM), a Senior Member of IEEE, past President and Trustee of the International Joint Conference on Artificial Intelligence (IJCAI), and winner of the 2014 Robert S. Engelmore Award. He has decades of teaching experience in information integration. knoblock@isi.edu