2017 IEEE 33rd International Conference on Data Engineering

NOUS: Construction and Querying of Dynamic Knowledge Graphs

Sutanay Choudhury [#1] Khushbu Agarwal [#1] Sumit Purohit [#1] Baichuan Zhang [#2]
Meg Pirrung [#1] Will Smith [#1] Mathew Thomas [#1]
[#1] *Pacific Northwest National Laboratory, 902 Battelle Blvd, Richland, WA 99352*
[#2] *Indiana University Purdue University Indianapolis, IN*

***Abstract*—The ability to construct domain specific knowledge graphs (KG) and perform question-answering or hypothesis generation is a transformative capability. Despite their value, automated construction of knowledge graphs remains an expensive technical challenge that is beyond the reach for most enterprises and academic institutions. We propose an end-to-end framework for developing custom knowledge graph driven analytics for arbitrary application domains. The uniqueness of our system lies A) in its combination of curated KGs along with knowledge extracted from unstructured text, B) support for advanced trending and explanatory questions on a dynamic KG, and C) the ability to answer queries where the answer is embedded across multiple data sources.**

I. Introduction

Data-driven applications critically depend on human experts who learn about a given domain through their interaction with data over time. An expert market analyst can quickly answer questions about recently trending products or explain the reason behind a new trend. However, manual approaches stops scaling as the data volume, throughput and diversity surges. Consequently, organizations are building custom knowledge bases (KB) using a combination of human-in-the-loop and data-driven techniques. These efforts range across diverse domains such as cyber-security [17], medical diagnosis [10] and retail [5].

Domain specific KB construction frequently begins with knowledge extraction from unstructured data such as publications, web pages or social media. The extracted knowledge is often aimed at improving the quality of search or recommendation algorithms. Another relevant area is Question-Answering (QA), which involves answering natural language queries on a large-scale text corpus [7]. Understanding domain specific vocabularies, correct classification of entities and their relationships is key to accomplishing these goals. Therefore, building on top of curated KBs such as FreeBase, YAGO etc. and augmenting with knowledge extracted by techniques such as Open Information Extraction [1] has emerged as a natural path for custom KG construction.

In short, our work is set in the following paradigm: 1) data arrives in streaming fashion and knowledge extraction happens continuously, 2) extracted knowledge is combined with curated knowledge to form a dynamic KG, and 3) a set of queries are executed on the dynamic KG. Executing queries on the KGs provides an unique analytics perspective. Since each relationship in a KG is potentially extracted from different data sources, it allows us to connect the dots across multiple data sources. In contrast, systems that return a passage of text from the best matching document does not provide such capabilities.

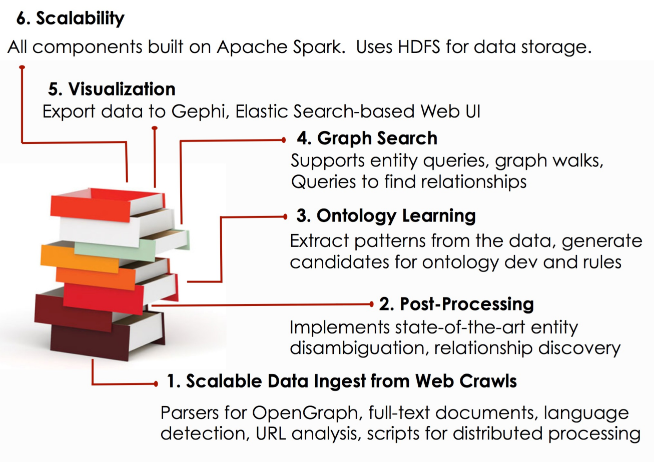
6. Scalability
All components built on Apache Spark. Uses HDFS for data storage.
5. Visualization
Export data to Gephi, Elastic Search-based Web UI
4. Graph Search
Supports entity queries, graph walks, Queries to find relationships
3. Ontology Learning
Extract patterns from the data, generate candidates for ontology dev and rules
2. Post-Processing
Implements state-of-the-art entity disambiguation, relationship discovery
1. Scalable Data Ingest from Web Crawls
Parsers for OpenGraph, full-text documents, language detection, URL analysis, scripts for distributed processing

Fig. 1. Various components of NOUS.

A. Technical contributions

Our proposed demonstration will showcase NOUS [1] (Figure 1), an open source system [2]. Following are the primary contributions of NOUS.

1) We develop a framework to build domain specialized knowledge graphs by fusing curated KBs with extracted knowledge. This is in contrast to most systems who either leverage on curated KBs such as Freebase or work on extracted knowledge. We view the Knowledge Graph construction as an incremental process and develop a family of algorithms designed for dynamic graphs.
2) We focus on two popular classes of domain-specific application needs: 1) discovering trends in streaming data and 2) answering explanatory (why-like) questions. We implement the former via a novel algorithm for streaming graph mining (section 3.5). We implement (2) by augmenting state of the art path-ranking algorithms with a *coherence* metric based approach (section 3.6).

[1] NOUS means "experiential knowledge" in Greek, one that is gathered over time.

[2] **https://github.com/streaming-graphs/NOUS**

2375-026X/17 $31.00 © 2017 IEEE
DOI 10.1109/ICDE.2017.228

3) We execute the queries on a dynamically updated Knowledge Graph. This allows NOUS to support queries whose answers are composed from multiple data sources.

Rest of the paper is organized as follows. Section 1.2 illustrates a primary use case for motivation. Section 2 provides an overview of related literature. Section 3 describes the details of various technical components and section 4 concludes with specific features of the demonstration.

B. *Use Case*

Civilian use of Drones is a key emerging technology today. Finance analysts, law enforcement agencies need to track emerging drone manufactures, novel applications of drones, and identify safety issues. Our goal is to build a system that can ingest information from web crawls and articles from trusted news sources to continuously stay abreast of information around drones. From the user perspective, an finance analyst may hypothesize about a startup being the acquisition target for a novel drone-based technology, or a security analyst will want to reason about why a non-military organization such as Windermere may employ drones in their operations (Figure 2). Figure 2 shows a sample of the drone graph generated by fusing knowledge from YAGO2 and Wall street journal articles. The lines in red and blue indicate facts available from curated KB and facts learned from web data respectively. Each fact is assigned a probability value of it being true, learned using the Link Prediction module.

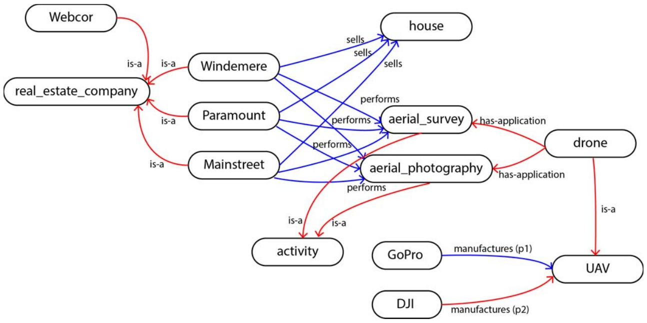

Fig. 2. Example of a Knowledge Graph tracking emerging technologies.

II. Background and Related Work

Knowledge Graphs with their ability to represent complex relationships between real world entities have become the de facto standard to store KBs. Knowledge graph construction from web data [6], [5], [7], [11] has been studied comprehensively over the last decade. *Deshpande et al* [5] provide an in depth discussion of the process and associated challenges. Openly available KBs like YAGO [12], Freebase [2] and NELL [3] provide massive amount of highly confident triples. We view these general purpose KB's as complimentary, to be used in conjunction with NOUS to build a custom domain KG. Most of these KGs or frameworks are limited in their querying capabilities. While algorithms for querying or mining dynamic graphs have been studied [4], much of the research happened without addressing Knowledge Graph specific issues.

III. NOUS Overview

This section describes various components of NOUS's knowledge graph construction pipeline as shown in Figure 1. The user interface and the question-answering system is discussed in the next section. In the text we emphasize on components where NOUS is making a research contribution and point the reader to the literature [5], [6] for a thorough discussion of the knowledge graph construction process and challenges.

A. *Data Sources*

Algorithms in NOUS are being used for developing custom knowledge graphs for diverse domains : 1) business intelligence applications via news articles and web crawls, 2) insider threat detection using various log data sources from enterprises and 3) citation analytics from bibliography databases. We will restrict our discussion to Wall Street Journal articles and general web crawls for the purpose of this demonstration.

B. *Triple Extraction from Natural Language Text*

We extract the text from every input document (such as a news article or blog post), and then process it sentence by sentence for entity and relation extraction. For relationship extraction, we used Open Information Extraction (OpenIE) [1] technique to obtain binary or n-ary relational tuples from every sentence. We also perform named entity extraction and co-reference resolution, and used this information to implement heuristics for triple extraction.

C. *Mapping Raw Triples to Knowledge Graph*

Our goal is to combine the extracted triples with a high-quality, openly available KG such as YAGO. Therefore, we need to map the subjects and objects in the triples to entities present in YAGO2, or else create a new node or relation into our custom knowledge graph.

A challenge with OpenIE like techniques is that they produce too many relations. We implement a distant supervision based approach to learn a rule-based model for each predicate and map the predicates from raw triples to the target ontology. Following the *Extreme Extraction* work by Freedman et al[8], we bootstrap each predicate model with 5-10 seed examples and expand the set of training examples for each predicate in a semi-supervised fashion. This is still an active area of refinement for NOUS.

We implement a variation of the AIDA algorithm proposed by *Hoffart et al* [9] for entity disambiguation [18], [16]. AIDA was chosen due to its high accuracy, scalability and ease of implementation in Spark platform. We adapted AIDA's context based similarity score that was originally based on comparing entity's Wikipedia article to the text surrounding the entity mention. As new entities from online articles are added to the knowledge graph, we use only the entity neighborhood in the knowledge graph to calculate contextual similarity.

D. *Confidence Estimation via Link Prediction*

Triples extracted from the text data sources are extremely noisy, and simply adding noisy facts to the knowledge graph will destroy its purpose. In addition to tracking source level trust, we implemented a Link Prediction approach [15] to quantitatively measure confidence in a triple using the prior state of the knowledge graph. For every predicate we build a latent feature embedding model using Bayesian Personalized Ranking (BPR) as the optimization criteria. Given an input triple, the model produces a real-valued score between 0 and 1.

E. Rule Learning via Frequent Graph Mining

Frequent Graph Mining (FGM) is the process of finding highly frequent subgraphs that help to discover structural association between entities and functional dependencies. A major research contribution of NOUS is the development of a distributed algorithm for streaming graph mining. Another novelty of our implementation is its ability to simultaneously support the curated KB and the extracted knowledge, and discover patterns by combining both structures.

The algorithm accepts the stream of incoming triples as input, a window size parameter that represents the size of a sliding window over the stream and reports the set of closed frequent patterns present in the window. As the stream characteristics change and some patterns turn from frequent to infrequent, our algorithm supports reconstruction of smaller frequent patterns from larger patterns that just turned infrequent. Distinct from transaction setting based algorithms such as gSpan [14], initial benchmarking of our work against distributed graph mining systems such as Arabesque [13] suggests 3x speedup on selected datasets.

F. Question Answering

The usability of NOUS depends on its ability to answer questions of interest to domain users. We implemented a novel path search algorithm for Knowledge Graphs. The algorithm accepts three arguments as input: a source s and a target entity t, and a relationship constraint, which typically is a predicate from the target ontology. Given this input, the algorithm returns a set of top-K paths to explain the relationship between s and t.

Our graph search algorithm is implemented on top of the distributed property graph model available from Apache Spark's GraphX library. The GraphX implementation allows us to store arbitrary properties with the vertices and edges. We utilize the text datasets available for each vertex in the graph (such as bag-of-words extracted from the Wikipedia page of an entity) and assign a topic distribution to every entity by executing the Latent Dirichlet Allocation (LDA) algorithm on the "document-term" matrix constructed from the text. During the graph walk, we perform a look-ahead search at every hop and select nodes with least topic divergence to the target node. Finally, we compute a "coherence" score for every path between the source and target, and the path with least amount of divergence is chosen.

IV. Demonstration Features: What to Expect

NOUS is implemented using the Scala programming language (version 1.5) on top of Apache Spark [3]. Our demonstration will use a Spark cluster running inside PNNL's compute cloud. We will use the Wall Street Journal (WSJ) corpus from 2010-2015 comprising 342,411 articles as the primary dataset and provide hands-on demonstration of the following features. The overarching goal of the demonstration will be to show how NOUS can be used to build custom knowledge graphs from web scale data and answer unique domain-specific queries.

1. Develop custom relation extractors and illustrate the trade-off from various heuristics.

2. Visualize the resultant graph and summarization of quality-related statistics (such as confidence distributions, and understanding how the structure of the underlying data influence the output quality).

3. Develop custom quality control modules for a new domain.

4. Execute queries for pattern discovery and graph search using both web and command line interface.

[3] http://spark.apache.org

References

[1] M. Banko, M. J. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni. Open information extraction for the web. In *IJCAI*, volume 7, pages 2670–2676, 2007.

[2] K. Bollacker et al. Freebase: A collaboratively created graph database for structuring human knowledge. In *Proc. International Conference on Management of Data*, SIGMOD '08.

[3] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. R. Hruschka, and T. M. Mitchell. Toward an architecture for never-ending language learning. In *AAAI*, 2010.

[4] S. Choudhury et al. A selectivity based approach to continuous pattern detection in streaming graphs. *EDBT*, 2015.

[5] O. Deshpande et al. Building, maintaining, and using knowledge bases: A report from the trenches. SIGMOD '13.

[6] X. Dong et al. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. KDD '14.

[7] D. Ferrucci et al. Building watson: An overview of the deepqa project. *AI magazine*, 31(3):59–79, 2010.

[8] M. Freedman et al. Extreme extraction: machine reading in a week. In *EMNLP*, 2011.

[9] J. Hoffart et al. Robust disambiguation of named entities in text. EMNLP '11.

[10] A. Lally et al. WatsonPaths: Scenario-based Question Answering and Inference over Unstructured Information. Ibm research report, 2014.

[11] F. Niu, C. Zhang, C. Ré, and J. W. Shavlik. Deepdive: Web-scale knowledge-base construction using statistical learning and inference. *VLDB*, 2012.

[12] F. M. Suchanek, G. Kasneci, and G. Weikum. Yago: A core of semantic knowledge. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07.

[13] C. H. Teixeira et al. Arabesque: A system for distributed graph mining-extended version. *arXiv preprint arXiv:1510.04233*, 2015.

[14] X. Yan and J. Han. gspan: Graph-based substructure pattern mining. In *ICDM*, 2002.

[15] B. Zhang, S. Choudhury, M. A. Hasan, X. Ning, K. Agarwal, S. Purohit, and P. G. P. Cabrera. Trust from the past: Bayesian personalized ranking based link prediction in knowledge graphs. *SDM Workshop on Mining Networks and Graphs*, 2016.

[16] B. Zhang, M. Dundar, and M. A. Hasan. Bayesian non-exhaustive classification a case study: Online name disambiguation using temporal record streams. In *CIKM*, pages 1341–1350, 2016.

[17] B. Zhang, N. Mohammed, V. Dave, and M. A. Hasan. Feature selection for classification under anonymity constraint. *Transactions on Data Privacy*, 2017.

[18] B. Zhang, T. K. Saha, and M. Al Hasan. Name disambiguation from link data in a collaboration graph. In *ASONAM*, pages 81–84. IEEE, 2014.