Uncovering Algorithmic Approaches in Open Information Extraction: A Literature Review

Injy Sarhan[1,2] and Marco R. Spruit[2]

[1] Computer Engineering Department, College of Engineering, Arab Academy for Science, Technology and Maritime Transport (AAST), Abukir, Alexandria 1029, Egypt
[2] Department of Information and Computing Sciences, Utrecht University, Princetonplein 5, 3584 CC Utrecht, The Netherlands
{i.a.a.sarhan,m.r.spruit}@uu.nl

Abstract. The explosion of mostly unstructured data has further motivated researchers to focus on Natural Language Processing (NLP), hereby encouraging the development of Information Extraction (IE) techniques that target the retrieval of crucial information from unstructured texts. In this paper we present a literature review on Open Information Extraction (OIE). We compare both machine learning and handcrafted rules-based algorithmic approaches and identify the recently proposed Neural OIE approach as a particularly promising area for further research.

Keywords: Open Information Extraction, Deep Learning, Machine Learning, Hand-crafted Rules, Shallow Syntactic Analysis.

1 Introduction

As the demand for a fast and efficient method to extract pivotal information from text increases day by day, researches are encouraged towards IE tasks. OIE is the process of extracting relation tuples from text, it targets to ease the process of identifying domain-independent relations extracted from texts that scales to large-size data. OIE executes either a single or consistent number of passes over its corpus that results in capturing vital relationships represented in each clause in the form of relational tuples [1].

The key difference between Relation Extraction (RE) task and OIE is that OIE doesn't require a specific predefined relation domain, simply, the relation extracted is the text that links the two arguments together. In a domain-specific RE approach the relation in interest should be pre-specified. For instance, given the sentence *"Barack Obama born August 4, 1961 in Hawaii served as the 44th President of the United States"*. A (BornIn-Loc) relation will extract the following arguments (Barack Obama, Hawaii). In contrast to OIE that will extract the following relation triples in the format (argument 1, relation, argument 2):

- (Barack Obama, BornIn-Loc, Hawaii)
- (Barack Obama, BornIn-Year, August 4, 1961)
- (Barack Obama, Served-as, President of the United States)

The extracted tuples can be binary, ternary or n-ary, where the relationship is expressed between more than 2 entities such as Person-Location-Organization relation (John Smith, California, XYZ Company). OIE can be represented in two broad categories, approaches that require machine generated data to train a classifier and approaches that rely on hand-crafted rules [2]. Each category is further divided into two sub-categories as shown in Fig.1.

This paper presents analyzes different OIE approaches and a glimpse to the future of OIE. The remainder of this paper is structured as follows; Section 2 describes the methods and search strategy, while Section 3 presents the first type of OIE that utilizes machine learning classifiers, followed by the second paradigm that is based on hand-crafted rules in Section 4. Section 5 briefly discusses OIE challenges. Section 6 explores the new trends in OIE. Finally, Section 7 concludes the paper.

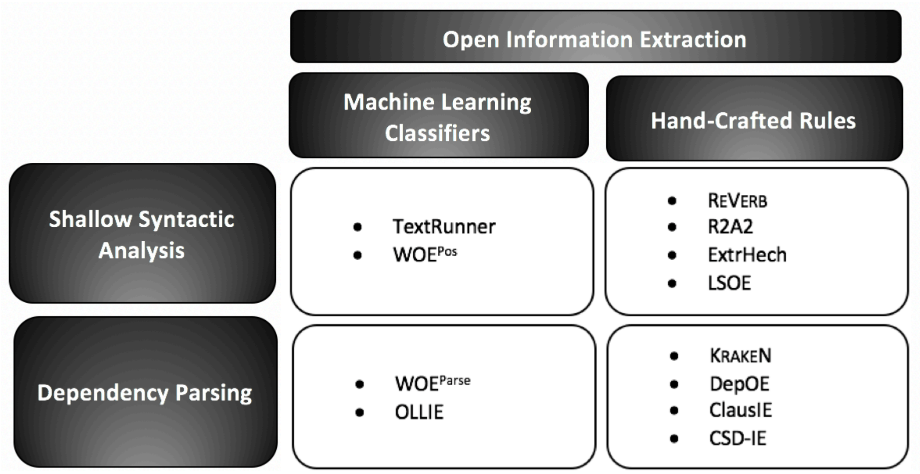

Fig. 1. Open Information Extraction Categories.

2 Methods and Search Strategies

The structured search was carried out in May-June 2018. The Snowball method was employed to serve this literature survey along with citation searching. After extensive research, we included papers that made an impact in the field of OIE, with the example of TextRunner [3] as the initial OIE system. As the field of OIE has grown rapidly in the last decade, with the exception of [3], only articles that were published in the last 8 years are included in our survey paper.

3 Machine Learning Classifiers

In this section we overview the OIE systems that we have studied that utilize automatically generated data to train a classifier. This methodology is further divided into two subcategories; those approaches that uses shallow syntactic analysis and approaches that utilize dependency parsing.

3.1 Shallow Syntactic Analysis

TextRunner, the first OIE system is a fully implemented, highly adaptable system introduced by Banko et al. in 2007 [3]. TextRunner utilizes a Naïve Bayes model to determine if the heuristically selected tokens that lie between two entities indicate a relationship or not. It exploits a domain-independent technique to extract relation tuples from a text corpus. Afterwards, it identifies domain-specific terms in the tuples by utilizing class recognizers, thus learning relation mapping rules and finally transform the tuples into domain relations [4].

A corpus of 9 million web pages is the sole input to TextRunner which then executes the extraction process in 3 key steps [5]:

1. A self-supervised learner: Low demand of hand-labeled training data is required due to the self-supervised nature of the system. The learner produces a Conditional Random Field [CRF] based classifier that exploits unlexicalized features in order to extract relations from the corpus.
2. A single pass extractor: The system extracts all possible relation tuples by making a single pass over the corpus using the CRF classifier. The tuples that are classified as trustworthy are retained by the extractor.
3. A redundancy-based assessor: The extracted tuples are re-ranked based on a probabilistic model of redundancy- similar to one used in KnowItAll [6]-. The assessor allocates a confidence score to each extracted tuple based on its number of occurrences in the text.

All these components enable TextRunner to be a high performance, general, and high-quality extractor for heterogeneous web text. Subsequent work showed that utilizing a linear-chain CRF [7] or Markov Logic Network [8] leads to further improvements over TextRunner [9].

Wu and Weld introduced the WOE (Wikipedia-based Open Extractor) system [10] that can operate in two modes: WOE^{Pos} and WOE^{Parse}. The WOE^{Pos} approach uses a CRF extractor trained with shallow syntactic features, unlike WOE^{Parse} that's discussed later in the next section 3.2. WOE^{Pos} system enhances TextRunner's performance by utilizing Wikipedia to train data for their extractors. The primary concept behind WOE is the automatic assembly of training examples by heuristically matching Wikipedia info box values with corresponding text. When compared to TextRunner, WOE^{Pos} increases the F-Measure by almost 34% owing to a finer training data from Wikipedia via self-supervision.

3.2 Dependency Parsing

Wu and Weld [10] additionally demonstrated that dependency parse features cause a massive increase in both recall and precision when compared to shallow linguistic features, thus they introduced the aforementioned model WOE^{Parse}. The parser-based extractor -WOE^{Parse}- utilizes a plentiful dictionary of dependency path patterns acquired from heuristic extractions produced from Wikipedia. WOE^{Parse} reaches an F-measure between 72% and 91% higher than that of TextRunner, the main reason behind this increase is because difficult sentences with complicated distance relations are handled better using a parser and this results in WOE^{Parse} to maintain a decent recall with only tolerable loss of precision, it also outperforms WOE^{Pos}. Albeit, it runs around 30 times slower than TextRunner owing to the time required for parsing.

OLLIE (Open Language Learning for Information Extraction) system, introduced by Mausam et al. [11], overcomes the main drawbacks of REVERB [12] — discussed in section 4.1 — and WOE, where extraction of both non-factual tuples and relations are only intervened by verbs. OLLIE bootstraps an immense training set from a number of high precision seed tuples obtained from REVERB to learn semi-lexicalized pattern templates. Those pattern templates are features in a dependency parse as they determine both the argument and the relation phrase. They are later put into use during the extraction phase [13]. The concept behind the learning component is to retrieve a large number of example sentences that assert a specific tuple, to ensure that all the important information had been captured. Eventually, OLLIE investigates the text around the tuple to append more details (attribution and clausal modifiers). The authors of OLLIE signal out certain features that seem to capture nearly all of the sentences with attribution and clausal modification. For both attribution and clausal, a feature and a filter are needed to remove false positives. Finally, OLLIE's confidence function is trained to lessen the confidence of an extraction if its surrounding text indicates that there's a possibility that it is non-factual. OLLIE achieved an area under the curve (AUC) 2.7 times higher than REVERB and 1.9 times larger than WOE^{Parse}.

4 Based on Hand-crafted Rules

The second category of OIE methods makes use of hand-crafted rules or heuristics to extract relation triples. Alike the first category of unsupervised training of classifiers, it is also further divided into two subcategories; those approaches that uses shallow syntactic analysis and approaches that utilizes dependency parsing.

4.1 Shallow Syntactic Analysis

In 2011, Fader et al. [12] proposed REVERB, which resembles TextRunner approach by utilizing computationally efficient surface patterns over tokens. REVERB implements a general model of verb-based relation phrases, by applying two simple constraints to extract binary facts. This algorithm consists of three major steps:

1. Identify relation phrases that meet syntactic and lexical constraints.
2. Locates a pair of Noun Phrase (NP) arguments for each identified relation phrase.
3. A confidence score is then allocated to the resulting extractions using a logistic regression classifier trained on 1,000 random Web sentences with shallow syntactic features.

In other words, REVERB takes as input a POS-tagged and NP-chunked sentence and outputs a set of (NP1, Relation, NP2) extraction triples. The inclusion of syntactic constraints aided to reduce uninformative extractions. Furthermore, a lexical constraint is used to separate valid relation phrases from over specified relation phrases. This algorithm deviates from the TextRunner algorithm in four significant manners: First, the relation phrase is recognized "holistically" rather than word-by-word as in TextRunner. Second, possible relation phrases are filtered based on statistics over a sizable corpus. Third, the REVERB algorithm works on extracting the relation first, then it extracts the argument, hereby avoiding the confusion between a noun in the relation phrase for an argument and finally, by introducing lexical and syntactic constraints resulting in doubling the area under the precision-recall curve when compared to TextRunner and WOE^{Pos}.

Succeeding the aforementioned approaches, Etzioni et al. [9] introduced the second generation of OIE, R2A2, through merging REVERB with an argument identifier - ARGLEARNER - to enhance argument extraction for the relation phrases. ARGLEARNER is a learning-based system, when given a sentence and a relation phrase pair it identifies arguments by utilizing patterns as features. In both TextRunner and REVERB, the arguments are the two adjacent NPs, while R2A2 utilizes ARGLEARNER to learn independent extractors for left and right boundaries of each argument using three classifiers, two of which identifies the left and right bounds of ARG1 and the third classifier identifies the right bound of ARG2 [9]. The system is then compared against REVERB on two datasets each consisting of 200 random sentences. R2A2 increased the area under the precision-recall curve by almost 100% from 0.45 to 0.9 [9].

Xavier et al. debate that it is not compulsory to have an immense list of patterns or various kinds of linguistic labels to perform OIE. In order to prove their proclaimed theory, they developed LSOE [14] (Lexical Syntactic pattern based Open Extractor) a novel unsupervised OIE approach that implements lexical-syntactic patterns to POS-tagged texts to extract relation triples (Arg1, Relation, Arg2).

The strategy is based on two types of patterns:

1. Generic patterns to identify non-specific relations.
2. Rule-based patterns to learn Qualia structure.

LSOE performance was compared with two state-of-the-art Open IE systems: REVERB and DepOE [15]. The latter is discussed in the next section. LSOE attained a higher precision when compared to the aforementioned state-of-art approaches.

Another approach that exploits shallow syntactic analysis based on hand-crafted rules is ExtrHech [16]. The latter approach acquires Part of Speech (POS) tagged text as input and applies syntactic constraints as regular expressions and outputs a set of relation triples. It is worth nothing that ExtrHech is a multilingual system that's applied on Spanish language as well as English. When compared against REVERB on a 68-sentence dataset, ExtrHech outperformed REVERB in terms of precision and recall.

4.2 Dependency Parsing

KRAKEN [17], an OIE methodology particularly purposed to capture complete N-ary facts, in addition, to examining fact completeness and correctness, which reflects on the quality of the extracted data. Given a Stanford-parsed sentence as an input to the system, KRAKEN conducts the following three main steps:

1. Fact phrase detection: KRAKEN locates fact phrases as a series of verbs, modifiers or prepositions.
2. Detection of argument heads: Using type-paths, heads of arguments can be identified. Every type-path indicates one or more links, as well as the direction of each link, to follow to find an argument head.
3. Detection of full argument: Recursively trail all downward links from the argument head to get the full argument.

Hand-crafted rules are used to locate relational phrases and their corresponding argument over typed dependency parses. Provided that a fact phrase has at least one argument, the system extracts it as a fact. When KRAKEN is compared against REVERB, KRAKEN almost doubles the number of recognized complete and true facts. It achieves notable results for binary, ternary and 4-ary facts.

DepOE (Dependency-Based Open Information Extraction) [15] is a multi-lingual OIE system specifically designed to extract verb-based triples from Wikipedia in four languages: Portuguese, Spanish, Galician, and English. The latter system embraced the features of both Machine Reading by creating an efficient and fast system guaranteeing scalability as the corpus grows and Learning by Reading by utilizing a dependency-based parser beneficial to obtaining fine-grained information. DepOE relies on the following steps:

1. Dependency parsing: All the sentences of the input text are inspected by the dependency-based parser by a multilingual tool.
2. Clause constituents: For each parsed sentence, it discovers the verb clauses it contains and, then, for each clause, it locates the verb candidates [subject, direct object, attribute and prepositional complements]
3. Extraction rules: A group of rules are employed on the clause constituents that are extracted from the previous step to extract the target triples.

Nevertheless, the dependency-based parser that was used in the first step has insufficient grammar that results in partial parsing lacking deep analysis. As a result, the number of extracted triples by DepOE is fewer than REVERB. It is worth nothing that DepOE has more accurate extractions of the two arguments as opposed to REVERB, that suffers from the erroneous identification of the first argument and extraction of an incomplete part of the second argument.

In the following year, Del Corro and Gemulla introduced ClausIE (Clause-based Open Information Extraction) [18]. ClausIE benefits from the linguistic knowledge about the grammar of the English language to identify clauses in an input sentence and afterwards determines the category of each clause to be consistent with the grammatical function of its constituents. Given an input sentence ClausIE performs the following [18]:

1. Computes the dependency parse of the input sentence to discover its syntactical structure.

2. Specify the set of clauses using the dependency parse.
3. Learn the set of coherent derived clauses based on the dependency parse and small domain-independent lexica.
4. Generate one or more propositions for each clause.

ClausIE primarily differs from preceding systems in the way that it doesn't exploit any training data and also does not necessitate further processing to remove low-precision extractions in contrast to REVERB [12] and OLLIE [11], both use post-processing statistical techniques that aids in increasing the precision. ClausIE yields 2.5–3.5 times more correct extractions than OLLIE. The inclusion of low-confidence propositions declines precision, that explains why TextRunner's precision is significantly lower than that of REVERB, WOE, and ClausIE. The latter three extractors obtain high precision due to high-confidence propositions [18].

Bast and Haussmann demonstrate that contextual sentence decomposition, a method initially created for high-precision semantic search can also be utilized for OIE, hereby, introducing CSD-IE (Contextual Sentence Decomposition Information Extraction) [19]. CSD-IE is carried out in two primary steps:

1. Identification of fundamental building blocks of the desired contexts in the sentence constituent identification (SCI) phase. Thus, a tree expressing the semantics is derived.
2. Tree constituents are combined to form the contexts creating sentence constituent recombination (SCR).

Triples are then obtained from the contexts outcome by identifying the first explicit verb phrase and surrounding adverbs to be the predicate, with the prefix being the subject and the postfix is the object. This approach achieves a decent precision with high recall and very good coverage and minimality when compared against REVERB, OLLIE and Clause-IE.

Despite the fact that WOE [10] and OLLIE [11] both exploit dependency parsers, they yet fail to correctly determine the subject of the second clause, owing to their use of automatically learned dependency parser patterns, for instance the OLLIE system learns from REVERB. Another consequence from using automatically learned dependency parser patterns is the high number of incorrect extractions produced by OLLIE. Hand-crafted approaches using dependency parsing, therefore, seem the way to go. However, these approaches still suffer from error propagation caused by the employed patterns.

5 OIE Challenges

Extracting data from text might be challenging, the two most recurrent challenges that a great number of the aforementioned OIE approaches faces are uninformative and incoherent extractions [12].

Incorrect handling of relational phrases is the main root of uninformative extraction that results in leaving out crucial information. For further illustration, consider the following sentence *"John Smith signed a fixed-price contract with ABC company after a*

2-month negotiation period", uninformative extraction results in extracting (John Smith, signed, fixed-price contract) instead of extracting (John Smith, signed a contract, ABC company). Uninformative extractions make up almost 4% of WOE^{Parse} output, 6% of WOE^{Pos} output, and 7% of TextRunner's output [9].

Incoherent extraction is purposeless extractions that are derived from opaque relation phrases that the extractor fails to correctly identify as it's the case with major OIE state-of-art [3] [10]. Considering the previous sentence, an example of incoherent extraction would be (John Smith, signed a contract, 2-month negotiation period). Incoherent extractions form nearly 30% of WOE^{parse} output, 15% of WOE^{Pos} 13% of TextRunner's output [9]. Syntactic and lexical constraints aids in the reduction of uninformative extractions and excluding incoherent extractions in addition to decreasing overly-specified extraction.

To overcome this constraint REVERB [12] exploits a syntactic constraint that forces every multi-word relation phrase to start with a verb, end with a preposition, and be a neighboring sequence of words in the sentence. Hence, preventing the extractor from making a series of decisions to decide whether to include each word in the relation phrase or not, regularly resulting in unclear predictions.

The majority of the current OIE approaches center their research on the extraction of binary facts and suffer a notable quality deterioration when capturing higher order N-ary relations with exception of KRAKEN [17] that focuses on the extraction of N-ary facts.

6 Future Trends in OIE

Recently, Neural Networks (NN) methods have been gaining a massive amount of attention due to their proven success at tackling various NLP tasks [20-22]. Distinctive from the several OIE state-of-the-art systems that were discussed in this paper, Cui et al. [23] proposed a neural OIE paradigm that implements an encoder-decoder framework. The encoder-decoder infrastructure is a method for text generation and has already been utilized in other NLP tasks successfully [23]. Being implemented by a recurrent neural network, the encoder-decoder framework inputs a variable length sequence, then the decoder uses the resulting compressed representation vector to produce the output sequence. Both the encoder and decoder use a 3-layer Long Short-Term Memory (LSTM) [24]. Binary extractions with high confidence are used to train the proposed neural OIE approach bootstrapped from a state-of-the-art OIE system, resulting in the generation of high-quality tuples.

While several OIE approaches have been developed in the past decade with the aim of extracting relations from given corpora mainly in the English language, only few researchers target other languages [14, 15]. Future researches should be aimed towards developing a multi-lingual OIE paradigm.

Furthermore, as previously discussed in the previous section, until now the main focus has been on the extraction of binary relations, omitting the importance of extraction of higher order relations that has a high impact not only on the quality of the extracted relations but also its completeness and correctness.

7 Conclusion

With the ongoing advancements in the field of NLP, OIE gained a massive amount of attention in the past years. Practically, the current OIE paradigms either employs automatically assembled training data or hand-crafted heuristics.

We started by reviewing approaches that necessitate machine learning classifiers. TextRunner [3] and WOE^{Pos} [10] emphases improving the efficiency of the extracted triples by applying syntactic constraints as POS and chunking. WOE^{Parse} [10] and OLLIE [11] use dependency parse features to boost the recall and precision, even though this affects negatively on the extraction speed. OLLIE stood out by achieving a higher AUC when compared to REVERB and WOE^{Parse}.

REVERB [12] used Hand-crafted patterns and exploited lexical and syntactic constrains to extract relation triples achieving notable results. While R2A2 [9] further enhanced REVERB by employing an argument learner. The second type of hand-crafted rules relied on dependency parsing like ClausIE [18], KRAKEN [17] and DepOE [15]. A summary of all the discussed approaches can be found in Table 1.

The analysis of this survey appears to mostly support the second approach, using hand-crafted patterns as is shown in the evaluation of [17,18]. However, after reviewing OIE systems, we believe that future research should be more directed towards a neural networks approach. NN has already provided a boost to several NLP tasks. The model proposed by [23] was able to overcome the error propagation caused by hand-crafted rules.

In conclusion, there is still a room for improvement in OIE. OIE can't be regarded as a simple NLP task, it still faces a number of shortcomings that opens up many research questions. While we have tried to cover the most representative state-of-art approaches that have appeared in the modern literature, to get a complete picture, Christina Niklaus et al. [26] also recently reviewed and assessed the performance of a number of OIE approaches. Their findings are well in line with our work and underline the rapidly increasing interest in the quickly evolving OIE domain.

Table 1. Summary of OIE Approaches

Approach	Category (Sub-category)	Dataset Used	Advantages	Disadvantages
TextRunner [3]	Machine Learning Classifier (Shallow Syntactic Analysis)	9 Million Webpages	First OIE system with a single-pass over corpus.	Low precision caused by the addition of low confidence propositions.
WOE^{Pos} [10]	Machine Learning Classifier (Shallow Syntactic Analysis)	Penn Treebank, Wikipedia, and the general Web.	Unlike WOE^{parse}, it avoids extracting false high- confidence triples.	Doesn't use deep processing during extractions.
WOE^{Parse} [10]	Machine Learning Classifier (Dependency Parsing)	Penn Treebank, Wikipedia, and the general Web.	Proved that parsed features play an important role in informative extractions.	WOE^{Parse} processes each sentence with a dependency parser, thus requiring in a longer processing time.
OLLIE [11]	Machine Learning Classifier (Dependency Parsing)	300 sentences from 3 sources: News, Wikipedia and Biology textbook.	Applies deep syntactic analysis to extract new relations.	Significantly slower than other state-of-art OIE systems.
REVERB [12]	Hand-Crafted Rules (Shallow Syntactic Analysis)	Same dataset as TextRunner trained on Penn Treebank	Employing syntactic constraint to avoid uninformative extractions.	REVERB restricts subjects to noun phrases without prepositions.
R2A2 [9]	Hand-Crafted Rules (Shallow Syntactic Analysis)	20,000 sentences.	Utilize ARGLEARNER to better Identify the arguments.	Does not identify long tail of patterns thus misses important recall from verb-based relations with long range dependencies
LSOE [14]	Hand-Crafted Rules (Shallow Syntactic Analysis)	9 million Web documents.	Identifying non-specific relations using generic patterns.	The lack of lexical syntactic patterns results in missing any relations expressed by verbs.
ExtrHech [16]	Hand-Crafted Rules (Shallow Syntactic Analysis)	68 sentences from FactSpaCIC.	Multi-lingual OIE system (English and Spanish).	System is evaluated on a small corpus.
KRAKEN [17]	Hand-Crafted Rules (Dependency Parsing)	500 sentences sampled from the Web using Yahoo's random link service.	Extraction of n-ary facts.	More than quarter of the evaluation set was skipped since the dependency parser employed doesn't indicate uncertain grammatical relationships.
DepOE [15]	Hand-Crafted Rules (Dependency Parsing)	Sports and Biology articles from Wikipedia.	Supports Multi-lingual extractions (English, Spanish, Portuguese, and Galician) by utilizing a multilingual rule-based parser.	Unable to correctly extract arguments in the English version due to the inefficiency of the named entity recognition system.

ClausIE [18]	Hand-Crafted Rules (Dependency Parsing)	3 datasets: REVERB dataset, 200 random sentences from Wikipedia, 200 random sentence form NY Times.	Extracts non-verb-mediated propositions.	Incorrect dependency parses and Implementation tends to miss essential adverbials.
CSD-IE [19]	Hand-Crafted Rules (Dependency Parsing)	2 datasets from ClausIE: 200 random sentences from the Wikipedia, and 200 random sentences from the NY Times.	Achieves minimality, to increase the relevance extracted arguments and relation by further decreasing its size.	Errors due to incorrect parsing.
Neural OIE [23]	Neural Network	Benchmark dataset from [25] that contains 3,200 sentences	Avoiding error propagation cause by hand-crafted pattern by employing an encoder-decode framework.	Only supports binary extractions and non-nested relations.

References

1. Christensen, J., Soderland, S., & Etzioni, O.: An analysis of open information extraction based on semantic role labeling. In Proceedings of the sixth international conference on Knowledge capture (pp. 113-120). ACM (2011).
2. Gamallo, P.: An Overview of Open Information Extraction (Invited talk). In OASIcs-Open Access Series in Informatics. Vol. 38. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, (2014).
3. Banko, M., Cafarella, M. J., Soderland, S., Broadhead, M., & Etzioni, O.: Open information extraction from the web. In IJCAI (Vol. 7, pp. 2670-2676) (2007).
4. Soderland, S., Roof, B., Qin, B., Xu, S., & Etzioni, O.: Adapting open information extraction to domain-specific relations. In AI magazine, 31(3), 93-102) (2010).
5. Christensen, J., Soderland, S., & Etzioni, O.: "Semantic role labeling for open information extraction." In Proceedings of the NAACL HLT 2010 First International Workshop on Formalisms and Methodology for Learning by Reading. Association for Computational Linguistics, (2010).
6. Etzioni, O., Cafarella, M., Downey, D., Popescu, A. M., Shaked, T., Soderland, S., ... & Yates, A.: Unsupervised named-entity extraction from the web: An experimental study. In Artificial intelligence (165.1: 91-134) (2005).
7. Banko, M., and Etzioni, O.: The tradeoffs between open and traditional relation extraction. In Proceedings of ACL-08: HLT (28-36) (2008).
8. Zhu, K., San Wong, Y., & Hong, G. S.: Multi-category micro-milling tool wear monitoring with continuous hidden Markov models. In Mechanical Systems and Signal Processing 23.2 (547-560) (2009).
9. Etzioni, O., Fader, A., Christensen, J., Soderland, S., & Mausam, M. Open information extraction: The second generation. In IJCAI (Vol. 11, pp. 3-10) (2011).

10. Wu, F. and Weld, D, S.: Open information extraction using Wikipedia. In Proceedings of the 48th annual meeting of the association for computational linguistics. Association for Computational Linguistics, (2010).
11. Schmitz, M., Bart, R., Soderland, S., & Etzioni, O: Open language learning for information extraction. In Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. Association for Computational Linguistics, (2012).
12. Fader, A., Soderland, S., & Etzioni, O.: Identifying relations for open information extraction." Proceedings of the conference on empirical methods in natural language processing. Association for Computational Linguistics, (2011).
13. Mausam, M.: Open information extraction systems and downstream applications. In Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, pp. 4074-4077. AAAI Press, (2016).
14. Xavier, C. C., de Lima, V. L. S., & Souza, M: Open information extraction based on lexical-syntactic patterns. In Brazilian Conference on Intelligent Systems, pp 189– 194, (2013).
15. Gamallo, P., Garcia, M., & Fernández-Lanza, S.: Dependency-based open information extraction. In Proceedings of the joint workshop on unsupervised and semi-supervised learning in NLP (pp. 10-18). Association for Computational Linguistics. (2012).
16. Zhila, A., & Gelbukh, A.: Comparison of open information extraction for English and Spanish. In 19th Annual International Conference Dialog pp. 714-722 (2013).
17. Akbik A, Löser A.: Kraken: N-ary facts in open information extraction. In Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction, pp. 52– 56, (2012).
18. Del Corro, L., & Gemulla, R.: Clausie: clause-based open information extraction. In Proceedings of the 22nd international conference on WWW pp. 355-366. ACM, (2013).
19. Bast, H., & Haussmann, E.: Open information extraction via contextual sentence decomposition. In Semantic Computing (ICSC), IEEE Seventh International Conference on. pp. 355-366. IEEE, (2013).
20. Rush, A. M., Chopra, S., & Weston, J.: A neural attention model for abstractive sentence summarization. In Proceedings of the Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, Lisbon, Portugal, pages 379–389. http://aclweb.org/anthology/D15-1044 (2015).
21. Gehring, J., Auli, M., Grangier, D., Yarats, D., & Dauphin, Y. N.: Convolutional Sequence to Sequence Learning. ArXiv e-prints. (2017).
22. Meng, R., Zhao, S., Han, S., He, D., Brusilovsky, P., & Chi, Y.: Deep key phrase generation. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Association for Computational Linguistics, Vancouver, Canada, pages 582–592. http://aclweb.org/anthology/P17-1054. (2017).
23. Cui, Lei, Wei, F. and Zhou, M.: Neural Open Information Extraction. arXiv preprint arXiv:1805.04270 (2018).
24. Hochreiter, S., & Schmidhuber, J. Long short-term memory. Neural Computing. 9(8):1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735. (1997).
25. Stanovsky, G., & Dagan, I.: Creating a large benchmark for open information extraction. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. ACL, Texas, pages 2300–2305. https://aclweb.org/anthology/D16-1252. (2016).
26. Niklaus, C., Cetto, M., Freitas, A., & Handschuh, S.: A Survey on Open Information Extraction." arXiv preprint arXiv:1806.05599 (2018).