Knowledge and Information Systems (2026) 68:47
https://doi.org/10.1007/s10115-025-02631-y

RESEARCH

Text2AMR2FRED, converting text into RDF/OWL knowledge graphs via abstract meaning representation

Aldo Gangemi[1] · Arianna Graciotti[2] · Antonello Meloni[3] · Andrea Giovanni Nuzzolese[4] · Valentina Presutti[2,4] · Diego Reforgiato Recupero[3] · Alessandro Russo[4]

Received: 5 December 2024 / Revised: 17 October 2025 / Accepted: 27 October 2025

Abstract
Converting natural language text into structured, logically coherent knowledge graphs (KGs) enhances the ability to retrieve, organize, and analyze vast amounts of information at scale. This paper introduces Text2AMR2FRED, a text-to-KG pipeline that converts multilingual natural language text into logically coherent, interoperable KGs. Designed to support large-scale information retrieval and knowledge extraction, this pipeline addresses key limitations of existing semantic parsers and machine readers, including issues with logical consistency and interoperability. By adhering to Semantic Web standards, Text2AMR2FRED systematically structures text-based information and enhances it through integration with external knowledge sources, delivering enriched, semantically sound KGs ready for diverse applications. We obtain the output KGs by leveraging Abstract Meaning Representation (AMR) as an intermediate semantic parsing formalism, exploiting the progress achieved by text-to-AMR parsers employing pre-trained language models. We produce a manually validated

✉ Arianna Graciotti
arianna.graciotti@unibo.it

Aldo Gangemi
aldo.gangemi@unibo.it

Antonello Meloni
antonello.meloni@unica.it

Andrea Giovanni Nuzzolese
andrea.nuzzolese@istc.cnr.it

Valentina Presutti
valentina.presutti@unibo.it

Diego Reforgiato Recupero
diego.reforgiato@unica.it

Alessandro Russo
alessandro.russo@istc.cnr.it

1 Department of Philosophy, University of Bologna, Bologna, Italy

2 Department of Modern Languages, Literatures, and Cultures, University of Bologna, Bologna, Italy

3 Mathematics and Computer Science Department, University of Cagliari, Cagliari, Italy

4 Institute for Cognitive Sciences and Technologies, National Research Council (CNR-ISTC), Rome, Italy

Published online: 06 January 2026

KGs bank created by transforming a dataset of natural language sentences into KGs using Text2AMR2FRED and applying an intrinsic evaluation method that leverages Open Knowledge Extraction motifs.

Keywords Knowledge graphs · Abstract meaning representation · Natural language processing · Semantic frames

1 Introduction

Converting natural language text into structured, logically coherent knowledge graphs (KGs) enhances the ability to retrieve, organize, and analyze vast amounts of information at scale [1]. By transforming unstructured text into a network of interconnected entities and relationships, KGs enable more efficient and accurate information retrieval, facilitate advanced semantic queries, and support deeper insights from large text corpora [2]. This process bridges the gap between human language and machine-readable data, making it a critical tool for natural language processing (NLP), AI-driven applications, and decision-making systems across various domains. The NLP and Semantic Web (SW) communities have invested substantial efforts in developing text-to-KG pipelines [3–5]. The NLP community, in particular, has leveraged advancements in machine learning (ML) and neural networks (NNs) to enhance semantic parsing and understanding. By incorporating these sophisticated models, NLP has significantly improved the extraction of structured knowledge from unstructured text. Meanwhile, the SW community has focused on formalizing the representation of knowledge and ensuring its interoperability across different domains. The integration of these fields is driving innovation in automated knowledge extraction, paving the way for more intelligent and context-aware systems. Graph-based semantic parsing has garnered significant attention, largely due to its potential to provide general-purpose, interpretable representations of meaning, such as abstract meaning representation (AMR) [6]. AMR, in particular, offers a graph-structured format that captures the semantics of entire sentences by abstracting away from surface linguistic variations, enabling more uniform and flexible knowledge extraction. This approach allows for deeper semantic understanding and alignment across languages and domains, making it a powerful tool for tasks ranging from machine translation to information extraction and reasoning within KGs [6]. Text-to-AMR transduction, leveraging neural machine translation and sequence-to-sequence (seq2seq) models, has demonstrated promising results in various domains [7]. Notable advancements include English-centric AMR parsing models, such as SPRING [8] and transition-based AMR parsing [9], as well as multilingual [10] and multi-formalism approaches like SGL [11]. Despite these successes, neural semantic parsers face significant challenges when it comes to making the extracted knowledge interoperable and practically usable. A key obstacle is the fragmentation of different formalisms, often referred to as “balkanization” [12], which hinders the integration of diverse semantic representations. Additionally, the lack of rigorous logical consistency in neural models further complicates the exploitation of the extracted knowledge in downstream tasks, limiting its applicability in more structured and logic-dependent systems.

The SW provides mechanisms to formally represent extracted knowledge using interoperable ontologies, facilitating knowledge augmentation with heterogeneous knowledge bases (KBs) and enabling alignment with other ontological frameworks. This ensures that knowledge from diverse sources can be integrated, reused, and extended in a coherent and standardized manner. Tools like the SW machine reader FRED [5] take advantage of these standards by formally interpreting the extracted information as logical axioms using ontology design patterns [13], and encoding them with SW protocols such as RDF and OWL. The resulting KGs allow for the exploration, retrieval, and reasoning over facts extracted from heterogeneous text corpora through structured queries (e.g., SPARQL). Furthermore, by aligning these KGs to other ontologies and knowledge resources, it becomes possible to enrich and expand the KB, supporting applications such as data interoperability, knowledge discovery, and advanced semantic querying across multiple domains. This alignment supports the discovery of explicit knowledge that would otherwise remain hidden within unstructured texts, revealing deeper insights and facilitating advanced reasoning. However, FRED's reliance on complex NLP pipelines presents significant challenges. These pipelines are often cumbersome, difficult to maintain, and lack flexibility, making them unsuitable for scaling in multilingual and diverse linguistic contexts.

To address these limitations, this paper introduces Text2AMR2FRED, a new architecture for transforming natural language into OWL-compliant RDF KGs. Unlike the original FRED pipeline, which directly converted text to RDF using rule-based NLP components, Text2AMR2FRED adopts a two-step approach: it first uses state-of-the-art, end-to-end neural AMR parsers to generate abstract semantic graphs, and then applies a custom transformation layer that maps AMR structures to RDF triples, enriched with semantic properties and ontological links. While inspired by the design and semantics of FRED, our approach does not pass AMR graphs directly into FRED. Instead, it implements tailored post-processing rules to produce FRED-style structured representations from the output of AMR parsers.

Although AMR and RDF are both graph-based representations, they differ fundamentally in their purposes and semantics. AMR is a surface-level semantic representation aimed at human interpretability, whereas RDF, especially when combined with OWL, offers a formal semantics grounded in model theory. A naive transformation of AMR into RDF triples would simply reify the graph structure without enabling formal reasoning or interoperability with other KBs.

Our approach goes beyond such syntactic conversion. It provides a formal model-theoretic interpretation of AMR by mapping it into a structured RDF/OWL ontology. This enables logical inference, semantic integration, and compatibility with linked data principles, making AMR content usable in knowledge-driven applications.

Therefore, this architecture mitigates the typical error propagation found in component-based NLP pipelines and benefits from AMR's abstraction over lexical and syntactic variation, enabling more generalized and robust semantic representations without relying on data augmentation strategies such as lexical substitution [14, 15]. The result is a more scalable and consistent pipeline across different domains and text types.

Our approach introduces multiple knowledge modeling advancements that go beyond traditional NLP pipelines. Specifically:

- High-quality RDF KG construction from text is achieved via the combination of AMR parsing and a custom transformation layer that maps AMR structures to RDF triples, inspired by the design and semantics of FRED, which is rooted in cognitively founded frame semantics and ontology design patterns (ODPs).
- Explicit linking to external and public semantic resources (e.g., DBpedia, WordNet, FrameNet, and DOLCE-based ontologies) ensures interoperability and reuse, which are key principles in knowledge engineering.
- Support for automated reasoning and entailment, made possible by the formal nature of the generated graphs and their alignment with SW standards (e.g., OWL and RDFS).

- A novel role of KGs as grounding structures for generative AI, as discussed in [16], where such graphs are used to guide or constrain language generation, a shift from traditional use, concentrating on querying or inference.

Furthermore, Text2AMR2FRED benefits from the multilingual capabilities of modern AMR parsers, allowing our architecture to support multiple languages and enabling cross-linguistic knowledge extraction, an important step beyond the original FRED system, which was limited to English. While our method is designed to be language-agnostic, enabling users to apply it to different languages, assuming a suitable AMR parser is available, direct comparisons in multilingual settings are currently infeasible. This is due to the lack of gold-standard benchmarks and the fact that no other publicly available systems, to the best of our knowledge, perform the same task of converting natural language text into OWL-compliant RDF KGs with comparable semantic depth and ontological grounding. Existing alternatives either lack the semantic expressivity of FRED or do not target RDF generation. As such, we limited our comparison to FRED, the most relevant baseline in terms of design and intended output, even though it relies on a different pipeline. Despite these limitations, our system's extensible architecture encourages experimentation across languages and facilitates the integration of multilingual resources into the SW.

In detail, our contributions in this paper are the following:

- We present Text2AMR2FRED, a pipeline that transforms natural language into RDF/OWL KGs by combining neural AMR parsers with a custom AMR-to-RDF transformation layer that models semantic relations inspired by FRED;
- We support multiple languages, submitting text to neural parsers such as SPRING (for English) and USeA [17] (for other languages). These generate AMR graphs grounded in PropBank frames[1] and formalized in the Framester [19] linguistic KG;
- We enrich the resulting KGs via post-processing techniques, aligning AMR-derived RDF triples to external KBs like DBpedia,[2] Wikidata,[3] and VerbAtlas.[4] Word Sense Disambiguation (WSD) is performed using EWISER,[5] linking AMR nodes to WordNet synsets and semantically typed OWL properties. All reused resources are either formalized in Framester or mapped to it;
- We present a publicly accessible web application[6] and corresponding APIs. Users can input text, view AMR graphs, and convert them to enriched RDF/OWL, with options to customize outputs (e.g., enable/disable enhanced entity linking with the specialized model BLINK [20], select serialization formats);
- We introduce an intrinsic evaluation methodology for KGs produced by Text2AMR2FRED, using Open Knowledge Extraction (OKE) motifs [13], elementary graph patterns that reflect standard logical constructs in the SW. This allows systematic comparison with outputs from FRED and other tools;
- We release a manually validated KG bank,[7] built from natural language sentences processed by Text2AMR2FRED and annotated using our evaluation method. This dataset provides a valuable benchmark for future work on KG construction and assessment.

1 https://propbank.github.io/ [18].

2 https://www.dbpedia.org/

3 https://www.wikidata.org/wiki/Wikidata:Main_Page.

4 https://verbatlas.org/.

5 https://github.com/SapienzaNLP/ewiser.

6 https://arco.istc.cnr.it/txt-amr-fred/.

7 https://github.com/infovillasimius/amr2Fred/blob/master/text2amr2fred_result_evaluation.zip.

The remainder of this paper is structured as follows: Section 2 reviews the related work on machine reading systems and research leveraging AMR. Section 3 details the architecture of the proposed tool, outlining its components and explaining the system's workflow. In Sect. 4, we present the evaluation process, describing the dataset we annotated, the systems we tested, and the results we obtained, including an in-depth error analysis of the tested tools. Finally, Sect. 5 concludes the paper, summarizing key findings and discussing potential future directions for this research.

2 Related work

Machine reading [21] is a general paradigm of NLP focused on the automatic, unsupervised understanding of text. Machine reading encompasses a number of solutions which are often classified as open information extraction (OIE) [22] systems. The main goal of these solutions is to extract frequent triplet patterns from extensive shallow parsing of textual corpora, to build a large KB of triplets made up of text chunks. Early approaches to machine reading based on OIE began with the Open Mind Common Sense project [23] and continued with Never-Ending Language Learning [24] (NELL). The Open Mind Common Sense project implemented a crowdsourcing and games-with-a-purpose tool to build a large, informal KB of facts represented in triplet-based natural language. Instead, NELL is a learning tool that has been processing the Web since 2010 to build a continuously evolving structured ontology of identified entities and predicates from the acquired facts. Notable works that hybridize SW with NLP are mainly focused on ontology learning and population (OL&P) task [25, 26]. Examples of such methods can be found in [27–30]. Most of these approaches are built on machine learning techniques, making them typically data-intensive, requiring large corpora to learn rules for automatic ontology construction. These rules are established through a training process, which can be time-consuming. Other approaches to OL&P use either lexico-syntactic patterns [31] or hybrid lexical-logical techniques [32]. Instead, FRED [5] is a state-of-the-art formal machine reader that produces RDF graphs from text, which are (i) domain- and task-independent, and (ii) designed according to the frame semantics [33] and ontology design patterns [34].

Deep learning has significantly improved machine reading, specifically in the creation of KGs. As an example, this trend is fairly evident by the number of works published in workshop series like "Deep Learning for Knowledge Graphs" [35] or "Deep Learning meets Ontologies and Natural Language Processing" [36]. Deep learning has entered the space of KG generation from text through specialized models that focus on individual tasks, such as NER [37], EL [38], and RE [39]. Examples of similar systems are OpenNRE [40] for neural relation extraction with limited Wikidata alignment, DeepKE [41] for NER, RE and attribute extraction, and UIE [42] for unified text-to-structure generation across a broader range of IE tasks (entities, relations, events, and sentiment extraction). These systems employ supervised learning approaches: OpenNRE requires fine-tuning on labeled relation types for predefined binary relations, DeepKE provides knowledge base population capabilities with predefined schemas, while UIE handles entities, relations, events, and sentiment through schema-based prompting. Semantic parsing strategies offer approaches that overcome the limitation of using separate systems for individual tasks by providing unified semantic representations that encompass broader recognition and disambiguation of entities and concepts of a sentence, as well as their mutual relationships.

Among those strategies, Text-to-AMR (i.e., Abstract Meaning Representation) semantic parsing has proven well-suited for a wide range of engineering applications, as reviewed in the recent survey by [7]. In particular, Text-to-AMR parsers based on neural machine translation and sequence-to-sequence (seq2seq) models achieved excellent results both in scenarios limited to English and AMR parsing, i.e., SPRING, transition-based AMR parsing [9], and in multilingual [10] and multi-formalisms scenarios, i.e., SGL [11]. Text-to-AMR parsing systems can be employed in text-to-KG generation scenarios with no further supervision. However, semantic parsing output—AMR being no exception—is usually informal. It does not comply with SW knowledge representation standards. Consequently, machine readers for the SW pursue the objective of leveraging SotA semantic parsers while aligning their output to knowledge representation best practices in OWL-compliant RDF KGs [43].

In recent years, the landscape of KG construction has undergone significant shifts with the advent of LLMs and transformer-based approaches. For example, in the realm of KG completion, Yao et al. [44] introduced KG-BERT, a model that leverages pre-trained language models for predicting missing links in KGs. This approach treats triples as textual sequences, enabling the application of BERT for KG tasks. Instead, the authors in [45] provide a comprehensive survey on generative information extraction using LLMs, highlighting their capabilities in transforming unstructured text into structured knowledge representations. Similarly, in [46], the authors discuss the unification of LLMs and KGs, presenting frameworks where LLMs enhance KGs and vice versa. Compared to these recent approaches, Text2AMR2FRED offers a robust alternative that maintains formal semantics throughout the pipeline. It achieves this by leveraging AMR graphs as intermediate semantic structures and translating them into OWL-compliant RDF KGs aligned with resources like PropBank, WordNet, DBpedia, and DOLCE. This design enables formal reasoning, semantic querying, and ontology alignment—capabilities largely absent in LLM- or transformer-based approaches.

Additionally, our approach aligns with and complements the emerging field of logic-augmented and neuro-symbolic generation. The authors in [16] demonstrate the use of RDF graphs generated by FRED as grounding inputs for LLMs, opening pathways for hybrid neuro-symbolic AI. Building on this, the authors in [47] propose logic-augmented generation (LAG), a framework for injecting formal semantics into neural text generation tasks. Text2AMR2FRED shares this vision by producing formally structured, ontology-aligned KGs that can serve a similar role as input to neuro-symbolic pipelines. However, it extends this approach by offering a multilingual, AMR-based pipeline that improves scalability, logical consistency, and semantic richness, thereby strengthening the foundation for downstream tasks such as logic-augmented generation.

From a general perspective, AMR, which stands at the core of Text2AMR2FRED, offers distinct advantages over OIE for KG creation. Automatic KG creation strategies based on OIE, particularly when driven by LLMs, yield promising results with in-context learning. However, they present additional challenges: similar approaches typically struggle to generate KB-grounded relations and entities [48], necessitating post hoc canonicalization efforts. The advantages of AMR-based approaches derive from the fundamental difference between AMR and OIE in their operational approach. OIE systems identify text segments expressing specific user-provided relations (e.g., "cause for" or "located in"), whereas AMR extracts all semantic relations (encoded as PropBank frames) expressed in the text. These approaches differ significantly in their level of abstraction: OIE achieves, in principle, maximal abstraction from syntactic patterns by focusing on user-specified semantic relations regardless of surface realization, while AMR maintains closer ties to the lexical and syntactic structures of the original text, but grounds them in standardized PropBank frames.

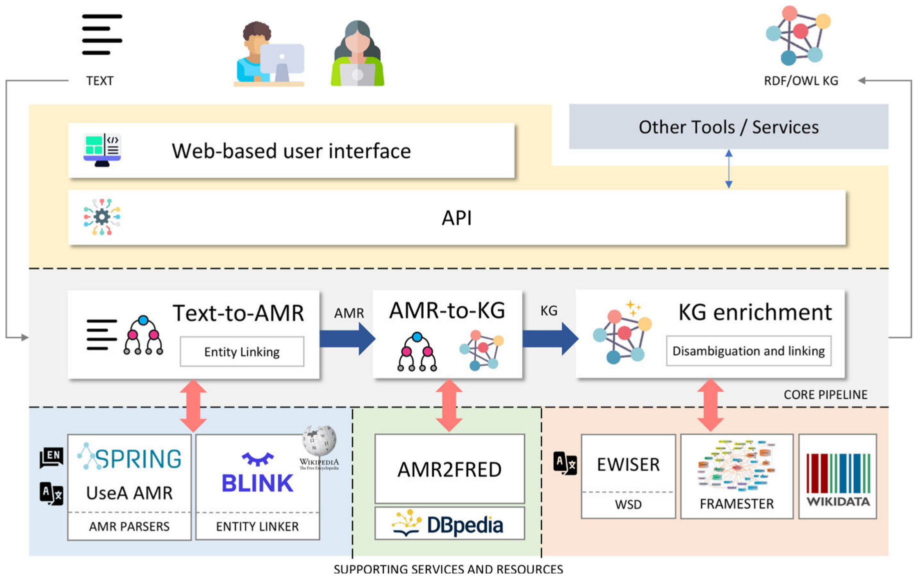


Fig. 1 Architectural model and pipeline of Text2AMR2FRED

3 Architecture of Text2AMR2FRED

Our approach to transforming natural language text into semantic RDF/OWL KGs relies on and implements the architectural framework and pipeline shown in Fig. 1. This reference pipeline combines the capabilities and outcomes of different tools and NLP tasks, as well as chaining transformation and enrichment steps that produce a KG from a given input text.

In brief, the KG generation process from natural language text can be summarized in three major stages, described in detail in the rest of this section: (i) the parsing of text into an AMR graph, relying on a text-to-AMR parser; (ii) the conversion of the AMR graph into an RDF/OWL KG, in line with FRED's knowledge representation patterns as implemented by the AMR2FRED tool; (iii) the semantic enrichment of the KG, by post-processing the RDF/OWL KG with the support of the Framester ontological resource.

The key processing stages of our pipeline are also summarized in the TEXTTOAMRTOFRED function, presented in Listing 1. The three core steps (lines 2–4) are subsequently described in the corresponding subsections (see Listings 2, 3 and 4).

The capabilities of our Text2AMR2FRED tool implementing this pipeline are made available through an API,[8] which offers programmatic access to the generation of AMR and RDF graphs from natural language text and allows integrating the KG generation process into other applications and workflows. This makes it possible for tools such as the Machine Reading Suite[9] to exploit the text-to-KG capabilities and generate RDF-named graphs from input text sentences or paragraphs in batches. On top of the API, a web interface[10] provides a user-friendly, interactive environment for users to input natural language text and visualize the

8 https://arco.istc.cnr.it/txt-amr-fred/api/docs.

9 https://github.com/anuzzolese/machine-reading.

10 https://arco.istc.cnr.it/txt-amr-fred.

Listing 1 TextToAmrToFred – Build knowledge graph from text

Input:

text | text to be processed
amrParser | text-to-AMR parser (`SPRING` or `USeA`)
kgFormat | format of the resulting KG (`TURTLE`, `RDF/XML`, `N-TRIPLES`, or `GRAPHICAL`)

Output:

kg | Knowledge graph built from *text* in the requested *kgFormat*

```
function TextToAmrToFred(text, amrParser, kgFormat)
    amr ← TextToAmr(text, amrParser)          ▷ produce AMR graph
    kg ← AmrToRdf(amr)                         ▷ produce KG
    kg ← EnrichKG(text, kg)                    ▷ enrich KG
    if kgFormat = GRAPHICAL then
        kg ← RdfToGraph(kg)                    ▷ build visual SVG graph
    end if
    return kg
end function
```

resulting KGs, as well as define various settings to configure the pipeline, such as choosing the AMR parser and the preferred output format and serialization of the RDF/OWL KG.

3.1 Text2AMR

As a first fundamental step in our pipeline, the input text is parsed into an AMR graph using a text-to-AMR parser. This step aims at capturing and representing the meaning of the text in a structured, graph-based format. The resulting symbolic representation abstracts away from surface-level details of the input text (like word order and syntactic or grammatical variations) to focus on the core concepts and relationships between entities, events, actions, and attributes. Core concepts and relations are represented in AMR graphs as PropBank rolesets (roughly equivalent to *frames* in frame semantics), and their semantic roles as PropBank argument (e.g., *agent*, *patient*, or *instrument*). In our implementation of the text-to-AMR module, sentences in English are parsed into AMR graphs using the SPRING[11] parser [8]. In contrast, for sentences in other languages, we rely on the AMR parser of the USeA[12] framework [49], which supports 100 languages.

Both SPRING and USeA are built on state-of-the-art sequence-to-sequence large language models (LLMs) based on the Transformer architecture. In particular, SPRING leverages BART, a pre-trained encoder–decoder transformer fine-tuned for AMR parsing and generation, while USeA integrates a multilingual extension of SPRING alongside XLM-RoBERTa for its WSD and SRL modules. Consequently, the underlying architectures, training data, and computational requirements of these models are not under our direct control. While SPRING and USeA represent the current state of the art in AMR parsing, their limitations inevitably influence the performance of our Text2AMR2FRED pipeline. Parsing errors may arise from incorrect or incomplete role assignment, fragmented AMR structures, or mislinked named entities, which in turn propagate to the RDF/OWL graph. For multilingual input, USeA's performance varies across languages depending on training data availability, leading to less accurate AMRs for low-resource languages. Moreover, entity linking remains a challenging aspect: Even with BLINK integration, mistakes in disambiguation can reduce semantic precision in the generated knowledge graphs. Consequently, the overall quality of

[11] http://nlp.uniroma1.it/spring/.

[12] https://github.com/SapienzaNLP/usea.

```
(z0 / unveil-01
    :ARG0 (z1 / company
            :wiki "Apple_Inc."
            :name (z2 / name
                      :op1 "Apple"))
    :ARG1 (z3 / watch
            :ARG1-of (z4 / revolutionary-04)))
```

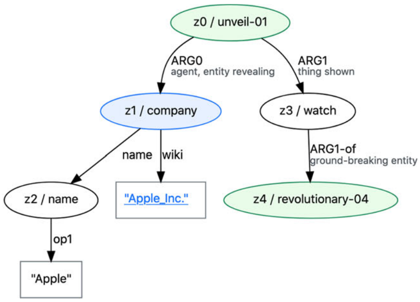


Fig. 2 AMR graph (PENMAN notation, left; graphical representation, right) resulting from the text2AMR parsing of the sentence *"Apple unveils a revolutionary watch"*

Text2AMR2FRED is bounded by the accuracy of upstream AMR parsers. Nevertheless, our framework is parser-agnostic: as more accurate AMR parsers become available, they can be seamlessly integrated into our pipeline, directly improving downstream knowledge graph quality.

The Text-to-AMR module manages named entity recognition, classification, and linking. AMR annotation guidelines specify how to encode named entities from natural language text into AMR graphs.[13] These guidelines indicate to canonicalize named entities using Wikipedia as the standard reference. Consequently, AMR gold-standard datasets contain entity linking information, enabling seq2seq AMR parsers used in our framework to recognize named entities according to AMR annotation guidelines and associate them to Wikipedia pages. However, entity linking is frequently inaccurate in practice, as seq2seq approaches alone often struggle with this task. Therefore, following the approach adopted for SPRING and summarized in [8], we integrate the BLINK Entity Linker[14] into the pipeline to enhance entity linking for English text.

This process is illustrated in Listing 2, where the TextToAmr function (lines 1–7) demonstrates how input text is parsed into an AMR graph using a selected text-to-AMR parser. For English text (when using the SPRING parser), entity linking in the resulting AMR graph is further enriched. The key steps for improving entity linking are shown in the EnrichEntityLinking function (lines 8–19), which implements the "wikification" approach used in SPRING.[15] The function processes the input text and its corresponding AMR graph by first identifying all nodes representing named entities—those containing both `:name` and `:wiki` attributes. For each such node, if the entity name appears in the original text, the BLINK entity linker disambiguates the mention and retrieves the most likely corresponding Wikipedia entity (LinkEntityWithBlink). When a linked entity is found, the `:wiki` value of the node in the AMR graph is updated accordingly (SetWiki). The result is an enriched AMR graph with more accurate Wikipedia references.

As a simple running example, we consider the sentence *"Apple unveils a revolutionary watch"*. Figure 2 shows the corresponding AMR graph in PENMAN notation[16] and using a graphical representation as provided by the Text2AMR2FRED web interface. The example highlights the frame-centric nature of AMR graphs, characterized by the occurrences of

13 https://github.com/amrisi/amr-guidelines/blob/master/amr.md#named-entities.

14 https://github.com/facebookresearch/BLINK.

15 https://github.com/SapienzaNLP/spring/blob/main/bin/blinkify.py.

16 https://penman.readthedocs.io/en/latest/notation.html.

Listing 2 TextToAmr – Build AMR graph from text with text-to-AMR parser

Input:
text | text to be processed
amrParser | text-to-AMR parser (`SPRING` or `USeA`)
Output:
amr | AMR graph in PENMAN notation built from *text*

```
function TextToAmr(text, amrParser)
    amr ← amrParser.parse(text)                        ▷ produce AMR graph
    if amrParser = SPRING then                         ▷ English text
        amr ← EnrichEntityLinking(text, amr)           ▷ wikification via Entity Linking
    end if
    return amr
end function
```

Input:
text | text to be processed
amr | AMR graph in PENMAN notation built from *text*
Output:
amr | AMR graph with enhanced entity linking (wikification)

```
function EnrichEntityLinking(text, amr)
    namedEntities ← GetNamedEntities(amr)              ▷ AMR nodes with :wiki and :name
    for all entity in namedEntities do
        if entity.name appears in text then
            wikiEntity ← LinkEntityWithBLink(name, text)
            if wikiEntity ≠ None then
                SetWiki(amr, entity, wikiEntity)
            end if
        end if
    end for
    return amr
end function
```

`unveil-01`,[17] the meaning of the verb *unveil*, and `revolutionary-04`[18] as the meaning of the adjective *revolutionary*, represented as PropBank rolesets and their associated arguments/roles (`ARG0`, `ARG1`), as well as the presence of a named entity (the Apple company) linked to the corresponding Wikipedia page. Yet, the meaning of some entities is not fully specified: for example, the variable `z3` denotes something that is a (specifically, is an instance of) `watch`, but the sense or semantics of what a `watch` is is not defined (via entity linking or any other semantic alignment or disambiguation). This aspect is further discussed in Section 3.3.

The output of the text-to-AMR module is thus an AMR graph in PENMAN notation that can be directly used as input for the AMR2FRED tool to produce the corresponding RDF/OWL KG, as detailed in the next section.

3.2 AMR2FRED

AMR2FRED [50] is a software tool designed to transform AMR into RDF, leveraging the knowledge extraction and logical and ontological representation patterns of the FRED

[17] https://propbank.github.io/v3.4.0/frames/unveil#unveil.01.

[18] https://propbank.github.io/v3.4.0/frames/revolt#revolutionary.04.

machine reading system [5], and the Framester KG [19] as a public ontological resource to disambiguate the formal semantics of concepts and relations.

To better support the output of recent AMR parsers such as SPRING and USeA, which generate frame-based AMRs grounded in PropBank, the original AMR2FRED tool was extensively refactored to interpret PropBank roles and aligned with current AMR annotation standards. Additionally, it was revised to improve resilience to edge cases and to enhance the semantic quality and ontological precision of the resulting RDF/OWL graphs.

AMR2FRED processes valid AMR strings in PENMAN notation, producing RDF/OWL triples in multiple serialization formats, including RDF/XML, Turtle, and N-Triples. This guarantees a formal interpretation and logic-based services to check consistency and perform complex inferences of the extracted graphs. Additionally, it offers graphical representations of the output graph as SVG and PNG images, according to user preferences. AMR2FRED is available as both a web application and an API,[19] with comprehensive API documentation provided. The tool is open-source, and its codebase is publicly accessible via its GitHub repository.[20]

The modular, component-based architecture of AMR2FRED is depicted in Fig. 3. The Web Module provides a graphical interface for users, relying on the API to return the AMR translation in the format selected by the user. The API Module sends the AMR in PENMAN format to the AMR2FRED Main Module, orchestrating the interaction between all other modules. The AMR2FRED Main Module forwards the AMR text to the AMR Parser Module, which parses the input and returns a graph where nodes and relations are labeled with a conventional set of prefixes. The AMR Parser interacts with the PropBank Module to translate PropBank frames and roles. Once the graph is parsed, the Main Module sends it to the RDF Writer Module for serialization in the user-specified format. The RDF writing process relies on Apache Jena,[21] an open-source Java framework for building SW and Linked Data applications. If the user selects a graphical format, the Main Module sends the graph to the Digraph Writer Module, which generates the graphical output. This module relies on Graphviz,[22] an open-source graph visualization software, to produce visual representations of the KG in SVG or PNG format.

The conversion pipeline that produces an RDF/OWL KG from AMR consists of several key phases: translation, mapping, processing, and the integration of best practices from the SW and Linked Data paradigms.

The first step of the conversion involves parsing the AMR graph and translating it into an acyclic graph. This is accomplished by creating a node for each AMR variable and connecting them using the specified relations. In the example of Fig. 2, the variables from `z0` to `z4` represent the nodes, while the relations include `/`, `:ARG0`, `:ARG1`, `:ARG1-of`, `:name`, `:wiki` and `:op1`. A special treatment is applied to the `/` relation, which indicates the *instance of* relation, and is interpreted as a `rdf:type` relation (the $\in$ membership in description logics (DL)). This relation appears only once for each variable, and its corresponding value is substituted for all occurrences of that variable throughout the graph. For example, all references to the variable `z3` are replaced with the instance name `watch`, followed by an underscore and a progressive number. If `z3` is referenced in the AMR graph, it will be translated as `watch_1` for all occurrences of `z3`. If a new variable, such as `v`, refers to another instance

19 http://framester.istc.cnr.it/amr-2-fred.

20 https://github.com/infovillasimius/amr2Fred.

21 https://jena.apache.org/.

22 https://graphviz.org/.

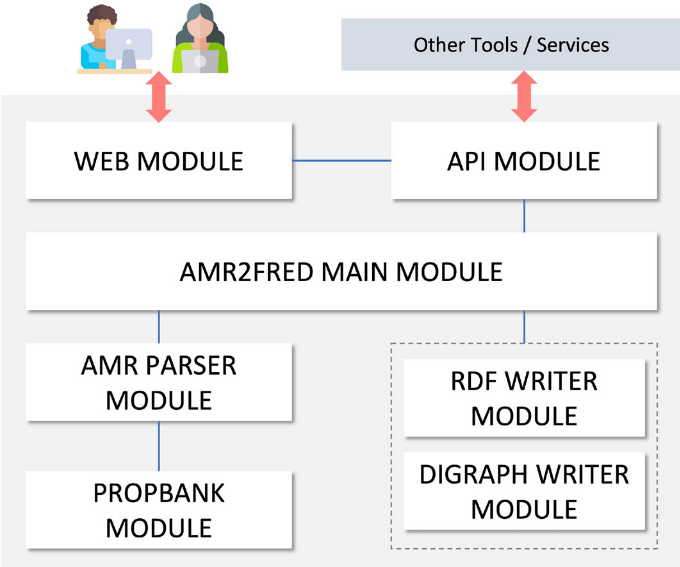


Fig. 3 AMR2FRED architecture

of `watch`, it will be translated as `watch_2`, and so on. This ensures consistency in how instance references are represented in the output.

Additionally, inverse relations such as `:ARG1-of` are not directly processed as they are. To handle these cases, if the parent of the inverse relation is not the root of the graph, a new node representing the subject of the relation is introduced, and the relation is reversed to maintain a coherent structure. However, if the parent is the root of the graph, the nodes are simply swapped, reversing the relation without introducing additional nodes.

Nodes like `unveil-01` represent PropBank rolesets, and connected relations of type `ARG`*n* are translated into corresponding PropBank local roles using a predefined conversion table. This table includes all possible cases, ensuring that each `ARG`*n* relation is accurately mapped to its appropriate role within the roleset, facilitating a consistent and semantically correct conversion process. The local roles and mapping rely on the RDF/OWL representation of PropBank and AMR rolesets and roles, available as part of the Framester resource. In our example, the core role `ARG0` for the `unveil-01` roleset refers to the agent, i.e., the entity revealing something, and targets the variable `z1` representing a `company`. According to the predefined conversion table, derived from the RDF representation of the `unveil-01` roleset,[23] the `:ARG0` role is translated into the local role `pblr:unveil-01.agent`. Similarly, the relation `:ARG1` is translated into the local role `unveil-01.thing-shown`, signifying that the `watch` (denoted by variable `z3`) is the thing being unveiled in the context of the `unveil-01` event. In the semantics of description logics, this knowledge is formalized as: `unveil_1` ∈ `unveil-01` and `unveil_1 unveil-01.thing-shown watch_1`.

Other non-core roles, such as `:name` in the given example, along with their associated nodes, are processed through a comprehensive translation pipeline consisting of 15 distinct steps. This pipeline is designed to handle complex relationships and currently incorporates over 200 condition checks to ensure precise processing. At each stage, various conditions are checked to account for all known edge cases and structural nuances, ensuring that both

23 https://w3id.org/framester/data/propbank-3.4.0/RoleSet/unveil-01.

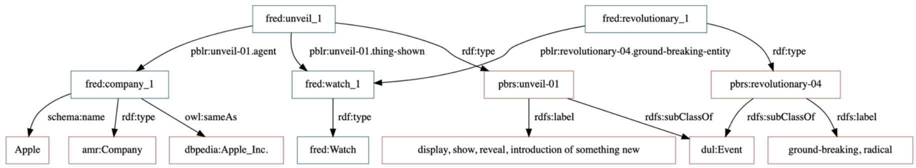


Fig. 4 Graphical representation of the core of the KG produced by AMR2FRED for the sentence *"Apple unveils a revolutionary watch"*

nodes and relations are accurately translated in logically valid OWL semantics. This step-by-step approach preserves semantic integrity and guarantees alignment with the overall graph structure, ensuring that the meaning embedded in the AMR graph is fully reflected in the resulting RDF/OWL KG.

The core logic of the AMR-to-RDF transformation is encapsulated in the function `AmrToRdf(amr)`, which coordinates the end-to-end translation process from a PENMAN-formatted AMR graph to an RDF serialization or visual output. This function orchestrates a sequence of operations, beginning with parsing and graph construction via `ParseAndTranslate` and proceeding through graph validation, semantic enrichment, and output generation. Within this process, key transformations are delegated to specialized functions, such as `SemanticTranslation`, which applies structural and semantic adjustments to align with OWL and Linked Data principles. These include the verification of dominance structures, handling of modifiers, processing of list elements and inverse relations, and elaboration of typed instances. A high-level overview of this procedure is presented in Listing 3. A detailed specification of all the specialized functions invoked in the pseudocode would result in an overly long listing, so only the top-level logic is presented here for clarity.

For transparency and reproducibility, interested readers can inspect the complete implementation in the code (`parse` method).[24]

Figure 4 shows the core of the KG produced by AMR2FRED for our running example. Variables in the AMR graph representing instances of PropBank rolesets become entities in the KG, typed with the corresponding roleset (e.g., `fred:unveil_1 rdf:type pbrs:unveil-01`) and related to other entities with the proper roles. Adhering to the principles of the SW and Linked Data paradigm, the named entity representing the Apple company is linked (with the `owl:sameAs` property) to the corresponding DBpedia entity, based on wikification in the AMR graph. The resulting KG is then further processed, as detailed in the next section, to augment the axiomatization of entities and classes.

3.3 KG post-processing and semantic enrichment

The KG produced by AMR2FRED is further processed through a semantic enrichment step. This enhances the KG by linking it to a broader set of concepts and knowledge sources, improving its semantic expressivity and interoperability with other datasets and KGs. The post-processing stage, whose goal is to refine the semantic content of the KG by disambiguating entities and enriching it with links to authoritative KBs, involves two key operations: (1) linking named entities in the KG to the corresponding Wikidata entities; and (2) disambiguating underspecified concepts in the graph and linking them to WordNet synsets (as formalized

[24] Java version: https://github.com/infovillasimius/amr2Fred/blob/master/amr2Fred/src/amr2fred/Parser.java. Python version: https://github.com/infovillasimius/py_amr2fred/blob/main/py_amr2fred/parser.py.

Listing 3 AMRTORDF – Translate AMR graph to RDF

Input:
amr | AMR graph in PENMAN format
Output:
RDF serialization or visualization string

```
function AMRTORDF(amr)
    if amr is empty or too short then
        return "Syntax error"
    end if
    root ← PARSEANDTRANSLATE(amr)
    if root = null then
        return "Syntax error"
    end if
    if graph status indicates unresolved nodes then
        CHECKGRAPH(root)
    end if
    if RDF output mode is active then
        return WRITERDF(root)
    else
        return VISUALIZEGRAPH(root)
    end if
end function
```

```
function PARSEANDTRANSLATE(amr)
    root ← BUILDGRAPHFROMAMR(amr)
    if infinite recursion is detected then
        return CREATEERRORNODE("Recursive")
    end if
    root ← ADDMISSINGINSTANCES(root)
    root ← SPLITMULTISENTENCES(root)
    root ← SEMANTICTRANSLATION(root)
    root ← DISAMBIGUATEVERBS(root)
    root ← INSERTTOPICIFMISSING(root)
    root ← FIXRESIDUALERRORS(root)
    root ← INTEGRATELOGICTRIPLES(root)
    return root
end function
```

```
function SEMANTICTRANSLATION(root)
    if root is empty then
        return null
    end if
    MARKDECLAREDVARIABLES(root)
    root ← VERIFYDOMINANCESTRUCTURE(root)
    root ← CHECKOPSROLES(root)
    root ← VERIFYBULLETLISTS(root)
    root ← APPLYINVERSERELATIONS(root)
    root ← VERIFYMODIFIERS(root)
    root ← PROCESSCHILDREN(root)
    root ← ADDPARENTLINKS(root)
    root ← ELABORATEINSTANCES(root)
    return root
end function
```

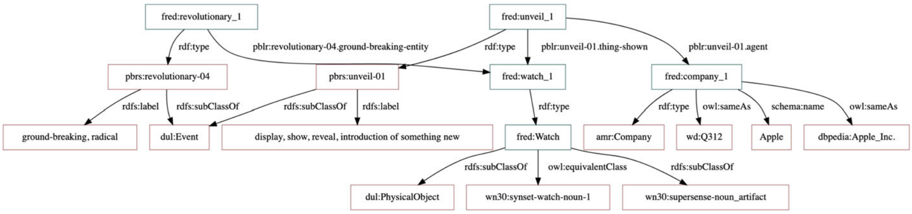


Fig. 5 Graphical representation of the core of the final KG produced by Text2AMR2FRED for the sentence *"Apple unveils a revolutionary watch"*

in Framester) and to DOLCE+DnS Ultralite[25] (DUL) and DOLCE-Zero[26] (d0) foundational classes. These two operations are outlined in the EnrichKG function in Listing 4.

The linking of named entities to Wikidata (summarized in the LinkToWikidata function) is eased by the presence in the KG of links to DBpedia entities, as produced by AMR2FRED exploiting the alignments to Wikipedia in the AMR graph. Entities in the KG having a link to DBpedia are retrieved (GetEntitiesDBpedia, line 7 in Listing 4) from the KG via a SPARQL query. For each entity with a link to DBpedia, a `owl:sameAs` link to the corresponding Wikidata entity (if any) is added (lines 8–13 in Listing 4). This step is performed with the help of the `wikimapper` Python package,[27] able to map Wikipedia pages to Wikidata IDs.

Figure 5 shows the core of the final KG produced for our running example. The `fred:company_1` entity linked to `dbpedia:Apple_Inc.` after the AMR2FRED conversion is also linked to the corresponding entity in Wikidata (`wd:Q312`) after the post-processing stage.

An additional enrichment step (summarized in the Disambiguate function in Listing 4) has the goal of producing semantic alignments for the entities in the KG resulting from the KG generation process (i.e., with the `fred:` or user-provided prefix/namespace) that are not yet typed (i.e., are not instances of any class) nor linked to external entities (via `owl:sameAs` predicates).

The KG is first queried via SPARQL to retrieve the entities that match the aforementioned criteria (usually classes originating from nouns and adjectives in the input text) and thus require disambiguation (line 17). Next, disambiguation with respect to WordNet synsets relies on a WSD step performed over the initial input text. This step disambiguates textual spans in the input text, producing a set of candidate terms/lemmas mapped to WordNet synsets (line 18). For each entity to be disambiguated, if its local name[28] corresponds to the lemma of a disambiguated textual span in the input text, an `owl:equivalentClass` link is added between the entity and the WordNet sysnet identified via WSD (lines 20–26).

WSD over the input text is performed relying on EWISER,[29] a WSD system well-suited for multilingual scenarios due to its state-of-the-art performance in both all-words English and multilingual WSD tasks. EWISER builds on transformer-based LLMs (e.g., BERT), which are fine-tuned for sense prediction, achieving state-of-the-art performance in both all-words English and multilingual WSD tasks.

25 http://www.ontologydesignpatterns.org/ont/dul/DUL.owl.

26 http://www.ontologydesignpatterns.org/ont/d0.owl.

27 https://github.com/jcklie/wikimapper.

28 the part in the URI after the prefix.

29 https://github.com/SapienzaNLP/ewiser.

Table 1 Motifs used in Text2AMR2FRED evaluation

Motif	Type	Description	Description logic
IDENTITY	Edge	Represents that two different individuals i_1 and i_2 refer to the exact same entity, asserting that the subjects are interchangeable in any context	$i_1 = i_2$
TYPE	Edge	Indicates that a particular entity i_1 is an instance of a specified class C	$i_1 \in C$
SUBCLASS	Edge	Specifies that a class C is a subclass of another class D, meaning all instances of the subclass are also instances of the superclass	$C \sqsubseteq D$
EQUIVALENCE	Edge	Asserts that two classes C and S are equivalent, meaning they contain exactly the same set of instances	$C \equiv D$
ROLE	Edge	Defines the role R of an entity x of type T within an event or situation e_1 of type E, specifying how the entity participates in that context	$R(e_1, x)$ $(E \sqsubseteq \exists \mathbf{R}.T)$
PROPERTY	Edge	Represents a binary fact or domain property assertion, relating two entities through a property, and excludes roles, modality, and negation	$P(e_1, e_2)$
MODALITY	Edge	Applies a modality mark to an event or situation, indicating the mode or manner in which the event or situation occurs	$M(e_1, m)$
NEGATION	Edge	Applies a negation mark to an event or situation, indicating that the event or situation did not occur or is not true	$N(e_1, bool)$

The RDF representation of WordNet synsets is part of the Framester KG. By exploiting Framester and the interlinking between linguistic and ontological resources it provides, disambiguated entities can be further enriched with links (using the `rdfs:subclassOf` property) to WordNet "supersenses" and DUL+d0 foundational classes. This alignment step (line 27) is carried out through a SPARQL CONSTRUCT query over the Framester KG to produce the additional triples that semantically enrich the disambiguated entities. More generally, the links to resources in Framester (from PropBank rolesets to WordNet synsets and DUL+d0 classes) act as entry points to the rich, multifaceted knowledge represented therein.

Concerning automated reasoning, both automated inference and logical consistency and coherence of the extracted models can be achieved by adding an off-the-shelf OWL reasoner at the end of the process. Consider that the ontologies (WordNet, Propbank) used to disambiguate word senses do not have any disjointness axiom, then no inconsistency may be generated. However, Text2AMR2FRED aligns classes and instances to public RDF resources that are on their turn aligned to Framester and the DOLCE-Zero OWL foundational ontology, which contain full-fledged disjointness axioms for a rigorous consistency and coherence verification.

In the example of Fig. 5, the `fred:Watch` class is semantically aligned to the corresponding WordNet synset denoting "a small portable timepiece" (`wn30:synset-watch-noun-1`[30]), as well as further qualified as a subclass of the *artifact* WordNet super-sense

[30] https://w3id.org/framester/wn/wn30/instances/synset-watch-noun-1.

Listing 4 ENRICHKG – Enrich the KG with links to Wikidata and WSD

```
Input:
    text | text to be processed
    kg | RDF/OWL knowledge graph built from text
Output:
    kg | RDF/OWL knowledge graph enriched with links to Wikidata and WSD

function ENRICHKG(text, kg)
    kg ← LINKTOWIKIDATA(kg)          ▷ add links to Wikidata for named entities
    kg ← DISAMBIGUATE(text, kg)      ▷ perform WSD and link to Framester
    return kg
end function
```

```
Input:
    kg | RDF/OWL knowledge graph
Output:
    kg | RDF/OWL knowledge with links to Wikidata

function LINKTOWIKIDATA(kg)
    entitiesDbp ← GETENTITIESDBPEDIA(kg)          ▷ entities with owl:sameAs to DBpedia
    for all entity in entitiesDbp do
        wikidataId ← DBPEDIATOWIKIDATA(entity.dbpediaUri)          ▷ wikimapper
        if wikidataId ≠ NONE then
            ADDSAMEASLINK(kg, entity.uri, wikidataId)          ▷ owl:sameAs to Wikidata
        end if
    end for
    return kg
end function
```

```
Input:
    text | text to be processed
    kg | RDF/OWL knowledge graph built from text
Output:
    kg | RDF/OWL knowledge with WSD to WordNet and links to Framester

function DISAMBIGUATE(text, kg)
    entities ← GETENTITIESTODISAMBIGUATE(kg)          ▷ query kg via SPARQL
    lemmaToSynset ← WSDTOWORDNET(text)          ▷ WSD wrt WordNet
    entityToWordnetSynMap ← ∅
    for all entity in entities do
        if entity.name in lemmaToSynset then
            wnSynUri ← lemmaToSynset[entity.name]          ▷ Synset for lemma
            ADDEQUIVALENCELINK(kg, entity.uri, wnSynUri)          ▷ Link to WordNet
            entityToWordnetSynMap[entity.uri] ← wnSynUri
        end if
    end for
    ALIGNTOWNDUL(kg, entityToWordnetSynMap)          ▷ CONSTRUCT query with Framester
    return kg
end function
```

(`wn30:supersense-noun_artifact`[31]) and of the `dul:PhysicalObject` class in DUL. `rdfs:subClassOf` is equivalent to the $\sqsubseteq$ relation in description logics.

[31] https://w3id.org/framester/wn/wn30/instances/supersense-noun_artifact.

4 Evaluation

Evaluating formal knowledge extraction is a resource-intensive task, as each extracted triple must be carefully assessed, often by multiple annotators. Even a short sentence of about 10 words can yield 30–50 RDF triples, each requiring manual inspection for correctness. Consequently, we opted for a dataset of short sentences that enabled a controlled and fine-grained evaluation while keeping the annotation workload feasible. Nonetheless, our tool is fully capable of processing longer texts, such as paragraphs or full news articles. The underlying AMR parser handles long-form input by splitting it into sentence-level parses, grouped under a top-level multi-sentence node. As a result, the triple generation and evaluation procedure remains structurally consistent regardless of input length, since each sentence is parsed and translated individually. Performance differences would more likely emerge in tasks involving cross-sentence knowledge, where relations span multiple sentences and require discourse-level inference. While such advanced reasoning is not the primary focus of our current evaluation, our approach lays the groundwork for future integration of discourse-aware relation extraction, as discussed in the related work (e.g., the role of grounding graphs in generative AI presented in [16]).

Therefore, in this section, we explain the method we designed and followed to perform an intrinsic evaluation of the KGs output by Text2AMR2FRED. We start this section introducing Open Knowledge Extraction (OKE) [51] *motifs* [13], elementary graph patterns corresponding to basic logical patterns used in SW. *Motifs* constitute the building blocks of our evaluation method and permit a cross-tool evaluation, allowing us to compare the quality of Text2AMR2FRED and FRED output KGs. In subsection 4.2, we describe the dataset of sentences that we transformed into KGs for the evaluation. In subsection 4.3, we recall the specifics of the systems under test (Text2AMR2FRED and FRED). In subsection 4.4, we outline the manual assessment process we applied to validate each triple. Based on the triple assessment, we could calculate the overall and per-motif accuracy for the two tools. In subsection 4.5, we report the outcomes of the comparative evaluation of the tools. In subsection 4.6, we report a detailed error analysis.

4.1 Motif-based evaluation

Text2AMR2FRED output KGs have been evaluated using a method that revolves around OKE motifs. These motifs represent the fundamental structural elements of RDF/OWL graphs for formal knowledge extraction from text. In this evaluation, as shown in Table 1, we focused on "edge motifs", which consist of single triple motifs. The edge motifs employed include `IDENTITY`, which asserts that two individuals refer to the same entity, as in `fred:person_1 owl:sameAs wiki:Q3427977`; `TYPE`, indicating that an entity is an instance of a class, as in `fred:opine_1 rdf:type pbrs:opine-01`; `SUBCLASS`, specifying class hierarchies, as in `fred:Soccer rdfs:subClassOf d0:Activity`; `EQUIVALENCE`, for class equivalence, as in `fred:Musician owl:equivalentClass wn30instances:synset-musician-noun-1`; `ROLE`, to define entity roles within events or situations, as in `fred:write_1 pblr:write-01.thing-written fred:document_1`; `PROPERTY`, to relate entities through binary assertions, as in (`fred:technology_1 dul:hasQuality fred:Various`. Additionally, we included `MODALITY` for mode indications, as in `fred:situation_2 boxing:hasModality boxing:Possible` and `NEGATION`, as in `fred:pretend_1 boxing:hasTruthValue boxing:False`, for annotating negated events or situations.

4.2 Dataset

To evaluate Text2AMR2FRED, we utilized a collection of previously assembled sentences for assessing FRED, the earlier version of the same system. This dataset, described in [13], comprises multi-domain content: sentences extracted from Wikipedia definitions, news, scientific reports and social media posts. Such a selection of diversified sources aims to demonstrate the tool's suitability for OKE.

To ensure consistency comparison with FRED, we transformed the same 41 sentences assessed in its evaluation campaign, described in [13], into KGs through the Machine Reading suite, which allows to process batches of text excerpts through the Text2AMR2FRED API and outputs RDF-named graphs (in NQuad format). Each named graph identifies a sentence with a proper URI and lists all the extracted triples for that sentence.

4.3 Systems tested

We comparatively tested two tools: Text2AMR2FRED, the system described in this paper, and FRED, the previous version of the same tool. FRED is a machine reader for the SW that integrates and represents multiple semantic parsing results, including frame detection, semantic role labeling, entity linking, and taxonomy induction. Applications include semantic sentiment analysis, relation augmentation, and discovery.

4.4 Procedure

We collected all triples included in the KGs output by the two tools. These triples were mapped to the motifs in Table 1. An expert annotator assessed their correctness one by one. Triples not mapped to a motif were excluded from the evaluation.

Also, a distinction was made between *functional/semantic* triples and *enrichment/alignment* triples. The former represents the core knowledge transformed into KG by the tool, such as PropBank frames with their arguments and expansions, nouns and adjectives aligned to WordNet synsets, and named entities recognized, typified, and linked to Wikipedia, DBpedia, and Wikidata. The latter are, for example, alignments between PropBank and VerbAtlas. Only the functional/semantic triples were mapped to motifs and evaluated, while enrichment/alignment triples were considered out of scope for the evaluation.

4.5 Results

Text2AMR2FRED consistently demonstrated higher accuracy across all the motifs, achieving 95% accuracy over $1,598$ triples, while FRED achieved 78% overall accuracy over $1,075$ triples.

Text2AMR2FRED produced a higher number of triples mapped to `ROLE`, `SUBCLASS`, `TYPE` and `IDENTITY` motifs as compared to FRED. In all of them, it achieved higher accuracy. FRED produced a higher number of triples mapped to `EQUIVALENCE`, `IDENTITY` and `PROPERTY` as compared to Text2AMR2FRED. In all of them, it achieved a lower accuracy.

Table 2 Comparison of Text2AMR2FRED and FRED accuracy per each different *motif*

Motif	Text2AMR2FRED #triples	Accuracy	FRED #triples	Accuracy
EQUIVALENCE	130	0.99*	162	0.61
IDENTITY	82	0.90	50	0.88
MODALITY	–	–	8	0.87
NEGATION	7	1.00	7	0.87
PROPERTY	163	0.90*	293	0.79
ROLE	312	0.92*	150	0.83
SUBCLASS	479	0.99*	263	0.82
TYPE	425	0.97*	142	0.77
All	1598	0.95*	1075	0.78

* indicates statistical significance ($p < 0.05$)

Noticeably, FRED produces more triples mapped to `PROPERTY` motif, while Text2AMR2FRED produces more triples mapped to `ROLE` motif. The reason for that is that Text2AMR2FRED employs, as a semantic parsing layer, an AMR parser. AMR formalism adheres to a "generalized framing", where most semantic structures in a sentence are represented as predicates with their roles and argument structures. This approach leads to more triples mapped to the `ROLE` motif. In contrast, FRED does not use AMR-based parsing and relies on different semantic parsing techniques, resulting, instead, in more triples mapped to the `PROPERTY` motif.

Table 2 shows a comparison between Text2AMR2FRED and FRED accuracy per each different motifs. We conducted Chi-square tests to determine statistical significance across all motif categories. The Chi-square analysis confirms that Text2AMR2FRED demonstrates a statistically significant improvement in overall accuracy compared to FRED (95% vs. 78%, χ^2 = 176.90, p < 0.0001). More specifically, statistically significant improvements were found in 5 out of the 7 motifs for which the comparison of the two systems was possible:

- EQUIVALENCE (99% vs. 61%, χ^2 = 59.03, p < 0.0001)
- PROPERTY (90% vs. 79%, χ^2 = 8.72, p = 0.0031)
- ROLE (92% vs. 83%, χ^2 = 8.04, p = 0.0046)
- SUBCLASS (99% vs. 82%, χ^2 = 71.21, p < 0.0001)
- TYPE (97% vs. 77%, χ^2 = 55.47, p < 0.0001)

The non-significant differences in IDENTITY (90% vs. 88%, p = 0.9086) and NEGATION (100% vs. 87%, p = 1.0000) motifs are likely attributable to the relatively simple nature of these motifs, where FRED already performed well. Additionally, the small sample size for NEGATION (only 7 triples in each system) limits the statistical power to detect significant differences.

We acknowledge that part of the improvement of Text2AMR2FRED may be attributed to the fact that the Text2AMR parsers, such as SPRING and USeA, are more recent and benefit from advanced neural architectures with access to large, high-quality training datasets. We need to consider that the original FRED extractor uses a hybrid parser (CCG+Boxer), which may fail to recognize many patterns that are more easily addressed by a transformer-based parser. Nonetheless, the transparency and explainability of CCG+Boxer are not available with the current AMR parsers, and it is also notable that the performance of the old CCG+Boxer parser is still substantial [52].

Table 3 Example of incorrect triples extracted by Text2AMR2FRED from the sentence *Sculpture is the branch of the visual arts that operates in three dimensions and one of the plastic arts*

Motif	Subject	Predicate	Object	Error explanation
EQUIVALENCE	fred:Branch	owl:equivalentClass	wn30_instances:synset-branch-noun-1	Here, "branch" refers to a subdivision of an art discipline
SUBCLASS	fred:Branch	rdfs:subClassOf	dul:Organization	Branch is not a kind of organization in this sentence

Moreover, the formalization/alignment layer of Text2AMR2FRED makes these improvements visible in a formal way, so that we can propose a modular and extensible pipeline that benefits from the intermediate AMR representation. Indeed, the better performance is not solely due to the AMR parsers but also to:

- The generalization capabilities of AMR, which serve as an abstraction that reduces the dimensionality of syntactic patterns and favors KG interoperability.
- The AMR-to-FRED post-processing, which, as with FRED, enables logical formalization, enrichment, and disambiguation.
- The integration of external resources like WordNet, Framester, DOLCE-Zero, etc. that, as with FRED, provide additional semantic features and interoperability.

Additionally, we note that the AMR-to-RDF step has been specifically optimized to work with AMRs produced by SPRING and USeA, which are based on PropBank frames. This optimization ensures that the mappings from AMR to RDF are more effective and semantically rich, as PropBank frames provide well-defined semantic roles (now integrated in the Framester formal KG of lexical knowledge).

Finally, the MODALITY motif is only indirectly supported in Text2AMR2FRED and does not appear in the evaluation because it is not directly comparable to FRED's implementation. In fact, the AMR-driven frame detection applies also to modal operators, making them modal frames on top of factual frames. For example, an *ought to* modality might be mapped to an occurrence of the `pbrs:recommend-01` frame.

4.6 Error analysis

To evaluate Text2AMR2FRED's limitations, we conducted an error analysis on the incorrect triples it extracted. We report in this subsection some examples.

One notable example, reported in Table 3, comes from the KG generated from the sentence *"Sculpture is the branch of the visual arts that operates in three dimensions and one of the plastic arts."*. Text2AMR2FRED pipeline wrongly disambiguated the word *branch*, leading to erroneous triples for the motifs `EQUIVALENCE` and `SUBCLASS`. For the same sentence, under the `EQUIVALENCE` motif, Text2AMR2FRED incorrectly linked `fred:Branch` to the WordNet synset for *branch* `synset-branch-noun-1`, which refers to a division within an organization. In this context, however, *branch* refers specifically to a subdivision of the visual arts. In particular, this failure can be traced back to an erroneous disambiguation performed by EWISER, the WSD component of Text2AMR2FRED. Secondly, under the `SUBCLASS` motif, Text2AMR2FRED produced a triple that incorrectly maps

Table 4 Example of incorrect triples extracted by Text2AMR2FRED from the sentence *The architecture of Saxon at this time was to interpret the DOM of the source stylesheet directly instead of building an expression tree*

Motif	Subject	Predicate	Object	Error explanation
`IDENTITY`	fred:country_1	owl:sameAs	wiki:Q1202	Refer to Saxony instead of Saxon XSLT
`TYPE`	fred:country_1	rdf:type	amr:Country	Saxon in this sentence is not a country
`PROPERTY`	fred:architecture_1	dul:hasQuality	fred:country_1	Saxon in this sentence is not a country

Table 5 Example of incorrect triples extracted by Text2AMR2FRED from the sentence *I'm sorry Brentwood I strongly dislike soccer but that's my opinion.*

Motif	Subject	Predicate	Object	Error explanation
`ROLE`	fred:say_1	pblr:say-01.hearer	free:city_district_1	The *hearer* of the predicate `say-01` is not a city district. It is a person

`fred:Branch` to a subclass of `dul:Organization`, suggesting that a *branch* is a type of organization. This wrong classification stems from the erroneous WSD of *branch*.

To further assess the system's limitations, in Table 4, we analyzed other incorrect triples extracted from the sentence: *"The architecture of Saxon at this time was to interpret the DOM of the source stylesheet directly instead of building an expression tree."* The tool misinterpreted the term *Saxon*, leading to several erroneous triples. Under the `IDENTITY` motif, the system generated a triple which incorrectly refers the node `fred:country_1` to *Saxon* as a country, linking it via `owl:sameAs` to `wiki:Q1202`, which represents Saxony. In the context of the sentence, however, *Saxon* refers to *Saxon XSLT*, a software tool for processing XML and XSLT. Contextually, under the `TYPE` motif, the system produced the triple `fred:country_1 rdf:type amr:Country`, which erroneously states that the named entity *Saxon* refers to the AMR type `country`. The erroneous named entity classification and linking arise from SPRING, the AMR parser used in Text2AMR2FRED, and BLINK, the entity linker employed by SPRING, which incorrectly links Saxon to the Wikipedia entry for Saxony. Similarly, under the `PROPERTY` motif, the system mistakenly produced the triple `fred:architecture_1 dul:hasQuality fred:country1_1`. This error reiterates the wrong caused by the improper named entity recognition, classification and linking of *Saxon*, as the tool failed to recognize that in this context, *Saxon* is not a country but a software application.

Further examining the system's limitations, in Table 5, we analyzed incorrect triples extracted from the sentence: *"I'm sorry Brentwood I strongly dislike soccer but that's my opinion."* The tool misinterpreted the term *Brentwood*, leading to erroneous triples due to issues in named entity recognition and linking. Under the `ROLE` motif, the system generated a triple which incorrectly identifies `free:city_district_1` as the *hearer* (ARG0) of the PropBank predicate `say-01`. The named entity *Brentwood* was erroneously classified as a city district, likely due to its common association with a geographical location. However, in the context of the sentence, *Brentwood* is being addressed as a person, not a place. The error, which again originates from the named entity recognition, classification, and linking components performed by SPRING and BLINK, led to the incorrect assignment of the *hearer* role to a city district, which is semantically inappropriate for the given sentence.

5 Conclusion and future work

Text2AMR2FRED addresses several key limitations in prior knowledge extraction and ontology learning approaches:

- **Low abstraction and poor generalization**: Traditional systems like open information extraction and early ontology learning and population methods often extract shallow, surface-level triples. Text2AMR2FRED, by combining deep AMR parsing with cognitively grounded frame semantics from FRED, produces more abstract and semantically rich representations that generalize better across domains.
- **Lack of pragmatic adequacy:** Prior systems struggled to represent implicit or inferred knowledge. Our approach enables the extraction of pragmatically meaningful structures (e.g., events, roles, quantifiers), which align more closely with human interpretation and facilitate complex reasoning and entailment.
- **Fragmented or unlinked output:** Many earlier pipelines produced isolated or poorly linked triples. Text2AMR2FRED ensures semantic grounding and interlinking by integrating public knowledge resources (e.g., DBpedia, FrameNet), thus improving interoperability and supporting multilingual, cross-domain use.

By integrating domain-specific ontologies, it enhances the informativeness of KGs, supporting complex queries and revealing implicit knowledge within text. This approach also facilitates the enrichment of output KGs with external KBs. We have shown its pipeline, whose capabilities are available through a publicly accessible API. The evaluation we have carried out includes OKE motifs, which permit a cross-tool evaluation allowing us to compare the quality of Text2AMR2FRED and FRED output KGs. The dataset we have used includes a collection of previously assembled sentences for assessing FRED. After collecting the triples included in the KGs output by the two tools, they were mapped to the motifs, and their correctness was annotated. The results showed that Text2AMR2FRED consistently demonstrated higher accuracy across all motifs, achieving 95% accuracy over 1,598 triples, while FRED achieved an overall accuracy of 78% over 1,075 triples.

The Text2AMR2FRED pipeline has demonstrated considerable uptake and impact across multiple research domains, including NLP, SW technologies, and machine reading comprehension. A recent survey [7] highlights how Text2AMR2FRED advances research by enabling seamless integration of AMR graphs into question-answering models. It enables the transformation of AMR representations into RDF format, thereby facilitating direct deployment in knowledge-based QA pipelines. This capability has been successfully leveraged in the UNIQORN framework [53] for unified question answering over both RDF KGs and natural language text.

Comparative evaluations have validated the system's effectiveness through studies examining distinct approaches to free-form question answering [54]. These assessments revealed that while ChatGPT exhibits popularity bias through diminished performance on low-frequency entity questions, the structured knowledge representations generated by

Text2AMR2FRED demonstrate greater robustness against such biases. The pipeline's ability to encode knowledge across diverse entity frequency distributions ensures more balanced performance for both common and rare entities.

The pipeline has also proven valuable in digital humanities applications [55, 56], where it has been employed to transform cultural heritage textual collections into KGs accessible via public SPARQL endpoints.[32] These resources have enabled digital humanities scholars, historians, and musicologists to develop compelling visual narratives using data.[33]

Additionally, it supports neuro-symbolic integration efforts focused on achieving grounded multimodal, knowledge-augmented meaning representations [16, 57], further expanding its research applications.

As future work, we aim to establish a standardized methodology for text-to-KG construction, addressing a critical gap that currently hinders consistent evaluation and comparison across different approaches. This objective directly targets the most significant limitation of our study: the narrow scope of our comparative evaluation, which presently encompasses only FRED, the foundational system from which Text2AMR2FRED evolved. To broaden our evaluation framework, we propose developing a comprehensive graph-based Retrieval-Augmented Generation (RAG) system that would enable fair comparison between Text2AMR2FRED and LLM-based pipelines. This approach would involve integrating diverse structured resources into an LLM architecture, with training focused either on AMR-to-FRED patterns (for AMR-mediated cases) or directly on FRED structural motifs (for direct transformation approaches). While designing such a pipeline presents considerable complexity and requires extensive parameter tuning and data integration, it offers a promising research direction, setting the path to continue the comparison of knowledge extraction tools for the SW initiated by the seminal study presented in [58]. Additionally, we plan to explore the potential of employing LLMs for post-processing outputs from other knowledge extraction tools, including OpenNRE, DeepKE, and UIE, as discussed in our related work section. This approach could enhance the quality and consistency of knowledge graph construction across different extraction methodologies.

Additionally, we plan to enhance the entity recognition capabilities of our system and expand the evaluation dataset to better assess the generalizability and effectiveness of the proposed approach.

We also plan to analyze the contribution of individual components in the pipeline through targeted ablation studies, focusing on semantic enrichment, ontology alignment, and the robustness of the AMR parsing process under controlled degradations.

Acknowledgements This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement for the Polifonia (a digital harmoniser for musical heritage knowledge) project (grant agreement No 101004746) and from FAIR – Future Artificial Intelligence Research Foundation as part of the grant agreement MUR n. 341, code PE00000013 CUP 53C22003630006.

Author Contributions Aldo Gangemi did conceptualization, formal analysis, data curation, methodology, supervision, writing—review, editing, supervision, funding acquisition; Arianna Graciotti done data curation, methodology, software, resources, validation, writing—original draft, and writing—review and editing; Antonello Meloni contributed to data curation, methodology, software, resources, validation, writing—original draft, and writing—review and editing; Andrea Giovanni Nuzzolese done software, resources, writing—original draft, and writing—review and editing; Valentina Presutti was involved in conceptualization, formal analysis, supervision, funding acquisition, and writing—review and editing; Diego Reforgiato Recupero did conceptualization, formal analysis, supervision, funding acquisition, writing—review and editing, and

[32] https://polifonia.disi.unibo.it/MusicBO/sparql.

[33] https://projects.dharc.unibo.it/melody/MusicBO/music_in_bologna_knowledge_graph_overview.

project administration; Alessandro Russo done software, resources, visualization, writing—original draft, and writing—review and editing.

Funding Open access funding provided by Alma Mater Studiorum - Università di Bologna within the CRUI-CARE Agreement.

Data Availability All the codes for our software and experiments are publicly available on GitHub in the various repositories referenced in the paper (https://github.com/anuzzolese/machine-reading; https://github.com/infovillasimius/amr2Fred). The main contribution, Text2AMR2FRED, is available online via APIs (https://arco.istc.cnr.it/txt-amr-fred/api/docs) and a WebApp (https://arco.istc.cnr.it/txt-amr-fred).

Declarations

Conflict of interest The authors declare no conflict of interest.

References

1. Baradaran R, Ghiasi R, Amirkhani H (2020) A survey on machine reading comprehension systems. Nat Lang Eng 28:683–732
2. Hogan A, Blomqvist E, Cochez M, D'amato C, Melo G, Gutierrez C, Kirrane S, Gayo J, Navigli R, Neumaier S, Ngomo A, Polleres A, Rashid S, Rula A, Schmelzeisen L, Sequeda J, Staab S, Zimmermann A (2021) Knowledge graphs. ACM Comput Surv 54(4):1–37. https://doi.org/10.1145/3447772. (**Place: New York, NY, USA Publisher: Association for Computing Machinery**)
3. Jaradeh MY, Singh K, Stocker M, Both A, Auer S (2023) Information extraction pipelines for knowledge graphs. Knowl Inf Syst 65(5):1989–2016. https://doi.org/10.1007/s10115-022-01826-x
4. Dessì D, Osborne F, Recupero DR, Buscaldi D, Motta E (2022) SCICERO: a deep learning and NLP approach for generating scientific knowledge graphs in the computer science domain. Knowl. Based Syst. 258:109945. https://doi.org/10.1016/J.KNOSYS.2022.109945
5. Gangemi A, Presutti V, Recupero DR, Nuzzolese AG, Draicchio F, Mongiovì M (2017) Semantic web machine reading with FRED. Semantic Web 8(6):873–893. https://doi.org/10.3233/SW-160240
6. Banarescu L, Bonial C, Cai S, Georgescu M, Griffitt K, Hermjakob U, Knight K, Koehn P, Palmer M, Schneider N (2013) Abstract meaning representation for sembanking. In: Proc. of the 7th linguistic annotation workshop and interoperability with discourse, ACL, Sofia, Bulgaria, pp 178–186. https://aclanthology.org/W13-2322
7. Wein S, Opitz J (2024) A survey of AMR applications. In: Al-Onaizan, Y., Bansal, M., Chen, Y.-N. (eds.) Proceedings of the 2024 conference on empirical methods in natural language processing, Association for Computational Linguistics, Miami, Florida, USA, pp 6856–6875. https://doi.org/10.18653/v1/2024.emnlp-main.390
8. Bevilacqua M, Blloshmi R, Navigli R (2021) One SPRING to rule them both: symmetric AMR semantic parsing and generation without a complex pipeline. Proc. AAAI Conf Artif Intell 35(14):12564–12573
9. Zhou J, Naseem T, Fernandez Astudillo R, Lee Y-S, Florian R, Roukos S(2021) Structure-aware fine-tuning of sequence-to-sequence transformers for transition-based AMR parsing. In: Proc. of the 2021 conference on empirical methods in natural language processing, ACL, Online and Punta Cana, Dominican Republic, pp 6279–6290. https://doi.org/10.18653/v1/2021.emnlp-main.507
10. Blloshmi R, Tripodi R, Navigli R (2020) XL-AMR: enabling cross-lingual AMR parsing with transfer learning techniques. In: Proc. of the 2020 conference on empirical methods in natural language processing (EMNLP), pp 2487–2500. ACL, Online. https://doi.org/10.18653/v1/2020.emnlp-main.195

11. Procopio L, Tripodi R, Navigli R (2021) SGL: speaking the graph languages of semantic parsing via multilingual translation. In: Proc. of the 2021 conference of the North American chapter of the association for computational linguistics: human language technologies, pp 325–337. ACL, Online. https://doi.org/10.18653/v1/2021.naacl-main.30
12. Oepen S, Abend O, Abzianidze L, Bos J, Hajic J, Hershcovich D, Li B, O'Gorman T, Xue N, Zeman D (2020) MRP 2020: the second shared task on cross-framework and cross-lingual meaning representation parsing. In: Proc. of the CoNLL 2020 shared task: cross-framework meaning representation parsing, pp 1–22. ACL, Online. https://doi.org/10.18653/v1/2020.conll-shared.1
13. Gangemi A, Recupero DR, Mongiovì M, Nuzzolese AG, Presutti V (2016) Identifying motifs for evaluating open knowledge extraction on the web. Knowl Based Syst 108:33–41. https://doi.org/10.1016/j.knosys.2016.05.023. (**New Avenues in Knowledge Bases for Natural Language Processing**)
14. Lacerra C, Tripodi R, Navigli R (2021) Genesis: a generative approach to substitutes in context. In: Proc. of the 2021 conference on empirical methods in natural language processing, pp 10810–10823. ACL, Online and Punta Cana, Dominican Republic. https://doi.org/10.18653/v1/2021.emnlp-main.844
15. Lacerra C, Pasini T, Tripodi R, Navigli R (2021) Alasca: an automated approach for large-scale lexical substitution. In: Zhou, Z.-H. (ed.) Proceedings of the thirtieth international joint conference on artificial intelligence, IJCAI-21, pp 3836–3842. International joint conferences on artificial intelligence organization, Montreal, Canada. https://doi.org/10.24963/ijcai.2021/528 . Main Track
16. De Giorgis S, Gangemi A, Russo A (2025) Neurosymbolic graph enrichment for grounded world models. Inf Process Manag 62(4):104127. https://doi.org/10.1016/j.ipm.2025.104127
17. Orlando R, Conia S, Faralli S, Navigli R (2022) Universal semantic annotator: the first unified API for WSD, SRL and semantic parsing. In: Calzolari, N., Béchet, F., Blache, P., Choukri, K., Cieri, C., Declerck, T., Goggi, S., Isahara, H., Maegaard, B., Mariani, J., Mazo, H., Odijk, J., Piperidis, S. (eds.) Proceedings of the thirteenth language resources and evaluation conference, pp 2634–2641. European Language Resources Association, Marseille, France. https://aclanthology.org/2022.lrec-1.282/
18. Pradhan S, Bonn J, Myers S, Conger K, O'gorman T, Gung J, Wright-bettner K, Palmer M (2022) PropBank comes of Age—Larger, smarter, and more diverse. In: Nastase, V., Pavlick, E., Pilehvar, M.T., Camacho-Collados, J., Raganato, A. (eds.) Proceedings of the 11th joint conference on lexical and computational semantics, pp 278–288. Association for Computational Linguistics, Seattle, Washington. https://doi.org/10.18653/v1/2022.starsem-1.24
19. Gangemi A, Alam M, Asprino L, Presutti V, Recupero DR (2016) Framester: a wide coverage linguistic linked data hub. In: EKAW 2016, Springer, Bologna, Italy, pp 239–254
20. Wu L, Petroni F, Josifoski M, Riedel S, Zettlemoyer L (2020) Scalable zero-shot entity linking with dense entity retrieval. In: Webber, B., Cohn, T., He, Y., Liu, Y. (eds.) Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP), pp 6397–6407. Association for Computational Linguistics, Online. https://doi.org/10.18653/v1/2020.emnlp-main.519
21. Etzioni O, Banko M, Cafarella MJ (2006) Machine reading. AAAI 6:1517–1519
22. Etzioni O, Banko M, Soderland S, Weld DS (2008) Open information extraction from the web. Commun ACM 51(12):68–74
23. Singh P, et al (2002) The public acquisition of commonsense knowledge. In: Proceedings of AAAI Spring Symposium: Acquiring (and Using) Linguistic (and World) Knowledge for Information Access, vol 3
24. Mitchell T, Cohen W, Hruschka E, Talukdar P, Yang B, Betteridge J, Carlson A, Dalvi B, Gardner M, Kisiel B et al (2018) Never-ending learning. Commun ACM 61(5):103–115
25. Cimiano P (2010) Ontology learning and population from text: algorithms, evaluation and applications vol. 27. Springer, Berlin/Heidelberg, Germany. https://dl.acm.org/doi/10.5555/1942117
26. Al-Aswadi FN, Chan HY, Gan KH (2020) Automatic ontology construction from text: a review from shallow to deep learning trend. Artif Intell Rev 53(6):3901–3928. https://doi.org/10.1007/s10462-019-09782-9
27. Cimiano P, Völker J (2005) A framework for ontology learning and data-driven change discovery. In: Proceedings of the 10th international conference on applications of natural language to information systems (NLDB), Springer, pp 227–238. https://doi.org/10.1007/11428817_21
28. Witte R, Khamis N, Rilling J (2010) Flexible ontology population from text: The OwlExporter. In: Calzolari, N., Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., Rosner, M., Tapias, D. (eds.) Proceedings of the seventh international conference on language resources and evaluation (LREC'10). European Language Resources Association (ELRA), Valletta, Malta. https://aclanthology.org/L10-1633/
29. Tanev H, Magnini B (2006) Weakly supervised approaches for ontology population. In: Proceedings of 11th conference of the European chapter of the association for computational linguistics, pp 17–24
30. Watróbski J (2020) Ontology learning methods from text-an extensive knowledge-based approach. Procedia Comput Sci 176:3356–3368

31. Hearst M (1992) Automatic acquisition of hyponyms from large text corpora in proc. In: 14th international conference computational linguistics, Nantes France
32. Völker J, Rudolph S (2008) Lexico-logical acquisition of owl dl axioms—an integrated approach to ontology refinement. In: Proceedings of ICFCA 2008. Lecture Notes in Artificial Intelligence, vol 4933
33. Fillmore CJ et al (2006) Frame semantics. Cogn Linguist Basic Read 34:373–400
34. Gangemi A, Presutti V (2009) In: Staab, S., Studer, R. (eds.) Ontology Design Patterns, Springer, Berlin, Heidelberg, pp 221–243. https://doi.org/10.1007/978-3-540-92673-3_10
35. Alam M, Buscaldi D, Cochez M, Osborne F, Recupero D, et al (2023) Preface-proceedings of the 6th workshop on deep learning for knowledge graphs (dlkg2023) co-located with international semantic web conference 2023. In: CEUR workshop proceedings, vol 3559, CEUR-WS, pp 1–4
36. Groth P, Rula A, Schneider J, Tiddi I, Simperl E, Alexopoulos P, Hoekstra R, Alam M, Dimou A, Tamper M (2022) The semantic web: Eswc 2022 satellite events. Lect Notes Comput Sci. https://doi.org/10.1007/978-3-031-11609-4
37. Hu Z, Hou W, Liu X (2024) Deep learning for named entity recognition: a survey. Neural Comput Appl 36(16):8995–9022. https://doi.org/10.1007/s00521-024-09646-6
38. Shen W, Li Y, Liu Y, Han J, Wang J, Yuan X (2023) Entity linking meets deep learning: techniques and solutions. IEEE Trans Knowl Data Eng 35(3):2556–2578. https://doi.org/10.1109/TKDE.2021.3117715
39. Han X, Gao T, Lin Y, Peng H, Yang Y, Xiao C, Liu Z, Li P, Zhou J, Sun M (2020) More data, more relations, more context and more openness: a review and outlook for relation extraction. In: Wong, K.-F., Knight, K., Wu, H. (eds.) Proceedings of the 1st conference of the asia-pacific chapter of the association for computational linguistics and the 10th international joint conference on natural language processing, pp 745–758. Association for Computational Linguistics, Suzhou, China. https://doi.org/10.18653/v1/2020.aacl-main.75
40. Han X, Gao T, Yao Y, Ye D, Liu Z, Sun M (2019) OpenNRE: an open and extensible toolkit for neural relation extraction. In: Padó, S., Huang, R. (eds.) Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP): system demonstrations, pp 169–174. Association for Computational Linguistics, Hong Kong, China. https://doi.org/10.18653/v1/D19-3029
41. Zhang N, Xu X, Tao L, Yu H, Ye H, Qiao S, Xie X, Chen X, Li Z, Li L (2021) DeepKE: a deep learning based knowledge extraction toolkit for knowledge base population. In: Che, W., Shutova, E. (eds.) Proceedings of the 2022 conference on empirical methods in natural language processing: system demonstrations, pp 98–108. Association for Computational Linguistics, Abu Dhabi, UAE. https://doi.org/10.18653/v1/2022.emnlp-demos.10
42. Lu Y, Liu Q, Dai D, Xiao X, Lin H, Han X, Sun L, Wu H (2022) Unified structure generation for universal information extraction. In: Muresan, S., Nakov, P., Villavicencio, A. (eds.) Proceedings of the 60th annual meeting of the association for computational linguistics (Volume 1: Long Papers), pp 5755–5772. Association for Computational Linguistics, Dublin, Ireland. https://doi.org/10.18653/v1/2022.acl-long.395
43. Martinez-Rodriguez JL, Hogan A, Lopez-Arevalo I (2020) Information extraction meets the semantic web: a survey. Semantic Web 11(2):255–335. https://doi.org/10.3233/SW-180333
44. Yao L, Mao C, Luo Y (2019) Kg-bert: Bert for knowledge graph completion. arXiv:abs/1909.03193
45. Zhong L, Wu J, Li Q, Peng H, Wu X (2023) A comprehensive survey on automatic knowledge graph construction. ACM Comput. Surv. 56(4):1–62. https://doi.org/10.1145/3618295
46. Pan S, Luo L, Wang Y, Chen C, Wang J, Wu X (2024) Unifying large language models and knowledge graphs: a roadmap. IEEE Trans Knowl Data Eng 36(7):3580–3599
47. Gangemi A, Nuzzolese AG (2025) Logic augmented generation. J Web Semant 85:100859. https://doi.org/10.1016/j.websem.2024.100859
48. Zhang B, Soh H (2024) Extract, define, canonicalize: An LLM-based framework for knowledge graph construction. In: Al-Onaizan, Y., Bansal, M., Chen, Y.-N. (eds.) Proceedings of the 2024 conference on empirical methods in natural language processing, pp 9820–9836. Association for Computational Linguistics, Miami, Florida, USA. https://doi.org/10.18653/v1/2024.emnlp-main.548
49. Orlando R, Conia S, Faralli S, Navigli R (2022) Universal Semantic Annotator: the First Unified API for WSD, SRL and Semantic Parsing. In: Proc. of LREC 2022, pp 2634–2641. European Language Resources Association, Marseille, France. https://aclanthology.org/2022.lrec-1.282
50. Meloni A, Reforgiato Recupero D, Gangemi A (2017) AMR2FRED, A tool for translating abstract meaning representation to motif-based linguistic knowledge graphs. In: The Semantic Web: ESWC 2017 Satellite Events, Springer, Portorož, Slovenia, pp 43–47
51. Van Durme B, Schubert L (2008) Open knowledge extraction through compositional language processing. In: Bos, J., Delmonte, R. (eds.) Semantics in Text Processing. STEP 2008 Conference Proceedings, pp 239–254. College Publications. https://aclanthology.org/W08-2219/

52. Curran J, Clark S, Bos J (2007) Linguistically motivated large-scale NLP with C&C and boxer. In: Ananiadou, S. (ed.) Proceedings of the 45th annual meeting of the association for computational linguistics companion volume proceedings of the demo and poster sessions, pp 33–36. Association for Computational Linguistics, Prague, Czech Republic. https://aclanthology.org/P07-2009/
53. Pramanik S, Alabi J, Roy RS, Weikum G (2024) Uniqorn: unified question answering over RDF knowledge graphs and natural language text. J Web Semant 83:100833. https://doi.org/10.1016/j.websem.2024.100833
54. Graciotti A, Presutti V, Tripodi R (2024) Latent vs explicit knowledge representation: How ChatGPT answers questions about low-frequency entities. In: Calzolari, N., Kan, M.-Y., Hoste, V., Lenci, A., Sakti, S., Xue, N. (eds.) Proceedings of the 2024 joint international conference on computational linguistics, language resources and evaluation (LREC-COLING 2024), pp 10172–10185. ELRA and ICCL, Torino, Italia. https://aclanthology.org/2024.lrec-main.888
55. Gangemi A, Graciotti A, Marzi E, Meloni A, Nuzzolese A, Presutti V, Reforgiato Recupero D, Russo A, Tripodi R (2024) MusicBO, an application of Text2AMR2FRED to the Musical Heritage domain. In: 20th extended semantic web conference. CEUR Workshop Proceedings, Crete, Greece
56. Santini C (2024) Combining language models for knowledge extraction from Italian TEI editions. Front Comput Sci 6:1472512. https://doi.org/10.3389/fcomp.2024.1472512
57. Gangemi A, Nuzzolese AG (2025) Logic augmented generation. J Web Semant 85:100859. https://doi.org/10.1016/j.websem.2024.100859
58. Gangemi A (2013) A comparison of knowledge extraction tools for the semantic web. In: The semantic web: semantics and big data, Springer, Berlin, Heidelberg, pp 351–366

Aldo Gangemi is full professor at University of Bologna and associate researcher at Italian National Research Council, Rome. He has co-founded the Semantic Technology Lab at ISTC-CNR. His research focuses on Semantic Technologies as an integration of methods from Knowledge Engineering, the Semantic Web, Linked Data, Cognitive Science, and Natural Language Processing. His theoretical interests concentrate upon the representation and discovery of knowledge patterns from data, ontology, natural language, and cognition, across multiple application domains. He has published more than 200 papers in international peer-reviewed journals, conferences and books, and seats as EB member of Semantic Web, Applied Ontology, and Web Semantics journals.

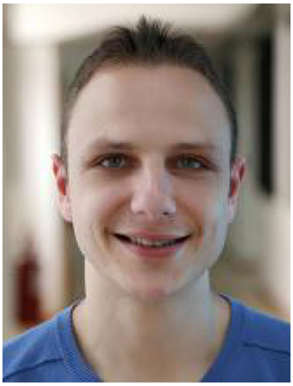
Arianna Graciotti is a PhD candidate at the University of Bologna, enrolled in the Computer Science and Engineering program and involved in Polifonia H2020 Project. Her project, at the crossroads of Natural Language Understanding and Semantic Web, focuses on knowledge extraction from multilingual and historical documents.

Antonello Meloni received his B.S., M.S., and PhD in Mathematics and Computer Science from the University of Cagliari. His research focuses on Natural Language Processing (NLP), Knowledge Graphs, and Semantic Web technologies. He contributed to the development of AMR2FRED and the py-amr2fred library, which transforms text into OWL-compliant RDF graphs. Additionally, he works on AIDA-Bot, a conversational agent designed for exploring the Academia/Industry DynAmics (AIDA) Knowledge Graph.

Andrea Giovanni Nuzzolese is a permanent researcher at the Semantic Technology Laboratory (STLab) of the National Research Council (CNR) in Rome, Italy. He received a PhD in Computer Science in 2014 from the University of Bologna (Italy). His research interests concern Knowledge Extraction, Ontology Design Patterns, Linked Data, and Semantic Web. He has been a researcher in the EU-funded projects IKS (Interactive Knowledge Stack) and MARIO (Managing active and healthy aging with use of caring service robots). He has been the reference researcher for STLab in the S&TDL (Science and Technology Digital Library) in which he contributed by introducing semantic technologies in the CNR's digital library. He has been the principal investigator for STLab in the context of the TURBO project founded by TURBO Adv S.r.l. and aimed at developing a semantic engine for Web page classification. Currently, he is the principal investigator of the project MIRA (Measuring the Impact of Research - Alternative indicators), which is founded by the Italian Agency for the Evaluation of Research (ANVUR). He is one of main developer of Apache Stanbol software stack, which provides a set of reusable components for semantic content management. He is a co-founder of the ScholarlyData initiative, which is about the publication of scholarly data as Linked Data. He has published scientific papers in international journals and conferences.

Valentina Presutti is an Associate Professor at the University of Bologna and an Associate Researcher at ISTC-CNR. She coordinates STLab. She holds a PhD in Computer Science (2006). Research topics include AI and Semantic Web, knowledge graphs and ontologies, automatic knowledge extraction, and knowledge engineering. She is PI of Polifonia (EU H2020 2021–2024). She has led national and European projects (MARIO, IKS, ArCo). During her post-doc in NeOn, she created ontologydesignpatterns.org and the WOP series and key references for the Semantic Web community. She teaches "Knowledge Engineering" and did M.Sc. in "Artificial Intelligence" at University of Bologna. She has published +150 papers. She is EiC of the Journal of Web Semantics (Elsevier) and serves as editorial member of Data Intelligence (MIT Press), Intelligenza Artificiale (IOS Press), and Semantic Web Studies (IOS Press). She co-directs ISWS and had various organizational and scientific roles in AI and Semantic Web-related events.

Diego Reforgiato Recupero has been a Full Professor at the Department of Mathematics and Computer Science of the University of Cagliari, Italy, since February 2022. In December 2015 he joined the University of Cagliari as Associate Professor. He is director and creator of the Human-Robot-Interaction Laboratory (http://hri.unica.it), co-director and creator of the Artificial Intelligence and Big Data Laboratory (http://aibd.unica.it), co-director of the Semantic Web laboratory (http://swlab.unica.it/), and member of the Commission for start-up and spin-off of the University. He is a member of different HYPERLINK "https://www.unica.it/unica/it/dip_matinfo_s1_ss4_sss7.page" HYPERLINK "https://www.unica.it/unica/it/dip_matinfo_s1_ss4_sss7.page"commissions within the Department of Mathematics and Computer Science. Moreover, he is the coordinator and co-founder of the new bachelor's degree "Applied Computer Science and Data Analytics" of the University of Cagliari. He teaches "Computers Architecture" for the bachelor degree in Computer Science, "Big Data" and "Deep Learning and Applications" for the master degree in Computer Science. He is a programmer, software developer, automation, and ICT expert. He holds a double bachelor from the University of Catania in computer science and a doctoral degree from the Department of Computer Science of the University of Naples Federico II. He got the National qualification for Computer Science Engineering, and he is a Computer Science Engineer. In 2005 he was awarded a 3-year post-doc fellowship with the University of Maryland where in 2006 he won the Computer World Horizon Award in the USA for the best research project on OASYS, an opinion analysis system that was commercialized by SentiMetrix (US company he co-founded). In 2008, he won a Marie Curie International Grant, PROVIDEO, that allowed him to come back to Italy and was able to fund a 3-year post-doc fellowship within the Department of Electrical, Electronic, and Computer Science Engineering (DIEEI) at the University of Catania. There he won the "Best Researcher Award 2012" for a project about the development of a green router nearing commercialization. In the same year, he got to the winning podium of the "Startup Weekend" event held in Catania and was a winner of Telecom Italia Working Capital Award with a grant of 25k euros for the "Green Home Gateway" project. In 2012 he co-founded the Italian company R2M Solution s.r.l. In 2013 he won a Post-Doctoral Researcher position within the Semantic Technology Laboratory (STLAB) of the Institute of Cognitive Science and Technologies (ISTC) of the National Research Council (CNR) where he worked on Semantic Web and Linked Open Data; he is still an associated researcher at STLAB (ISTC-CNR) (http://stlab.istc.cnr.it/), where he collaborates within the Semantic Web and NLP domains. In 2013 he published a paper on Science related to the energy efficiency techniques on the Internet. Besides SentiMetrix inc. (US company founded in 2007) and R2M Solution s.r.l. (Italian company founded in 2012), he co-founded R2M Solution ltd. (UK company founded in 2014), La Zenia s.r.l. (Italian company founded in 2014 for management of sports and recreational events), B-UP (Italian company and spin-off of the CNR founded in 2016 together with colleagues of CNR related to the development of software within the Semantic Web domain), and VISIOSCIENTIAE (a spin-off of the University of Cagliari founded in 2018 and related to the application of artificial intelligence to the financial domain). In 2018 together with Philips Research and other four institutions he wrote and won HYPERLINK "https://www.philhumans.eu/" HYPERLINK "https://www.philhumans.eu/"PhilHumans, a European Industrial Doctorate Marie Curie to fund 8 international PhD students and for which he coordinates several tasks, and, first in its kind ever won by the University of Cagliari. In 2021 he coordinated and won Dr. VCoach, a Marie Curie Individual Fellowship two-year project that allowed the researcher Nino Cauli to come back to Sardinia after several years spent abroad for research. Prof. Reforgiato is a patent co-owner in the field of data mining and sentiment analysis (20100023311). Since October 2025 he has been coordinating HYPERLINK "https://www.cost.eu/actions/CA24121/"KGELL. He was awarded more than 10 Best Research Paper Awards in prestigious International Conferences. He has industrial and consulting experience in several national and international industries. He has research experience (writing of the proposals and coordination after their acceptance and funding) across a wide array of industrial, Italian, FP7, and H2020 research projects (with R2M, where he covered the role of managing director, about 40 funded FP7 and H2020 projects) most recently in the fields of ICT, Energy Saving in Telecommunication Networks and Smart Grids, Robotics, and Semantic Web. He is the author of more than 170 conference and journal papers in these research fields, with more than 2000 citations. He is co-editor of the books: 1) S. Consoli, D. Reforgiato Recupero, and M. Petkovic (2019) Data Science for Healthcare: Methodologies and Applications, Springer Nature. 2) S. Consoli, D. Reforgiato Recupero, and M. Saisana (2021) Data Science for Economics and Finance, Springer Nature. He has been cited and interviewed in several national and international media (radio, television broadcasts, newspapers) for the awards and achievements he scored.



Alessandro Russo is a technologist at the Institute of Cognitive Sciences and Technologies (ISTC) of the National Research Council (CNR), where he has been a member of the Semantic Technology Laboratory (STLab) since 2015. In 2014, he obtained a PhD in Engineering in Computer Science from Sapienza University of Rome, and since 2016 he has been a co-founding partner of BUP srl, a CNR spin-off. Throughout his career, Alessandro has contributed to various national and international research projects, technology transfer activities, and tenders, including the EU-funded projects WORKPAD (FP6), SMARTVORTEX (FP7), CoMiFin (FP7), MARIO (H2020), HACID (Horizon Europe), and MAREGRAPH (Digital Europe Programme). His activity and experience include languages and architectures for process management, as well as methods and tools for knowledge representation and management, with a particular focus on Semantic Web languages and technologies, also in relation to application scenarios involving Artificial Intelligence and machine learning for the development of neurosymbolic architectures. He is involved in projects and activities focused on the production and management of heterogeneous data and the use of knowledge graphs to support software systems and services in various contexts.