Graphene: A Context-Preserving Open Information Extraction System

Matthias Cetto[1], Christina Niklaus[1], André Freitas[2], and **Siegfried Handschuh[1]**
[1] Faculty of Computer Science and Mathematics, University of Passau
{matthias.cetto, christina.niklaus, siegfried.handschuh}@uni-passau.de
[2] School of Computer Science, University of Manchester
andre.freitas@manchester.ac.uk

Abstract

We introduce Graphene, an Open IE system whose goal is to generate accurate, meaningful and complete propositions that may facilitate a variety of downstream semantic applications. For this purpose, we transform syntactically complex input sentences into clean, compact structures in the form of core facts and accompanying contexts, while identifying the rhetorical relations that hold between them in order to maintain their semantic relationship. In that way, we preserve the context of the relational tuples extracted from a source sentence, generating a novel lightweight semantic representation for Open IE that enhances the expressiveness of the extracted propositions.

1 Introduction

Information Extraction (IE) is the task of turning the unstructured information expressed in natural language (NL) text into a structured representation in the form of relational tuples consisting of a set of arguments and a phrase denoting a semantic relation between them: ⟨*arg1*; *rel*; *arg2*⟩. 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. Since its introduction by Banko et al. (2007), a large body of work on the task of Open IE has been described. By analyzing the output of state-of-the-art systems (e.g., (Mausam et al., 2012; Del Corro and Gemulla, 2013; Angeli et al., 2015)), 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. Third, state-of-the-art Open IE systems lack the expressiveness needed to properly represent complex assertions, resulting in incomplete, uninformative or incoherent propositions that have no meaningful interpretation or miss critical information asserted in the input sentence.

To overcome these limitations, we developed an Open IE framework called ”Graphene” that transforms syntactically complex NL sentences into clean, compact structures that present a canonical form which facilitates the extraction of accurate, meaningful and complete propositions. The contributions of our work are two-fold. First, to remove the complexity of determining intricate predicate-argument structures with variable arity from syntactically complex input sentences, we propose a two-layered transformation process consisting of a clausal and phrasal disembedding layer. It removes clauses and phrases that convey no central information from the input and converts them into independent context sentences, thereby reducing the source sentence to its main information. In that way, the input is transformed into a **novel hierarchical representation in the form of core facts and accompanying contexts**. Second, we **identify the rhetorical relations by which core sentences and their associated contexts are connected in order to preserve their semantic relationship**. These two innovations enable us to enrich extracted relational tuples of the form ⟨*arg1*; *rel*; *arg2*⟩ with contextual information that further specifies the tuple and to establish semantic links between them, resulting in a novel lightweight semantic representation for Open IE that provides highly informative extractions and thus supports their interpretability in downstream applications. The source code is available at `https://github.com/Lambda-3/Graphene`.

2 The System in a Nutshell

Graphene makes use of a two-layered transformation stage consisting of a clausal and phrasal disembedding layer, which is followed by a final relation extraction (RE) stage. It takes a text document as an input and returns a set of semantically typed and interconnected relational tuples. The workflow of our approach is displayed in Figure 1.

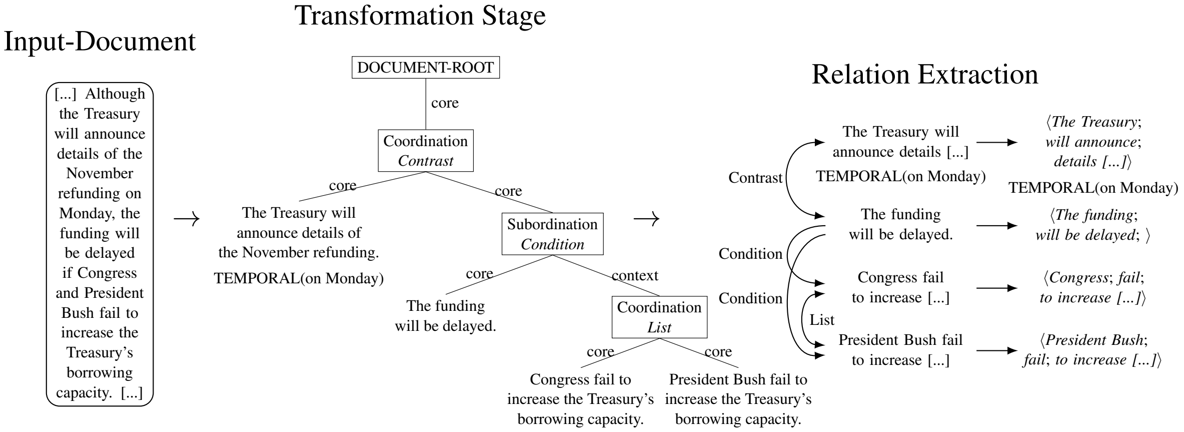
Input-Document
[...] Although the Treasury will announce details of the November refunding on Monday, the funding will be delayed if Congress and President Bush fail to increase the Treasury's borrowing capacity. [...]
Transformation Stage
DOCUMENT-ROOT
core
Coordination Contrast
core
core
The Treasury will announce details of the November refunding.
TEMPORAL(on Monday)
Subordination Condition
core
context
The funding will be delayed.
Coordination List
core
core
Congress fail to increase the Treasury's borrowing capacity.
President Bush fail to increase the Treasury's borrowing capacity.
Relation Extraction
Contrast
Condition
Condition
List
The Treasury will announce details [...]
TEMPORAL(on Monday)
The funding will be delayed.
Congress fail to increase [...]
President Bush fail to increase [...]
⟨The Treasury; will announce; details [...]⟩
TEMPORAL(on Monday)
⟨The funding; will be delayed; ⟩
⟨Congress; fail; to increase [...]⟩
⟨President Bush; fail; to increase [...]⟩

Figure 1: Extraction workflow for an example sentence.

2.1 Transformation Stage

During the transformation process, source sentences that present a complex linguistic structure are converted into a hierarchical representation of core facts and associated contexts that are connected by rhetorical relations capturing their semantic relationship similar to Rhetorical Structure Theory (RST) (Mann and Thompson, 1988). These compact, syntactically sound structures ease the problem of recognizing predicate-argument relations that are contained in the input without losing their semantic dependencies.

Clausal Disembedding. In the clausal disembedding layer, we split up complex multi-clause sentences that are composed of *coordinated* and *subordinated clauses*, *relative clauses*, or *attributions* into simpler, stand-alone sentences that contain one clause each. This is done in a recursive fashion so that we obtain a hierarchical structure of the transformation process comparable to the diagrams used in RST. As opposed to RST, however, the transformation process is carried out in a top-down fashion, starting with the input document and using a set of hand-crafted syntactic rule patterns that define how to split up, transform and recurse on complex syntactic patterns[1]. Each split will create two or more simplified sentences that are connected with information about (1) their constituency type depicting their semantic relevance (*coordinate* or *subordinate*) and (2) the rhetorical relation that holds between them. The constituency type infers the concept of nuclearity from RST, where coordinate sentences (which we call *core sentences*) represent nucleus spans that embody the central part of information, while subordinate sentences (*context sentences*) represent satellite spans that provide background information on the nucleus. The classification of the rhetorical relations is based on both syntactic and lexical features. While

[1] The complete rule set can be found online: `https://github.com/Lambda-3/Graphene/blob/master/wiki/supplementary/syntactic-simplification-patterns.pdf`

former are manifested in the phrasal composition of a sentence's phrasal parse tree, latter rely on a set of manually defined cue phrases. In this way, a hierarchical tree representation of the recursive transformation process for the whole document is constructed which we denote as *discourse tree*. Its leaf nodes represent the simplified sentences that were generated during the clausal disembedding layer.

Phrasal Disembedding. After recursively dividing multi-clause sentences into stand-alone sentences that contain one clause each, they are further simplified on a phrasal level. For this purpose, sentences are processed separately and transformed into simpler structures by extracting the following phrasal components from the input: *prepositional phrases, participial phrases, adjectival/adverbial phrases, appositive phrases, lead noun phrases, coordinations of verb phrases, enumerations of noun phrases* and *purposes*. This task is assisted by the sentence simplification system described in Niklaus et al. (2016).

2.2 Relation Extraction

After the transformation stage, RE is performed by using the simplified sentences as an input. The framework is designed to accept any type of RE implementation which is able to extract relational tuples from single sentences. The identified rhetorical relations from the transformation stage are then mapped to the corresponding relational tuples in the form of simple and linked contextual arguments (see Section 3). As a result, different approaches for RE can be complemented with contextual information that further specifies the extracted relational tuples. In that way, a new layer of semantics is added to the task of RE that can be used in other NLP tasks (see Section 6).

3 Output Format

In order to represent contextual relations between propositions, the default representation of a relational tuple of the form ⟨*arg1*; *rel*; *arg2*⟩ needs to be extended. Therefore, we present a novel lightweight semantic representation for Open IE that is both machine processable and human readable. It extends a binary subject-predicate-object tuple $t \leftarrow (rel, arg_{subj}, arg_{obj})$ 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*), yielding the final representation of (id, cl, t, C_S, C_L) tuples. The *context-layer* cl encodes the contextual hierarchy of core and contextual facts. Propositions with a context-layer of 0 carry the core information of a sentence, whereas propositions with a context-layer of $cl > 0$ provide contextual information about propositions 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 $c_S \in C_S, c_S \leftarrow (s, r)$ contains a textual expression s that is classified by the semantic relation r, a linked contextual argument $c_L \in C_L, c_L \leftarrow (id(z), r)$ refers to the content expressed in another proposition z.

To facilitate the inspection of the extracted propositions, a human-readable format, called *RDF-NL*, is generated by Graphene (see Figure 2). In this format, propositions are grouped by sentences in which they occur and are represented by tab-separated strings for the identifier id, context-layer cl and the core extraction that is represented by the binary relational tuple $t \leftarrow (rel, arg_{subj}, arg_{obj})$: subject argument arg_{subj}, relation name r and object argument arg_{obj}. Contextual arguments (C_S and C_L) are indicated by an extra indentation level to their parent tuples. The representation of a contextual argument 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*), if present. The content of a simple contextual argument is the textual expression, whereas the content of a linked contextual argument is the identifier of the target proposition.

Besides, the framework can materialize its relations into a graph serialized under the N-Triples[2] specification of the Resource Description Framework (RDF) standard. In that way, the consumption of the extracted relations by downstream applications is facilitated. A detailed description as well as some examples of the machine-readable RDF format are available online[3].

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

[3] https://github.com/Lambda-3/Graphene/blob/master/wiki/RDF-Format.md

```
Although the Treasury will announce details of the November refunding on Monday, the funding
will be delayed if Congress and President Bush fail to increase the Treasury's borrowing capacity.

#1    0    the Treasury    will announce    details of the November refunding
    S:TEMPORAL    on Monday
    L:CONTRAST     #2

#2    0    the funding    will be delayed
    L:CONTRAST     #1
    L:CONDITION    #3
    L:CONDITION    #4

#3    1    Congress    fail    to increase the Treasury 's borrowing capacity

#4    1    president Bush    fail    to increase the Treasury 's borrowing capacity
```

Figure 2: Proposed representation format (RDF-NL) - human readable representation.

4 Usage

Graphene can be either used as a Java API, imported as a Maven dependency, or as a service which we provide through a command line interface or a REST-like web service that can be deployed via docker. A demonstration video is available online[4].

5 Benchmarking

We evaluated the performance of our Open IE system Graphene using the benchmark framework proposed in Stanovsky and Dagan (2016), which is based on a QA-Semantic Role Labeling corpus with more than 10,000 extractions over 3,200 sentences from Wikipedia and the Wall Street Journal[5]. This benchmark allowed us to compare our framework with a set of state-of-the-art Open IE approaches in recall and precision (see Figure 3). With a score of 50.1% in average precision, Graphene achieves the best performance of all the systems in extracting accurate tuples. Considering recall, our framework (27.2%) is able to compete with the best-performing baseline approaches (32.5% and 33.0%). The interested reader can refer to Cetto et al. (2018) for more details.

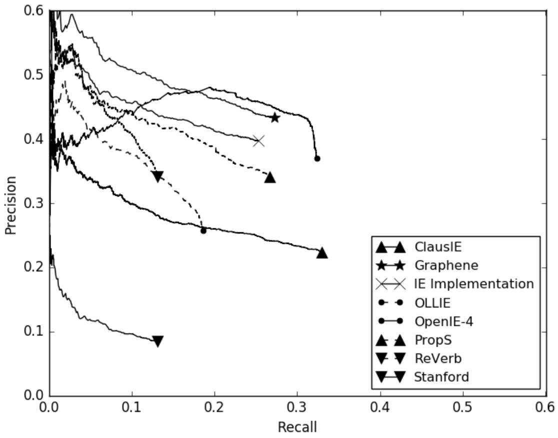

Figure 3: Performance of Graphene.

6 Application Scenarios of the Lightweight Semantic Open IE representation

The resulting lightweight semantic representation of the source text in the form of a two-layered hierarchy of semantically-linked relational tuples can be used to facilitate a variety of artificial intelligence tasks, such as building QA systems, creating text summarization applications or supporting semantic inferences.

For example, QA systems could build upon the semantically typed and interconnected relational tuples produced by our Open IE system Graphene to investigate the dependencies between extracted propositions (such as causalities, attributions and local or temporal contexts) and map specific question types to the corresponding semantic relationships when querying the underlying data. Based on the example given in Figure 2, one can imagine the following user query:

Under which circumstances will the funding be delayed?

Here, the system could infer from the interrogative expression *"Under which circumstances?"* to search for propositions that are linked to the extraction stating that ⟨*the funding*; *will be delayed*; ∅⟩

[4] https://asciinema.org/a/bvhgIP8ZEgDwtmRPFctHyxALu?speed=3

[5] available under https://github.com/gabrielStanovsky/oie-benchmark

by a conditional (CONDITION) relation. Accordingly, in this scenario the system is expected to return propositions #3 and #4 of Figure 2.

7 Conclusion

We presented Graphene, an Open IE system that transforms sentences which present a complex linguistic structure into a novel hierarchical representation in the form of core facts and accompanying contexts which are connected by rhetorical relations capturing their semantic relationship. In that way, the input text is turned into clean, compact structures that show a canonical form, thus facilitating the extraction of accurate, meaningful and complete propositions based on a novel lightweight semantic representation consisting of a set of semantically typed and interconnected relational tuples. In the future, we aim to port this idea to languages other than English.

References

Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D. Manning. 2015. 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*, pages 344–354. ACL.

Michele Banko, Michael J. Cafarella, Stephen Soderland, Matt Broadhead, and Oren Etzioni. 2007. Open information extraction from the web. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, pages 2670–2676. Morgan Kaufmann Publishers Inc.

Matthias Cetto, Christina Niklaus, André Freitas, and Siegfried Handschuh. 2018. Graphene: Semantically-linked propositions in open information extraction. In *Prooceedings of COLING 2018. To appear.*

Luciano Del Corro and Rainer Gemulla. 2013. Clausie: Clause-based open information extraction. In *Proceedings of the 22Nd International Conference on World Wide Web*, pages 355–366. ACM.

William C Mann and Sandra A Thompson. 1988. Rhetorical structure theory: Toward a functional theory of text organization. *Text-Interdisciplinary Journal for the Study of Discourse*, 8(3):243–281.

Mausam, Michael Schmitz, Stephen Soderland, Robert Bart, and Oren Etzioni. 2012. 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*, pages 523–534. ACL.

Christina Niklaus, Bernhard Bermeitinger, Siegfried Handschuh, and André Freitas. 2016. A sentence simplification system for improving relation extraction. In *Proceedings of COLING 2016: System Demonstrations*.

Gabriel Stanovsky and Ido Dagan. 2016. Creating a large benchmark for open information extraction. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. ACL.