Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 225 (2023) 2224–2233

Procedia Computer Science

www.elsevier.com/locate/procedia

27th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES 2023)

SailGenie: SAiling expertIse to knowLedge Graph through opEN Information Extraction

Salvatore Carta[a], Pietro Fariello[b], Alessandro Giuliani[a], Leonardo Piano[a], Alessandro Sebastian Podda[a], Sandro Gabriele Tiddia[a,*]

[a]*Department of Mathematics and Computer Science, University of Cagliari, 09032, Cagliari, Italy.*
[b]*Sailmaster S.r.l., 09124, Cagliari, Italy.*

Abstract

This work is focused on the sailing domain, for which several innovative technologies are being adopted to improve sailing efficiency, performance, and safety. In this context, a knowledge graph could be used, for example, to represent information about different types of boats, sailing techniques, maritime safety, or weather conditions. Although numerous construction methods or ready-to-go knowledge graphs have been proposed in many fields, the sailing domain still needs to be explored. As the most effective methods rely on domain-specific datasets, the absence of suitable and available sailing datasets is one of the main challenges. Although several Open Information Extraction (OpenIE) methods may generate relevant triplets (the elementary units composing a knowledge graph) from arbitrary text without any additional information about its topic, such methods usually generate many incorrect triplets. In this paper, we aim (i) to address the aforementioned problem by proposing an innovative method that combines in an improved and strengthened way different OpenIE tools to generate proper triplets from domain-specific sources and, in particular, (ii) to build and release a suitable dataset for the sailing domain. Results confirm that our proposal can maximize the extracted information and infer unique information irretrievable by the classical OpenIE tools and, furthermore, that the generated dataset is significantly valuable for the sailing scenario.

Keywords: Artificial Intelligence; Knowledge graphs; Sailing; Domain-specific dataset.

1. Introduction

The recent advancement of web technologies led to an exponential increase in the volume of data in many real-world scenarios. Although, on the one hand, this scenario conveys several benefits in various fields, on the other hand, it comes with several challenges. One of the main drawbacks is information overload — i.e., when dealing with an

[*] Corresponding author. Tel.: +39-675-8705
E-mail address: sandrog.tiddia@unica.it

10.1016/j.procs.2023.10.213

excessive amount of information causes extreme difficulty in making a decision — in those willing to analyze it and complete disinterest in those less so. A practical approach to solve this is to develop a suitable knowledge graph (KG), a powerful tool aimed at organizing, accessing, understanding, and utilizing information in an efficient and effective way. In detail, a knowledge graph is a visual representation of knowledge and all the relationships among different entities, concepts, and data points. Nowadays, KGs are a prominent area of Artificial Intelligence [20]. Although many widespread general-purpose KGs permit to capture and represent generic knowledge of real-world data, such as Wikidata[1], DBPedia[2], or YAGO[3], domain-specific KGs are becoming necessary for targeted real-world scenarios [16], e.g., in online news platforms [23], fact-checking [3], health and life sciences [10], or, in fact, sailing. On the one hand, although the aforementioned generic KGs can cover and represent a broad range of information, they are usually not suitable nor sufficiently exhaustive for the needs of the specific domains. On the other hand, developing a custom KG is highly challenging and needs significant domain expertise. Indeed, one of the main issues is that the most effective strategies for KG construction rely on domain-specific datasets, which are usually not publicly available or even absent altogether. To our knowledge, there are no suitable and available sailing datasets. A popular approach to overcome the absence of available data is to rely on Open Information Extraction (OpenIE), the task of extracting relevant triplets from text without any domain knowledge information or human expertise [1], a *triplet* being the elementary unit of KGs composed of three components: subject, predicate, and object. Let us point out that triplets and KGs are strictly related, as a KG uses a graph-based model to organize information carried out by the triplets into nodes and edges, where nodes are entities (subject or object), and edges are the predicates. To this end, as triplet generation is the key to constructing a knowledge graph, in this paper we mainly focus on triplets. Several OpenIE methods have been proposed for generating KGs from free and open-domain texts. However, many shortcomings in terms of coverage and applicability to numerous specific domains affect state-of-the-art methods [24]. Therefore, most extracted triplets are often irrelevant or incorrect.

This work addresses such issues in a specific scenario, i.e., the *sailing* domain. The sailing world is an extended and passionate universe that had people involved for millennia. All sailing activities, from the simple entertainment of a sundown sail with friends or families to the more demanding scenarios of sailing races, require a unique combination of physics and mental skills, together with an in-depth knowledge of every aspect of sailing, e.g., a deep understanding of the vessel, wind, water, and weather conditions. The sailing world has evolved, mainly in recent decades, supported by the advent of modern design tools and sustained by continuous technological evolution. In particular, the digitalization era allowed a substantial transformation in the sailing domain, as sailors nowadays may access a vast number of resources and information that support them in making the sailing experience more enjoyable and safe. For example, sailors may now easily plan routes, track other vessels' positions, and receive real-time oceanographic and weather data. Furthermore, developing advanced digital sensors and monitoring systems may provide more detailed information about the condition of vessels and equipment, which can prevent failures and accidents and improve sailing performance. In addition, data overloading also affects the sailing world, considering that rapidity can be crucial in such a scenario, especially in emergencies.

To this end, using KGs may support and enhance the sailing experience for all sailors, as it can link together a large amount of heterogeneous information, such as the types of vessels, sailing procedures, navigation hazards, weather conditions, and many more, in a proper and easily interpretable structure. It might be used, for example, to integrate the already developed technology of weather routing — planning a route based on the forecast — with boat structural limits and the people's comfort/enjoyment.

In this paper, we focus on the problem of devising a proper KG for the sailing domain, addressing the lack of specific domain ready-to-go resources (e.g., KGs or datasets). In particular, we yield the following scientific contributions:

- we analyze and discuss the challenges and the limitations of the available resources in the sailing context, motivating the need for developing a proper KG in this scenario;
- we devised an innovative method for automatically generating proper triplets based on an enhanced combination of classical OpenIE models, intending to reduce the human effort;

[1] https://www.wikidata.org/
[2] https://www.dbpedia.org/
[3] https://yago-knowledge.org/

- starting from several sources, we construct a domain-specific ground truth, i.e., a set of relevant triplets representing the basic information for the sailing scenario;
- we build a suitable dataset, focused on two main real-world sailing areas, i.e., the knowledge of *sailing basics* and *navigation safety*. The dataset is composed of manually annotated correct and incorrect triplets.

The rest of the paper is organized as follows: Section 2 reports the related work. Section 3 describes the proposed methodology for generating relevant triplets for the sailing domain and the developed dataset, whereas Section 4 reports the experiments. Section 5 ends the paper with the conclusions.

2. Related Work

Sailing is a complex activity that requires much training and expertise. Several efforts have been made in recent years to make sailing safer and improve the performance of skippers on board through intelligent systems [13]. In this regard, knowledge graphs can enhance sailor safety and security by representing information about potential hazards, risks, and emergencies and providing personalized recommendations. The automatic construction of KGs directly from text has attracted considerable attention from the research community over the last few years [29] whereby a multitude of Knowledge Graph Construction (KGC) pipelines have been presented in a wide range of domains, including education [2], geoscience [28], food [12] and financial news [9]. KGC pipelines leverage information extraction methods, which can be distinguished into two macro-categories, Closed Information Extraction (ClosedIE) and Open Information Extraction (OpenIE). The former identifies instances from a fixed set of corpora, considering only a closed set of relationships between two arguments [18]. Thus, building domain-specific KGs with ClosedIE methods uses a pre-defined schema, an agreed set of specific concept types, and relation types for vertices and edges [15] . In contrast, Open Information Extraction (OpenIE) systems adopt a domain-agnostic method and can extract entities and relationship triples from any sentence written in natural language. Basically, OpenIE methods try to identify linguistic extraction patterns, either hand-crafted or automatically learned from the data. ClausIE [7] uses linguistic knowledge about the English language grammar (i.e., dependency parsing) first to detect clauses in an input sentence and, subsequently, to extract propositions. REVERB, introduced by Fader et al. [11], extracts tuples by singling out relation phrases that satisfy syntactic and lexical constraints. End-to-end neural systems have been proposed to bypass error accumulation in the ruled-based pipelines. The first attempt has been made by Stanovsky et al. [25], where the author formulates OpenIE as a sequence labeling problem and applies an LSTM-transducer to extract OpenIE triplets automatically. Similarly, Kolluru et al. resolved OpenIE as a sequence labeling problem [17], and Ro et al. propose a multilingual OpenIE system that exploits BERT and multi-head attention [22]. OpenIE has also been addressed as a sequence generation problem where a Sequence-to-Sequence model is trained to ”translate” an input sequence to a tuple by adding field demarcators [6]. Generating Domain-specific KGs using Open Information Extraction only is currently an open and complex challenge. Few researchers have already faced this problem. Jain et al. combined existing OpenIE methods to build an art-historic KG that can facilitate data exploration for domain experts [14]. Similarly, Muhammad et al. explored an Open Information Extraction approach for constructing literature knowledge graphs [19].

3. Methodology

With the goal of extracting relevant triplets from sailing-related content, we focused on addressing two critical issues that make this task particularly challenging:

- **the absence of specialized domain tools**. To overcome this issue, we rely on OpenIE approaches to capture relevant sailing entities and relations and generate meaningful triplets.
- **the lack of annotated datasets**. We deal with this lack by building a suitable dataset, which includes (i) the input corpus, i.e., the set of textual excerpts extracted from several information sources, (ii) a suitable ground truth, i.e., a set composed of domain triplets manually inferred by the corpus, and (iii) a set of triplets generated with the OpenIE-based tools, annotated and evaluated by human assessors.

3.1. Triplet Generation

This Section describes our approach for generating relevant triplets starting from domain-specific resources. As already pointed out, we rely on an improved and innovative combination of OpenIE methods to overcome their problem of extracting too many irrelevant or incorrect triplets. Figure 1 depicts an overview of the system architecture.

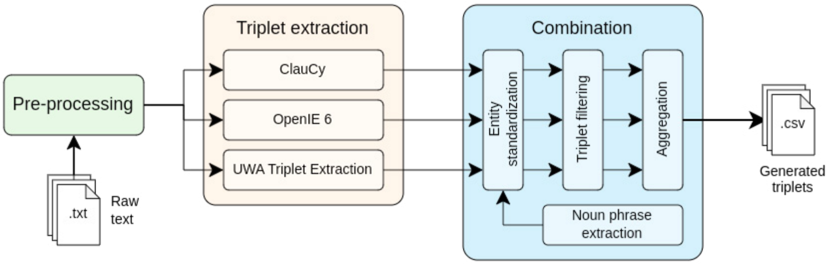

Fig. 1: Schema overview of our OpenIE method.

The approach is sequentially organized into three main stages: *Pre-processing*, *Triplet extraction*, and *Combination*. Each stage is detailed in the following.

3.1.1. Pre-processing

To extract more meaningful triplets, we pre-process the input sentences with NeuralCoref[4] [4], which annotates and resolves co-reference clusters using a neural network and particularly helps in replacing part of the pronouns with the original noun (hence, a potential entity) they are masking.

3.1.2. Triplet extraction

OpenIE tools, as anticipated in Section 2, extract triplets following domain agnostic approaches that only leverage the knowledge about the language grammar and the observable patterns between related entities in the sentence, whether this knowledge is manually defined or automatically apprehended. Thus, each approach can potentially perceive different information not necessarily captured by every other tool. According to this insight, we decided to rely on a combination of OpenIE tools that significantly differ in their implementation, aiming to maximize the coverage of the extracted information. Let us point out that although we are aware that at the state of the art there are numerous OpenIE methods, the focus, in this stage, is not to compare all of them but rather to prove that the combination of two or more approaches improves the quality of the generated triplets. Therefore, we selected some of the most performative approaches, described below:

- **ClauCy**[5], a Python implementation of the previously introduced ClausIE [7], works upon a set of "clauses" from each sentence and their corresponding "type" (identified according to the grammatical function of the clause constituents), extracting triplets from the clause constituents themselves;
- **OpenIE6**[6] [17] uses a neural sequence labeling OpenIE approach that assigns the triplet role labels (subject, predicate, and object) to the sentence tokens according to their grammatical function, using a 2D grid setting where the sentence tokens define the columns, and the rows allow the extraction of multiple triplets;
- **UWA**[7] [26] [27] first identifies, through POS patterns, the head and tail entities with their relations to create a list of in-sentence triplets, and then creates a graph to uncover the relations among entities in separate sentences.

The abovementioned approaches are among the most widely used and well-established methods and provide a solid benchmark. In particular, the UWA implementation provided by Stewart et al. won the first prize in the 2019 ICDM/ICBK contest about automatic KGs construction [30].

[4] https://github.com/huggingface/neuralcoref
[5] https://github.com/mmxgn/spacy-clausie
[6] https://github.com/dair-iitd/openie6
[7] https://github.com/Michael-Stewart-Webdev/Seq2KG/tree/master/sourcecode/candidate_extraction

3.1.3. Combination

Although we have observed a broader retrieval of relevant information by simply aggregating the outcomes of different OpenIE implementations, only a part of all the additional triplets uncovered new unique information. Indeed, some triplets may just be slightly different from other discovered triplets, being, therefore, redundant. Moreover, due to the nature of OpenIE, numerous extracted triplets from each selected tool are irrelevant or incorrect, resulting in a conspicuous number of noisy triplets. To address this problem, we introduced a stage in which we define a standard form for the entities, creating a first step towards entity resolution, and we also apply an additional (and more conservative) set of triplets filtering rules, intending to remove the noisy elements.

Entity standardization. The standard entity form relies on a reference set of noun phrases[8] that we use to map our subjects and objects. In detail, we exploit the widespread Spacy[9] noun chunker to extract the reference noun phrases. Let us note that each tool processes the text in a different way; hence, to have a reliable mapping, we define the following strategy:

1. we first normalize each string by character cleansing, i.e., lowering and applying some regular expressions to remove punctuation, special characters, unnecessary spaces, and the starting articles, and, subsequently, performing a lemmatization;
2. for each subject/object composing the extracted triplets, we search for its corresponding noun phrase by retrieving the one that contains or is contained in the subject/object;
3. in case of multiple matches, we empirically assign a score α to each match between an entity e and a noun phrase p as (`len` is the length of a string):

   $$\alpha =\mid len(e) - len(p) \mid$$

   We will keep the match with the minimum score. To overcome the issue of retrieving too long entities or no one, if no matches have $\alpha \leq 0.5 * len(e)$, we keep the original subject/object.

Triplet filtering. After obtaining a standardized form for each subject and object, we apply filtering rules designed to cut out the triplets with the highest probability of being incorrect or uninformative. Therefore, we keep the triplets compliant with the following set of rules:

- subject and object must not be null or contain empty strings;
- subject and object must differ;
- subject and object must not be pronouns (`PRON`), subordinating conjunctions (`SCONJ`) or cardinal numbers (`CD`);
- subject and object must contain a noun (`NOUN` or `PROPN`);

Aggregation. Entity standardization converts similar triplets to a unique form. Nevertheless, there may still be triplets whose subjects or objects are contained in another triplet with the same predicate. Although appearing quite different, they still refer to the same concept, i.e., they are *redundant*. In this step, we aim to unify also these triplets. To this end, we identify the sets of redundant triplets, and, for each set, we keep the triplets with the longest subject and object, discarding all the others. For example, from the sentence “Boat will often sail a zig-zag course downwind” the following triplets may be extracted:

- (boat; sail; course)
- (boat; sail; zig-zag course)
- (boat; sail; zig-zag course downwind)

In this example, we would keep only the third triplet, being the more informative. Merging the triplets solves the redundancy issue and defines the final set of triplets. We describe the experiments and discuss the results in Section 4.

[8] Noun phrases are segments of text which include nouns and any words that depend on and accompany nouns.

[9] https://github.com/explosion/spaCy

3.2. Domain dataset building

The further main contribution of our work is building a sailing-specific dataset, aiming to address the lack of any specific resources in this domain. In detail, the proposed dataset encompasses three main components, i.e., the original *corpus*, the *groundtruth*, and the generated *set of triplets*. We deem that the resulting dataset, which is publicly available[10], may represent a primary reference for further developments of specialized and more powerful tools operating in the sailing domain.

3.2.1. Corpus

The underlying component of the dataset is the corpus we built by identifying a set of reference texts on the sailing domain. In particular, we manually retrieved content from two different types of sources:

- *WEB*: a set of sailing articles extracted from Wikipedia and Discover Boating[11], the latter being a website containing vast information for supporting people in getting started with boating experiences;
- *YT*: a collection of sailing video tutorials from various YouTube channels of sailing experts and enthusiasts, from which we extracted the textual transcription with Whisper[12] [21].

Although we rely on textual representation for both types of content, using multimedia sources permits us to introduce sentences with different styles and structures, which vary deeply from encyclopedic articles to speeches in video tutorials, leading to augmenting the representativeness of the selected samples. Moreover, we also organized the collected content according to the covered topic, distinguishing between two different subdomains:

- *BASICS*: related to fundamental knowledge for beginners about navigation, behavior, and maneuvers;
- *SAFETY*: includes information on measures, legal requirements, instruments, and best practices specifically required to ensure everyone's safety during sailing.

From this collection, we manually identified a subset of paragraphs, aiming to define a more concise and relevant selection of sentences containing valuable knowledge, removing redundant or irrelevant information, e.g., the intros and outros of articles and videos or excerpts not particularly related to sailing concepts. Table 1 summarizes the distribution of the final set of sentences.

Table 1: Number of sentences collected for each category.

Sentences	BASICS	SAFETY	Total
WEB	134	76	210
YT	58	66	124
Total	192	142	334

3.2.2. Ground truth

Starting from the selected sentences, we create the ground truth by manually identifying the triplets from each sentence. In detail, several human annotators identified a set of relevant triplets according to the following criteria:

- domain triplets only, i.e., only triplets with entities and relations related to sailing;
- subject and objects should not be pronouns;
- split the subjects/objects with an "and/or" conjunction into multiple triplets;
- avoid triplets with overly complex subjects and objects;
- predicates should identify relations by using words explicitly found in the sentence.

[10] https://zenodo.org/record/7794131

[11] https://www.discoverboating.com/

[12] https://github.com/openai/whisper.git

Throughout the annotation process, ongoing interactive comparison sessions have been held between the annotators, where annotations were shared and commented on by the annotators to ensure mutual agreement on the predefined criteria. The final product is a set of triplets for each domain, including their reference to the sentences they have been extracted from. We extracted 509 triplets for BASICS (average of 2.65 triplets/sentence) and 368 triplets for SAFETY (average of 2.59 triplets/sentence).

3.2.3. Annotated triplets

We extracted triplets from our dataset corpus using both our method and the individual tools listed in Section 3.1.2. We then manually annotated a subset of the generated triplets as valid or invalid, aiming to define a set of positive and negative samples that serve as an additional contribution to the dataset.

Table 2: Number of extracted triplets on the sailing corpus for each sub-domain.

	BASICS	SAFETY	Total
ClaucyIE	448	319	767
OpenIE 6	538	339	877
UWA	528	389	917
SailGenie	839	538	1377
Sum	2353	1585	3938
Unique	2070	1363	3433
Sample	292	215	507

In detail, for each domain and triplet, we kept a trace of the tool that generated it, i.e., ClauCy, OpenIE6, UWA, or our combination of them. Then, we grouped the triplets while aggregating the information about their related tool in a set, removing the occurrence of repetitive triplets that could lead to a mismatching annotation by the same annotator at different moments. We obtained 2070 unique triplets from the BASICS domain and 1363 from SAFETY. To reduce the human effort of the annotation but, at the same time, keep a significative sample, we selected the triplets from a random selection of 50 sentences (25 for each domain), obtaining 292 unique triplets from the BASICS domain and 215 from SAFETY. We report a summary in Table 2. We involved three annotators to remove subjective biases in the evaluation of the 507 isolated triplets, agreeing to label as valid the only triplets that:

- represent true facts;
- involve entities and relations related to sailing;
- do not contain pronouns or stop-words as entities.

The most frequent label among the three annotators determines the final label and the validity of the triplet. For fairness, we hid the information about the tool in the triplets presented to the annotators. As an example, from the BASICS sentence "*The boat has to sail at an angle to the wind, with the sails acting like wings and generating lift.*" the tools have extracted valid triplets like:

- the boat; has; sail
- the sails; acting; like wings
- the sails; generating; lift
- the boat; has to sail at; angle

In Figure 2 is reported an excerpt of a feasible KG built with valid triplets from SailGenie only.

Annotator agreement. Let us note that, in the case of manual annotations, the risk is dealing with a high disagreement among annotators, which may lead to an untrustworthy evaluation. To this end, we evaluated the inter-annotator agreement by computing Cohen's Kappa [5] statistic on the resulting labels. The metric compares the agreement between two annotators, calculating the degree of their agreement while excluding the probability of consistency expected by chance. In detail, applying Cohen's Kappa on all combinations of annotators, we obtain an average of 0.607. As a rule of thumb, a value above 0.6 (the score varies in the [0, 1] range) represents a satisfactory annotator agreement.

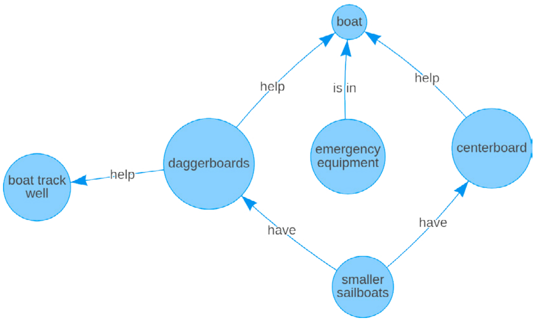

Fig. 2: Excerpt of a feasible KG using SailGenie triplets.

4. Experiments

This Section reports the performed experiments, which have the twofold aim of (i) assessing the effectiveness of our approach, demonstrating that using a suitable combination of OpenIE approaches is more reliable in capturing domain-relevant entities and relations than the classical methods, and (ii) applying our method also for estimating the usefulness of the released dataset, performing a proper comparison with the classical OpenIE methods listed in Section 3.1.2, which represent the *baselines* tools, basing on a human-based assessment.

4.1. Preliminary experiments

Before evaluating SailGenie, which implements our approach, on the specific sailing domain, we estimated the usefulness of the tool combination on publicly available datasets. In doing so, we assumed the same experimental setup employed by Stewart and Liu [27] in the performance evaluation of their UWA tool.

Datasets. Stewart and Liu [27] addressed the same lack of datasets available for evaluating triplet extraction systems creating three different datasets, named *Restaurant Services* (CS), *Automotive Engineering* (AE), and *BBN*. CS and AE were built from scratch by scraping several news websites about the catering and automotive industries; BBN is an existing and standard benchmarking dataset for entity typing, from which a small portion of the corpus was selected. All corpora were annotated with triplet memberships and released publicly as datasets[13]. For each dataset, we refer to the triplets manually extracted from the test partitions of the datasets, i.e., their ground truth.

Results. To assess the quality of the generated triplets, we rely on the same metric proposed by Stewart and Liu [27]: the *embedding similarity* (ESim), which aims to combine structural and semantic similarity between the generated and ground truth triplets into a single score by comparing the paths in graphs constructed by the two sets of triplets. We report all *ESim* scores in Table 3, which displays all the computed scores, showing that our tool performs best for each dataset. The best scores are highlighted in bold. These results highlight how SailGenie is more reliable than the individual tools in capturing the semantics and the structure characterizing the ground truth for different domain-specific datasets.

Table 3: Embeddings similarities: comparison with the literature datasets.

	AE	BBN	CS
ClaucyIE	0.9055	0.9192	0.9097
OpenIE 6	0.9109	0.9374	0.9208
UWA	0.9162	0.9367	0.9242
SailGenie	**0.9194**	**0.9379**	**0.9244**

[13] https://github.com/Michael-Stewart-Webdev/Seq2KG/tree/master/datasets

4.2. Experiments on sailing data

To assess the effectiveness of our method, we focused on evaluating SailGenie in the scenario of generating triplets starting from sailing-related data. In doing so, we relied on our released dataset for testing and individually comparing SailGenie with the baselines listed in Section 3.1.2 in their vanilla status.

Dataset. We evaluated the abovementioned models using the domain-specific dataset we built, described in Section 3.2.1, consisting of a collection of sailing-domain sentences from multimedia sources, organized into the two different sub-domains BASICS and SAFETY, with respectively 192 and 142 sentences each.

Evaluation metrics. Although automated evaluation can be helpful for evaluating particular aspects of a KG, manual assessments provide a more fine and comprehensive evaluation that considers the complexity, subjectivity, and context of the knowledge being represented. To this end, instead of relying on the embedding similarities-based metric, which can only help in estimating the equality between two sets of triplets (i.e., generated and ground truth), we decided to rely on the manual annotations described in Section 3.2.3. According to previous works [8], we adopt well-known evaluation metrics that perfectly fit our scenario. In detail, the classical Machine Learning metrics *precision* and *recall* are described as follows: given a tool k belonging to the set of all tools $\mathcal{K}$, let T_k refer to the triplets generated by the underlying tool, and G refer to mappings in the *gold standard*, being, in our case, the set of triplets generated by all the tools and labeled as “valid” by annotators. *Precision* is defined as $P_k =\mid T_k \cap G \mid / \mid T_k \mid$ and *recall* as $R_k =\mid T_k \cap G \mid / \mid G \mid$. In other words, P_k is the ratio between the number of valid triplets and the total number of triplets generated from k, and R_k is the ratio between the number of valid triplets generated by k and the total number of relevant triplets. Furthermore, we also computed the F_1 measure, being the equally weighted harmonic mean of both values, i.e., $F_{1,k} = 2 \cdot P_k \cdot R_k/(P_k + R_k)$.

Results. We report all the computed metrics in Table 4. As expected, compliant with the exploratory experiments, SailGenie obtained the best precision for each sub-domain, indicating how it is able to generate more relevant triples. Furthermore, SailGenie was the best tool considering the recall in both sub-domains without being penalized in precision, which indicates the positive support in retrieving more information while limiting noisy triplets provided by our combination process.

Table 4: Evaluation metrics scores for the different tools and sub-domains.

	BASICS			SAFETY		
	precision	*recall*	F_1	*precision*	*recall*	F_1
ClaucyIE	0.3065	0.1863	0.2317	0.1250	0.0968	0.1091
OpenIE 6	0.3067	0.2255	0.2599	0.3273	0.2903	0.3077
UWA	0.3611	0.2549	0.2989	0.2909	0.2581	0.2735
SailGenie	**0.3770**	**0.4510**	**0.4107**	**0.4231**	**0.5323**	**0.4714**

Summarizing, Table 4 highlights how SailGenie outperformed the OpenIE baselines, providing a clear insight that it can effectively maximize the extracted information.

5. Conclusions

In this paper, we proposed SailGenie, an effective tool for generating suitable triplets representing a domain-specific Knowledge Graph in the sailing scenario, and we built and released a proper dataset that addresses the lack of specific domain resources in this domain. In particular, the proposed tool combines different OpenIE tools to generate relevant triplets from domain-specific sources, whereas the dataset contains a ground truth and a set of correct and incorrect triplets belonging to the sailing basics and the navigation safety contexts. Experiments proved the effectiveness of our method in capturing sailing-specific knowledge. The generated dataset can be a valuable resource for the scientific community in further developments of specialized models and services operating in the sailing domain.

Acknowledgements

This work has been partially carried out thanks to the Ministerial Decree no. 351 of 9th April 2022, based on the NRRP – funded by the European Union - NextGenerationEU - Mission 4 “Education and Research”, Component 1 “Enhancement of the offer of educational services: from nurseries to universities” - Investment 4.1, that provided a financial support for the Leonardo Piano’s doctoral pathway.

References

[1] Banko, M., Cafarella, M.J., Soderland, S., Broadhead, M., Etzioni, O., 2007. 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. pp. 2670–2676.

[2] Chen, P., Lu, Y., Zheng, V.W., Chen, X., Li, X., 2018. An automatic knowledge graph construction system for k-12 education, in: Proceedings of the Fifth Annual ACM Conference on Learning at Scale, Association for Computing Machinery, New York, NY, USA. URL: https://doi.org/10.1145/3231644.3231698, doi:10.1145/3231644.3231698.

[3] Ciampaglia, G.L., Shiralkar, P., Rocha, L.M., Bollen, J., Menczer, F., Flammini, A., 2015. Computational fact checking from knowledge networks. PLOS ONE 10, 1–13. URL: https://doi.org/10.1371/journal.pone.0128193, doi:10.1371/journal.pone.0128193.

[4] Clark, K., Manning, C.D., 2016. Deep reinforcement learning for mention-ranking coreference models, in: Conference on Empirical Methods in Natural Language Processing.

[5] Cohen, J., 1960. A coefficient of agreement for nominal scales. Educational and Psychological Measurement 20, 37–46. doi:10.1177/001316446002000104.

[6] Cui, L., Wei, F., Zhou, M., 2018. Neural open information extraction, in: Annual Meeting of the Association for Computational Linguistics.

[7] Del Corro, L., Gemulla, R., 2013. Clausie: clause-based open information extraction, in: Proceedings of the 22nd international conference on World Wide Web, pp. 355–366.

[8] Dutta, A., Meilicke, C., Niepert, M., Ponzetto, S., 2013. Integrating open and closed information extraction: Challenges and first steps, in: Proceedings of the 2013th International Conference on NLP & DBpedia - Volume 1064, CEUR-WS.org, Aachen, DEU. p. 50–61.

[9] Elhammadi, S., Lakshmanan, L.V.S., Ng, R.T., Simpson, M., Huai, B., Wang, Z., Wang, L., 2020. A high precision pipeline for financial knowledge graph construction, in: International Conference on Computational Linguistics.

[10] Ernst, P., Meng, C., Siu, A., Weikum, G., 2014. Knowlife: A knowledge graph for health and life sciences, in: 2014 IEEE 30th International Conference on Data Engineering, pp. 1254–1257. doi:10.1109/ICDE.2014.6816754.

[11] Fader, A., Soderland, S., Etzioni, O., 2011. Identifying relations for open information extraction, in: Conference on Empirical Methods in Natural Language Processing.

[12] Haussmann, S., Seneviratne, O.W., Chen, Y., Ne'eman, Y., Codella, J., Chen, C.H., McGuinness, D.L., Zaki, M.J., 2019. Foodkg: A semantics-driven knowledge graph for food recommendation, in: International Workshop on the Semantic Web.

[13] van Hillegersberg, J., Vroling, M., Smit, F., 2017. Improving decision making in ocean race sailing using sensor data .

[14] Jain, N., Sierra-Múnera, A., Streit, J., Thormeyer, S., Schmidt, P., Lomaeva, M., Krestel, R., 2022. Generating domain-specific knowledge graphs: Challenges with open information extraction .

[15] Josifoski, M., Cao, N.D., Peyrard, M., West, R., 2021. Genie: Generative information extraction. ArXiv abs/2112.08340.

[16] Kejriwal, M., 2019. Domain-Specific Knowledge Graph Construction. 1st ed., Springer Publishing Company, Incorporated.

[17] Kolluru, K., Adlakha, V., Aggarwal, S., Mausam, Chakrabarti, S., 2020. Openie6: Iterative grid labeling and coordination analysis for open information extraction. ArXiv abs/2010.03147.

[18] Liu, G., Li, X., Wang, J., Sun, M., Li, P., 2020. Extracting knowledge from web text with monte carlo tree search. Proceedings of The Web Conference 2020 .

[19] Muhammad, I., Kearney, A., Gamble, C., Coenen, F., Williamson, P.R., 2020. Open information extraction for knowledge graph construction, in: DEXA Workshops.

[20] Paulheim, H., 2017. Knowledge graph refinement: A survey of approaches and evaluation methods. Semantic Web 8, 489–508.

[21] Radford, A., Kim, J.W., Xu, T., Brockman, G., McLeavey, C., Sutskever, I., 2022. Robust speech recognition via large-scale weak supervision. ArXiv abs/2212.04356.

[22] Ro, Y., Lee, Y., Kang, P., 2020. Multiˆ2oie: Multilingual open information extraction based on multi-head attention with bert. ArXiv abs/2009.08128.

[23] Rudnik, C., Ehrhart, T., Ferret, O., Teyssou, D., Troncy, R., Tannier, X., 2019. Searching news articles using an event knowledge graph leveraged by wikidata, in: Amer-Yahia, S., Mahdian, M., Goel, A., Houben, G., Lerman, K., McAuley, J.J., Baeza-Yates, R., Zia, L. (Eds.), Companion of The 2019 World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019, ACM. pp. 1232–1239.

[24] Schneider, R., Oberhauser, T., Klatt, T., Gers, F.A., Löser, A., 2017. Analysing errors of open information extraction systems, in: Proceedings of the First Workshop on Building Linguistically Generalizable NLP Systems, Association for Computational Linguistics, Copenhagen, Denmark. pp. 11–18. URL: https://aclanthology.org/W17-5402, doi:10.18653/v1/W17-5402.

[25] Stanovsky, G., Michael, J., Zettlemoyer, L., Dagan, I., 2018. Supervised open information extraction, in: North American Chapter of the Association for Computational Linguistics.

[26] Stewart, M., Enkhsaikhan, M., Liu, W., 2019. Icdm 2019 knowledge graph contest: Team uwa. 2019 IEEE International Conference on Data Mining (ICDM) , 1546–1551.

[27] Stewart, M., Liu, W., 2020. Seq2kg: An end-to-end neural model for domain agnostic knowledge graph (not text graph) construction from text, in: International Conference on Principles of Knowledge Representation and Reasoning.

[28] Wang, C., Ma, X., Chen, J., Chen, J., 2018. Information extraction and knowledge graph construction from geoscience literature. Comput. Geosci. 112, 112–120.

[29] Wu, X., Wu, J., Fu, X., Li, J., Zhou, P., Jiang, X., 2019a. Automatic knowledge graph construction: A report on the 2019 icdm/icbk contest, in: 2019 IEEE International Conference on Data Mining (ICDM), IEEE. pp. 1540–1545.

[30] Wu, X., Wu, J., Fu, X., Li, J., Zhou, P., Jiang, X., 2019b. Automatic knowledge graph construction: A report on the 2019 icdm/icbk contest. 2019 IEEE International Conference on Data Mining (ICDM) , 1540–1545.