Open Information Extraction in Digital Libraries: Current Challenges and Open Research Questions

Hermann Kroll[1], Judy Al-Chaar[1] and Wolf-Tilo Balke[1]

[1]Institute for Information Systems, TU Braunschweig, Mühlenpfordtstr. 23, 38106, Braunschweig, Germany,

Abstract

A central challenge for digital libraries is to provide effective access paths to ever-growing collections of mostly textual, i.e., unstructured information. The traditional, yet expensive way to manage, categorize, and annotate such collections is extensive manual metadata curation to semantically enrich library items. The ability to convert textual information automatically into a structured representation would be extremely beneficial, allowing for novel access paths as well as supporting semantically meaningful discovery. This paper investigates opportunities and challenges that the latest techniques for *open information extraction* offer for digital libraries. Open information extraction promises to work out-of-the-box and does not require domain-specific training data. To evaluate how well such tools perform, we perform a qualitative evaluation in two domains: general news and biomedicine. Our research shows current benefits, but also reveals serious challenges for practical applications. In particular three research questions still have to be addressed to reliably use open information extraction in digital library projects.

Keywords

Digital Libraries, Open Information Extraction, Performance Measurement, Metadata Quality

1. Introduction

Digital libraries want to offer structured access to information and knowledge over constantly growing collections. And indeed, there is a growing amount of structured databases, knowledge graphs, or linked open data sources available for retrieval in some domains. Moreover, offering such structured information is also vital for several downstream applications, such as supporting complex graph queries in DBpedia [1], or enabling literature-based discovery methods to infer new knowledge [2, 3, 4, 5]. Yet, the majority of knowledge in digital library collections today is still hidden in textual form, and effective methods to harvest structured knowledge from books, journal articles, conference proceedings, etc. are rare. What are the main reasons?

It usually boils down to the costs vs. quality trade-off: Today's intelligent learning techniques enable domain experts to design reliable entity linking and relation extraction for harvesting pre-designed relations between entities from texts, see e.g., [6]. However, these systems to a large degree rely on supervised learning and thus need large-scale training data that cannot be readily transferred across domains. That means experts have to give ten thousands of examples to train an extraction system for a single relation. In brief, although supervised methods for entity recognition/linking and relation extraction have been shown to be up to the job with reasonable quality, their practical application comes at a high cost requiring huge amounts of training data [7]. Hence, even when limiting it down to specialized domains only, automatically structuring textual collections is still rarely performed in library practice.

In contrast to designing extraction systems for each domain, methods for unsupervised information extraction (OpenIE) promise to change the game. OpenIE aims to extract knowledge from texts without knowing the entity and relation domains a-priori [6]. Thus, OpenIE can be understood as an unsupervised method that could be efficiently applied across different domains. Yet, although OpenIE tools claim to be ready-to-use and suggest a high extraction precision, they are still rarely used in digital library projects. Is it because they are not quite as "ready-to-use" as is commonly expected? In previous work, we have proposed a toolbox that utilizes OpenIE tools to harvest knowledge from texts [8]. The toolbox contains novel algorithms to clean OpenIE outputs, and we performed a quantitative evaluation on biomedical benchmarks. In contrast to our previous works [8, 9], here we analyze the performance of OpenIE on a qualitative level, i.e., what are the main challenges in OpenIE for digital libraries? We do this by performing an evaluation in two common yet very different domains to allow for some generalizability: news articles from the New York Times and scientific articles from PubMed. The contribution of this position paper is a discussion of the future challenges and open research questions of OpenIE in digital libraries. In particular, we formulate three open research questions, which need to be answered before OpenIE can be readily used throughout collections.

DISCO@JCDL2021, September 27–30, 2021, Online
kroll@ifis.cs.tu-bs.de (H. Kroll); j.al-chaar@tu-bs.de (J. Al-Chaar); balke@ifis.cs.tu-bs.de (W. Balke)
0000-0001-9887-9276 (H. Kroll); 0000-0002-5443-1215 (W. Balke)

2. Investigating the Practical Performance of OpenIE

OpenIE was built as a versatile set of tools for extracting information from unstructured texts. The word *open* in OpenIE refers to the fact that OpenIE systems do not require pre-defined domains, relations, and named entities to extract new information. However, a system that perfectly transforms unstructured into structured information is nowhere to be found, and research is still ongoing [10, 11, 6]. Older OpenIE tools are based on simple machine learning and rule-based methods. In general, a rule-based system is a system that applies rules to store, sort, and manipulate data, e.g., the Stanford CoreNLP tool [10]. These systems use hand-crafted syntactic or semantic linguistic rules such as POS, and parsers, which usually cause errors in propagation and compounding at each stage. Modern systems build on neural architectures to increase extraction quality [11], i.e., a neural system's task can be seen as a classification problem or a sequence tagging problem. The main idea of a neural OpenIE system is to learn arguments and relation tuples bootstrapped from a state-of-the-art OpenIE system. The most recent and best-performing OpenIE neural system is OpenIE6 2020 [11]. We analyze both OpenIE tools in the following, namely Stanford CoreNLP and OpenIE6.

2.1. Evaluation Corpus

We have randomly selected articles from two different domains for our qualitative evaluation to allow for some generalizability of our findings. In particular, we investigate ten articles from the New York Times and 17 biomedical articles from PubMed. Topic-wise, the news articles are political, environmental, space & cosmos, and opinion articles. Various sentences were chosen from these articles based either on their structure or context. Regarding the structure of sentences, we feature five types of structures: simple, compound, and complex sentences. The purpose of this approach is to go from easy-to-understand sentences to more and more difficult ones. In addition, nested sentences and sentences that contain any type of negation, such as *not*, are selected. We selected 20 sentences for each category in both corpora.

2.2. Extraction Quality

The evaluation assessed whether the extraction includes *all essential* and *only reasonable* information that should be extracted. We employed three referees to rate the extracted information by both OpenIE tools. For each extraction, they decided whether the sentence's original information is retained completely, only partially, or is erroneously extracted: *Full* means that the statement carries the main message of the sentence. *Partial* means that some essential information is missed in the extraction. *Not* means an erroneous extraction that does not yield correct or useful information.

In a sentence with negation, extractions should always include the negation to retain the original information. We take the majority vote of the raters for reporting. Table 1 includes all sentence categories per tool and the number of sentences selected from a corpus. In addition, each category is manually evaluated by finding the percentage of extractions that *full*, *partial* or *not* show correct and reasonable tuples. From the 200 sentences, five representative sentences were selected for this paper to explain our five categories and to give the reader an intuition about the extraction results.

Simple. A simple sentence is a sentence that includes only one independent clause, i.e., subject, verb, and optionally an object. An example of this category is the following sentence: *1. The naysayers raised fair points.* CoreNLP yields the following two extractions: (naysayers; raised; fair points) and (naysayers; raised; points). CoreNLP tends to extract multiple, sometimes redundant, tuples (fair points or points as the objects). For our evaluation, we have always selected the largest CoreNLP tuples, e.g., we have selected the tuple that contains *fair points*. CoreNLP's extracted tuple contains all the essential information that should be extracted. As for OpenIE6, it extracted the following tuple: (The naysayers; raised; fair points). As for the evaluation of CoreNLP in simple sentences in the NYT corpus, 62% of its extractions consist of a complete statement, and 19% of the extractions were partially complete. Thus, 19% of the extractions showed an incomplete statement missing important parts of it. On the other hand, OpenIE6 showed very good results (100%) when run on simple sentences.

Compound. In contrast to simple sentences, a compound sentence consists of two independent clauses that are joined using a comma, semicolon, or any conjunction. For example, *India has about 10 million coronavirus cases now, and schools have been offering online instruction since March* [12]. The extractions expected from this sentence basically have to be two extractions (one for each independent clause). In this case, however, CoreNLP's only extraction is (India; has now; about 10 million coronavirus cases.); the second phrase in this sentence is not extracted at all. In contrast, OpenIE6 yields the following extractions: (India; has; about 10 million coronavirus cases now,) and (schools; have been offering; online instruction since March.). Both extractions cover the two independent clauses in the sentence. When the system was run on the complete set of compound sentences, 81% (NYT) and 76% (PubMed) of OpenIE6's extractions were complete, and 19% (NYT) and 14% (PubMed) were at least partially informative. In CoreNLP, however, only 24% (NYT) and 15% (PubMed) of the extractions were fully extracted.

Table 1
Evaluation results for OpenIE extraction quality. Three experts rate the extraction quality for CoreNLP and OpenIE6 on a scale between full (all information is kept), partial (relevant parts are missing) and not (information is wrongly or not extracted). The report below is based on a majority vote over the individual ratings.

Corpus	Sent. Category	#Sent.	CoreNLP			OpenIE6		
			Full	Partial	Not	Full	Partial	Not
NY Times	Simple	20	62%	19%	19%	**100%**	0%	0%
	Compound	20	24%	41%	35%	**81%**	19%	0%
	Complex	20	15%	53%	32%	**78%**	18%	4%
	Nested	20	4%	54%	42%	**80%**	18%	2%
	Negation	20	5%	5%	90%	**73%**	10%	17%
PubMed	Simple	20	52%	38%	10%	**100%**	0%	0%
	Compound	20	15%	44%	41%	**76%**	14%	10%
	Complex	20	38%	48%	14%	**56%**	13%	31%
	Nested	20	22%	63%	15%	**89%**	11%	0%
	Negation	20	5%	33%	62%	**81%**	15%	4%

Complex. A complex sentence is a sentence that consists of one independent clause and at least one dependant clause. A complex sentence might look like: *Relentless advertising campaigns are telling Indian parents that coding is critical because making children code will develop their cognitive skills* [12]. A good extraction, in this case, would be if either one extraction that included the entire sentence was produced or multiple extractions for each dependent and independent clause. CoreNLP's most informative extractions for this sentence are: (Relentless advertising campaigns; are telling; Indian Parents), (coding; is; critical) and (making children code; will develop; their cognitive skills). Nevertheless, the extraction (Relentless advertising campaigns; are telling; Indian Parents) seems unclear and incomplete. As for OpenIE6, we have the following tuples: (Relentless advertising campaigns; are telling; Indian parents that coding is critical because making children code will develop their cognitive skills), (coding; is; critical because making children code will develop their cognitive skills) and finally (making children code; will develop; their cognitive skills). All the previous extractions do not miss any important information but are quite long. As for the complex sentences, most of CoreNLP's extractions miss important parts of the sentence. For the news corpus, only 15% of the extractions were fully extracted. On the other hand, 78% of OpenIE6's extractions were complete.

Nested. Next, a sentence was selected to test whether the provided tools are able to handle nested extractions, too. Consider the following sentence: *As a result, many marine species are impeccably adapted to detect and communicate with sound* [13]. As we can see here, this sentence consists of only one subject and one relation; however, the rest of the sentence can be divided into two arguments. Here, the nested information that species adapt to detect and adapt to communicate should be retained ideally. CoreNLP's only extraction was (many marine species; are impeccably adapted; to detect with sound) and OpenIE6 extracted the following tuples: (many marine species; are impeccably adapted; to communicate with sound) and (many marine species; are impeccably adapted; to detect with sound). Thus, CoreNLP misses the second phrase, whereas OpenIE6 keeps the complete information. CoreNLP extracts only the first component from a set of nested sentences in most cases, ignoring the conjunction and everything that came after it. And therefore, 4% (NYT) and 22% (PubMed) of CoreNLP's extractions were complete; however, in OpenIE6, 80% (NYT) and 89% (PubMed) of the extractions were complete.

Negation. Last but not least, the last kind of sentences selected were sentences containing any type of negation, such as *not*, *no*, *none* or *neither*. This category was selected to analyze how each tool reacts to negations in a sentence. In this case, a sentence with the negation *not* was selected, e.g., *Recent studies show that man was not always the hunter* [14]. CoreNLP's extraction was (Recent studies; show; man). Whereas OpenIE6's extractions were (Recent studies; show; that man was not always the hunter) and (man; was not; always the hunter). So, CoreNLP ignores the negation part completely, whereas OpenIE6 keeps the negation correctly. The tools were also tested on multiple sentences containing negations such as *not*. In CoreNLP, some of these sentences did not have any extractions at all. If the sentence had any extractions, then the negative part was either entirely ignored and not extracted or extracted but without the negation. On the contrary, most of the OpenIE6 extractions included the negation. Still, in negation, 17% (NYT) and 4% (PubMed) of OpenIE6's extractions were erroneously extracted. Nevertheless, compared to OpenIE6, CoreNLP showed a much higher percentage of extractions full of errors: 90% (NYT) and 62% (PubMed) of CoreNLP's yielded extractions were incomplete or wrong.

Table 2
Evaluation results for the extracted OpenIE arguments (subjects and objects). Three experts rate whether the argument represents a single concept of interest or a complex concept, where a complex concept consists of multiple concepts.

Corpus	Argument Type	CoreNLP		OpenIE6	
		Single	Complex	Single	Complex
NY Times	Subject	98%	2%	89%	11%
	Object	80%	20%	32%	68%
PubMed	Subject	99%	1%	76%	24%
	Object	75%	25%	47%	53%

2.3. Argument Complexity

Having a closer look at the extractions, it seems that CoreNLP tends to extract smaller arguments (subjects or objects) than OpenIE6. For example, CoreNLP yields the triple (making children code; will develop; their cognitive skills) whereas OpenIE6 extracts (coding; is; critical because making children code will develop their cognitive skills). The last extraction may be hard-to-handle in a downstream application because the object contains a whole sentence fragment (obviously not structured). The latter one should ideally be broken into smaller pieces. To understand how often arguments are complex, we asked our three experts to rate all extracted arguments again. They assessed whether an argument represents a single concept of interest or a complex concept. For example, a single concept might be a city, a person, an article, a drug, etc. A complex concept consists of multiple smaller concepts, e.g., a person doing something, a location plus date information, an action plus date information, etc. The results are reported in Tab. 2. For example, 98% of CoreNLP's extracted subjects on NYT are actually single concepts. OpenIE6 extracts 89% subjects being single concepts. OpenIE6 extracts complex objects more often than CoreNLP: 68% vs. 20% (NYT) and 53% vs. 25% (PubMed).

3. Discussion

We analyzed CoreNLP and OpenIE6 on five sentence categories in two domains: The New York Times and PubMed. In addition, we put it into perspective with the main findings of our previous work [8].

Extraction Accuracy. First, OpenIE6 outperforms CoreNLP for every sentence category. This finding is not surprising because Kolluru et al. have proposed OpenIE6 as the best performing OpenIE system in 2020 [11]. They have evaluated OpenIE6 against ten different OpenIE tools on four established benchmarks. Their findings show that OpenIE6 achieves an F1-measure between 46.4% (CaRB 1-1) and 65.6% (OIE16-C). However, our previous evaluation reveals that CoreNLP is much faster, i.e., CoreNLP requires around 8.5 minutes to process 52k sentences, whereas OpenIE6 requires a modern GPU (Nvidia GTX 1080TI) and around one hour to process the same sentences [8]. Kolluru et al. have reported that OpenIE6 can process up to 31.7 sentences per second on a Tesla V100 GPU [11]. For comparison, an older system called RnnOIE can process up to 149.2 sentences per second but come with a lower F1-measure between 39.5% (CaRB 1-1) and 56.0% (OIE16-C).

Open Research Question 1. *What is the best trade-off between extraction runtime and accuracy?*

Extraction Arguments. Our qualitative evaluation has revealed that OpenIE tools may extract complex arguments, i.e., an argument that involves multiple concepts. Handling complex arguments can be challenging when using OpenIE in a digital library project, e.g., complex arguments will not represent a precise entity for a knowledge graph. Thus, post-processing is necessary to filter arguments by some domain-specific rules or pre-known vocabularies. One example might be entity-based filters like in [8]. The core idea was to keep only domain-specific concepts in arguments that are found in pre-known entity vocabularies. In addition, complex concepts could be also be handled by hand-crafted rules, e.g., store a date in an argument as additional information about the actual extraction.

Open Research Question 2. *How should extracted arguments be handled? And, may post-processing here be helpful to handle, filter or repair complex arguments?*

Not Canonicalized Outputs. OpenIE's extractions are not canonicalized, i.e., different subjects might refer to the same real-world concept (New York, NY, NYC, etc.). The same holds for relations: multiple verb phrases might represent the same relation, e.g., is born on, has birthdate. Vashishth et al. propose a tool called CESI to canonicalize Open Knowledge Bases (a collection of OpenIE extractions) [15]. Their goal is to identify and resolve synonymous subjects, relations, and objects that refer to the same real-world concept. They utilize side information like the Paraphrase database and entity linking information to embed the open knowledge base into a high-dimensional embedding space. Then, agglomerative clustering is used to find synonymous subjects, relations, and objects. But, canonicalizing complex arguments might be especially challenging, i.e., how can a

whole sentence fragment be canonicalized correctly. In addition, clustering results might be hard-to-interpret, e.g., which relation is hidden behind a set of verb phrases. We have thus proposed to integrate domain experts in the canonicalizing process, i.e., domain experts build a reliable relation vocabulary to canonicalize verb phrases [8].

Open Research Question 3. *How can OpenIE extraction be canonicalized to reliably resolve synonymous noun and verb phrases?*

Conclusion. OpenIE offers a way to bring more structure in otherwise unstructured document collections. Our evaluation shows that in simple settings, modern OpenIE tools like OpenIE6 can indeed already extract information with good quality. However, as sentences become more complex, the resulting extractions usually lack important information or do not retain precise semantics.

Still, we believe that OpenIE tools are extremely valuable because their advantage of not requiring domain-specific training examples is necessary for scalability over large digital libraries and especially for more heterogeneous collections. Moreover, combined with methods for filtering unnecessary information or detecting important domain-specific concepts, their overall quality in a concrete application may be drastically increased. To this end, we have formulated three demanding research questions for future research. More research will be necessary to bridge the gap between unstructured and structured information while bypassing the need for supervision as much as possible.

References

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

[2] R. Zhang, M. J. Cairelli, M. Fiszman, G. Rosemblat, H. Kilicoglu, T. C. Rindflesch, S. V. Pakhomov, G. B. Melton, Using semantic predications to uncover drug–drug interactions in clinical data, Journal of Biomedical Informatics 49 (2014) 134–147.

[3] D. R. Swanson, Complementary structures in disjoint science literatures, in: Proceedings of the 14th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '91, Association for Computing Machinery, 1991, p. 280–289. doi:10.1145/122860.122889.

[4] C. Wise, V. N. Ioannidis, M. R. Calvo, X. Song, G. Price, N. Kulkarni, R. Brand, P. Bhatia, G. Karypis, Covid-19 knowledge graph: Accelerating information retrieval and discovery for scientific literature, 2020. arXiv:2007.12731.

[5] D. Hristovski, A. Kastrin, D. Dinevski, T. C. Rindflesch, Constructing a graph database for semantic literature-based discovery, Studies in health technology and informatics 216 (2015) 1094.

[6] G. Weikum, X. L. Dong, S. Razniewski, F. Suchanek, Machine knowledge: Creation and curation of comprehensive knowledge bases, Foundations and Trends® in Databases 10 (2021) 108–490. doi:10.1561/1900000064.

[7] H. Kilicoglu, D. Shin, M. Fiszman, G. Rosemblat, T. C. Rindflesch, SemMedDB: a PubMed-scale repository of biomedical semantic predications, Bioinformatics 28 (2012) 3158–3160.

[8] H. Kroll, J. Pirklbauer, W.-T. Balke, A toolbox for the nearly-unsupervised construction of digital library knowledge graphs, in: To appear In: Proceedings of the ACM/IEEE Joint Conference on Digital Libraries in 2021, JCDL '21, Association for Computing Machinery, 2021.

[9] H. Kroll, D. Nagel, M. Kunz, W.-T. Balke, Demonstrating narrative bindings: Linking discourses to knowledge repositories, in: Fourth Workshop on Narrative Extraction From Texts, Text2Story@ECIR2021, volume 2860 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2021, pp. 57–63. URL: http://ceur-ws.org/Vol-2860/paper7.pdf.

[10] C. D. Manning, M. Surdeanu, J. Bauer, J. R. Finkel, S. Bethard, D. McClosky, The Stanford CoreNLP natural language processing toolkit, in: Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations, 2014, pp. 55–60.

[11] K. Kolluru, V. Adlakha, S. Aggarwal, Mausam, S. Chakrabarti, OpenIE6: Iterative grid labeling and coordination analysis for open information extraction, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP, Online, 2020, pp. 3748–3761.

[12] N. Misra, Do children really need to learn to code?, https://www.nytimes.com/2021/01/02/opinion/teaching-coding-schools-india.html, NY Times (April 2021).

[13] S. Imbler, In the oceans, the volume is rising as never before, https://www.nytimes.com/2021/02/04/science/ocean-marine-noise-pollution.html, NY Times (April 2021).

[14] A. Newitz, What new science techniques tell us about ancient women warriors, https://www.nytimes.com/2021/01/01/opinion/women-hunter-leader.html, NY Times (April 2021).

[15] S. Vashishth, P. Jain, P. Talukdar, CESI: Canonicalizing open knowledge bases using embeddings and side information, in: Proceedings of the 2018 World Wide Web Conference, WWW '18, 2018, p. 1317–1327. doi:10.1145/3178876.3186030.