Knowledge-Based Systems 268 (2023) 110455

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

A Canonical Context-Preserving Representation for Open IE: Extracting Semantically Typed Relational Tuples from Complex Sentences

Christina Niklaus [a,*], Matthias Cetto [a], André Freitas [b,c], Siegfried Handschuh [a,d]

[a] University of St.Gallen, Rosenbergstrasse 30, 9000 St.Gallen, Switzerland
[b] Idiap Research Institute, Rue Marconi 19, 1920 Martigny, Switzerland
[c] University of Manchester, Oxford Rd, Manchester, M13 9PL, UK
[d] University of Passau, Innstraße 33, Passau, Germany

ARTICLE INFO

Article history:
Received 10 August 2022
Received in revised form 6 October 2022
Accepted 7 March 2023
Available online 15 March 2023

Keywords:
Information Extraction
Open IE
Text Simplification
Semantic representation
Complex sentences
Discourse relations
Predicate-argument structures
Relational tuples
Semantic hierarchy
Minimal propositions
Relation extraction
Meaning representation
Rhetorical structures

ABSTRACT

Modern systems that deal with inference in texts need automatized methods to extract meaning representations (MRs) from texts at scale. Open Information Extraction (IE) is a prominent way of extracting all potential relations from a given text in a comprehensive manner. Previous work in this area has mainly focused on the extraction of isolated relational tuples. Ignoring the cohesive nature of texts where important contextual information is spread across clauses or sentences, state-of-the-art Open IE approaches are thus prone to generating a loose arrangement of tuples that lack the expressiveness needed to infer the true meaning of complex assertions.

To overcome this limitation, we present a method that allows existing Open IE systems to *enrich their output with additional meta information*. By leveraging the semantic hierarchy of minimal propositions generated by the discourse-aware Text Simplification (TS) approach presented in Niklaus et al. (2019), we propose a mechanism to *extract semantically typed relational tuples from complex source sentences*. Based on this novel type of output, we introduce a *lightweight semantic representation for Open IE* in the form of normalized and context-preserving relational tuples. It extends the shallow semantic representation of state-of-the-art approaches in the form of predicate-argument structures by capturing *intra-sentential rhetorical structures and hierarchical relationships* between the relational tuples. In that way, the semantic context of the extracted tuples is preserved, resulting in more informative and coherent predicate-argument structures which are easier to interpret.

In addition, in a comparative analysis, we show that the semantic hierarchy of minimal propositions benefits Open IE approaches in a second dimension: the canonical structure of the simplified sentences is easier to process and analyze, and thus *facilitates the extraction of relational tuples*, resulting in an improved precision (up to 32%) and recall (up to 30%) of the extracted relations on a large benchmark corpus.

1. Introduction

Every day, huge amounts of data are generated. According to estimates by the World Economic Forum, around 463 exabytes of data will be created each day globally by 2025, most of it in the form of unstructured text data.[1] That is a lot of relevant and potentially useful information, of course — but too much for humans to access without further ado. Instead, automatic processing of this data by machines would be beneficial. Text data, however, is not an ideal data source for feeding into machines. Rather, we need to put it into a form that machines can read and analyze automatically, for example, by converting it into a knowledge graph that represents the data in a structured way. Such a representation allows information to be processed automatically and relationships between elements to be discovered in order to gain new insights.

One possibility to transform unstructured information into a structured representation is the task of Open Information Extraction (IE). According to [1], Open IE is "perhaps the simplest form of a semantic analysis". It aims to obtain a shallow semantic representation of texts in the form of predicates and their arguments. Such predicate-argument structures are regarded as fundamental components of a semantic representation of a sentence, which is often also called a meaning representation (MR). Thus, the task

* Corresponding author.
E-mail addresses: christina.niklaus@unisg.ch (C. Niklaus), matthias.cetto@unisg.ch (M. Cetto), andre.freitas@idiap.ch, andre.freitas@manchester.ac.uk (A. Freitas), siegfried.handschuh@unisg.ch, siegfried.handschuh@uni-passau.de (S. Handschuh).

[1] https://www.weforum.org/agenda/2019/04/how-much-data-is-generated-each-day-cf4bddf29f/.

https://doi.org/10.1016/j.knosys.2023.110455

of Open IE can be seen as the first step towards a richer semantic analysis [1].

By analyzing the output of state-of-the-art Open IE systems (e.g., [1–4]), we observed three common shortcomings. First, relations often span over long nested structures or are presented in a non-canonical form that cannot be easily captured by a small set of extraction patterns. Therefore, such relations are commonly missed by state-of-the-art approaches. Second, current Open IE systems tend to extract propositions with long argument phrases that can be further decomposed into meaningful propositions, with each of them representing a separate fact. Overly specific constituents that mix multiple, potentially semantically unrelated propositions are difficult to handle for downstream applications, such as question answering (QA) or textual entailment tasks. Instead, such approaches benefit from extractions that are as compact as possible [5]. Third, most state-of-the-art Open IE systems lack the expressiveness needed to properly represent complex assertions, resulting in incomplete, uninformative or incoherent extractions that have no meaningful interpretation or miss critical information asserted in the input sentence.

To overcome these limitations, we developed an Open IE framework that transforms complex text data into a novel lightweight semantic representation in the form of normalized and context-preserving relational tuples. The contributions of our work are two-fold. First, to remove the complexity of determining complex predicate-argument structures with variable arity from syntactically complex input sentences, we make use of the transformation process proposed in [6]. It converts complex source sentences into smaller units with a simple canonical structure, using clausal and phrasal disembedding mechanisms. The resulting normalized subject–predicate–object structure of the simplified sentences reduces the complexity of the relation extraction task, leading to an improved precision and recall of the extracted relational tuples. Second, to support the extraction of highly informative tuples that allow for a meaningful interpretation in downstream applications, we leverage the approach described in [6] to establish a semantic hierarchy between the extracted relations. By capturing the semantic context and relationships to other units in the form of rhetorical relations, the method presented in [6] puts a semantic layer on top of the simplified sentences that we utilize to extract relational tuples within their semantic context. In that way, we enrich the output of state-of-the-art Open IE systems with additional meta information that extends the shallow semantic representation of current approaches, producing an output that preserves important contextual information of the extracted relational tuples. We thus introduce an innovative lightweight semantic representation for Open IE that supports the extraction of accurate, meaningful and complete relational tuples.

The contributions of our work can be summarized as:

(i) A comprehensive comparative evaluation of state-of-the-art Open IE systems that demonstrates that the canonical structure of the simplified sentences *reduces the complexity of the relation extraction task*, leading to an improved precision (up to 32%) and recall (up to 30%) of the extracted relational tuples on a large benchmark corpus.[2]

(ii) A systematic qualitative analysis of the output that shows that the semantic hierarchy of minimal propositions can be leveraged to *extract relational tuples within their semantic context*, resulting in more informative and coherent predicate-argument structures.

(iii) The introduction of a *novel lightweight semantic representation for complex text data in the form of normalized and context-preserving relational tuples* that extends the shallow semantic representation of state-of-the-art Open IE approaches in the form of predicate-argument structures by capturing *intra-sentential rhetorical structures and hierarchical relationships* between the relational tuples.

This manuscript is organized as follows: in Section 2, we present a critique of the state of the art in Open IE. In this context, we compare previously proposed approaches and draw attention to their particular strengths and weaknesses, opening new research directions to be explored in this work. Next, in Section 3, we describe how the discourse-aware TS approach proposed in [6] can be leveraged for the task of Open IE to extract semantically typed relational tuples from complex text data, resulting in a novel lightweight semantic representation for Open IE. In Section 4, we then report on the results of the experiments that were conducted to evaluate the performance of our proposed approach with regard to (i) improving the coverage and accuracy of the relational tuples extracted from complex input sentences, and (ii) enriching the output with semantic information. In Section 5, we present Complex QA as a use case for our proposed lightweight semantic representation. Finally, in Section 6, we conclude our findings and present directions for future work.

2. Related work

IE turns the unstructured information expressed in NL text into a structured representation [8] in the form of relational tuples consisting of a set of arguments and a phrase denoting a semantic relation between them: ⟨*arg1*; *rel*; *arg2*⟩. Traditional approaches to IE focus on answering narrow, well-defined requests over a predefined set of target relations on small, homogeneous corpora. To do so, they take as input the target relation along with hand-crafted extraction patterns or patterns learned from hand-labeled training examples (e.g., [9–11]). Consequently, shifting to a new domain requires the user to not only name the target relations, but also to manually define new extraction rules or to annotate new training data by hand. Thus, those systems rely on extensive human involvement. In order to reduce the manual effort required by IE approaches, [12] introduced a new extraction paradigm: Open IE. Unlike traditional IE methods, Open IE is not limited to a small set of target relations known in advance, but rather extracts all types of relations found in a text. In that way, it facilitates the domain-independent discovery of relations extracted from text and scales to large, heterogeneous corpora such as the Web. Hence, [12] identified three major challenges for Open IE systems:

Automation Open IE systems must *rely on unsupervised extraction strategies*, i.e. instead of specifying target relations in advance, possible relations of interest must be automatically detected while making only a single pass over the corpus. Moreover, the manual labor of creating suitable training data or extraction patterns must be reduced to a minimum by requiring only a small set of hand-tagged seed instances or a few manually defined extraction patterns.

Corpus Heterogeneity Heterogeneous datasets form an obstacle for tools that perform a deep linguistic analysis, such as syntactic or dependency parsers, since they commonly work well when trained on and applied to a specific domain, but are prone to produce incorrect results when used in a different genre of text. Furthermore, NER is unsuitable

[2] The evaluation presented in Section 4.2.1 represents a significant expansion of the experiments performed in [7]. It uses a larger set of baseline Open IE systems, additional datasets and a modified matching process for mapping predicted extractions to reference extractions.

to target the variety and complexity of entity types on the Web. As Open IE systems are intended for *domain-independent usage*, such tools should be avoided in favor of shallow parsing methods such as part-of-speech taggers.

Efficiency In order to *readily scale to large amounts of text*, Open IE systems must be computationally efficient. Enabling fast extraction over huge datasets, shallow linguistic features, like part-of-speech tags, are to be preferred over deep features that are derived from parse trees.

These criteria were first implemented in the Open IE system TextRunner, which was presented together with the task definition in [12]. This seminal work triggered a lot of research effort in this area, resulting in a multitude of proposed approaches that often do not strictly adhere to these initial guidelines. For example, to date, Open IE systems are commonly evaluated on rather small-scale, domain-dependent corpora. In addition, recent approaches frequently rely on the output of a dependency parser to identify extraction patterns, thereby hurting the domain independence and efficiency assumptions.

2.1. Approaches to open information extraction

A large body of work on the task of Open IE has been described since its introduction by [12]. Existing Open IE approaches make use of a set of patterns in order to extract relational tuples from a sentence, each consisting of a set of arguments and a phrase that expresses a semantic relation between them. Such extraction patterns are either hand-crafted or learned from automatically labeled training data, as shown below.

2.1.1. Early approaches based on hand-crafted extraction patterns and self-supervised learning

Early approaches to the task of Open IE are either based on hand-crafted extraction patterns, where a human expert manually defines a set of extraction rules, or self-supervised learning, where the system automatically finds and labels its own training examples.

Binary relations. The line of work on Open IE begins with TextRunner [12], a self-supervised learning approach that consists of three modules. First, given a small sample of sentences from the Penn Treebank [13], the learner applies a dependency parser to heuristically identify and label a set of extractions as positive and negative training examples. This data is then used as input to a Naive Bayes classifier which learns a model of trustworthy relations using unlexicalized part-of-speech and noun phrase chunk features. The self-supervised nature mitigates the need for hand-labeled training data, and unlexicalized features help scale to the multitudes of relations found on the Web. The second component, the extractor, then generates candidate tuples from unseen text by first identifying pairs of noun phrase arguments and then heuristically designating each word in between as part of a relational phrase or not. Next, each candidate extraction is presented to the classifier, which keeps only those labeled as trustworthy. The restriction to the use of shallow features in this step makes TextRunner highly efficient. Finally, a redundancy-based assessor assigns a probability to each retained tuple based on the number of sentences from which each extraction was found, thus exploiting the redundancy of information in Web text and assigning higher confidence to extractions that occur multiple times.

WOE [14] also learns an open information extractor without direct supervision. It makes use of Wikipedia as a source of training data by bootstrapping from entries in Wikipedia infoboxes, i.e. by heuristically matching infobox attribute–value pairs with

$$V \mid VP \mid VW^*P$$
V = verb particle? adv?
W = (noun | adj | adv | pron | det)
P = (prep | particle | inf. marker)

Fig. 1. ReVerb's part-of-speech-based regular expression for reducing incoherent and uninformative extractions.

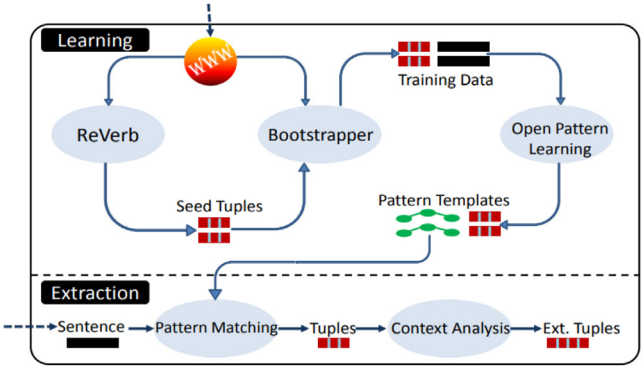

Fig. 2. Ollie's system architecture [2]. Ollie begins with seed tuples from ReVerb, uses them to build a bootstrap learning set, and learns open pattern templates, which are applied to individual sentences at extraction time.

corresponding sentences in the article. This data is then used to learn extraction patterns on both part-of-speech tags (WOE^{pos}) and dependency parses (WOE^{parse}). Former extractor utilizes a linear-chain CRF to train a model of relations on shallow features which outputs certain text between two noun phrases when it denotes a relation. Latter approach, in contrast, makes use of dependency trees to build a classifier that decides whether the shortest dependency path between two noun phrases indicates a semantic relation. By operating over dependency parses, even long-range dependencies can be captured. Accordingly, when comparing their two approaches, [14] show that the use of dependency features results in an increase in precision and recall over shallow linguistic features, though, at the cost of extraction speed, hence negatively affecting the scalability of the system.

[15] propose ReVerb, a shallow extractor that addresses three common errors of previously presented Open IE approaches: the output of such systems frequently contains a great many of *uninformative extractions* (i.e., extractions that omit critical information), *incoherent extractions* (i.e., extractions where the relational phrase has no meaningful interpretation) and *overly-specific relations* that convey too much information to be useful in downstream semantic tasks. ReVerb improves over those approaches by introducing a syntactic constraint that is expressed in terms of simple part-of-speech-based regular expressions (see Fig. 1), covering about 85% of verb-based relational phrases in English texts, as a linguistic analysis has revealed. In that way, the amount of incoherent and uninformative extractions is reduced. Moreover, in order to avoid overspecified relational phrases, a lexical constraint is presented which is based on the idea that a valid relational phrase should take many distinct arguments in a large corpus. Besides, while formerly proposed approaches start with the identification of argument pairs, ReVerb follows a relation-centric method by first determining relational phrases that satisfy above-mentioned constraints, and then finding a pair of noun phrase arguments for each such phrase.

Ollie [2] follows the idea of bootstrap learning of patterns based on dependency parse paths. However, while WOE relies on Wikipedia-based bootstrapping, Ollie applies a set of high precision seed tuples from its predecessor system ReVerb to bootstrap

Table 1
Evolution of early Open IE approaches illustrated by means of the relations extracted from the following input sentence: *"The novelist Franz Kafka is the author of a short story entitled 'The Metamorphosis'."*.

Approach	Extracted tuple	Pros	Cons
TextRunner	⟨*The novelist Franz Kafka*; *is*; *the author of a short story*⟩		Uninformative relational phrases
ReVerb	⟨*The novelist Franz Kafka*; *is the author of*; *a short story*⟩	Extraction of coherent and meaningful relational phrases	Limitation to verb-mediated relations
Ollie	⟨*Franz Kafka*; *is the author of*; *a short story*⟩, ⟨*Franz Kafka*; *is*; *a novelist*⟩, ⟨*a short story*; *be entitled*; *'The Metamorphosis'*⟩	Wider syntactic scope of relational phrases	Still limited to binary relations

Table 2
From binary to n-ary relations. The extractions above compare ReVerb's binary relational tuple with Exemplar's n-ary relational tuple when given the following input sentence: *"Franz Kafka was born into a Jewish family in Prague in 1883."*, showing that the latter is much more informative and complete.

Approach	Extracted tuple	Pros	Cons
ReVerb	⟨*Franz Kafka*; *was born into*; *a Jewish family*⟩		Limitation to binary relations
Exemplar	⟨*Franz Kafka*; *was born*; *(into) a Jewish family*; *(in) Prague*; *(in) 1883*⟩	Extraction of complete facts by gathering the full set of arguments for a relational phrase (n-ary relations)	Prone to erroneous extractions on syntactically complex sentences

a large training set over which it learns a set of extraction pattern templates using dependency parses (see Fig. 2). In contrast to previously presented systems that fully ignore the context of a tuple and thus extract propositions that are not asserted as factual, but are only hypothetical or conditionally true, Ollie includes a context-analysis step in which *contextual information* from the input sentence around an extraction is analyzed to expand the output representation by adding attribution and clausal modifiers, if necessary, and thus increasing the precision of the system (for details, see Section 2.1.3). Moreover, Ollie is the first Open IE approach to identify not only verb-based relations, but also relationships mediated by nouns and adjectives (see Ollie's extractions in Table 1). In that way, it *expands the syntactic scope of relational phrases* to cover a wider range of relation expressions, resulting in a much higher yield (at comparable precision) as compared to previous systems.

Table 1 illustrates by means of an example sentence how the proposed Open IE systems evolve over time by addressing particular issues that were identified in prior approaches.

N-ary relations. The Open IE systems presented above focus on the extraction of binary relations, which commonly leads to extraction errors such as incomplete, uninformative or erroneous tuples. To address this problem, [16] present KrakeN. It is the first approach that is specifically built for capturing complete facts from sentences by *gathering the full set of arguments for each relational phrase* within a sentence, thus producing tuples of arbitrary arity. The identification of relational phrases and their corresponding arguments is based on hand-written extraction rules over typed dependency parses.

Exemplar [17] applies a similar approach for extracting n-ary relations, using hand-crafted patterns based on dependency parse trees to detect a relation trigger and the arguments connected to it. Based on the task of SRL, whose key idea is to classify semantic constituents into different semantic roles [18], it assigns each argument its corresponding role (such as subject, direct object or prepositional object).

Table 2 depicts the added value of extracting n-ary relations that capture the full set of arguments for a relational phrase, thus generating informative extractions that represent complete facts from the input. In contrast, Open IE approaches that are limited to the extraction of binary relations are prone to missing some important aspects of a fact or event stated in the source sentence.

Table 3 summarizes the properties of early approaches to the task of Open IE, taking into account the features that they use to specify or learn the extraction patterns, the arity of the relations that they capture and the syntactic scope of relational phrases that they cover.

2.1.2. Paraphrase-based approaches

Noticing that previously proposed Open IE approaches are *prone to erroneous extractions on syntactically complex sentences*, more recent work is based on the idea of incorporating a *sentence re-structuring stage*. Its goal is to transform complex sentences, where relations are spread over several clauses or presented in a non-canonical form, into a set of syntactically simplified independent clauses that are easy to segment into Open IE tuples. Thus, they aim to improve the coverage and accuracy of the extractions. An example of such a paraphrase-based Open IE approach is ClausIE [1], which exploits linguistic knowledge about the grammar of the English language to map the dependency relations of an input sentence to clause constituents. In that way, a set of coherent clauses presenting a simple linguistic structure is derived from the input. Then, the type of each clause is determined by combining knowledge of properties of verbs (with the help of domain-independent lexica) with knowledge about the structure of input clauses. Finally, based on its type, one or more tuples are extracted from each clause, each representing different pieces of information. The basic set of patterns used for this task is shown in Table 4.

In the same vein, [19] propose a strategy to break down structurally complex sentences into simpler ones by decomposing the original sentence into its basic building blocks via chunking. The dependencies of each two chunks are then determined (one of "connected", "disconnected" or "dependent") using either manually defined rules over dependency paths between words in different chunks or a Naive Bayes classifier trained on shallow features, such as part-of-speech tags and the distance between chunks. Depending on their relationships, chunks are combined into simplified sentences, upon which the extraction process is carried out.

[3] present Stanford Open IE, an approach in which a classifier is learned for splitting a sentence into a set of logically entailed shorter utterances by recursively traversing its dependency tree and predicting at each step whether an edge should yield an independent clause or not. In order to increase the usefulness of the extracted tuples for downstream applications, each self-contained clause is then maximally shortened by running natural logic inference [20] over it. In the end, a small set of 14 hand-crafted patterns are used to extract a predicate-argument triple from each utterance. An illustration of this approach is depicted in Fig. 3.

Arguing that it is hard to read out from a dependency parse the complete structure of a sentence's propositions, since, amongst others, different predications are represented in a non-uniform

Table 3
Properties of early approaches to Open IE.

Approach	Features	Arity	Type of relations
TextRunner	Shallow syntax (part-of-speech tagging and noun-phrase chunking)	Binary	Verb-based
WOE	Shallow syntax (part-of-speech tagging for WOE^{pos}) and dependency parse (for WOE^{parse})	Binary	Verb-based
Ollie	Dependency parse	Binary	Mediated by verbs, nouns and adjectives
ReVerb	Shallow syntax (part-of-speech tagging and noun phrase chunking)	Binary	Verb-based
KrakeN	Dependency parse	n-ary	Verb-based
Exemplar	Dependency parse	Binary and n-ary	Verb- and noun-based

Table 4
ClausIE's basic patterns for relation extraction [1]. S: Subject, V: Verb, C: Complement, O: Direct object, A: Adverbial, V_i: Intransitive verb, V_c: Copular verb, V_e: Extended-copular verb, V_{mt}: Monotransitive verb, V_{dt}: Ditransitive verb, V_{ct}: Complex-transitive verb.

	Pattern	Clause type	Example	Derived clauses
S_1:	SV_i	SV	AE died.	(AE, died)
S_2:	SV_eA	SVA	AE remained in Princeton.	(AE, remained, in Princeton)
S_3:	SV_cC	SVC	AE is smart.	(AE, is, smart)
S_4:	$SV_{mt}O$	SVO	AE has won the Nobel Prize.	(AE, has won, the Nobel Prize)
S_5:	$SV_{dt}O_iO$	SVOO	RSAS gave AE the Nobel Prize.	(RSAS, gave, AE, the Nobel Prize)
S_6:	$SV_{ct}OA$	SVOA	The doorman showed AE to his office.	(The doorman, showed, AE, to his office)
S_7:	$SV_{ct}OC$	SVOC	AE declared the meeting open.	(AE, declared, the meeting, open)

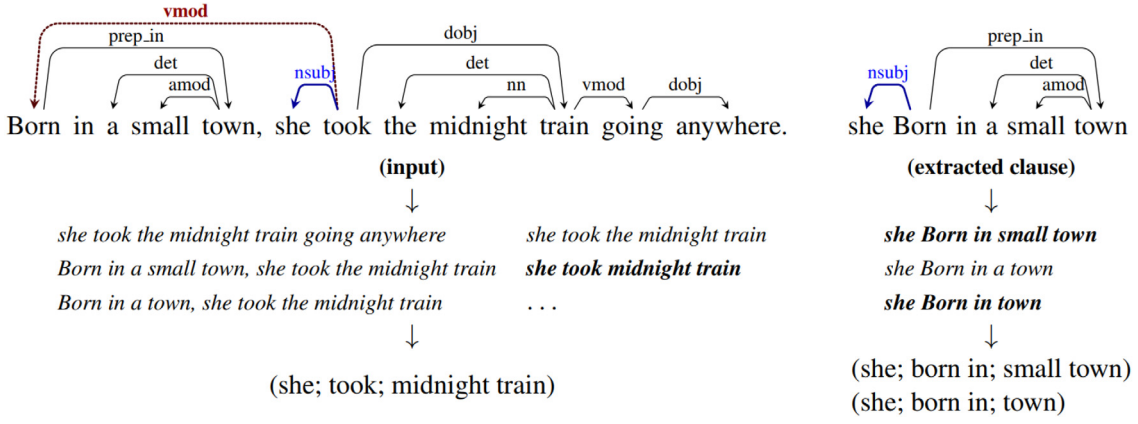

Fig. 3. An illustration of Stanford Open IE's approach. From left to right, a sentence yields a number of independent clauses. From top to bottom, each clause produces a set of entailed shorter utterances, and segments the ones which match an atomic pattern into a relational tuple [3].

Table 5
Properties of paraphrase-based approaches to Open IE.

Approach	Features	Arity	Type of relations
ClausIE	Dependency parse	Binary and n-ary	Verb-based relations and noun-mediated relations from appositions and possessives
[19]	Two modes: • shallow syntax (part-of-speech tags) • dependency parse	Binary and n-ary	Verb- and noun-based
Stanford Open IE	Dependency parse	Binary	Verb- and noun-based
PropS	Dependency parse	n-ary	Mediated by verbs, nouns and adjectives

manner and proposition boundaries are not easy to detect, [21] suggest PropS. They introduce a semantically-oriented sentence representation that is generated by transforming a dependency parse tree into a directed graph which is tailored to directly represent the proposition structure of an input sentence. Consequently, extracting relational tuples from this novel output format is straightforward. The conversion of the dependency tree into the proposition structure is carried out by a rule-based converter.

Table 5 gives an overview of the properties of the paraphrase-based Open IE approaches described above.

2.1.3. Systems that capture inter-proposition relationships

Aforementioned Open IE systems *lack the expressiveness needed for a proper interpretation of complex assertions*, since they ignore the context under which a proposition is complete and correct. Thus, they do not distinguish between information asserted in a sentence and information that is only hypothetical or conditionally true. For example, extracting the relational tuple ⟨*the earth*; *is the center of*; *the universe*⟩ from the sentence "*Early scientists believed that the earth is the center of the universe.*" would be inappropriate, since the input is not asserting this proposition, but only noting that it was believed by early scientists [22].

To properly handle such cases, Ollie introduces an attribution context to indicate that an extracted relational tuple is reported or claimed by some entity:

(⟨*the earth*; *be the center of*; *the universe*⟩;
`AttributedTo` *believe*; *Early astronomers*)

In that way, it extends the default Open IE representation of $\langle arg_1; rel; arg_2 \rangle$ tuples with an extra field. Beyond attribution relationships, Ollie targets the identification of clausal modifiers, such as:

(⟨*Romney*; *will be elected*; *President*⟩;
`ClausalModifier` *if*; *he wins five key states*)

Both types of modifiers are identified by matching patterns with the dependency parse of the input sentence. Clausal modifiers are determined by an adverbial clause edge and filtered lexically (the first word of the clause must match a list of cue terms, e.g. *if*, *when*, or *although*), while attribution modifiers are identified by a clausal complement edge whose context verb must match one of the terms given in VerbNet's list of common verbs of communication and cognition [2].

A similar type of output is produced by Ollie's successor OpenIE-4 [22], which combines SrlIE [18] and RelNoun [23]. The former is a system that converts the output of an SRL system into an Open IE extraction by treating the verb as the relational phrase, while taking its role-labeled arguments as the corresponding Open IE argument phrases. The latter, in contrast, represents a rule-based Open IE system that extracts noun-mediated relations, thereby paying special attention to demonyms and compound relational nouns. In addition, OpenIE-4 marks up temporal and spatial arguments by assigning them a *T* or *S* label, respectively. For instance, it extracts the following tuple from the example sentence from Table 2: ⟨*Franz Kafka*; *was born*; *into a Jewish family;* ***L:*** *in Prague;* ***T:*** *in 1883*⟩. Lately, its successor OpenIE-5 was released. It integrates BONIE [24] and CALMIE [25]. While the former focuses on extracting tuples where one of the arguments is a number or a quantity-unit phrase, the latter targets the extraction of relational tuples from conjunctive sentences.

Similar to Ollie, [5], who explore the use of CSD for Open IE, advocate to further specify relational tuples with information on which they depend. Their system CSD-IE is based on the idea of paraphrasing-based approaches described in Section 2.1.2. Using a set of hand-crafted rules over the output of a constituent parser, a sentence is first split into sub-sequences that semantically belong together, forming so-called "contexts". Each such context now contains a separate fact, yet it is often dependent on surrounding contexts. In order to preserve such inter-proposition relationships, tuples may contain untyped references to other extractions. However, as opposed to Ollie, where additional contextual modifiers are directly assigned to the corresponding relational tuples, [5] represent contextual information in the form of separate, linked tuples. To do so, each extraction is given a unique identifier that can be used in the argument slot of an extraction for a later substitution with the corresponding fact by a downstream application. An example for an attribution is shown below [5]:

#1: ⟨*The Embassy*; *said*; *that #2*⟩
#2: ⟨*6700 Americans*; *were*; *in Pakistan.*⟩

Another current approach that captures inter-proposition relationships is proposed by [26], who present a nested representation for Open IE that is able to capture high-level dependencies, allowing for a more accurate representation of the meaning of an input sentence. Their system NestIE uses bootstrapping over a dataset for textual entailment to learn both binary and nested triple representations for n-ary relations over dependency parse trees. These patterns can take on the form of binary tuples $\langle arg_1; rel; arg_2 \rangle$ or nested tuples, such as $\langle \langle arg_1; rel; arg_2 \rangle; rel_2; arg_3 \rangle$ for n-ary relations. Using a set of manually defined rules, contextual links between extracted tuples are inferred from the dependency parse in order to generate a nested representation of assertions that are complete and closer in meaning to the original statement. Similar to Ollie, such contextual links are identified as clausal complements, conditionals and relation modifiers. Linked tuples are represented by arguments that refer to the corresponding extractions using identifiers, as shown in the following example [26]:

#1: ⟨*body*; *appeared to have been thrown*; ∅⟩
#2: ⟨#1; *from*; *vehicle*⟩

Moreover, the example below denotes a temporal relationship between the extractions #1 and #2, based on the following sentence:

"*After giving 5000 people a second chance at life, doctors are celebrating the 25th anniversary of Britain's first heart transplant.*"

#1: ⟨*doctors*; *are celebrating*; *the 25th anniversary of Britain's first heart transplant*⟩
#2: ⟨*doctors*; *giving*; *second chance at life*⟩
#3: ⟨#1; *after*; #2⟩

MinIE [27], another recent Open IE system, is built on top of ClausIE, a system that was found to often produce overspecified extractions. Such overly specific constituents that combine multiple, potentially semantically unrelated propositions in a single relational or argument phrase may easily hurt the performance of downstream NLP applications, such as QA or textual entailment. Instead, those approaches benefit from extractions that are as compact as possible. Therefore, MinIE aims to minimize both relational and argument phrases by identifying and removing parts that are considered overly specific. For this purpose, MinIE provides four different minimization modes which differ in their aggressiveness, thus allowing to control the trade-off between precision and recall. Moreover, it semantically annotates extractions with information about polarity, modality, attribution and quantities instead of directly representing it in the actual extractions, as the following example shows [27]:

"*Pinocchio believes that the hero Superman was not actually born on beautiful Krypton.*"

#1: ⟨*Superman*; *was born actually on*; *beautiful Krypton*⟩
Annotation: factuality, (− [not], certainty), attribution (Pinocchio, +, possibility [believes])
#2: ⟨*Superman*; *was born on*; *beautiful Krypton*⟩
Annotation: factuality, (− [not], certainty), attribution (Pinocchio, +, possibility [believes])
#3: ⟨*Superman*; *"is"*; *hero*⟩
Annotation: factuality, (+, certainty)
with + and − signifying positive and negative polarity, respectively.

In that way, the output generated by MinIE is further reduced to its core constituents, producing maximally shortened, semantically enriched extractions.

Table 6 gives a summary of the properties of above-mentioned Open IE approaches that capture inter-proposition relationships by incorporating the semantic context between the extracted tuples. In that way, important contextual information that is needed to infer the true meaning of a statement is preserved, resulting in an output that increases the informativeness of the extracted relational tuples.

Table 6
Properties of Open IE approaches capturing inter-proposition relationships.

Approach	Features	Arity	Type of relations
Ollie	Dependency parse	Binary	Mediated by verbs, nouns and adjectives
OpenIE-4/5	• shallow syntax (part-of-speech tagging and noun phrase chunking) • constituency parse	n-ary	Mediated by verbs, nouns and adjectives
CSD-IE	Constituency parse	Binary	Verb- and noun-based
NestIE	Dependency parse	n-ary	Verb- and noun-based
MinIE	Dependency parse	Binary	Verb-based relations and noun-mediated relations from appositions and possessives

Table 7
Properties of end-to-end neural approaches to Open IE.

Approach	Features	Arity	Type of relations
[28]	Seq2Seq (RNN)	Binary	Mediated by verbs, nouns and adjectives
RnnOIE	SL	n-ary	Verb-based

2.1.4. End-to-end neural approaches

With [4,28], two end-to-end neural approaches for the task of Open IE are proposed. In [28], Open IE is cast as a sequence-to-sequence generation problem, where the input sequence is the sentence and the output sequence consists of the corresponding relational tuples. Pairs of input and output sequences, serving as training instances for learning arguments and their corresponding relational phrases, are obtained from highly confident tuples that are bootstrapped from the OpenIE-4 system. Encoder and decoder are implemented using RNNs with an attention-based copying mechanism. In that way, the dependencies on other NLP tools (e.g., syntax parsers) are reduced, while bypassing hand-crafted patterns and alleviating error propagation. However, [28]'s framework is restricted to the extraction of binary tuples, disregarding more complex structures such as n-ary and nested extractions, often resulting in long and complex argument phrases.

Similar to [28], with RnnOIE, [4] present an encoder–decoder based Open IE approach. The authors reformulate Open IE as an SL task, using a custom BIO scheme [29] inspired by a recent state-of-the-art deep learning approach for SRL [30].

Table 7 summarizes the properties of end-to-end neural approaches to the Open IE task.

2.2. Core principles of a formal representation for relational tuples

Though a multitude of systems for Open IE have been developed over the last decade, a clear formal specification of what constitutes a valid relational tuple is still missing. This lack of a well-defined, generally accepted task definition has long prevented the creation of an established, large-scale annotated corpus serving as a gold standard collection for an objective and reproducible cross-system comparison. A first attempt in standardizing the Open IE evaluation was made by [31] by providing a large gold benchmark corpus. It is based on a set of consensual guiding principles that underlie most Open IE approaches proposed so far, as they have identified. Those principles cover the core aspects of the task of Open IE, allowing for a clearer formulation of the problem to be solved. The three key features to consider are the following:

Assertedness The assertedness principle states that each tuple extracted from the input should be asserted by the original sentence. Usually, instead of inferring predicate-argument structures from implied statements, e.g. the tuple ⟨*Sam*; *convinced*; *John*⟩ from ⟨*Sam*; *succeeded in convincing*; *John*⟩, Open IE systems tend to extract the full relational phrase (⟨*Sam*; *succeeded in convincing*; *John*⟩), incorporating matrix verbs ("*succeeded*") and other elements, such as negations or modals (e.g., ⟨*John*; *could not join*; *the band*⟩). Therefore, systems should not be penalized for not finding inferred relations.

Minimality In order to benefit downstream NLP tasks, it is favorable for Open IE systems to extract compact, self-contained tuples that do not combine several unrelated facts. Therefore, systems should aim to generate valid extractions with minimal spans for both relation and argument slots, while preserving the meaning of the input. For instance, the coordination in the sentence "*Bell distributes electronic and building products*" should ideally yield the following two relational tuples: ⟨*Bell*; *distributes*; *electronic products*⟩ and ⟨*Bell*; *distributes*; *building products*⟩.

Completeness and Open Lexicon The completeness and open lexicon principle aims to extract all relations that are asserted in the input text. This principle was one of the fundamental ideas that have been introduced in the work of [12] together with the Open IE terminology. In their work, the Open IE task was defined as a domain-independent task that extracts all possible relations from heterogeneous corpora, instead of only extracting a set of pre-specified classes of relations. The majority of state-of-the-art Open IE systems realize this challenge by considering all possible verbs as potential relations. Accordingly, their scope is often limited to the extraction of verbal predicates, while ignoring relations mediated by more complex syntactic constructs, such as nouns or adjectives.

These guiding principles were later adopted by [32,33] in the annotation process of the WiRe57 and CaRB corpora. However, the three basic principles described above provide only a rough framework for what properties a valid relational tuple should have. In this work, we take a first step towards *normalizing relational tuples* by introducing a novel lightweight semantic representation for the task of Open IE. We specifically target complex text data, with the objective of transforming it into a structured representation in the form of canonical and context-preserving relational tuples. To this end, we create a standardized output scheme that

(i) normalizes the extracted relations by reducing the predicate and argument slots to their essential components, resulting in tuples with a canonical subject–predicate–object structure, and
(ii) captures intra-sentential rhetorical structures and hierarchical relationships between tuples, thus incorporating their semantic context.

3. Extracting semantically typed relational tuples from complex sentences

As described above, the Open IE task has been evolving along the years to capture relations beyond simple subject–predicate–object triples. Still, a closer look into the extraction of the linkage between clauses and phrases within a complex sentence has been neglected. This work aims to address this gap. It is targeted on capturing complementary linguistic phenomena, including intra-sentential discourse relations, predicate-argument structures, as well as clausal and phrasal linkage patterns.

3.1. Semantic hierarchy of minimal propositions

The method that we propose is based on the output produced by the discourse-aware TS approach presented in [6], using its reference implementation DisSim [34]. It splits and rephrases complex English sentences within the semantic context in which they occur. Based on a linguistically grounded transformation stage that uses clausal and phrasal disembedding mechanisms, complex sentences are transformed into *shorter utterances with a simple canonical structure* that can be easily analyzed by downstream applications. In this context, the notion of minimality is introduced, as one of the main objectives of the proposed approach is to decompose source sentences into a set of *self-contained minimal semantic units*. To avoid breaking down the input into a disjointed sequence of statements that is difficult to interpret because important contextual information is missing, the semantic context between the split propositions is incorporated in the form of *hierarchical structures and semantic relationships*. In that way, sentences with a complex linguistic structure are converted into a semantic hierarchy of minimal propositions that leads to a novel representation of complex assertions that puts a semantic layer on top of the simplified sentences (see Fig. 4).

Data model. The resulting semantic hierarchy of minimal propositions is represented in the form of linked proposition tree (LPT). Its basic structure is depicted in Fig. 5. A linked proposition tree is a labeled binary tree[3] $LPT = (V, E)$.

Let $V \in \{REL, PROP\}$ be the set of nodes, where $PROP$ is the set of leaf nodes denoting the set of minimal propositions.

A $prop \in PROP$ is a triple $(s, v, o) \in CT$, where CT represents the set of clause types: $CT = \{SV, SVA, SVC, SVO, SVOO, SVOA, SVOC\}$ (see Table 4).

$s \in S$ denotes a subject, $v \in V$ a verb and $o \in \{O, A, C, OO, OA, OA, \emptyset\}$ a direct or indirect object (O), adverbial (A) or complement (C) (or a combination thereof).

Hence, a minimal proposition $prop \in PROP$ is a simple sentence that is reduced to its clause type. Thus, it represents a minimal unit of coherent information where all optional constituents are discarded, resulting in an utterance that expresses a single event consisting of a predicate and its core arguments. Accordingly, a minimal proposition presents a simple and regular structure that is easier to process and analyze for downstream Open IE tasks.

Furthermore, let $REL = \{Contrast, List, Disjunction, Cause, Result, Background, Condition, Elaboration, Attribution, Purpose, Temporal\text{-}Before, Temporal\text{-}After, Temporal, Spatial, Unknown\text{-}Coordination, Un\text{-}known\text{-}Subordination\}$ be the set of rhetorical relations, comprising the set of inner nodes.

A $rel \in REL$ represents the semantic relationship that holds between its child nodes. It reflects the semantic context of the associated propositions $prop \in PROP$. In that way, the coherence structure of the input is preserved.

Finally, let $E \in CL$ be the set of constituency labels, with $CL \in \{core, context\}$.

A $c \in CL$ represents a labeled edge that connects two nodes $V \in LPT$. It enables the distinction between core information and less relevant contextual information. In that way, hierarchical structures between the split propositions $prop \in PROP$ are captured.

3.2. Creating a canonical context-preserving representation for open information extraction

We leverage the semantic hierarchy of simplified sentences that was generated in the previous stage by the discourse-aware TS framework introduced in [6], using it as a pre-processing step for *extracting semantically typed relational tuples from complex input sentences*. This approach is based on the idea that the fine-grained representation of complex assertions in the form of hierarchically ordered and semantically interconnected sentences that present a simplified syntax may support the task of Open IE in two dimensions:

Coverage and Accuracy Based on the assumption that shorter sentences with a more regular structure are easier to process and analyze for downstream Open IE applications, we hypothesize that the complexity of the relation extraction task will be reduced when taking advantage of the split sentences instead of dealing directly with the raw source sentences. Hence, we expect to improve the performance of state-of-the-art Open IE systems in terms of the *precision* and *recall* of the extracted relational tuples when operating on the simplified sentences.

Context-Preserving Representation We hypothesize that the semantic hierarchy generated by the discourse-aware TS approach can be leveraged to extract relations within the *semantic context* in which they occur. By enriching the output of state-of-the-art Open IE systems with additional meta information, we intend to create a novel context-preserving representation for the task of Open IE which extends the shallow semantic representation of current approaches to produce an output that captures contextual information. In that way, we aim to generate a representation for relational tuples that is more informative and coherent, and thus easier to interpret.

Below, we explain how state-of-the-art Open IE approaches can make use of the discourse-aware TS approach proposed in [6] as a pre-processing step to leverage the syntactically simplified sentences for facilitating the extraction of relational tuples and to enrich their output with semantic information.

3.2.1. Enriching state-of-the-art open information extraction approaches with semantic information

To extract a set of semantically typed relational tuples from a complex assertion, each minimal proposition $prop \in PROP$ that was generated by the discourse-aware TS approach (represented as leaf nodes in the linked proposition tree LPT) is sent as input to the Open IE system whose output shall be enriched with semantic information. For instance, in case of the example sentence from Table 8, the input to an Open IE system are the six split sentences shown in the lower part of the table. When using the state-of-the-art framework RnnOIE, the following seven relational tuples are extracted:

[3] In rare cases, a node may have more than two children, e.g. when the input sentence contains a list of noun phrases enumerating more than two entities.

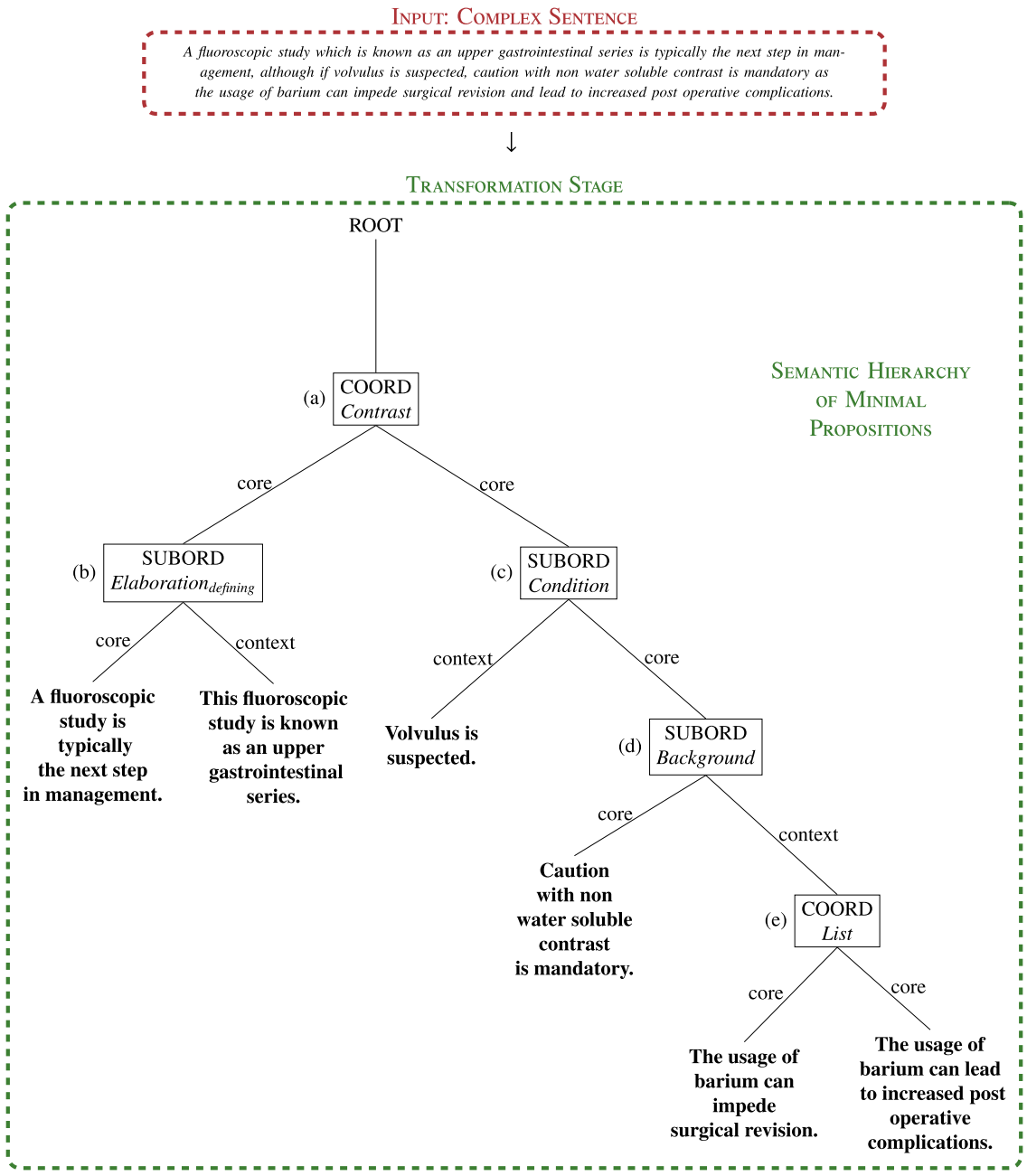

Fig. 4. Semantic hierarchy of minimal propositions representing the context-preserving output that is generated by the discourse-aware TS approach described in [6]. A complex input sentence is transformed into a semantic hierarchy of simplified sentences in the form of minimal, self-contained propositions that are linked to each other via rhetorical relations. The output presents a regular, fine-grained structure that preserves the semantic context of the input, allowing for a proper interpretation of complex assertions.

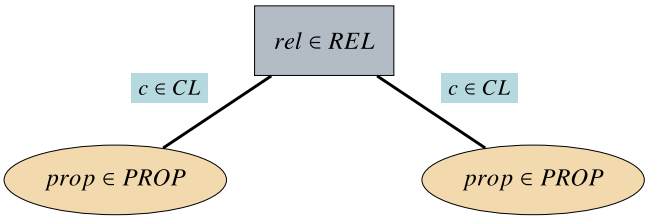

Fig. 5. Basic structure of a linked proposition tree *LPT*. It represents the data model of the semantic hierarchy of minimal propositions.

#1.1 ⟨*A fluoroscopic study*; *is*; *typically, the next step in management*⟩
#1.2 ⟨*A fluoroscopic study*; *is known*; *as an upper gastrointestinal series*⟩
#2.1 ⟨*Caution with non water soluble contrast*; *is*; *mandatory*⟩
#2.2.1 ⟨*Volvulus*; *is suspected*; ⟩
#2.2.2.1 ⟨*The usage of barium*; *can impede*; *surgical revision*⟩
#2.2.2.2 ⟨*The usage of barium*; *can lead*; *to increased post operative complications*⟩
#2.2.2.2 ⟨*The usage of barium*; *to increased*; *post operative complications*⟩

After identifying the predicate-argument structures of the input sentences, the extracted relational tuples need to be mapped to their semantic context. It is available from the linked proposition tree *LPT* in the form of semantic relationships *rel* ∈ *REL*

Table 8
Result of the sentence splitting subtask on an example sentence. A complex source sentence is decomposed into a set of minimal semantic units.

Complex source	A fluoroscopic study which is known as an upper gastrointestinal series is typically the next step in management, although if volvulus is suspected, caution with non water soluble contrast is mandatory as the usage of barium can impede surgical revision and lead to increased post operative complications.
Minimal propositions	A fluoroscopic study is typically the next step in management. This fluoroscopic study is known as an upper gastrointestinal series. Volvulus is suspected. Caution with non water soluble contrast is mandatory. The usage of barium can impede surgical revision. The usage of barium can lead to increased post operative complications.

between the split propositions $prop \in PROP$ and their hierarchical order $c \in CL$. As Open IE systems often extract more than one relational tuple at different levels of granularity from a given simplified sentence (e.g., #2.2.2.2), we need to determine which of them represents the main statement of the underlying minimal proposition. This decision is based on the following heuristic: an extraction is assumed to embody the main message of a sentence if

(1) the head verb of the input sentence is contained in the relational phrase *rel_phrase* of the extracted tuple (e.g., ⟨*Volvulus*; *is suspected*; ⟩); or
(2) the head verb of the input sentence equals the object argument arg_2 of the extracted tuple (e.g., ⟨*Volvulus*; *is*; *suspected*⟩).

For example, in the proposition "*Volvulus is suspected.*", the term "*suspected*" represents the head verb. With condition (1), we are able to match tuples such as ⟨*Volvulus*; *is suspected*; ⟩, where the head verb can be found in the predicate position. Condition (2), in contrast, covers tuples that contain the head verb in object position, which is often the case for Open IE approaches that extract verbal auxiliaries (here: "*is*") as the relational phrase and the main verb as the corresponding object argument, e.g. ⟨*Volvulus*; *is*; *suspected*⟩.

Any extraction that matches one of the above-mentioned criteria will be denoted as a head tuple *htuple(s)* of the corresponding simplified sentence *s*. The head verbs of the minimal propositions from our running example are underlined in the extracted tuples below. By applying the heuristic described above, we determine whether or not they represent the head tuple of one of the simplified sentences. If so, we map the extraction to their corresponding minimal proposition $prop \in PROP$ from the linked proposition tree *LPT*:

- ⟨*A fluoroscopic study*; *is*; *typically, the next step in management*⟩
  → *htuple*(#1.1)
- ⟨*A fluoroscopic study*; *is known*; *as an upper gastrointestinal series*⟩
  → *htuple*(#1.2)
- ⟨*Caution with non water soluble contrast*; *is*; *mandatory*⟩
  → *htuple*(#2.1)
- ⟨*Volvulus*; *is suspected*; ⟩
  → *htriplet*(#2.2.1)
- ⟨*The usage of barium*; *can impede*; *surgical revision*⟩
  → *htuple*(#2.2.2.1)
- ⟨*The usage of barium*; *can lead*; *to increased post operative complications*⟩
  → *htuple*(#2.2.2.2)
- ⟨*The usage of barium*; *to increased*; *post operative complications*⟩
  → ∅

Once the extractions are projected to the split sentences $prop \in PROP$ from the linked proposition tree *LPT*, the contextual hierarchy $c \in CL$ and the semantic relationship $rel \in REL$ that holds between two split sentences can be directly transferred to the extracted relational tuples, thus embedding them within the semantic context derived from the source sentence. Extractions that do not represent a head tuple, as is the case for the second extraction from sentence #2.2.2.2 in our example, are discarded.

In that way, any Open IE system can make use of the discourse-aware TS framework to enrich its extractions with semantic information. The semantically typed output produced by the state-of-the-art Open IE approach RnnOIE when applying the framework as a pre-processing step is shown in Fig. 6 in the form of a lightweight semantic representation. In this format, the extracted relations are grouped by the sentences in which they occur. Each extraction ((1), (2), (3), (4), (5) and (6)) is represented by a tab-separated string consisting of an identifier *id*, a context layer *cl* and the core tuple that is represented by a binary tuple $t \leftarrow (arg_{subj}, rel\_phrase, arg_{obj})$, where *rel_phrase* denotes a relational phrase, arg_{subj} a subject argument and arg_{obj} an object argument. Contextual information is provided by contextual arguments (C_S and C_L), which are indicated by an additional indentation level to their parent tuples (e.g., (1a) and (1b)). Their representation consists of a type string and tab-separated content. The type string encodes both the context type (S for a simple contextual argument $c_S \in C_S$, and L for a linked contextual argument $c_L \in C_L$), and the semantic relationship (e.g. Contrast, List) that holds between the contextual argument and the associated parent extraction. Finally, the content of a simple contextual argument is a textual expression in the form of a phrase, whereas the content of a linked contextual argument is the identifier of the target core tuple to which the respective parent relational tuple is linked. For further details on the output format, see Section 3.2.2.

To allow for a better comparison, we added the set of relational tuples that are returned by RnnOIE when directly operating on the raw source sentence (see Fig. 7). As can be seen, in this case, its output represents only a loose arrangement of relational tuples that are hard to interpret as they ignore the context under which an extraction is complete and correct, and thus lacks the expressiveness needed for a proper interpretation of the extracted predicate-argument structures. For instance, for a correct understanding of our running example, it is important to distinguish between information asserted in the sentence and information that is only conditionally true, since the information contained in the statement "*caution with non water soluble contrast is mandatory*" depends on the pre-posed if-clause. As the output depicted in Fig. 7 shows, RnnOIE is not able to capture this relationship, though. This lack of expressiveness impedes the interpretation of the extracted tuples and is vulnerable to incorrect reasoning in downstream Open IE applications. However, with the help of the semantic hierarchy generated by the discourse-aware TS approach, its output can easily be enriched with contextual information that allows to capture the semantic relationship between the set of extracted relations and, hence, preserve their interpretability in downstream tasks, as illustrated in Fig. 6.

Moreover, the output from Fig. 6 demonstrates that the TS approach supports the Open IE system under consideration in

```
RnnOIE (using discourse-aware TS framework):
(1) #1    0    A fluoroscopic study; is; typically, the next step in management
(1a)               L:ELABORATION    #2
(1b)               L:CONTRAST       #3
(2) #2    1    A fluoroscopic study; is known; as an upper gastrointestinal series
(3) #3    0    Caution with non water soluble; is; mandatory
(3a)               L:CONTRAST       #1
(3b)               L:CONDITION      #6
(3c)               L:BACKGROUND     #4
(3d)               L:BACKGROUND     #5
(4) #4    1    The usage of barium; can impede; surgical revision
(4a)               L:LIST           #5
(5) #5    1    The usage of barium; can lead; to increased post operative complications
(5a)               L:LIST           #4
(6) #6    1    Volvulus; is suspected;
```

Fig. 6. Semantically enriched output of RnnOIE when using the discourse-aware TS approach as a pre-processing step. The extracted relational tuples are augmented with contextual information in terms of intra-sentential rhetorical structures and hierarchical relationships, thus preserving the coherence structure of the complex source sentence.

```
RnnOIE (stand-alone):
(1) A fluoroscopic study; known; as an upper gastrointestinal series
(2) caution with non water soluble contrast; is; mandatory as the usage of barium
(3) as the usage; of barium can impede; surgical revision and lead
(4)  ; to increased; post operative complications
```

Fig. 7. Relational tuples extracted by RnnOIE *without* using the discourse-aware TS approach as a pre-processing step. The output represents a loose arrangement of relational tuples that are hard to interpret as they ignore the semantic context under which an extraction is complete and correct.

a second dimension: using it as a pre-processing step to split the input into a sequence of syntactically simplified sentences improves the precision and recall of the extracted relational tuples. For instance, the arguments in the object slots of extraction (2) and (3) in Fig. 7 are inaccurate, as they incorporate extra phrasal elements that should rather form the basis of a separate extraction, thus leading to a malformed relational tuple that has no meaningful interpretation. However, when operating on the decomposed sentences, RnnOIE successfully divides the information contained in tuple (2) in Fig. 7 into two individual extractions that are properly structured ((2) and (3) in Fig. 6); the same holds for extraction (3) in Fig. 7. In addition, when operating on the split components, RnnOIE succeeds in extracting the tuple ⟨*Volvulus*; *is suspected*; ⟩, which it was not able to detect when running over the original complex sentence.

3.2.2. Generating a formal semantic representation for open information extraction

The relational tuples that are extracted from a set of simplified sentences by the Open IE approaches are turned into a lightweight sentence-level meaning representation (MR) in the form of semantically typed predicate-argument structures. For this purpose, we augment the shallow semantic representation of state-of-the-art Open IE systems with

(1) *rhetorical structures* that establish a semantic link between the extracted tuples, and
(2) information about the *hierarchical level* of each extraction.

In order to represent contextual relationships between the extracted relational tuples, the default representation of an Open IE tuple needs to be extended. Therefore, we propose a novel *lightweight semantic representation for the output of Open IE systems* that is both machine processable and human readable.

In this representation, a binary relational tuple $t \leftarrow (arg_{subj}, rel\_phrase, arg_{obj})$ is extended with:

- a unique identifier *id*;
- information about the contextual hierarchy, the so-called *context layer cl*; and
- two sets of semantically classified contextual arguments C_S (*simple contextual arguments*) and C_L (*linked contextual arguments*).

Accordingly, the tuples extracted from a complex sentence are represented as a set of (id, cl, t, C_S, C_L) tuples. The *context layer cl* encodes the hierarchical order of core and contextual facts. Tuples with a context layer of 0 carry the core information of a sentence, whereas tuples with a context layer of $cl > 0$ provide contextual information about extractions with a context layer of $cl - 1$. Both types of contextual arguments, C_S and C_L, provide (semantically classified) contextual information about the statement expressed in t. Whereas a simple contextual argument $c_S \in C_S, c_S \leftarrow (s, rel)$ contains a textual span s that is classified by the semantic relation *rel*, a linked contextual argument $c_L \in C_L, c_L \leftarrow (id(z), rel)$ refers to the content expressed in another relational tuple z.[4]

Example. In the following, we will illustrate how the complex sentence from our running example from Table 8, whose semantic hierarchy is depicted in Fig. 8 in the form of a linked proposition tree, is transformed into the lightweight semantic representation described above.

In the relation extraction stage, the head tuple t is extracted from each leaf node of the linked proposition tree, i.e. from each simplified sentence. In our case, this process results in the following predicate-argument tuples:

[4] C_S represents a single phrasal element, while C_L denotes a full relational tuple with a separate subject and object argument. The identification of the type string consisting of a context type and a semantic relationship, as well as the generation of the proposition *prop* from which the contextual argument is extracted is encoded in the transformation patterns that are used during the pre-processing step where complex sentences are transformed into simplified propositions using clausal and phrasal disembedding mechanisms. The extraction of phrases results in simple contextual arguments, whereas splitting clauses leads to linked contextual arguments. For more details, the interested reader may refer to [6].

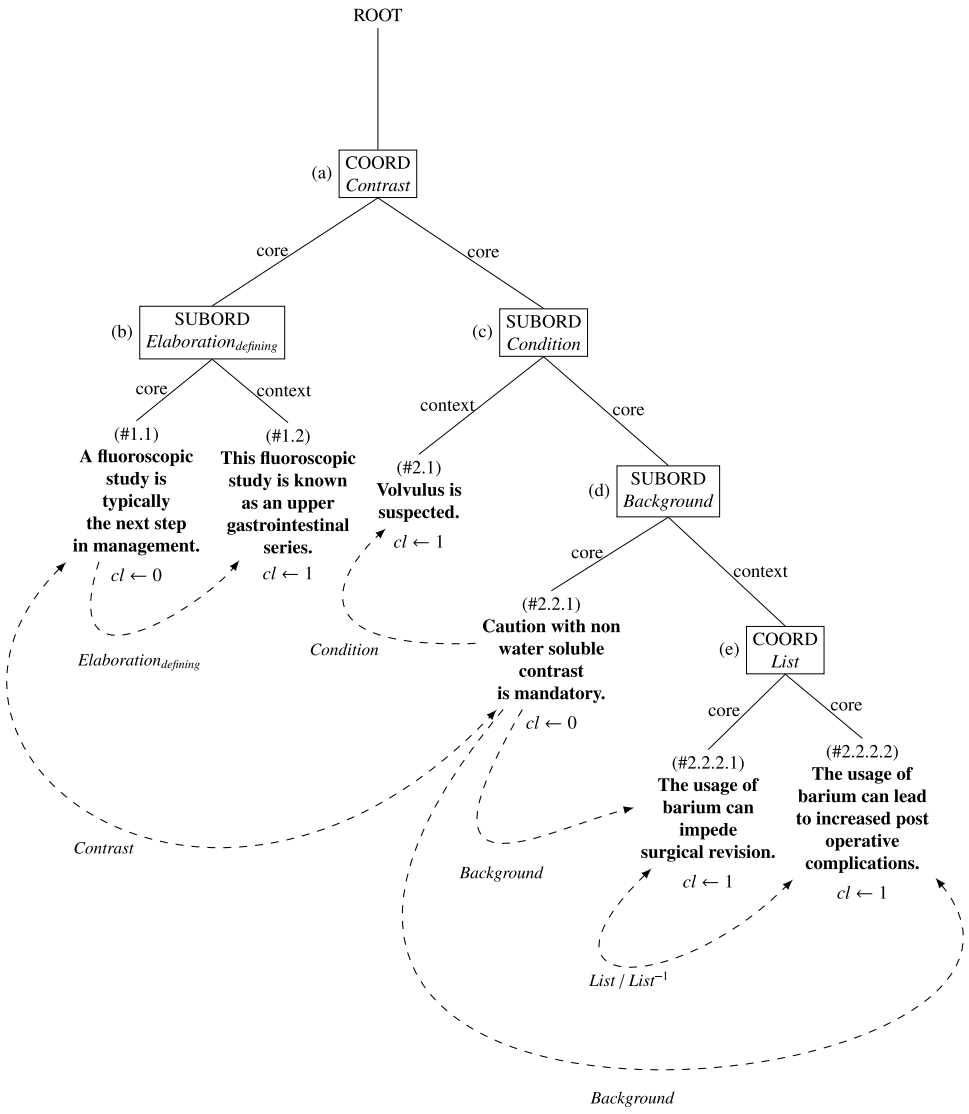

Fig. 8. Final linked proposition tree of our example sentence.

- t_1: ⟨*A fluoroscopic study; is typically; the next step in management*⟩
- t_2: ⟨*This fluoroscopic study; is known; as an upper gastrointestinal series*⟩
- t_3: ⟨*Volvulus; is suspected;*⟩
- t_4: ⟨*Caution with non water soluble contrast; is; mandatory*⟩
- t_5: ⟨*The usage of barium; can impede; surgical revision*⟩
- t_6: ⟨*The usage of barium; can lead; to increased post operative complications*⟩

Next, a tuple (id, cl, t, C_S, C_L) is created for every extraction as follows: The value of the context layer *cl* is determined by counting the number of context edges in the path from the root node to the corresponding simplified sentence in the linked proposition tree. Our approach is based on the assumption that the core information of any internal node *n* in the tree is represented by sentences that

(1) are leaf nodes of the subtree spanned by *n*, and
(2) are connected to *n* by a path that consists of core edges only.

Such a path is referred to as a *core chain*. The set of leaf nodes that are connected to *n* by a core chain are called *representatives* of *n*. For instance, the core information of node (a) in Fig. 8 can be found in the representatives (#1.1) and (#2.2.1) by following the core chains that pass through the nodes (a)(b)(#1.1) and (a)(c)(d)(#2.2.1). We further claim that any relationship *rel* (represented by an internal node *n*) that holds between the child nodes of *n* (let *a* and *b* denote these child nodes in the binary setting) also holds pairwisely between representatives of *a* and *b*. Fig. 9 illustrates an example. As visualized by the dashed arrows, the relation *rel* between the nodes *a* and *b* applies to the following pairs of sentences: {{1, 3}, {1, 4}, {2, 3}, {2, 4}}. Note that these arrows define a direction for each relation *rel* that is established between two sentences *x* and *y*. Such a directed edge (x, y) is called a *contextual link* for the relation *rel* in our approach.

The direction of the contextual link indicates how the relationship between the two sentences is represented: for each contextual link (x, y), we will attach *y* to *x* as a contextual argument, semantically classified by the corresponding relation *rel*. By default, the identifier of *y* will be added to the set of linked contextual arguments C_L of the sentence *x*.

The contextual argument assignment depends on the constituency type of the corresponding relational node, as shown in Fig. 9. For a coordination node *n*, we create two contextual links for each relation *rel* that holds between two representatives *x* and *y*: (x, y) for the relation *rel* and (y, x) for the inverse relation rel^{-1} (see Table A.13a in Appendix A). Consequently, core sentence

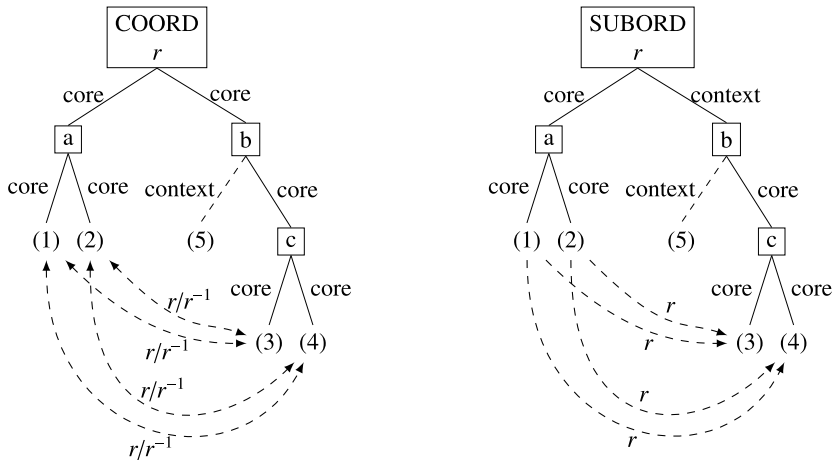

Fig. 9. Establishing contextual links between sentences connected by coordination nodes *(left)* and subordination nodes *(right)*.

```
(1) #1    0    A fluoroscopic study;    is typically;    the next step in management
(1a)              L:ELABORATION_defining    #2
(1b)              L:CONTRAST                #3
(2) #2    1    This fluoroscopic study;    is known;      as an upper gastrointestinal
                                                          series
(3) #3    0    Caution with non water soluble contrast;     is;    mandatory
(3a)              L:CONTRAST                #1
(3b)              L:CONDITION               #4
(3c)              L:BACKGROUND              #5
(3d)              L:BACKGROUND              #6
(4) #4    1    Volvulus;                 is suspected;
(5) #5    1    The usage of barium;      can impede;      surgical revision
(5a)              L:LIST                    #6
(6) #6    1    The usage of barium;      can lead;        to increased post operative
                                                          complications
(6a)              L:LIST                    #5
```

Fig. 10. RDF-NL representation of our running example.

representatives x and y are mutually connected to each other via linked contextual arguments classified as *rel* and rel^{-1}. In contrast, subordination nodes n establish a single contextual link (x, y) for each relation *rel* that holds between two representatives x and y, with y being inside the contextual subtree of n. As a result, a core sentence representative x is enriched with a linked contextual argument that refers to the context sentence representative y, but not vice versa.

The final output of our running example in the form of our proposed lightweight semantic representation is shown in Fig. 10.

To facilitate both a manual analysis and machine processing of the extracted relational tuples, a human and machine readable format, called RDF-NL, is generated by our framework (see Figs. 10 and 11). In this format, relational tuples are represented by tab-separated strings, specifying the identifier *id*, context layer *cl* and the core extraction (e.g., (1) and (2) in Fig. 10) that is represented by a binary relational tuple $t \leftarrow (arg_{subj}, rel\_phrase, arg_{obj})$:

- subject argument arg_{subj},
- relation name *rel*, and
- object argument arg_{obj}.

Contextual arguments (C_S and C_L) are indicated by an extra indentation level to their parent tuples (e.g., (1a) and (1b) in Fig. 10). Their representation consists of a type string and a tab-separated content. The type string encodes both the context type (S for a simple contextual argument $c_S \in C_S$ and L for a linked contextual argument $c_L \in C_L$) and the classified semantic relation (e.g., *Cause*, *Purpose*). The content of a simple contextual argument is a single phrase (e.g., (4a) in Fig. 11), whereas the content of a linked contextual argument is the identifier of the respective target extraction, which again represents an entire relational tuple (e.g., (4b) in Fig. 11).

In addition, our framework can materialize its extractions into a graph serialized under the N-Triples[5] specification of the RDF standard (see Fig. 12). In this format, each sentence is modeled as a blank node comprising the original sentence (1) and a list of contained extractions ((2) to (13)). Each relational tuple is represented as a blank node of RDF text resources that represent the subject argument arg_{subj} (6), the relation name *rel* (7) and the object argument arg_{obj} (8) of the relational tuple $t \leftarrow (arg_{subj}, rel\_phrase, arg_{obj})$. Information about the context layer *cl* (5) is stored as well. Contextual information is attached by RDF predicates that encode both the context type and the classified semantic relation, analogous to the RDF-NL format. These RDF predicates link to either text resources in the case of a simple contextual argument[6] or to other extractions (9, 11) in the case of a linked contextual argument. Finally, all RDF text resources are mapped to their unescaped literal values (12).

[5] https://www.w3.org/TR/n-triples.

[6] An example of a text resource denoting a simple contextual argument can be found in (8) in Fig. 13.

```
(1) #1    0    The house;          was once;     part of a plantation
(1a)           L:LIST                            #2
(2) #2    0    It;                 was;          the home of Josiah Henson
(2a)           L:ELABORATION_non_defining        #3
(2b)           L:LIST                            #1
(3) #3    1    Josiah Henson;      was;          a slave
(3a)           L:ELABORATION_defining            #4
(3b)           L:ELABORATION_defining            #5
(4) #4    2    A slave;            escaped;      to Canada
(4a)           S:TEMPORAL                        in 1830
(4b)           L:LIST                            #5
(5) #5    2    A slave;            wrote;        the story of his life
(5a)           L:LIST                            #4
```

Fig. 11. RDF-NL representation of the sentence "*The house was once part of a plantation and it was the home of Josiah Henson, a slave who escaped to Canada in 1830 and wrote the story of his life*.".

4. Evaluation

Assuming that the fine-grained representation of complex sentences in the form of hierarchically ordered and semantically interconnected sentences that present a simplified syntax supports the extraction of semantically typed relational tuples from complex input sentences, we examine the use of the discourse-aware TS approach introduced in [6] for downstream Open IE applications. For this purpose, we integrate the TS framework as a pre-processing step into a variety of state-of-the-art Open IE systems, including ReVerb, Ollie, ClausIE, Stanford Open IE, PropS, OpenIE-4, MinIE, OpenIE-5 and RnnOIE (see Section 4.1.1). We then determine whether these systems profit from the output generated by the proposed discourse-aware TS approach along two different dimensions:

(1) In a first step, we measure if their performance is improved in terms of *recall* and *precision* of the extracted relational tuples (see Section 4.1.3).[7]
(2) In a second step, we examine whether the semantic hierarchy generated by the TS framework is beneficial in *extending the shallow semantic representation of state-of-the-art Open IE approaches with additional meta information* to produce an output that preserves contextual information of the extracted relational tuples (see Section 4.1.5).

4.1. Experimental setup

In the following, we present the experimental setup for assessing the use of the semantic hierarchy of minimal propositions generated by the discourse-aware TS approach as a support for the task of Open IE.

4.1.1. Baselines

To demonstrate the merits of the discourse-aware TS approach for downstream Open IE tasks, we conduct a comparative analysis examining the output of the following state-of-the-art Open IE approaches[8]:

(i) ReVerb, an early Open IE approach that is based on hand-crafted extraction patterns;
(ii) its successor system Ollie;
(iii) ClausIE, which is the first Open IE system that is based on the idea of incorporating a sentence re-structuring stage;
(iv) Stanford Open IE, which follows a similar approach by splitting complex sentences into a set of logically entailed shorter utterances;
(v) PropS, an Open IE approach that applies a semantically-oriented sentence representation as the basis for the extraction of relational tuples;
(vi) Ollie's successor OpenIE-4;
(vii) MinIE, which is built on top of ClausIE with the objective of minimizing both relational and argument phrases;
(viii) OpenIE-5, which was presented as OpenIE-4's succeeding system; and
(ix) the first end-to-end neural approach to the task of Open IE, RnnOIE.

4.1.2. Benchmarks

To examine the use of sentence splitting as a pre-processing step for state-of-the-art Open IE systems, we conducted experiments on two commonly applied Open IE benchmarks:

OIE2016. The first dataset we used is [31]'s OIE2016 benchmark framework, which has become the de facto standard for the evaluation of the task of Open IE. It consists of 10,359 extractions over 3200 sentences from Wikipedia and newswire articles from the Wall Street Journal.[9]

CaRB. Furthermore, we assessed and compared the quality of the output produced by the Open IE systems under consideration on the CaRB benchmark, a crowdsourced Open IE dataset [33]. It is comprised of a subset of 1282 sentences from the OIE2016 corpus, which were manually annotated using Amazon Mechanical Turk.[10]

4.1.3. Comparative analysis of the outputs

To evaluate the impact of sentence splitting on the coverage and accuracy of state-of-the-art Open IE approaches, we conduct a comparative analysis using the Open IE benchmarks described above. We hypothesize that the complexity of the relation extraction task in Open IE systems will be reduced by taking advantage of the split sentences instead of dealing with the raw complex source sentences. This assumption is based on the idea that shorter sentences with a more regular structure are easier to process and analyze for downstream Open IE applications. Hence,

[7] Prior work has also shown that sentence splitting may be used as a preprocessing step to facilitate and boost the performance of further artificial intelligence (AI) tasks, including, for instance, Text Summarization [35] and Machine Translation [36], as it removes nested structures and brings related pieces of information closer together.

[8] We use the default configuration of each system.

[9] The OIE2016 benchmark dataset is available under https://github.com/gabrielStanovsky/oie-benchmark.

[10] The CaRB benchmark for Open IE is available for download under https://github.com/dair-iitd/CaRB.

```
A fluoroscopic study which is known as an upper gastrointestinal series is
typically the next step in management, although if volvulus is suspected,
caution with non water soluble contrast is mandatory as the usage of
barium can impede surgical revision and lead to increased post operative
complications.

(1)     _:fa89da284aee42469dafe015bb2b79e1
<http://lambda3.org/graphene/sentence#original-text>
"A fluoroscopic study which is known as an upper gastrointestinal series
is typically the next step in management , although if volvulus is
suspected , caution with non water soluble contrast is mandatory as the
usage of barium can impede surgical revision and lead to increased post
operative complications
."^^<http://www.w3.org/2001/XMLSchema#string> .
(2)     _:fa89da284aee42469dafe015bb2b79e1
        <http://lambda3.org/graphene/sentence#has-extraction>
(3)     _:7a43269ddb2a4489a76256278c2534df .
(4)     _:7a43269ddb2a4489a76256278c2534df
        <http://lambda3.org/graphene/extraction#extraction-type>
        "VERB_BASED"^^<http://www.w3.org/2001/XMLSchema#string> .
(5)     _:7a43269ddb2a4489a76256278c2534df
        <http://lambda3.org/graphene/extraction#context-layer>
        "0"^^<http://www.w3.org/2001/XMLSchema#integer> .
(6)     _:7a43269ddb2a4489a76256278c2534df
        <http://lambda3.org/graphene/extraction#subject>
        <http://lambda3.org/graphene/text#A+fluoroscopic+study> .
(7)     _:7a43269ddb2a4489a76256278c2534df
        <http://lambda3.org/graphene/extraction#predicate>
        <http://lambda3.org/graphene/text#is+typically> .
(8)     _:7a43269ddb2a4489a76256278c2534df
        <http://lambda3.org/graphene/extraction#object>
        <http://lambda3.org/graphene/text#the+next+step+in+management> .
(9)     _:7a43269ddb2a4489a76256278c2534df
        <http://lambda3.org/graphene/extraction#L-IDENTIFYING_DEFINITION>
(10)    _:27c44ff725af4dcd81b78d7026a464e9 .
(11)    _:7a43269ddb2a4489a76256278c2534df
        <http://lambda3.org/graphene/extraction#L-CONTRAST>
(12)    _:cb9dd041b06e4001b6a2b3947cdeb688 .
        <http://lambda3.org/graphene/text#A+fluoroscopic+study>
        <http://www.w3.org/1999/02/22-rdf-syntax-ns#value> "A fluoroscopic
        study"^^<http://www.w3.org/2001/XMLSchema#string> .
        <http://lambda3.org/graphene/text#is+typically>
        <http://www.w3.org/1999/02/22-rdf-syntax-ns#value> "is
        typically"^^<http://www.w3.org/2001/XMLSchema#string> .
        <http://lambda3.org/graphene/text#the+next+step+in+management>
        <http://www.w3.org/1999/02/22-rdf-syntax-ns#value> "the next step in
        management"^^<http://www.w3.org/2001/XMLSchema#string> .
(13)    _:fa89da284aee42469dafe015bb2b79e1
        <http://lambda3.org/graphene/sentence#has-extraction>
...
```

Fig. 12. RDF-Triples representation of our running example.

we expect to improve the performance of state-of-the-art Open IE systems in terms of the precision and recall of the extracted relational tuples when operating on the simplified sentences generated by the discourse-aware TS approach. To examine this hypothesis, we compare the performance of the Open IE approaches listed in Section 4.1.1 when directly operating on the raw input data with their performance when the reference TS implementation DisSim is used as a pre-processing step to split complex source sentences into a set of syntactically simplified sentences with a canonical subject–predicate–object structure.

The Open IE systems are evaluated according to the following metrics:

- We compute the PR curve by evaluating the systems' performance at different extraction confidence thresholds.
- We calculate the AUC as a scalar measure of the systems' overall performance.
- We report the average precision and recall of the extracted relational tuples, as well as their F_1 score.

```
The house was once part of a plantation and it was the home of Josiah Henson,
a slave who escaped to Canada in 1830 and wrote the story of his life.

(1)     _:206b3c28cdad40df8b995b05bad3d8ac
        <http://lambda3.org/graphene/sentence#original-text>
        "The house was once part of a plantation and it was the
        home of Josiah Henson , a slave who escaped to Canada
        in 1830 and wrote the story of his life
        ."^^<http://www.w3.org/2001/XMLSchema#string> .
...
(2)     _:206b3c28cdad40df8b995b05bad3d8ac
        <http://lambda3.org/graphene/sentence#has-extraction>
        _:ae87562d67c5425ba321927796091c74 .
(3)     _:ae87562d67c5425ba321927796091c74
        <http://lambda3.org/graphene/extraction#extraction-type>
        "VERB_BASED"^^<http://www.w3.org/2001/XMLSchema#string> .
(4)     _:ae87562d67c5425ba321927796091c74
        <http://lambda3.org/graphene/extraction#context-layer>
        "2"^^<http://www.w3.org/2001/XMLSchema#integer> .
(5)     _:ae87562d67c5425ba321927796091c74
        <http://lambda3.org/graphene/extraction#subject>
        <http://lambda3.org/graphene/text#A+slave> .
(6)     _:ae87562d67c5425ba321927796091c74
        <http://lambda3.org/graphene/extraction#predicate>
        <http://lambda3.org/graphene/text#escaped> .
(7)     _:ae87562d67c5425ba321927796091c74
        <http://lambda3.org/graphene/extraction#object>
        <http://lambda3.org/graphene/text#to+Canada> .

(8)     _:ae87562d67c5425ba321927796091c74
        <http://lambda3.org/graphene/extraction#S-TEMPORAL>
        <http://lambda3.org/graphene/text#in+1830+.> .
(9)     _:ae87562d67c5425ba321927796091c74
        <http://lambda3.org/graphene/extraction#L-LIST>
        _:ee5ea62c09664638801cf665aefdfeb0 .

(10)     <http://lambda3.org/graphene/text#A+slave>
        <http://www.w3.org/1999/02/22-rdf-syntax-ns#value> "A
        slave"^^<http://www.w3.org/2001/XMLSchema#string> .
(11)     <http://lambda3.org/graphene/text#escaped>
        <http://www.w3.org/1999/02/22-rdf-syntax-ns#value>
        "escaped"^^<http://www.w3.org/2001/XMLSchema#string> .
(12)    <http://lambda3.org/graphene/text#to+Canada>
        <http://www.w3.org/1999/02/22-rdf-syntax-ns#value> "to
        Canada"^^<http://www.w3.org/2001/XMLSchema#string> .
(13)    <http://lambda3.org/graphene/text#in+1830+.>
        <http://www.w3.org/1999/02/22-rdf-syntax-ns#value> "in 1830
        ."^^<http://www.w3.org/2001/XMLSchema#string> .
...
```

Fig. 13. RDF-Triples representation of the example sentence from Fig. 11.

To compare the performance of the Open IE systems, we integrated them into the following benchmarks:

OIE2016. To allow for a comparison with current approaches, we extended the OIE2016 benchmark by incorporating more recent Open IE systems, including MinIE, OpenIE-5 and RnnOIE.[11] Moreover, since MinIE and PropS do not provide confidence scores for their extractions, we apply the following confidence function[12]:

$$conf_s(n) \leftarrow \frac{1}{0.005n + 1} \tag{1}$$

This function is based on the assumption that the longer the input, the more difficult and thus error-prone is the extraction of relations and their corresponding arguments. For instance, it

[11] We used the latest version of the OIE2016 benchmark (commit 6d9885e50c7c02cbb41ee98–ccce9757fda94c89c on their github repository).

[12] Though the authors of ClausIE specify a method for calculating the confidence scores of their system's extractions, it is not implemented in the published source code. Therefore, we make use of the confidence function from Eq. (1) for ClausIE's extractions, too.

yields a confidence level of 80% for a sentence with 50 characters, while a sentence containing 200 characters results in a confidence score of only 50%.

To match predicted extractions to reference extractions, the OIE2016 benchmark serializes both the predicted and the gold tuples into a sentence and computes their lexical match. A pair of predicted-reference tuples is declared a match if the fraction of words of the prediction that is also present in the reference surpasses a pre-defined threshold (0.25 by default).[13] Based on this lexical matching function, a set of gold tuples is compared with a set of predicted tuples to estimate word-level precision and recall, from which the systems' AUC score and PR curve are derived.

However, the scoring method used in the OIE2016 benchmark for matching predictions to ground truth tuples has been criticized for not being robust [32,33]. One of its major weaknesses is that the matching function does not penalize extractions that are overly long. Since it only checks whether a sufficient number of tokens from the reference occurs in the predicted spans, extractions with long relational or argument phrases have a natural advantage over shorter extractions, as they are more likely to contain more words from the reference tuples. Moreover, the scoring function does not penalize extractions for misidentifying parts of a relation in an argument slot (or vice versa), since it concatenates the relational phrase and its arguments to a continuous text span. Finally, the scorer loops over gold tuples in an arbitrary order and matches them to predicted extractions in a sequential manner; once a gold matches to a predicted extraction, it is rendered unavailable for any subsequent, potentially better fitting extraction.

CaRB. To overcome the limitations of the scoring function used in the OIE2016 benchmark, [33] presented an improved matching function addressing the issues described above. Instead of a lexical match, CaRB's scorer utilizes a tuple match, i.e. it matches relational phrase with relational phrase and argument slot with argument slot. Moreover, CaRB creates an all-pair matching table, with each column representing a predicted tuple and each row representing a gold tuple, rather than greedily matching references to predictions in arbitrary order. Between each pair of tuples, precision and recall are computed. For determining the overall recall, the maximum recall score is taken in each row, and averaged, while for computing the overall precision, each system's predictions are matched one to one with gold tuples, in the order of best match score to worst; the precision scores are then averaged to get the overall precision score. To calculate the PR curve, this computation is done at different confidence thresholds of the predicted extractions. Note that CaRB uses a multi-match approach for recall computation, which prevents from overly penalizing systems that merge information from multiple gold tuples into a single extraction. In contrast, its precision computation is based on a single match approach between predictions and references, penalizing systems that produce multiple very similar and redundant extractions.

For the evaluation of the Open IE approaches, we do not use the CaRB gold tuples because they are ill-suited for our purposes for a variety of reasons. First, though atomicity was one of the declared goals when creating the ground truth reference, we noticed that CaRB's gold tuples contain a lot of overgenerated predicates and overly long argument phrases. This is contrary to our intentions, as one of the central goals of our approach is to generate extractions where each tuple represents an indivisible unit that cannot be further decomposed into meaningful propositions. Moreover, CaRB's reference tuples often convey inferred relations that are not explicitly stated in the input sentences. Conversely, our Open IE approach focuses on the extraction of relations that are explicitly expressed in the text, rather than extracting information that is merely implied by it. Finally, the annotators of the CaRB corpus only marked up binary relations (optionally augmented by temporal and local information), while our approach aims to capture the full context of a relation. Therefore, we chose to keep the ground truth relations of the OIE2016 corpus, and only use the improved evaluation metrics of the CaRB benchmark.

Similar to the OIE2016 benchmark, we integrated MinIE and RnnOIE into the CaRB framework to allow for a comparison with more recent approaches.

In two consecutive rounds, each Open IE system is quantitatively evaluated by calculating the PR curve of its extractions at different confidence thresholds, by computing the AUC score and by measuring average precision, recall and F_1 of the extracted relational tuples. In the first round, each system is executed on the original complex input sentences, whereas in the second round, the simplified sentences generated by the discourse-aware TS approach are taken as input. Based on the results we obtain we can make a statement about how sentence splitting affects the performance of state-of-the-art Open IE systems in terms of *coverage* and *accuracy* of the extracted relations when applied as a pre-processing step.

4.1.4. Qualitative analysis of the extracted relations

To complement the automatic evaluation from the previous section, which assessed the impact of sentence splitting on the systems' performances by means of a set of automatic metrics, we conduct a manual analysis for a more in-depth examination of the extracted relational tuples. In this context, we compare and discuss the specifics of the relations extracted by each approach using a representative example, and examine the impact on the generated output when operating on the semantic hierarchy of minimal propositions instead of dealing with the original source sentences.

4.1.5. Analysis of the lightweight semantic representation of relational tuples

Finally, we investigated whether the semantic hierarchy of minimal propositions generated by the discourse-aware TS approach can be leveraged to transform the shallow semantic representation of state-of-the-art Open IE systems into a canonical context-preserving representation of relational tuples. For this purpose, we analyzed the core characteristics of the resulting representation in the light of the experiments outlined above. The aim was to determine to what extent the semantic hierarchy of simplified sentences supports

(i) the *extraction of simplistic canonical predicate-argument structures* from complex source sentences, and
(ii) the *enrichment of the extracted relational tuples with additional meta information* to produce an output that preserves contextual information.

4.2. Results and discussion

In the section below, we examine how discourse-aware sentence splitting affects the performance of state-of-the-art Open IE systems, both in terms of the coverage and accuracy of the extracted relational tuples, and the enrichment of the output with contextual information.

[13] The original scoring function presented in [31] was more restrictive. Here, a prediction was classified as being correct if the predicted and the reference tuple agree on the grammatical head of all their elements, i.e. the predicate and the arguments. However, this scheme was later relaxed in their github repository to the lexical matching function described above, which has become the OIE2016 benchmark's default scorer in the current literature.

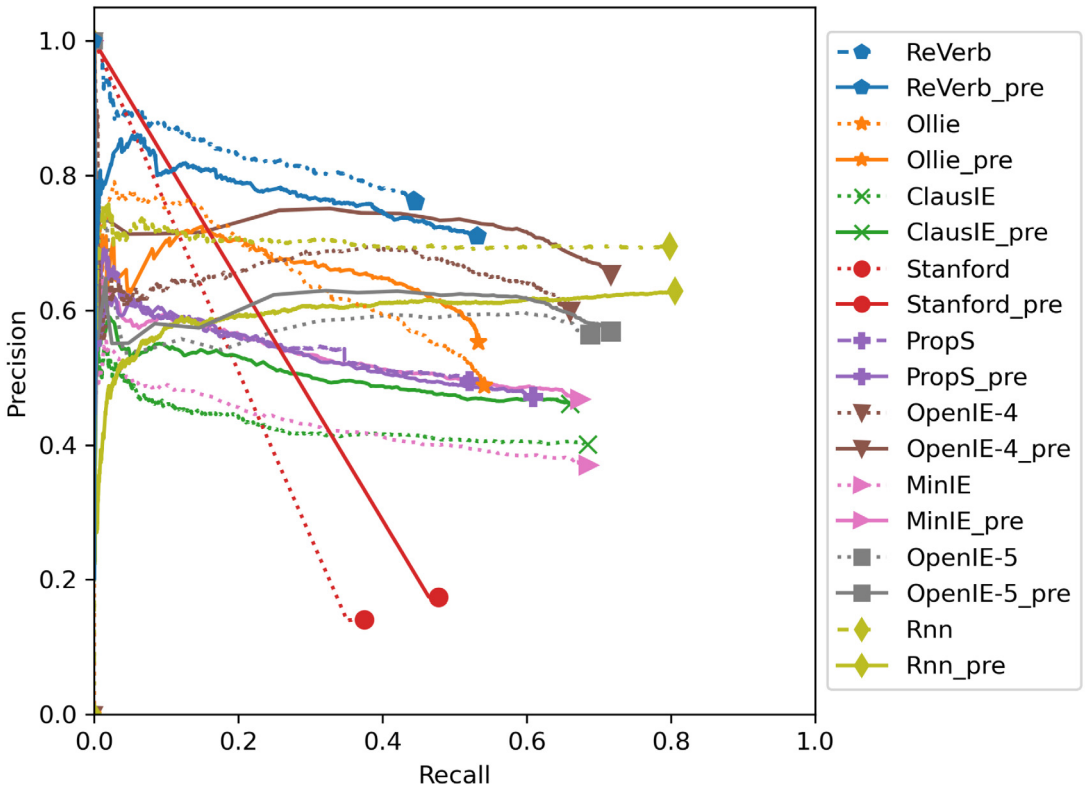

Fig. 14. Comparative analysis of the performance of state-of-the-art Open IE systems on the OIE2016 benchmark *with* (solid lines) and *without* (dotted lines) sentence splitting as a pre-processing step.

4.2.1. Comparative analysis of the outputs

In a first step, we examine the impact of the discourse-aware TS approach on the accuracy and coverage of the relations extracted by a set of state-of-the-art Open IE systems. For this purpose, we conduct a quantitative comparison of the systems' recall and precision scores on the OIE2016 and CaRB benchmarks with and without using the sentence splitting framework as a pre-processing step, and analyze whether their particular deficiencies can be eliminated when operating on the structurally simplified sentences.

OIE2016. Fig. 14 demonstrates the effectiveness of the TS process that splits complex sentences into a set of minimal propositions with a more regular syntax, when being applied as a pre-processing step for the task of Open IE. It shows the PR curves of the Open IE baseline systems that have been executed on the OIE2016 benchmark as a stand-alone system (dotted lines), and within our TS-Open IE pipeline, where they act as the relation extraction component after the transformation stage (solid lines). The average precision, recall, F_1 and AUC scores when operating on top of the simplified sentences are listed in Table 9, while the corresponding improvements – as compared to their stand-alone counterparts – are given in Table 10.

The scores in Table 10 show that when splitting complex sentences into a set of syntactically simplified propositions, all systems under consideration gain in F_1 and AUC, except for Ollie and RnnOIE. The highest improvement in AUC is achieved by ReVerb, yielding a 39.19% increase over the output produced when acting as a stand-alone system, followed by Stanford Open IE (+35.47%) and MinIE (+21.28%). On the contrary, Ollie's (−1.12%) and RnnOIE's (−14.13%) AUC scores somewhat decline. Regarding F_1, the picture is similar. While Ollie (−0.34%) and RnnOIE (−8.30%) lose in F_1, all other systems under consideration enhance their score, above all Stanford Open IE (+25.49%), MinIE (+12.48%) and OpenIE-4 (+9.39%). On closer examination, we find that some approaches mainly improve in precision, whereas others primarily profit from a boost in recall. The former include for example Stanford Open IE (+25.00%), MinIE (+23.67%) and ClausIE (+16.98%), while the latter comprises systems such as Stanford Open IE (+27.20%), ReVerb (+19.06%) and PropS (16.89%) (for details, see Section 4.2.2).

Table 9
Average precision, recall, F_1 and AUC scores of a variety of state-of-the-art Open IE approaches on the OIE2016 benchmark when using the reference TS implementation DisSim as a pre-processing step.

System	Precision	Recall	F_1	AUC
ReVerb	**0.776**	0.531	0.631	0.412
Ollie	0.667	0.533	0.593	0.353
ClausIE	0.503	0.660	0.571	0.332
Stanford Open IE	0.175	0.477	0.256	0.275
PropS	0.522	0.609	0.562	0.326
OpenIE-4	0.727	0.717	**0.722**	**0.519**
MinIE	0.533	0.674	0.595	0.359
OpenIE-5	0.610	0.717	0.659	0.435
RnnOIE	0.596	**0.806**	0.685	0.480

Table 10
Improvements in precision, recall, F_1 and AUC on the OIE2016 benchmark when using the reference TS implementation DisSim as a pre-processing step.

System	Precision	Recall	F_1	AUC
ReVerb	−7.18%	+19.06%	+8.42%	+**39.19**%
Ollie	+1.06%	−1.48%	−0.34%	−1.12%
ClausIE	+16.98%	−3.51%	+8.14%	+12.93%
Stanford Open IE	+**25.00%**	+**27.20%**	+**25.49%**	+35.47%
PropS	−6.12%	+16.89%	+4.46%	+12.41%
OpenIE-4	+9.98%	+8.64%	+9.39%	+19.59%
MinIE	+23.67%	−1.75%	+12.48%	+21.28%
OpenIE-5	+4.99%	+4.22%	+4.60%	+9.02%
RnnOIE	−14.98%	+0.88%	−8.30%	−14.13%

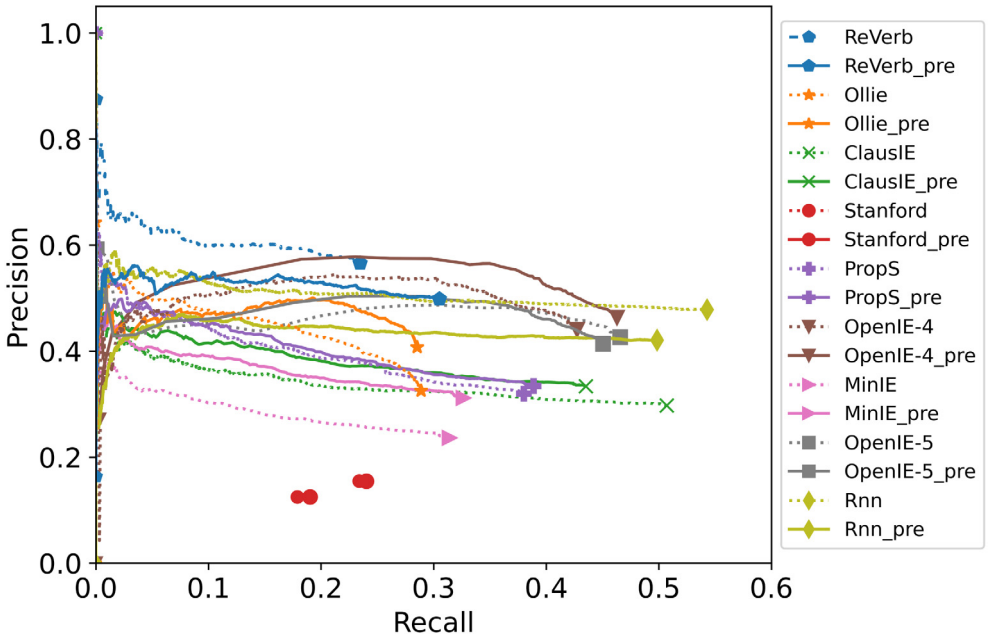

Fig. 15. Comparative analysis of the performance of state-of-the-art Open IE systems on the customized CaRB benchmark *with* (solid lines) and *without* (dotted lines) sentence splitting as a pre-processing step.

CaRB. To support our findings from the previous paragraph, we replicated our analysis on the customized version of the CaRB benchmark. Using its evaluation methodology in combination with the ground truth relations from the OIE2016 benchmark, we investigated the impact of applying the discourse-aware TS approach as a pre-processing step to split complex input sentences into a set of syntactically simplified propositions. Fig. 15 illustrates the outcome.

As before, it depicts the PR curves of the Open IE baseline systems that have been executed on the CaRB benchmark both as a stand-alone system (dotted lines), and within our TS-Open IE pipeline, where they act as the relation extraction component after the transformation stage (solid lines). The systems' overall precision, recall, F_1 and AUC scores when operating on top of the simplified sentences are listed in Table 11. The corresponding improvements achieved in comparison to their stand-alone counterparts are displayed in Table 12.[14]

Though the order of the tested Open IE systems slightly changes with respect to each evaluation metric, the overall trend is similar to what could already be observed on the OIE2016 benchmark in the previous section. As shown in Table 12, all approaches under consideration gain both in F_1 and AUC, except for ClausIE (+0.80%, −2.33%), OpenIE-5 (−3.34%, −1.38%) and RnnOIE (−10.24%, −21.17%). What stands out is that, as before, Stanford Open IE (+24.50% and +33.33%) and MinIE (17.71% and 33.33%) are among the systems that profit most from simplifying the structure of the input sentences.

Moreover, in contrast to the results obtained on the OIE2016 benchmark, Ollie now benefits from applying the TS approach as a pre-processing step with regard to all four metrics (+19.01%, +1.81%, +8.60%, +3.08%). This is likely due to the lexical matching function used in OIE2016 not penalizing extractions for misidentifying parts of an argument phrase in the relation slot,

Table 11
Precision, recall, F_1 (at maximum F_1 point) and AUC of a variety of state-of-the-art Open IE approaches on the CaRB benchmark when applying the reference TS implementation DisSim as a pre-processing step and using the gold tuples from the OIE2016 dataset in combination with the improved evaluation metrics from the CaRB framework.

System	Precision	Recall	F_1	AUC
ReVerb	**0.500**	0.305	0.379	0.160
Ollie	0.432	0.281	0.341	0.134
ClausIE	0.336	0.433	0.379	0.168
Stanford Open IE	0.154	0.240	0.188	0.136
PropS	0.342	0.380	0.360	0.158
OpenIE-4	0.478	0.454	**0.466**	**0.250**
MinIE	0.323	0.316	0.319	0.120
OpenIE-5	0.426	0.443	0.434	0.215
RnnOIE	0.421	**0.498**	0.456	0.216

Table 12
Improvements in precision, recall, F_1 and AUC scores on the CaRB benchmark when applying the reference TS implementation DisSim as a pre-processing step and using the gold tuples from the OIE2016 dataset in combination with the improved evaluation metrics from the CaRB framework.

System	Precision	Recall	F_1	AUC
ReVerb	−11.82%	**+29.79%**	+14.16%	+11.11%
Ollie	+19.01%	+1.81%	+8.60%	+3.08%
ClausIE	+11.63%	−13.75%	+0.80%	−2.33%
Stanford Open IE	+23.20%	+26.32%	**+24.50%**	**+33.33%**
PropS	+5.56%	+0.53%	+3.15%	+3.27%
OpenIE-4	+3.02%	+9.13%	+6.15%	+16.82%
MinIE	**+31.84%**	+4.29%	+17.71%	**+33.33%**
OpenIE-5	−4.27%	−2.42%	−3.34%	−1.38%
RnnOIE	−11.92%	−8.29%	−10.24%	−21.17%

[14] As shown in Tables 10 and 12, when applying the reference TS implementation DisSim as a preprocessing step, OpenIE-5 shows inconsistent performance improvement on the OIE2016 and CaRB benchmarks. This can be attributed to the fact that the two benchmarks use different scoring functions to determine recall and precision. As detailed in Section 4.1.3, CaRB's precision computation is based on a single match approach between predictions and references. Consequently, it penalizes systems that produce multiple very similar and redundant extractions, as is often the case with OpenIE-5 when operating on the simplified sentences where it tends to extract numerous tuples (i) that strongly resemble each other, and (ii) whose relational and argument slots are typically shorter compared to when taking the original complex source sentence as input. Regarding recall, CaRB's scoring function inherently penalizes shorter extractions since the probability of containing words from the gold tuples is lower for tuples with shorter relational and argument phrases. While this is also true for the OIE2016 benchmark, the effect is amplified for the CaRB benchmark due to its multi-match approach that is applied for the computation of the recall score.

therefore already assigning high precision scores to the stand-alone version of OLLIE. CaRB's matching function, however, utilizes a tuple match that punishes such a misplacement. Since OLLIE is particularly prone to extracting overgenerated predicates, it greatly benefits from the split sentences, as they prevent it from producing overly long relational phrases, which leads to a sharp rise in precision by 19.01%. The same applies to MinIE, explaining its high boost in precision (+31.84%). Conversely, OpenIE-5 shows a slight decline in all four metrics (−4.27%, −2.42%, −3.34%, −1.38%) when running on the syntactically simplified sentences instead of the raw complex input data.

Furthermore, ClausIE shows a comparatively larger drop in recall (−13.74%), resulting in a marginally lower AUC score (−2.33%) as compared to when operating on the raw source sentences. This result can be explained by CaRB's scoring function. Although the CaRB framework aims for atomic relational tuples that represent a minimal semantic unit each, it uses a multi-match approach for measuring a system's recall in order to avoid penalizing it too severely if it puts information from multiple gold tuples into a single extraction. Therefore, CaRB's matching function is likely to yield a relatively high recall when operating on the original data, upon which ClausIE tends to produce overly long argument phrases through combining and rearranging phrasal elements of the input. When instead applying the TS approach as a pre-processing step, there is a risk that some of the information contained in the source sentences is lost throughout the transformation process, resulting in a decreased recall. Though, the split sentences greatly help in generating indivisible argument phrases.

Summary. The automatic comparative evaluation conducted on a set of state-of-the-art Open IE baseline systems demonstrated the effectiveness of using the discourse-aware TS approach proposed in [6] as a pre-processing step. We showed that 78% (OIE2016 benchmark) and 67% (CaRB benchmark) of the tested approaches benefit (in terms of an improved F_1 and AUC score) from splitting complex sentences into a set of minimal propositions with a more regular syntax, upon which the relation extraction step is then performed.

4.2.2. Qualitative analysis of the extracted relations

Figs. 17 to 24 illustrate the outputs produced by the Open IE baselines on a representative example sentence, when using as input the set of syntactically simplified sentences from Fig. 16, which were generated by the discourse-aware TS approach. For all Open IE systems, we see that a substantial proportion of the extracted relational tuples are enriched with contextual information. In all cases, this supplementary information is correctly assigned to the extraction to which it refers. In the following, we will discuss the effects of applying the TS approach as a pre-processing step in more detail, together with its implications on the systems' overall precision and recall scores.[15] [16]

REVERB. When acting as a stand-alone system, REVERB achieves a particularly high precision of the extracted relational tuples, though at the cost of a comparatively low recall rate. The reason for this is that REVERB tends to make very conservative choices in the extraction process. However, when using the TS approach as a pre-processing step, REVERB succeeds in extracting a larger number of relations from the input sentences, resulting in an improved recall score. Fig. 17 supports these findings. It shows that REVERB now detects all the relations contained in the input, whereas before, when operating on the original sentence, it was only able to identify the two main relations of the source, without any additional contextual information (see Fig. B.30). Still, overall precision slightly decreases, most likely due to errors that were introduced during the transformation stage, leading to incorrectly split sentences.

OLLIE. Regarding recall and precision, OLLIE's performance does not change significantly whether or not applying the proposed TS approach as a pre-processing step. Using the reference implementation DISSIM, it slightly improves in precision, while losing somewhat in recall. However, as the output in Fig. 18 illustrates, the simplified sentence structure helps in solving OLLIE's main problem: overgenerated predicates. When operating on the split sentences, the number of quasi-redundant extractions which stem from OLLIE's focus on performing a great many of phrasal permutations for generating the relational phrases of its extractions, is dramatically reduced. Instead of rearranging phrasal components in the relational phrases in various ways (see Fig. B.29), they are now appended to the extracted tuples in the form of additional contextual information.

ClausIE. ClausIE primarily benefits from a push in precision, while its recall slightly drops. As we see in Fig. 19, this is no surprise. The syntactically simplified sentences prevent ClausIE from generating argument phrases by simply combining phrasal elements from the input in various different orders. In that way, its main problem – overly long argument phrases that subsume other self-contained facts (see Fig. B.27) – is resolved. Rather, ClausIE is likely to output arguments that represent an indivisible component in terms of a minimal semantic unit when acting on the split sentences provided by the TS approach.

Stanford open IE. Of all tested systems, Stanford Open IE shows the highest increase in both precision and recall. While it still tends to overproduce argument phrases by adding, deleting and restructuring optional phrasal complements – thereby generating a large number of redundant extractions –, it profits from the simplified sentences by enriching its extractions with contextual information (see e.g., (2) and (2a) in Fig. 20), leading to a higher recall of its output. As Fig. 20 shows, Stanford Open IE is now even able to detect relations that have not been identified before (see Fig. B.28) when operating on the raw input sentences ((1), (1a) and (1b) in Fig. 20). At the same time, the inaccurate extraction ⟨*she*; *was confirmed*; *becoming*⟩ is eliminated from the output.

PropS. PropS profits from a clear boost in recall when operating upon the structurally simplified sentences. This can be explained by PropS's tendency to incorporate relations in the argument slot of a superordinate extraction (e.g., ⟨*she*; *becoming*; *the first Supreme Court Justice of Hispanic descent*⟩ in (2) in Fig. B.33) when acting on the raw input sentences. In contrast, the split sentences encourage the extraction of a separate relational tuple for such cases (see (3) in Fig. 21). Moreover, while PropS is likely to produce a large number of n-ary relations when given the original complex sentences as input, it generates an output where additional arguments are preferably expressed in contextual arguments that are attached to the respective parent relation (e.g., (1a) and (1b) in Fig. 21) when applying the TS approach as a pre-processing step.

OpenIE-4 and OpenIE-5. OpenIE-4 and OpenIE-5 both benefit from the syntactically simplified sentences in terms of recall and precision. Although they already achieve a relatively high recall rate when operating on the original complex input sentences, they still tend to miss some of the contextual information contained in them (see Fig. B.31), which they are able to detect when given the simplified sentences as input (e.g., (1a) and (1b) in Fig. 22), leading to an increased recall score.

[15] We refer here to the precision and recall scores achieved on the OIE2016 benchmark.

[16] To allow for better comparison, we provide the outputs generated by the baseline Open IE approaches when operating directly on the complex source sentences in Appendix B.

```
DisSim:
(1)  #1    0    He nominated Sonia Sotomayor.
(1a)               S:PURPOSE     This was to replace David Souter.
(1b)               S:TEMPORAL    This was on May 26, 2009.
(2)  #2    0    She was confirmed.
(2a)               S:TEMPORAL    This was on August 6, 2009.
(3)  #3    0    She was becoming the first Supreme Court Justice of Hispanic descent.
```

Fig. 16. Simplified output generated by the reference TS implementation DisSim on the following example sentence: "*He nominated Sonia Sotomayor on May 26, 2009 to replace David Souter; she was confirmed on August 6, 2009, becoming the first Supreme Court Justice of Hispanic descent.*".

```
ReVerb (with pre-processing):
(1)  #1    0    He;     nominated;       Sonia Sotomayor
(1a)               S:PURPOSE     to replace David Souter
(1b)               S:TEMPORAL    on May 26 , 2009
(2)  #2    0    she;    was confirmed;
(2a)               S:TEMPORAL    on August 6 , 2009
(3)  #3    0    she;    was becoming;    the first Supreme Court Justice of Hispanic
                                           descent
```

Fig. 17. Relations extracted by ReVerb when using the reference TS implementation DisSim as a pre-processing step.

```
Ollie (with pre-processing):
(1) #1    0    he;     nominated;       Sonia Sotomayor
(1a)              S:PURPOSE     to replace David Souter
(1b)              S:TEMPORAL    on May 26 , 2009
(2) #2    0    she;    was becoming;    the first Supreme Court Justice of Hispanic
                                        descent
```

Fig. 18. Relations extracted by Ollie when using the reference TS implementation DisSim as a pre-processing step.

```
ClausIE (with pre-processing):
(1) #1    0    he;    nominated;    Sonia Sotomayor
(1a)            S:PURPOSE     to replace David Souter.
(1b)            S:TEMPORAL    on May 26, 2009.
(2) #2    0    she;    was confirmed;
(2a)            S:TEMPORAL    on August 6, 2009.
(3) #3    0    she;    was becoming;    the first Supreme Court Justice of Hispanic
                                        descent
```

Fig. 19. Relations extracted by ClausIE when using the reference TS implementation DisSim as a pre-processing step.

```
Stanford Open IE (with pre-processing):
(1)  #1    0    he;     nominated;       Sonia Sotomayor
(1a)               S:PURPOSE     to replace David Souter
(1b)               S:TEMPORAL    on May 26 , 2009
(2)  #2    0    she;    was;             confirmed
(2a)               S:TEMPORAL    on August 6 , 2009
(3) #3    0     she;    was becoming;    first Supreme Court Justice
(4) #4    0     she;    was becoming;    first Supreme Court Justice of Hispanic descent
(5) #5    0     she;    was becoming;    Supreme Court Justice
(6) #6    0     she;    was becoming;    Supreme Court Justice of Hispanic descent
```

Fig. 20. Relations extracted by Stanford Open IE when using the reference TS implementation DisSim as a pre-processing step.

```
PropS (with pre-processing):
(1)  #1    0    He;     nominated;       Sonia Sotomayor
(1a)               S:PURPOSE     replace David Souter
(1b)               S:TEMPORAL    on May 26 , 2009
(2)  #2    0    she;    confirmed;
(2a)               S:TEMPORAL    on August 6 , 2009
(3)  #3    0    she;    becoming;    the first Supreme Court Justice of Hispanic descent
```

Fig. 21. Relations extracted by PropS when using the reference TS implementation DisSim as a pre-processing step.

```
OpenIE-4 (with pre-processing):
(1) #1    0    he;    nominated;    Sonia Sotomayor
(1a)            S:PURPOSE     to replace David Souter.
(1b)            S:TEMPORAL    on May 26, 2009.
(2) #2    0    she;    was;    confirmed
(2a)            S:TEMPORAL    on August 6, 2009.
(3) #3    0    she;    was becoming;    the first Supreme Court Justice of Hispanic
                                        descent
```

Fig. 22. Relations extracted by OpenIE-4 when using the reference TS implementation DisSim as a pre-processing step.

```
MinIE (with pre-processing):
(1)  #1    0    He;     nominated;       Sonia Sotomayor
(1a)               S:PURPOSE     to replace David Souter
(1b)               S:TEMPORAL    on May 26 , 2009
(2)  #2    0    she;    was confirmed;
(2a)               S:TEMPORAL    on August 6 , 2009
(3)  #3    0    she;    was becoming the first Supreme Court Justice of;    Hispanic
                                                                            descent
```

Fig. 23. Relations extracted by MinIE when using the reference TS implementation DisSim as a pre-processing step.

```
RnnOIE (with pre-processing):
(1)  #1    0    He;     nominated;       Sonia Sotomayor
(1a)               S:PURPOSE     to replace David Souter
(1b)               S:TEMPORAL    on May 26 , 2009
(2)  #2    0    she;    confirmed;
(2a)               S:TEMPORAL    on August 6 , 2009
(3)  #3    0    she;    becoming;    the first Supreme Court Justice of Hispanic descent
```

Fig. 24. Relations extracted by RnnOIE when using the reference TS implementation DisSim as a pre-processing step.

MinIE. When using the TS approach as a pre-processing step, MinIE achieves a notable growth in precision, while losing somewhat in recall. This is due to the fact that, similar to Ollie, when operating on the original data, it is prone to producing overgenerated predicates (see Fig. B.32), a problem that vanishes when taking the simplified sentences as input. Rather than incorporating and permuting phrasal complements in the relational phrases, MinIE then adds them to the extracted tuples as supplementary contextual information, resulting in a set of extractions that is much smaller, but at the same time much more precise (see Fig. 23).

RnnOIE. RnnOIE is the only Open IE system under consideration that is clearly adversely affected by applying the TS approach as a pre-processing step. Its precision considerably drops by almost 15%, whereas there is a slight increase by less than 1% in recall. This might be due to the fact that RnnOIE was explicitly trained on the OIE2016 dataset and is therefore tailored to its specific features. Accordingly, the splitting errors introduced by the reference TS implementation DisSim have a particularly strong negative impact on its performance, leading to a sharp decrease in the precision of its extracted tuples. Fig. 24 illustrates the output produced by RnnOIE when operating on the simplified sentences.

In general, we note that in many cases, the specific characteristics of the Open IE systems with regard to the type and boundaries of both the relational and argument phrases dissipate when operating on the simplified sentences. For instance, the overgenerated predicates that can be frequently observed in the output of Ollie and MinIE disappear, as well as ClausIE's overly long argument phrases. Only Stanford Open IE still shows a strong tendency towards overproducing argument phrases. Hence, we conclude that by applying the discourse-aware TS approach as a pre-processing step, our TS-Open IE pipeline is capable of correcting erroneous extractions, as indicated by the improved overall precision. Moreover, the higher recall scores indicate that the split sentences substantially facilitate the extraction of the relations contained in the input.

4.2.3. Analysis of the lightweight semantic representation of relational tuples

In Section 3.2.2, we presented a novel formal semantic representation for Open IE that allows for a *canonical context-preserving representation of relational tuples*. In the following, we will briefly examine its main properties in the light of the experiments outlined above.

Simplistic canonical predicate-argument structure. The minimal propositions generated by the TS approach introduced in [6] not only reduce the complexity of the relation extraction step in the Open IE task, but also lead to a canonical predicate-argument structure of the extracted relational tuples.

Reduced complexity of the relation extraction step The formal semantic representation for relational tuples that we propose is primarily aimed at complex input sentences. Using the TS approach proposed in [6], they are first split into a set of minimal semantic units, where each proposition represents a separate fact. In that way, complex polypredicative constructions are transformed into monopredicative units and trimmed down to their essential constituents, i.e. elements that are also part of their clause types. Thus, long nested structures are broken up and transformed into a normalized subject–verb–object structure (or simple variants thereof, see Table 4). Consequently, the 1-to-n problem when extracting predicate-argument tuples from complex sentences is scaled down to a *1-to-1 predicate mapping problem*, where exactly one relational tuple is generated from each simplified sentence. Thus, the complexity

of the relation extraction step is reduced. Beyond that, shortening each simplified sentence to its core components further minimizes the complexity of this task, ensuring that overly specific argument phrases that subsume other self-contained facts are prevented from being generated.

According to the human analyses presented in [6], the discourse-aware TS approach succeeds in splitting complex source sentences into a fine-grained output of simplified sentences that are to a large extent syntactically well-formed and preserve the meaning of the input. In particular, it achieves an average score of 1.45 for the "simplicity" dimension on three TS corpora. This result is close to the maximum score of 2, which indicates that a given output sentence presents an atomic semantic unit. Accordingly, we conclude that the TS approach largely succeeds in decomposing complex input sentences into *minimal propositions with a simplistic canonical structure* that can be leveraged by downstream Open IE approaches.

As demonstrated by the empirical findings described in Section 4.2.1, the fine-grained representation of sentences in the form of atomic semantic units is easier to analyze for state-of-the-art Open IE approaches, resulting in a higher precision and recall of the extracted relational tuples with regard to six out of the nine tested systems (see Tables 10 and 12). Consequently, when operating on the simplified sentences, Open IE frameworks are more likely to return correct relational tuples and less prone to miss information from the source sentence in the extracted tuples.

Canonical predicate-argument structure The minimal propositions that present a simplistic canonical structure inherently lead to a normalized representation of the relational tuples extracted by an Open IE system. Each simplified sentence results in a *canonical binary predicate-argument structure* that is enriched with contextual information in order to avoid producing an isolated set of relational tuples that are hard to interpret (for details, see below).

When operating on the original complex input sentences, the characteristics of the output extracted by state-of-the-art Open IE systems vary widely. As outlined in the manual analysis in Appendix B, there is no standardized output scheme among the different approaches. While some of the frameworks are limited to the extraction of binary relations, others output n-ary relations (typically without establishing a semantic connection between associated tuples). Moreover, many Open IE systems tend to produce a large number of quasi-redundant extractions. This is due to the fact that they are prone to overgenerate predicate and argument phrases by adding, deleting and restructuring phrasal complements. The resulting output often not only is redundant but also presents overly specific predicates, as well as overly long argument units, generally hurting the performance of downstream applications [27].

As the analysis in Section 4.2.2 reveals, the semantic representation that we propose normalizes the extracted relational tuples by *reducing the predicate and argument slots to their essential components*, while *additional information can be found in associated context tuples*. In that way, overgenerated predicate and argument slots, as well as (quasi-)redundant extractions are prevented for the most part, resulting in a simplistic unified canonical representation of relational tuples.

Contextual information. Previous work in the area of Open IE has mainly focused on the extraction of isolated relational tuples, resulting in a loose arrangement of tuples that lack contextual information. To preserve the coherence of the extracted predicate-argument structures, the semantic representation that we propose extends the shallow semantic representation of state-of-the-art Open IE approaches by incorporating their semantic context. For this purpose, it captures intra-sentential rhetorical structures and hierarchical relationships between the relational tuples.

Contextual hierarchy To avoid producing a set of disconnected extractions, the proposed semantic representation specifies a hierarchical order between the extracted relations. For this purpose, it distinguishes *different levels of context*. Each tuple is assigned to a contextual layer — the lower the allotted layer, the more relevant is the information contained in it. In that way, related tuples are linked to one another, thus establishing a contextual hierarchy between them. According to the results presented in [37], the discourse-aware TS approach succeeds in assigning the correct hierarchical relationship between a pair of propositions in almost 90% of the cases.

The lower part of Fig. 25 illustrates an example. Tuples (6) and (8) are assigned to contextual layer 0, i.e. the lowest level of context. Accordingly, they depict the key information of the source sentence. Tuples (7), (9), (10) and (11), in contrast, are each allocated to context level 1, which means that they provide additional information about their respective parent tuple. This type of information can easily be exploited for example in downstream text summarization applications. In order to reduce the content of a given document to its core information, such approaches need only focus on the relations of the lowest contextual layers, containing the main information of the input.

Semantically typed relational tuples In addition to setting up a contextual hierarchy between the extracted relational tuples, the proposed semantic representation for Open IE supports the extraction of further meta information that determines how the tuples are semantically interconnected. For this purpose, it applies a selected set of rhetorical relations that link associated tuples. In that way, it allows for the representation of semantically typed relational tuples. Thus, the *semantic context* of the extracted tuples is preserved. The comparative analysis with the annotations contained in the RST-DT in [37] shows that the discourse-aware TS framework reaches an average precision of 70% for the classification of the rhetorical relations that hold between a pair of propositions.

As depicted in Fig. 25, OpenIE-4's default representation returns a disconnected set of predicate-argument structures. Our novel representation, in contrast, enriches the output with contextual information in terms of rhetorical relations. In that way, the extracted relations are put into a logical structure in the form of semantically typed relational tuples that preserve the semantic context of the extractions, resulting in a *more informative and coherent output* which is easier to interpret.

5. Complex question answering as a use case of the lightweight semantic representation

Our novel lightweight semantic representation for complex text data puts a semantic layer on top of the simplified sentences

```
OpenIE-4's default representation:
(1) A fluoroscopic study;    known;    as an upper gastrointestinal series
(2) the usage of barium;     lead;     to increased post operative complications
(3) volvulus;                is suspected;
(4) non water soluble contrast;    is;      mandatory
(5) the usage of barium;     can impede;    surgical revision


OpenIE-4's output when using our novel semantic representation:
(6)  #1    0    A fluoroscopic study;       is typically;    the next step in management
(6a)               L:ELABORATION   #2
(6b)               L:CONTRAST      #3
(7)  #2    1    This fluoroscopic study;    is known;        as an upper gastrointestinal
                                                             series
(8)  #3    0    non water soluble contrast; is;              mandatory
(8a)               L:CONTRAST      #1
(8b)               L:CONDITION     #6
(8c)               L:BACKGROUND    #4
(8d)               L:BACKGROUND    #5
(9)  #4    1    The usage of barium;    can impede;    surgical revision
(9a)               L:LIST          #5
(10) #5    1    The usage of barium;    can lead;      to increased post operative
                                                       complications
(10a)              L:LIST          #4
(11) #6    1    Volvulus;               is suspected;
```

Fig. 25. Comparison of the output generated by OpenIE-4 when using its *default representation* and when using our *proposed formal semantic representation.*

```
(1) #1   0   Donald Trump;                withdrew;           his sponsorship
                              S:TEMPORAL   in 1990
                              S:TEMPORAL   after the second Tour de Trump
                              L:CAUSE      #3
(2) #2   0   Donald Trump;                was;                US president
(3) #3   1   his other business ventures; were experiencing;  financial woes
```

Fig. 26. Example of our proposed lightweight semantic representation for complex QA.

by setting up a semantic hierarchy between them. It can be leveraged to extract relational tuples within their semantic context, facilitating a variety of artificial intelligence tasks, such as Question Answering (QA), Text Summarization or Natural Language Inference (NLI).

For instance, consider a QA system that is built upon the semantically typed and interconnected relational tuples. Our novel lightweight semantic representation of relational tuples allows to not only answer simple questions that are restricted to a single tuple, but also more complex questions where the relationship between multiple tuples is relevant, i.e. their context must be known.

Consider the following example sentence:

> *Trump withdrew his sponsorship after the second Tour de Trump in 1990 because his business ventures were experiencing financial woes.*

From this input, three relational tuples along with their semantic hierarchy can be extracted, resulting in the representation depicted in Fig. 26.

Based on this data, QA tasks can be executed. For example, imagine the following user queries:

- Q1: "***Who** withdrew his sponsorship?*"
  This query is processed by searching for tuples that have a "*withdrew*" relation and an object argument arg_2 that maps to the noun phrase "*his sponsorship*" in the query. As a result, we expect the subject argument arg_1 to be returned as an answer. In this example, tuple (1) matches and returns "*Donald Trump*".
- Q2: ***What** were his other business ventures experiencing?*
  Similarly to Q1, this query searches for an object argument arg_2 that is in an "*experiencing*" relationship with the subject argument "*his other business ventures*". In this case, tuple (3) matches, returning "*financial woes*" as an answer.
- Q3: "*What are activities of Donald Trump **after the second Tour de Trump**?*"
  In this example query, the user is interested in activities of Donald Trump after the second Tour de Trump. Here, the existence of reified contextual information comes into play. In order to return valid extractions, the system has to search for a tuple that maps to the subject argument "*Donald Trump*" and has a contextual argument that matches the phrase "*after the second Tour de Trump*". In this scenario, tuple (1) is returned.
- Q4: "***When** did Donald Trump withdraw his sponsorship?*"
  Another application that further makes use of the contexts' semantic annotations is a scenario where the user explicitly searches for contextual information, such as temporal or spatial information. Here, the word "*When*" indicates a temporal contextual information that is assigned to a relational tuple that matches the subject argument "*Donald Trump*", the relational phrase "*withdrew*" and the object argument "*his sponsorship*". In this example, the system retrieves the contexts "*in 1990*" and "*after the second Tour de Trump*" from tuple (1) based on their TEMPORAL classifications.

- Q5: "***Why*** *did Donald Trump withdraw his sponsorship in 1990?*"
  In addition to the previous example, classified semantic relations that hold between two propositions can also be used for finding semantically linked relational tuples. Possible use cases include, for example, causalities, temporal dependencies, attributions or intentions for specific actions. Here, the system infers from the word "*Why*" that it needs to search for a tuple that is linked via a causal (CAUSE) relation. In this scenario, the system is expected to return tuple (3).

While Q1 and Q2 represent comparatively simple questions that can be answered by taking into account only a single tuple, Q3 through Q5 represent more complex questions. In order to be able to answer them, several tuples have to be considered, including their context. Putting the extracted relations into a logical structure in the form of semantically typed relational tuples that preserve the semantic context of the extractions, our novel lightweight semantic representation of Open IE tuples supports the task of complex QA.

6. Conclusion

In this work, we present a method for taking advantage of the semantic hierarchy of minimal propositions generated by the discourse-aware TS approach proposed in [6] for the task of Open IE. The approach that we introduce uses the TS framework as a pre-processing step for extracting semantically typed relational tuples from complex assertions. The fine-grained representation of complex sentences in the form of hierarchically ordered and semantically interconnected propositions not only facilitates the extraction of relational tuples but also enables the enrichment of the output with semantic information.

6.1. Contributions

While the task of Open IE has been evolving over the years to capture relations beyond simple subject–predicate–object triples, a closer look into the extraction of the linkage between clauses and phrases within a complex sentence has largely been neglected. This work addresses this gap. It is targeted at capturing complementary linguistic phenomena, including intra-sentential discourse relations, predicate-argument structures, as well as clausal and phrasal linkage patterns to enrich the output of Open IE systems with additional meta information, thus preserving the semantic context of the extracted relational tuples. In addition, by taking advantage of the simplified sentences instead of dealing directly with the raw source sentences, the complexity of the relation extraction task is reduced, facilitating the process of extracting a formal semantic representation from unstructured text.

Accordingly, in a comprehensive evaluation, we demonstrated that state-of-the-art Open IE systems may leverage the fine-grained representation of complex sentences in the form of hierarchically ordered and semantically interconnected propositions for

(i) **facilitating the extraction of relational tuples**, leading to an improved precision (up to 32%) and recall (up to 30%) of the extracted relational tuples on a large benchmark corpus, and

(ii) **enriching their output with semantic information**, resulting in more informative and coherent predicate-argument structures. In this context, we introduce **a novel lightweight semantic representation for complex text data in the form of normalized and context-preserving relational tuples** that extends the shallow semantic representation of state-of-the-art Open IE approaches in the form of predicate-argument structures by capturing *intra-sentential rhetorical structures and hierarchical relationships* between the relational tuples.

6.2. Summary of the results

In a thorough analysis, we investigated how the performance of state-of-the-art Open IE systems is affected when taking advantage of the semantic hierarchy of minimal propositions generated by the discourse-aware TS approach introduced in [6]. For this purpose, we integrated the TS framework into existing Open IE systems as a pre-processing step for extracting semantically typed relational tuples from complex input sentences. Assuming that the fine-grained representation of complex assertions in the form of hierarchically ordered and semantically interconnected sentences with a simplified syntax improves the coverage and accuracy of the extracted tuples and supports the creation of a novel canonical context-preserving representation of relational tuples, we have examined two hypotheses, as detailed below.

6.2.1. Hypothesis 1: Coverage and accuracy

The minimal, self-contained propositions generated by the TS approach proposed in [6] improve the performance of state-of-the-art Open IE systems in terms of precision and recall.

Experimental support In a first step, we measured if the performance of state-of-the-art Open IE approaches was improved in terms of recall and precision of the extracted relational tuples when operating on the split sentences instead of dealing directly with the raw source sentences. For this purpose, we performed a comparative analysis that examined the output of nine existing Open IE approaches on two recently proposed benchmarks. To evaluate the impact of sentence splitting on the coverage and accuracy of the relational tuples extracted by them, we compared their performance when directly operating on the raw input data with their performance when the reference TS implementation DisSim was used as a pre-processing step to split complex source sentences into a set of syntactically simplified sentences with a more regular structure. To complement the automatic evaluation described above, we conducted a manual qualitative analysis of the extracted relational tuples in order to determine whether the particular deficiencies of the tested Open IE systems could be eliminated when taking advantage of the structurally simplified sentences.

Interpretation The results achieved in the automatic comparative evaluation conducted over a set of state-of-the-art Open IE systems demonstrated the effectiveness of using the proposed discourse-aware TS approach as a pre-processing step to improve the coverage and accuracy of the extracted relational tuples. We showed that 78% (OIE2016 benchmark) and 67% (CaRB benchmark) of the tested approaches benefit in terms of AUC and F_1 from splitting complex sentences into a set of minimal propositions with a canonical syntax, upon which the relation extraction step is then performed. Regarding the OIE2016 benchmark, the highest improvement in AUC was achieved by ReVerb, yielding a +39.19% increase over the output produced when acting as a stand-alone system, while Stanford Open IE was the system with the largest gain in F_1 (+25.49%). Considering the CaRB benchmark, an increase of up to +33.33% in AUC and +24.50% in F_1, respectively,

could be noted. When analyzing by hand the effects of applying the TS approach as a pre-processing step, we noted that in many cases the specific characteristics of the Open IE systems with regard to the type and boundaries of both the relational and argument phrases dissipated when operating on the simplified sentences. For instance, the overgenerated predicates that could be frequently observed in the output of Ollie and MinIE disappeared, as well as ClausIE's overly long argument phrases.

Conclusion Hence, we conclude that by applying the discourse-aware TS framework as a pre-processing step, Open IE systems are capable of correcting erroneous extractions, as indicated by the improved overall precision. Moreover, the higher recall scores indicate that the split sentences substantially facilitate the extraction of the relations contained in the input.

6.2.2. Hypothesis 2: Canonical context-preserving representation of relational tuples

The semantic hierarchy of minimal propositions generated by the discourse-aware TS approach

(a) reduces the complexity of the relation extraction step, resulting in a simplistic canonical predicate-argument structure of the output, and
(b) allows for the enrichment of extracted relational tuples with contextual information that supports their interpretability.

Experimental support In a second step, we assessed the merits of the discourse-aware TS approach in supporting the extraction of semantically typed relational tuples from complex assertions in downstream Open IE applications. We examined whether the semantic hierarchy of simplified sentences can be leveraged to transform the shallow semantic representation of state-of-the-art Open IE approaches into a canonical context-preserving representation of relational tuples. For this purpose, we analyzed the core characteristics of the resulting representation in the light of the experiments outlined above. The aim was to determine to what extent the semantic hierarchy of minimal propositions supports both the extraction of simplistic canonical predicate-argument structures from complex source sentences, and the enrichment of the extracted relational tuples with additional meta information to produce an output that preserves contextual information.

Interpretation We detailed that the semantic hierarchy of minimal propositions benefits the extraction of canonical context-preserving predicate-argument structures for the task of Open IE. Representing normalized monopredicative units, the simplified sentences reduce the complexity of the relation extraction step and inherently support the extraction of canonical predicate-argument structures. Thus, a standardized output scheme is created, where each simplified sentence results in a *normalized binary predicate-argument structure*, in which *both the predicate and the argument slots are reduced to their essential components*. In that way, the generation of overly specific predicate and argument phrases, as well as (quasi-)redundant extractions is prevented.

Moreover, we argued that the semantic hierarchy can be leveraged to incorporate contextual information of the extracted relational tuples, thus extending the shallow semantic representation of state-of-the-art Open IE systems. We pointed out that the semantic hierarchy supports the specification of a hierarchical order between the extracted relational tuples, as it enables to distinguish between *different levels of context* - the lower the allotted layer, the more relevant is the information contained in it. In addition, we delineated that the semantic hierarchy generated by the discourse-aware TS approach can be used to enrich the output of Open IE approaches with contextual information in terms of rhetorical relations, allowing for the representation of *semantically typed relational tuples*. Thus, the extracted relations are put into a *logical structure that preserves the semantic context of the extractions*, resulting in an output that is more informative and coherent, and thus easier to interpret.

Conclusion Hence, we conclude that the semantic hierarchy of minimal propositions generated by the discourse-aware TS approach can be leveraged to transform the shallow semantic representation of existing Open IE systems into a canonical context-preserving representation of relational tuples. The proposed representation allows for a simplistic unified representation of predicate-argument structures that can easily be enriched with contextual information in terms of intra-sentential rhetorical structures and hierarchical relationships between the extracted tuples, resulting in a set of interrelated semantically typed tuples that preserve the coherence of the output.

6.2.3. Summary

The analyses outlined above provide sufficient corroboration evidence to support hypothesis 1 and hypothesis 2. Accordingly, we conclude that the semantic hierarchy of minimal propositions generated by the discourse-aware TS approach benefits downstream Open IE applications by

(i) improving their performance in terms of the precision and recall of the extracted predicate-argument structures, and
(ii) transforming the shallow semantic representation of state-of-the-art systems into a canonical context-preserving representation of relational tuples.

6.3. Future work

In the future, we aim to further *expand the lightweight semantic representation* for Open IE that we proposed in this work. As of now, it is restricted to capturing inter-proposition rhetorical structures only. However, this does not allow for a more detailed semantic analysis at the level of individual propositions to further characterize a single event that is expressed by a relational tuple. By assigning thematic roles in the style of Semantic Role Labeling (SRL) to the tuples' arguments, the informativeness of the extracted relations may be increased. In that way, we can indicate exactly what semantic relations hold among a predicate and its associated arguments. Accordingly, the participants in an event, as well as other event properties can be further semantically classified, helping to better understand the exact meaning conveyed in the extracted relational tuples. Thus, we improve the use of the semantic representation generated by our framework for downstream NLP applications.

Moreover, as of now, our framework operates on the level of individual sentences. Accordingly, it only considers intra-sentential relationships when creating the semantic hierarchy of minimal propositions. A promising future research direction will be to examine how the approach we propose can be extended to capture inter-sentence relationships, resulting in a linked proposition tree that covers not only a single source sentence, but

Table A.13
Set of rhetorical relations and their inversions.

(a) Coordinations.

Rhetorical relation/Inverse relation	Core span	Following core span
Unknown-Coordination/ Unknown-Coordination	A syntactically coordinate span (default)	A syntactically coordinate span (default)
Contrast/Contrast	One alternate	The other alternate
Cause/Result	A situation	Another situation which causes that one
Result/Cause	A situation	Another situation which is caused by that one
List/List	An item	A next item
Disjunction/Disjunction	An item	An alternate item
Temporal-After/Temporal-Before (Sequence)	A situation	A situation that occurs after that
Temporal-Before/Temporal-After (Inverted-Sequence)	A situation	A situation that occurs before that

(b) Subordinations.

Rhetorical relation	Core span	Context span
Unknown-Subordination	The syntactically superordinate span (default)	The syntactically subordinate span (default)
Attribution	The reported message	The source of the attribution
Background	Text whose understanding is being facilitated	Text for facilitating understanding
Cause	A situation	Another situation which causes that one
Result	A situation	Another situation which is caused by that one
Condition	Action or situation whose occurrence results from the occurrence of the conditioning situation	Conditioning situation
Elaboration	Basic information	Additional information
Purpose	An intended action	The intent behind the situation
Temporal-After	A situation	A situation that occurs after that
Temporal-Before	A situation	A situation that occurs before that

rather a paragraph or even a full document, similar to rhetorical structure trees generated by RST parsers. Including intersentential semantic relations would allow to consider the wider semantic context of a proposition, beyond a single sentence. In that way, a relational tuple could be embedded in the context of a whole text, making it possible to draw broader conclusions.

CRediT authorship contribution statement

Christina Niklaus: Conceptualization, Software, Formal analysis, Investigation, Writing – original draft. **Matthias Cetto:** Methodology, Software, Writing – original draft. **André Freitas:** Conceptualization, Supervision, Writing – review & editing. **Siegfried Handschuh:** Supervision, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Appendix A. Rhetorical relations and their inversions

Since in the discourse-aware TS approach proposed in [6], the rhetorical relations are used to semantically classify the contextual sentences, we need to ensure that the context span (corresponding to the satellite span in RST [38]) is the unit that is characterized by the relation name. However, this differs from some of the definitions found in the literature. For instance, the "Cause", "Result", "Temporal-After" and "Temporal-Before" relations in the RST-DT refer to the situation that is presented in the nucleus [39]. Consequently, in these cases, the relationship has to be inverted. Moreover, for coordinate relations (corresponding to multi-nuclear relations in RST), the relation name specifies the span on the right position, according to the theory of RST. Hence, in order to allow for a classification of both spans in the case of a coordination, we introduced inverse relations, where the core span to the right is classified with the original relation from the definition, whereas its left counterpart is flagged with the corresponding inverse relation (see Table A.13a).

Appendix B. Qualitative analysis of the extracted relations

Below, we compare and discuss the characteristics of the output produced by the various Open IE systems that we included in our analyses when using the complex source sentences as input. Figs. B.27 to B.34 provide representative examples of the relational tuples extracted by the different approaches, given the following input sentence:

> "*He nominated Sonia Sotomayor on May 26, 2009 to replace David Souter; she was confirmed on August 6, 2009, becoming the first Supreme Court Justice of Hispanic descent.*"

ClausIE. ClausIE shows a tendency to extract overly long argument phrases that do not represent an atomic unit, but instead subsume other self-contained facts (e.g., (1) and (2) in Fig. B.27). In many cases, these arguments are generated by combining and rearranging phrasal components from the input sentence in various different orders.

Stanford open IE. Stanford Open IE, too, has a tendency to generate arguments through phrasal permutation. However, instead of combining multiple separate phrasal components into a single argument phrase, it typically sticks with a given noun phrase and adds or deletes optional elements, such as adjectives or adverbs (e.g., (3), (5), (6) and (7) in Fig. B.28).

Ollie. As opposed to ClausIE and Stanford Open IE, which focus on restructuring the argument slot, Ollie puts the emphasis on conveying a range of alternative relational phrases by including, excluding and rearranging phrasal components, resulting in comparatively long relational phrases (e.g., (2), (3) and (4) in Fig. B.29).

```
ClausIE:
(1) he;     nominated;        Sonia Sotomayor on May 26 2009 to replace David Souter
(2) she;    was confirmed;    on August 6 2009 becoming the first Supreme Court Justice
                              of Hispanic descent
(3) she;    was confirmed;    becoming the first Supreme Court Justice of Hispanic descent
```

Fig. B.27. Relations extracted by ClausIE.

```
Stanford Open IE:
(1) she;    was confirmed on;    August 6 2009
(2) she;    was confirmed;       becoming
(3) she;    becoming;            Supreme Court Justice of Hispanic descent
(4) she;    was confirmed;
(5) she;    becoming;            first Supreme Court Justice
(6) she;    becoming;            Supreme Court Justice
(7) she;    becoming;            first Supreme Court Justice of Hispanic descent
```

Fig. B.28. Relations extracted by Stanford Open IE.

```
Ollie:
(1) she;                 was confirmed on;                    August 6, 2009
(2) He;                  nominated Sonia Sotomayor on;        May 26
(3) He;                  nominated Sonia Sotomayor;           2009
(4) He;                  nominated 2009 on;                   May 26
(5) Sonia Sotomayor;     be nominated 2009 on;                May 26
(6) He;                  nominated 2009;                      Sonia Sotomayor
(7) 2009;                be nominated Sonia Sotomayor on;     May 26
```

Fig. B.29. Relations extracted by Ollie.

```
ReVerb:
(1) He;      nominated;           Sonia Sotomayor
(2) she;     was confirmed on;    August 6, 2009
```

Fig. B.30. Relations extracted by ReVerb.

ReVerb. Ollie's predecessor system ReVerb lays the focus on preventing the extraction of incoherent and uninformative relations, thereby making very conservative choices that lead to a high precision of the extracted relational tuples. However, its output often lacks in comprehensiveness, resulting in tuples that are commonly short and accurate, but miss a lot of information contained in the source sentences (see Fig. B.30).

OpenIE-4 and OpenIE-5. Similar to ReVerb, OpeIE-4 and OpenIE-5 commonly succeed in extracting highly accurate tuples that are composed of relational and argument phrases which represent minimal semantic units (e.g., (1) and (3) in Fig. B.31). However, as opposed to ReVerb, they take a less conservative approach in the extraction process, allowing for a higher recall of the extracted relations.

MinIE. MinIE was built on top of ClausIE with the goal of minimizing both relational and argument phrases by identifying and removing parts that are overly specific. Indeed, as Fig. B.32 shows, it succeeds in shortening the argument phrases to their core information (e.g., (1), (2), (5) and (6) in Fig. B.32), extracting a large number of correct tuples with atomic arguments that cover most of the information contained in the input. However, similar to Ollie, MinIE tends to produce overgenerated predicates that may impede the processing of the output in downstream applications.

PropS. PropS is one of the few Open IE systems under consideration that aim to extract n-ary relations with the objective

```
OpenIE-4:
(1) he;     nominated;       Sonia Sotomayor
(2) she;    was confirmed;
(3) she;    was becoming;    the first Supreme Court Justice of Hispanic descent
```

Fig. B.31. Relations extracted by OpenIE-4.

```
MinIE:
(1) He;    nominated Sonia Sotomayor on;                  May 26
(2) He;    nominated Sonia Sotomayor to replace;          David Souter
(3) He;    nominated;                                     Sonia Sotomayor
(4) He;    be replace;                                    David Souter
(5) she;   was confirmed on August 6 2009 becoming first Supreme Court Justice of;
                                                          Hispanic descent
(6) she;   was confirmed becoming first Supreme Court Justice of;   Hispanic descent
```

Fig. B.32. Relations extracted by MinIE.

```
PropS:
(1) He;      nominated;     Sonia Sotomayor; on May 26, 2009; to replace David Souter
(2) she;     confirmed;     on August 6, 2009; becoming the first Supreme Court Justice
                            of Hispanic descent
(3) ;        first;         the Supreme Court Justice of Hispanic descent
```

Fig. B.33. Relations extracted by PropS.

```
RnnOIE:
(1) He;      nominated;     Sonia Sotomayor; on May 26, 2009; to replace David Souter
(2) ;        replace;       David Souter
(3) she;     confirmed;     on August 6, 2009; becoming the first Supreme Court Justice
                            of Hispanic descent
(4) she;     becoming;      the first Supreme Court Justice of Hispanic descent
```

Fig. B.34. Relations extracted by RnnOIE.

of obtaining as complete an extraction as possible. It achieves to split the argument phrases into indivisible units, with each of them conveying a separate fact. Thus, PropS extracts highly accurate and complete facts from the source sentences (see Fig. B.33).

RnnOIE. Similar to PropS, RnnOIE targets the extraction of n-ary relations in order to capture facts as complete as possible from the input. In doing so, it follows a verb-centric approach that aims to extract a relation for every verbal predicate in the source sentence (see Fig. B.34).

References

[1] L. Del Corro, R. Gemulla, Clausie: Clause-based open information extraction, in: Proceedings of the 22nd International Conference on World Wide Web, WWW '13, Association for Computing Machinery, New York, NY, USA, 2013, pp. 355–366, http://dx.doi.org/10.1145/2488388.2488420.

[2] Mausam, M. Schmitz, S. Soderland, R. Bart, O. Etzioni, 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, Jeju Island, Korea, 2012, pp. 523–534, URL https://www.aclweb.org/anthology/D12-1048.

[3] G. Angeli, M.J. Johnson Premkumar, C.D. Manning, Leveraging linguistic structure for open domain information extraction, in: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Association for Computational Linguistics, Beijing, China, 2015, pp. 344–354, http://dx.doi.org/10.3115/v1/P15-1034, URL https://www.aclweb.org/anthology/P15-1034.

[4] G. Stanovsky, J. Michael, L. Zettlemoyer, I. Dagan, Supervised open information extraction, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), Association for Computational Linguistics, New Orleans, Louisiana, 2018, pp. 885–895, http://dx.doi.org/10.18653/v1/N18-1081, URL https://www.aclweb.org/anthology/N18-1081.

[5] H. Bast, E. Haussmann, Open information extraction via contextual sentence decomposition, in: 2013 IEEE Seventh International Conference on Semantic Computing, IEEE, 2013, pp. 154–159.

[6] C. Niklaus, M. Cetto, A. Freitas, S. Handschuh, Transforming complex sentences into a semantic hierarchy, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Florence, Italy, 2019, pp. 3415–3427, http://dx.doi.org/10.18653/v1/P19-1333, URL https://www.aclweb.org/anthology/P19-1333.

[7] M. Cetto, C. Niklaus, A. Freitas, S. Handschuh, Graphene: Semantically-linked propositions in open information extraction, in: Proceedings of the 27th International Conference on Computational Linguistics, Association for Computational Linguistics, Santa Fe, New Mexico, USA, 2018, pp. 2300–2311, URL https://www.aclweb.org/anthology/C18-1195.

[8] D. Jurafsky, J.H. Martin, Speech and Language Processing, second ed., Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2009.

[9] E. Agichtein, L. Gravano, Snowball: Extracting relations from large plain-text collections, in: Proceedings of the Fifth ACM Conference on Digital Libraries, DL '00, Association for Computing Machinery, New York, NY, USA, 2000, pp. 85–94, http://dx.doi.org/10.1145/336597.336644.

[10] S. Brin, Extracting patterns and relations from the world wide web, in: Selected Papers from the International Workshop on the World Wide Web and Databases, WebDB '98, Springer-Verlag, Berlin, Heidelberg, 1998, pp. 172–183.

[11] E. Riloff, R. Jones, Learning dictionaries for information extraction by multi-level bootstrapping, in: Proceedings of the Sixteenth National Conference on Artificial Intelligence and the Eleventh Innovative Applications of Artificial Intelligence Conference Innovative Applications of Artificial Intelligence, in: AAAI '99/IAAI '99, American Association for Artificial Intelligence, USA, 1999, pp. 474–479.

[12] M. Banko, M.J. Cafarella, S. Soderland, M. Broadhead, O. Etzioni, Open information extraction from the web, in: Proceedings of the 20th International Joint Conference on Artifical Intelligence, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA., 2007, pp. 2670–2676.

[13] M.P. Marcus, B. Santorini, M.A. Marcinkiewicz, Building a large annotated corpus of english: The penn treebank, Comput. Linguist. 19 (2) (1993) 313–330, URL https://www.aclweb.org/anthology/J93-2004.

[14] F. Wu, D.S. Weld, Open information extraction using wikipedia, in: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Uppsala, Sweden, 2010, pp. 118–127, URL https://www.aclweb.org/anthology/P10-1013.

[15] A. Fader, S. Soderland, O. Etzioni, Identifying relations for open information extraction, in: Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Edinburgh, Scotland, UK., 2011, pp. 1535–1545, URL https://www.aclweb.org/anthology/D11-1142.

[16] A. Akbik, A. Löser, Kraken: N-ary facts in open information extraction, in: Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-Scale Knowledge Extraction (AKBC-WEKEX), Association for Computational Linguistics, Montréal, Canada, 2012, pp. 52–56, URL https://www.aclweb.org/anthology/W12-3010.

[17] F. Mesquita, J. Schmidek, D. Barbosa, Effectiveness and efficiency of open relation extraction, in: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Seattle, Washington, USA, 2013, pp. 447–457, URL https://www.aclweb.org/anthology/D13-1043.

[18] J. Christensen, Mausam, S. Soderland, O. Etzioni, 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, Los Angeles, California, 2010, pp. 52–60, URL https://www.aclweb.org/anthology/W10-0907.

[19] J. Schmidek, D. Barbosa, Improving open relation extraction via sentence re-structuring, in: Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14), European Language Resources Association (ELRA), Reykjavik, Iceland, 2014, pp. 3720–3723, URL http://www.lrec-conf.org/proceedings/lrec2014/pdf/1038_Paper.pdf.

[20] J. Van Benthem, A Brief History of Natural Logic, LondonCollege Publications, 2008.

[21] G. Stanovsky, J. Ficler, I. Dagan, Y. Goldberg, Getting more out of syntax with props, 2016, CoRR abs/1603.01648 URL http://arxiv.org/abs/1603.01648.

[22] M. Mausam, Open information extraction systems and downstream applications, in: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI '16, AAAI Press, 2016, pp. 4074–4077.

[23] H. Pal, Mausam, Demonyms and compound relational nouns in nominal open IE, in: Proceedings of the 5th Workshop on Automated Knowledge Base Construction, Association for Computational Linguistics, San Diego, CA, 2016, pp. 35–39, http://dx.doi.org/10.18653/v1/W16-1307, URL https://www.aclweb.org/anthology/W16-1307.

[24] S. Saha, H. Pal, Mausam, Bootstrapping for numerical open IE, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Association for Computational Linguistics, Vancouver, Canada, 2017, pp. 317–323, http://dx.doi.org/10.18653/v1/P17-2050, URL https://www.aclweb.org/anthology/P17-2050.

[25] S. Saha, Mausam, Open information extraction from conjunctive sentences, in: Proceedings of the 27th International Conference on Computational Linguistics, Association for Computational Linguistics, Santa Fe, New Mexico, USA, 2018, pp. 2288–2299, URL https://www.aclweb.org/anthology/C18-1194.

[26] N. Bhutani, H.V. Jagadish, D. Radev, Nested propositions in open information extraction, in: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Austin, Texas, 2016, pp. 55–64, http://dx.doi.org/10.18653/v1/D16-1006, URL https://www.aclweb.org/anthology/D16-1006.

[27] K. Gashteovski, R. Gemulla, L. del Corro, MinIE: Minimizing facts in open information extraction, in: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Copenhagen, Denmark, 2017, pp. 2630–2640, http://dx.doi.org/10.18653/v1/D17-1278, URL https://www.aclweb.org/anthology/D17-1278.

[28] L. Cui, F. Wei, M. Zhou, Neural open information extraction, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Association for Computational Linguistics, Melbourne, Australia, 2018, pp. 407–413, http://dx.doi.org/10.18653/v1/P18-2065, URL https://www.aclweb.org/anthology/P18-2065.

[29] L. Ramshaw, M. Marcus, Text chunking using transformation-based learning, in: Third Workshop on Very Large Corpora, 1995, URL https://www.aclweb.org/anthology/W95-0107.

[30] L. He, K. Lee, M. Lewis, L. Zettlemoyer, Deep semantic role labeling: What works and what's next, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Vancouver, Canada, 2017, pp. 473–483, http://dx.doi.org/10.18653/v1/P17-1044, URL https://www.aclweb.org/anthology/P17-1044.

[31] G. Stanovsky, I. Dagan, Creating a large benchmark for open information extraction, in: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Austin, Texas, 2016, pp. 2300–2305, http://dx.doi.org/10.18653/v1/D16-1252, URL https://www.aclweb.org/anthology/D16-1252.

[32] W. Lechelle, F. Gotti, P. Langlais, Wire57 : A fine-grained benchmark for open information extraction, in: Proceedings of the 13th Linguistic Annotation Workshop, Association for Computational Linguistics, Florence, Italy, 2019, pp. 6–15, http://dx.doi.org/10.18653/v1/W19-4002, URL https://www.aclweb.org/anthology/W19-4002.

[33] S. Bhardwaj, S. Aggarwal, M. Mausam, CaRB: A crowdsourced benchmark for open IE, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Association for Computational Linguistics, Hong Kong, China, 2019, pp. 6262–6267, http://dx.doi.org/10.18653/v1/D19-1651, URL https://www.aclweb.org/anthology/D19-1651.

[34] C. Niklaus, M. Cetto, A. Freitas, S. Handschuh, DisSim: A discourse-aware syntactic text simplification framework for english and german, in: Proceedings of the 12th International Conference on Natural Language Generation, Association for Computational Linguistics, Tokyo, Japan, 2019, pp. 504–507, http://dx.doi.org/10.18653/v1/W19-8662, URL https://www.aclweb.org/anthology/W19-8662.

[35] N. Bouayad-Agha, G. Casamayor, G. Ferraro, S. Mille, V. Vidal, L. Wanner, Improving the comprehension of legal documentation: The case of patent claims, in: Proceedings of the 12th International Conference on Artificial Intelligence and Law, ICAIL '09, Association for Computing Machinery, New York, NY, USA, 2009, pp. 78–87, http://dx.doi.org/10.1145/1568234.1568244.

[36] S. Štajner, M. Popović, Improving machine translation of english relative clauses with automatic text simplification, in: Proceedings of the 1st Workshop on Automatic Text Adaptation (ATA), Association for Computational Linguistics, Tilburg, the Netherlands, 2018, pp. 39–48, http://dx.doi.org/10.18653/v1/W18-7006, URL https://www.aclweb.org/anthology/W18-7006.

[37] C. Niklaus, A. Freitas, S. Handschuh, Shallow discourse parsing for open information extraction and text simplification, in: Proceedings of the 3rd Workshop on Computational Approaches to Discourse, International Conference on Computational Linguistics, Gyeongju, Republic of Korea and Online, 2022, pp. 64–76, URL https://aclanthology.org/2022.codi-1.9.

[38] W.C. Mann, S.A. Thompson, Rhetorical structure theory: Toward a functional theory of text organization, Text-Interdiscip. J. Study Discourse 8 (3) (1988) 243–281.

[39] L. Carlson, D. Marcu, Discourse Tagging Reference Manual, Vol. 54, ISI Technical Report ISI-TR-545, 2001, p. 56, URL https://www.isi.edu/~marcu/discourse/tagging-ref-manual.pdf.