SpanIE: A Span-Based Approach to OpenIE

Alfio Ferrara[1,†], Darya Shlyk[1,†]

[1]Università degli Studi di Milano, Department of Computer Science

Abstract

We introduce SpanIE, a span-based approach to solve the task of Open Information Extraction. SpanIE is a two-stage framework, providing a new way to leverage linguistic knowledge in proposition extraction from unstructured text. Our system recovers the underlying argument-predicate structure of a sentence from its dependency tree, and relies on the resulting sentence representation to solve the OpenIE task. The intermediate representation generated by SpanIE, is meant to benefit the information extraction process, allowing for accurate and expressive extractions. We conduct a preliminary comparative evaluation and present the experimental results to prove the effectiveness of the proposed approach. Using the benchmark framework for OIE, we show that our implementation achieves comparable performance with the state-of-the-art Open IE systems on the relation extraction task.

1. Introduction

Open Information Extraction [1] defines an unsupervised approach to the general task of Information Extraction (IE). Information Extraction is concerned with providing a structured representation to the information embedded in text [2]. More specifically, the goal is to detect phrases denoting semantic relationships between entities mentioned in a sentence, and extract these relations along with their arguments in the form of relational tuples: <arg1, relation, arg2>. The output extraction has the same structure of an RDF triple, called a subject-predicate-object expression. For example, given the sentence, *Albert Einstein was born in Ulm*, the following relation can be extracted: <Albert Einstein, born-in, Ulm>

While most traditional approaches to IE rely on hand-crafted patterns to extract a limited set of predefined relation types, OpenIE offers a different extraction paradigm. It allows for domain independent discovery of any relation occurring in text. The main advantage of this approach is the ability to work with textual corpora coming from a variety of domains, with no need to specify a target relationship set in advance. The extractions produced with OpenIE systems naturally populate structured information resources, represented by relational databases. They also provide convenient data to build knowledge graphs representing specific domains [3, 4]. The ability to capture facts asserted in text, makes OpenIE propositions a valuable source of information for many downstream NLP applications, including question answering, text comprehension and textual entailment tasks [5, 6].

SEBD 2023: 31st Symposium on Advanced Database System, July 02–05, 2023, Galzignano Terme, Padua, Italy

[†]These authors contributed equally.

alfio.ferrara@unimi.it (A. Ferrara); darya.shlyk@unimi.it (D. Shlyk)

0000-0002-4991-4984 (A. Ferrara); 0009-0006-4051-6871 (D. Shlyk)

CEUR Workshop Proceedings (CEUR-WS.org)

In this paper, we aim to extend the existing line of work on the task of OpenIE. We propose an alternative method to leverage syntactic dependencies to extract relational tuples from text. Our framework recovers predicate-argument structure from the sentence dependency tree and generates a compact sentence representation, that facilitates meaningful extractions. In this discussion paper, we present the experimental results from the preliminary evaluation, that provide enough evidence to conclude the effectiveness of the developed approach.

The paper is organized as follows. In Section 2, we first provide an overview of several approaches that were recently proposed to address the Open Information Extraction task. In Section 3, we outline the SpanIE framework, discussing the building blocks of its extraction pipeline. Specifically, we describe the algorithm for the construction of an intermediate sentence representation, and discuss how this representation is used to solve the actual relation extraction task. In Section 4, we conduct a comparative evaluation of SpanIE with other prominent systems, using a well-known benchmark framework for OIE. Finally, in Section 5, we give our concluding remarks.

2. Related Works

Most prominent approaches addressing the task of relation extraction, adopt a set of hand-crafted extraction rules defined over lexical and linguistic features. It has been shown that matching patterns based on deep syntactic features rather than shallow part-of-speech tags, achieve higher precision and generate more accurate extractions [7]. As a result, many rule-based approaches prevalently work with dependency information obtained from pretrained parsers, and apply extraction patterns directly on the sentence parse tree [8, 9, 10, 11, 12].

Another effective strategy to improve the accuracy of OpenIE extraction consists in simplifying the structure of the input sentence through an intermediate transformation stage, before running the extraction algorithm. The basic idea is that the linguistic complexity of a sentence might hinder meaningful and accurate extractions. Thus, the workflow is split between two phases. First, the sentence is processed to obtain an intermediate representation, allowing to reduce the complexity of the actual extraction task. Then, the extractor is applied on the transformed sentence to produce the final relational tuples. ClausIE[13], Stanford Open IE[14] and Graphene[15] adopt different techniques to implement the same underlying idea. The aforementioned frameworks aim to simplify the source sentence, by breaking it down into a set of independent clauses, and extract relational tuples from those independent utterances. Graphene, takes a step further, in that it aims to preserve the semantic relationships between individual clauses, resulting from sentence simplification. When it performs clausal decomposition of the source sentence, the system distinguishes between core and contextual clauses and connects them in a hierarchical way by means of rhetorical relations, such as *Contrast*, *Condition* and *Enablement*. The intermediate representation provided to the extractor is a discourse tree of propositions, in the form of core facts and accompanying context. As a result of applying this transformation, Graphene returns a set of semantically related tuples for each input sentence. Similar in the spirit to the described approach, our algorithm for triple extraction also relies on a hierarchical representation of the input sentence. However, we adopt a different methodology to build this intermediate representation. Firstly, the transformation process in Graphene is

carried out in a top-down fashion. Conversely, our approach is based on a bottom-up traversal of the parse tree. Secondly, we design transformation rules, that apply to a different syntactic structure, which makes use of dependency rather than constituency grammar.

Most recently proposed frameworks for OpenIE extraction, adopt a modular architecture, and use external resources, like pretrained models, either as components of their extraction pipeline, or at the later post-processing stage. OLLIE's [16] successor OpenIE4 [17], for instance, combines RelNoun [18], a rule-based system for extracting noun-mediated relations, and SRLIE, that relies on the output of the state-of-the-art PropBank-trained SRL system. The main advantage of our solution is its operational independence. SpanIE is a self-contained framework, with all modules of the extraction pipeline being implemented internally. Working only over the dependency parse, our system can produce an interpretable structured representation of a sentence, and use it to extract not-only verb-based relations, but also other kind of relational constructs.

3. SpanIE System

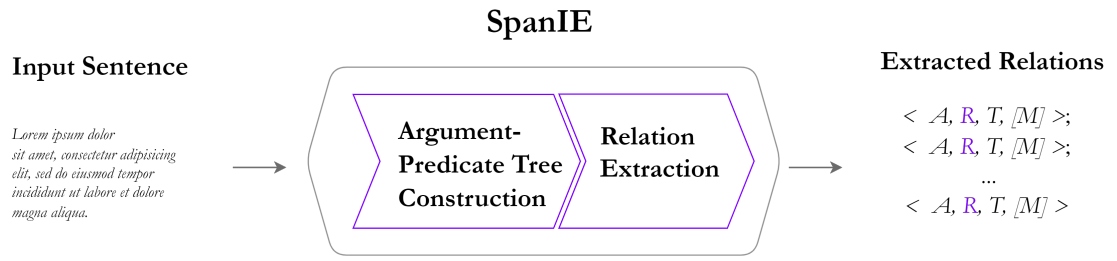
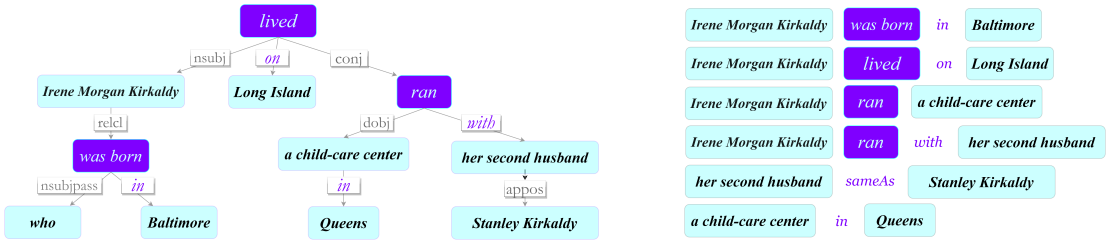


Figure 1: Overview of the SpanIE workflow

In this work, we extend the line of rule-based approaches on the task of OpenIE by proposing a new method for automatic extraction of binary relations from unstructured text. In Figure 1, we schematically outline the workflow employed by our system. Similar to existing solutions, SpanIE adopts a two stage approach to carry out the extraction process. Our novel idea is to simplify the task, by performing relation extraction on a sentence representation that directly reflect its underlying argument-predicate structure. This intermediate representation, which we call *argument-predicate tree*, enables the definition of simple extraction rules, that operate on a level of multi-token phrases instead of individual words (see Figure 2.1). For this purpose, we exploit the linguistic concepts of a nominal and predicate phrase, to reconstruct argument and predicate structures from automatically generated dependency parse of a sentence.

We use this representation as input to our relation extraction algorithm. The final output of the proposed framework is a set of relational tuples of the form $\langle A, R, T, [M] \rangle$ (Figure 2.2), where R denotes a relation phrase holding between two argument phrases, namely an *agent* A and a *target* T; M denotes an optional *modifier* that specifies the context associated with the relation R. Figure 2 provides an example of an argument-predicate tree and the final set



Argument-Predicate Tree (1) **Extracted relational tuples (2)**

Figure 2: An example of argument-predicate tree and corresponding tuples generated by SpanIE.

of relations generated by SpanIE from the sentence: *"Irene Morgan Kirkaldy, who was born in Baltimore, lived on Long Island and ran a child-care center in Queens with her second husband, Stanley Kirkaldy."*

3.1. Argument-Predicate Tree Construction

Given an input sentence, its argument-predicate tree is the result of a series of transformations defined over the syntactic structure of the sentence, represented by an automatically generated dependency tree, like the one in Figure 3, obtained with SpaCy[1] parser .

Our algorithm for the construction of the argument-predicate tree performs a bottom-up traversal of the sentence syntactic dependency tree, processing tokens one at a time based on their linguistic properties. For each token in the sentence, we first decide whether to skip it or keep it for further processing. Tokens considered irrelevant to the ultimate extraction task, such as punctuation and conjunctions, are discarded from the intermediate representation (see nodes colored in gray in Figure 3). If the token is retained, a decision is made on how to represent it in the argument-predicate tree. Depending on its syntactic dependency, a token can be folded into an arc (dashed nodes), or collapsed with its parent node (nodes with purples incoming edges). If none of these operations takes place, the token is included as is in the final representation. In particular, a token is folded if it is a preposition or a discourse word, like interjection, filler or non-adverbial discourse marker. In this case, the token is used to label an arc in the predicate-argument tree. Instead, a token is collapsed if its syntactic dependency is matching one of two distinct sets of patterns, corresponding to nominal or verbal phrases, respectively. In this case, we merge a token with its syntactic parent to build a multi-word phrase that represents the nominal or the verbal phrase in the final argument-predicate tree. For instance, compounds, determiners and nominal modifiers are included into the nominal phrase headed by the referent noun. Similarly, particles and auxiliaries make one verb phrase with the main verb.

Finally, in the argument-predicate tree (Figure 2.1), we distinguish between *predicate nodes*,

[1]https://spacy.io

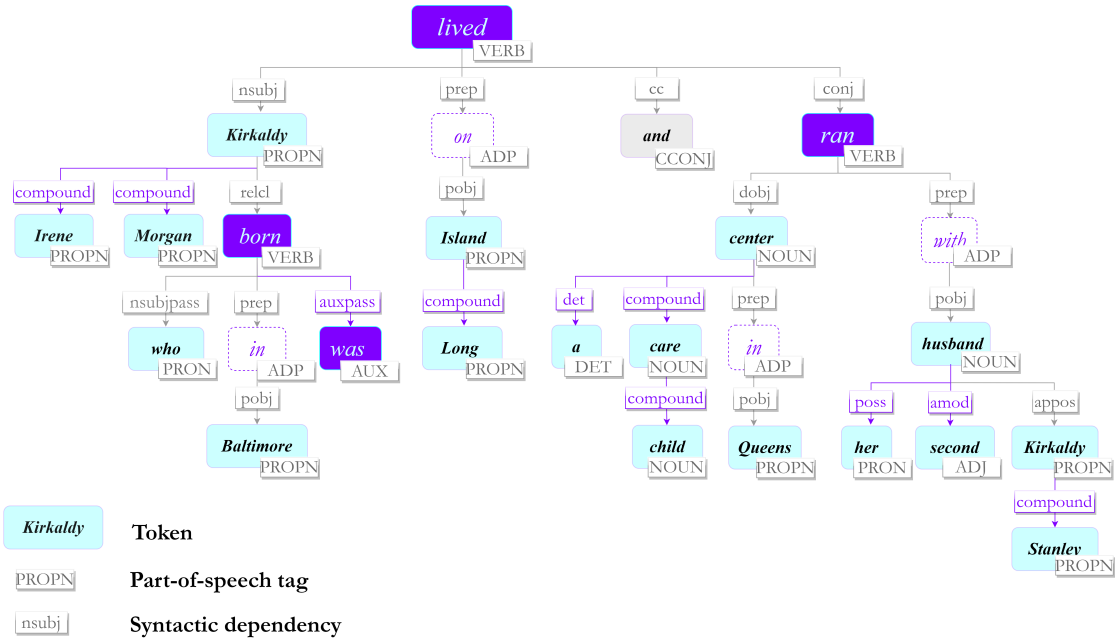


Figure 3: An example of a dependency parsing obtained with SpaCy.

expressed by verb phrases (in purple), and *argument nodes*, that are generically represented by nominal expressions (in cyan). Prepositions and discourse words add semantic information to this representation and are stored as labels on the arcs of the tree.

3.2. Relation extraction

In this section we describe the last stage of our SpanIE workflow. The extraction algorithm takes as input the argument-predicate tree and outputs a set of relational tuples of the form $\langle A, R, T, [M]\rangle$, where R represents a relation between and agent A and a target T, while M represents an (optional) modifier of the relation R. Usually, in the verb-based approach to relation extraction, R corresponds to a predicate phrase. SpanIE is not limited to verbal relations, but allows to additionally extract other relational constructs, like appositives (i.e., sameAs relations) or binary relations mediated by prepositions.

To perform the extraction task, we exploit the topology of the argument-predicate tree, matching extraction patterns against the simplified syntactic structure. Focusing on verbal relations, we perform a top-down traversal of the argument-predicate tree and collect all the predicate nodes (purple nodes). For each predicate, we inspect its local sub-tree, searching for argument nodes to use in the agent and target positions of the relational tuple. First, we identify the argument node A governed by the predicate via a *subject* relation. We call this argument the *agent* of the predicate. For each remaining argument node T_i in the argument set of the predicate, we build a relation $\langle A, R, T_i\rangle$. A predicate node can appear in the argument set of another predicate as its clausal complement (via *comp*, *xcomp*, and *pcomp* dependencies). In this

case, the predicate node is extracted twice, as a clausal argument of another predicate, leading to a nested extraction, and as a predicate in its own relational tuples. For instance, in the sentence *A team won the final, by cheating on its opponents.*, the predicate node *cheating*, appears in two extractions, ⟨A team, won, cheating, [by]⟩ and ⟨A team, cheating, its opponents, [on]⟩.

In the previous example, we see how SpanIE extracts relation modifiers such as prepositions. Sometimes, modifiers can be interpreted as an extension of the target argument nodes (as in the example, "by cheating" or "on its opponents"). This is not a binding behaviour. Our algorithm extracts prepositions as a separate component of a relation tuple, leaving the user free to decide where to place prepositions in the final output and how to use them to interpret the context of the relation R. Another option provided by our system, is the possibility to internally resolve pronouns that appear in the agent position of a relational tuple. For instance, in Figure 2.2 SpanIE outputs ⟨Irene Morgan Kirkaldy, was born, Baltimore, [in]⟩, instead of ⟨who, was born, Baltimore, [in]⟩. In addition, we also include patterns, that allow to optionally generate non verbal relational tuples. For example, we extract the appositive construct, ⟨her second husband, sameAs, Kirkaldy Stanely⟩, by matching on *appos* relation in the argument-predicate tree. In the same fashion, we can extract preposition-mediate relations, like ⟨a child-care center, in, Queens⟩.

4. Evaluation

In this section, we evaluate the performance of our system, using the OpenIE benchmark framework proposed in Stanovsky and Dagan (2016) [19]. We compare SpanIE with six prominent information extraction systems on a large benchmark corpus for OIE. The dataset comprises more than 10,000 tuples over 3,200 sentences from Wikipedia and the Wall Street Journal. The gold extractions were obtained for each verbal predicate, by converting the QA-Semantic Role Labeling annotations. Given the design of the reference benchmark, we evaluate our implementation only considering its predicate-based extractions. In Figure 4, we compare the performance of different systems by running the scripts used in Stanovsky and Dagan (2016)[2].

To perform the evaluation, the authors match the automatically generated extractions with gold extractions based on their lexical overlap. Unlike other systems in the reference benchmark, our current implementation does not estimate confidence score for different extractions. Thus, for the purpose of the evaluation, we assign a default 1.0 confidence score to all extractions generated by SpanIE. We report both the precision and recall (and F1) values and the corresponding precision-recall curves for all the benchmark systems, as shown in Figure 4. Compared to other systems, SpanIE exhibits the second best performance after OpenIE-4. However, it is important to stress the fact that the best performing system in the benchmark, OpenIE-4, heavily relies on the results of pre-trained models to perform relation extraction, while our self-contained framework is able to achieve comparable performance, working autonomously by an approach that is exclusively based on morphosyntactic rules. This makes the results promising and the system highly customizable.

[2]The scripts are available at: https://github.com/gabrielStanovsky/oie-benchmark.

	Precision	Recall	F1
ClausIE	0.414	0.679	0.514
OpenIE-4	0.615	0.686	0.648
Stanford	0.137	0.306	0.189
PropS	0.500	0.482	0.491
Graphene	0.437	0.335	0.379
OLLIE	0.500	0.514	0.507
SpanIE	0.554	0.677	0.609

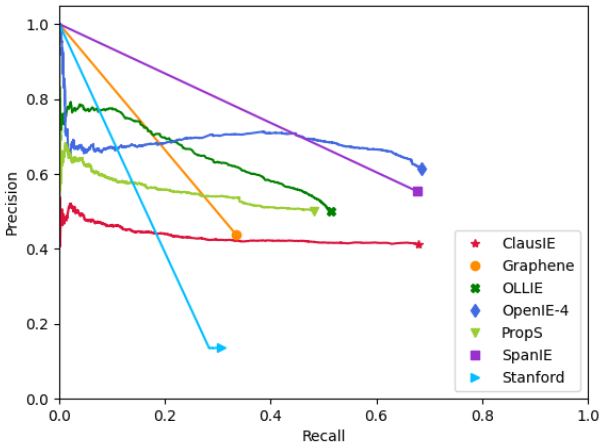


Figure 4: Precision Recall values and curves for OpenIE systems

5. Concluding remarks

In this paper, we presented SpanIE, a span-based approach to solve the task of Open Information Extraction. SpanIE is designed as a self-contained framework, which internally implements all the necessary modules of its extraction pipeline. The distinctive feature of our solution is the ability to limit the dependency on external models and solve the task autonomously, by only relying on a set of deterministic rules defined over the syntactic structure of a sentence. The preliminary results obtained from a comparative evaluation using the OIE benchmark, provide comforting evidence of the effectiveness of the proposed approach. Our system achieves a comparable performance with the state-of-the-art systems for OpenIE, while offering a novel way to leverage syntactic information in the task of relation extraction. In the future work, we plan to examine further strategies to improve the system performance and consider extending our framework with semantic information about different relational types. Additionally, we would like to investigate the idea of ontology construction from the knowledge graph of relation extractions produced with the proposed approach.

References

[1] O. Etzioni, M. Banko, S. Soderland, D. S. Weld, Open information extraction from the web, Communications of the ACM 51 (2008) 68–74.

[2] D. Jurafsky, J. H. Martin, Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition, 2023.

[3] S. Soderland, J. Gilmer, R. Bart, O. Etzioni, D. S. Weld, Open information extraction to kbp relations in 3 hours., in: TAC, 2013.

[4] B. D. Mishra, N. Tandon, P. Clark, Domain-targeted, high precision knowledge extraction, Transactions of the Association for Computational Linguistics 5 (2017) 233–246.
[5] J. Berant, I. Dagan, J. Goldberger, Global learning of typed entailment rules, in: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, 2011, pp. 610–619.
[6] A. Fader, L. Zettlemoyer, O. Etzioni, Open question answering over curated and extracted knowledge bases, in: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, 2014, pp. 1156–1165.
[7] F. Wu, D. S. Weld, Open information extraction using wikipedia, in: Proceedings of the 48th annual meeting of the association for computational linguistics, 2010, pp. 118–127.
[8] 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, 2011, pp. 1535–1545.
[9] 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), 2012, pp. 52–56.
[10] 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, 2013, pp. 447–457.
[11] G. Stanovsky, J. Ficler, I. Dagan, Y. Goldberg, Getting more out of syntax with props, arXiv preprint arXiv:1603.01648 (2016).
[12] S. Zhang, R. Rudinger, B. Van Durme, An evaluation of predpatt and open ie via stage 1 semantic role labeling, in: IWCS 2017—12th International Conference on Computational Semantics—Short papers, 2017.
[13] L. Del Corro, R. Gemulla, Clausie: clause-based open information extraction, in: Proceedings of the 22nd international conference on World Wide Web, 2013, pp. 355–366.
[14] G. Angeli, M. J. J. 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), 2015, pp. 344–354.
[15] M. Cetto, C. Niklaus, A. Freitas, S. Handschuh, Graphene: Semantically-linked propositions in open information extraction, arXiv preprint arXiv:1807.11276 (2018).
[16] M. Schmitz, S. Soderland, R. Bart, O. Etzioni, et al., 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, 2012, pp. 523–534.
[17] M. Mausam, Open information extraction systems and downstream applications, in: Proceedings of the twenty-fifth international joint conference on artificial intelligence, 2016, pp. 4074–4077.
[18] H. Pal, et al., Demonyms and compound relational nouns in nominal open ie, in: Proceedings of the 5th workshop on automated knowledge base construction, 2016, pp. 35–39.
[19] 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, 2016, pp. 2300–2305.