Semantic Web Machine Reading with FRED

Editor(s): Name Surname, University, Country
Solicited review(s): Name Surname, University, Country
Open review(s): Name Surname, University, Country

Aldo Gangemi [a,b,*], Valentina Presutti [b], Diego Reforgiato Recupero [b], Andrea Giovanni Nuzzolese [b], Francesco Draicchio [b] and Misael Mongiovì [b]

[a] *LIPN, Université Paris 13, Sorbonne Paris Cité CNRS UMR 7030, France*
E-mail: aldo.gangemi@lipn.univ-paris13.fr
[b] *Semantic Technology Lab, ISTC-CNR, Rome and Catania, Italy*
E-mail: {aldo.gangemi,valentina.presutti,diego.reforgiato,andrea.nuzzolese,misael.mongiovi}@istc.cnr.it

Abstract. FRED is a machine reader for extracting RDF graphs that are linked to LOD and compliant to Semantic Web and Linked Data patterns. We describe the capabilities of FRED as a semantic middleware for semantic web applications. It has been evaluated against generic tasks (frame detection, type induction, event extraction, distant relation extraction), as well as in application tasks (semantic sentiment analysis, citation relation interpretation).

Keywords: Machine Reading, Knowledge Extraction, Semantic Web, Linked Data, Event extraction

1. Introduction

This paper presents a detailed description of FRED [31], a machine reader for the Semantic Web, which solves a stack of native Semantic Web (SW) machine reading tasks.

The Machine Reading paradigm [9] relies on bootstrapped, self-supervised Natural Language Processing (NLP) performed on basic tasks, in order to extract knowledge from text. Machine reading is typically much less accurate than human reading, but can process massive amounts of text in reasonable time, can detect regularities hardly noticeable by humans, and its results can be reused by machines for applied tasks.

FRED performs a formal variety of machine reading which generates RDF graph representations out of the data extracted from text. FRED can be considered a *semantic middleware*, because (1) its graphs extend and improve the results of multiple Natural Language Processing (NLP) components, but it also aims at (2) linking those results to existing semantic web knowledge, and (3) providing a formal representation to that knowledge. In addition, (4) FRED graphs are typically customized for specific application tasks.[1] Finally, (5) FRED graphs include the RDF encoding of text annotations by reusing Earmark [28] and the NLP Interchange Format (NIF) [19], so providing an interface between syntactic and semantic automatic annotation. A diagram visualizing a sample FRED graph (showing only the subgraph for the semantic triples, i.e. without textual annotation and syntax) is depicted in Figure 1. The example stresses the n-ary, event-oriented representation style, which uses semantic roles, temporal relations, and event-event relations to maintain the connectedness of extracted entities.

[*] Corresponding author. E-mail: aldo.gangemi@lipn.univ-paris13.fr

[1] A basic task addresses a non-reducible problem, such as named entity recognition or relation extraction, while application tasks address complex or user-oriented problems, e.g. search engine optimization, opinion mining, linked data population, etc.)

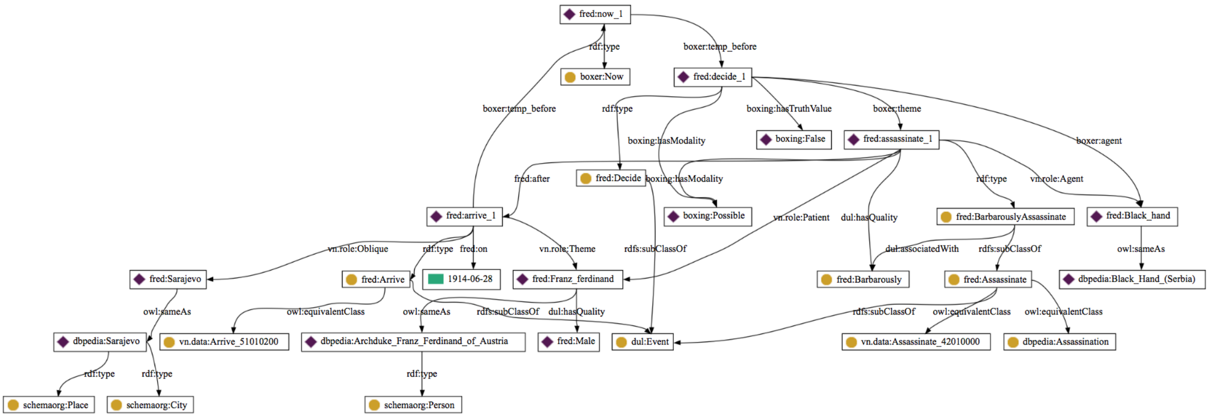


Fig. 1. An example of RDF graph produced by FRED for the sentence: *The Black Hand might not have decided to barbarously assassinate Franz Ferdinand after he arrived in Sarajevo on June 28th, 1914.*

FRED is freely accessible as a RESTful API[2] with RDF serialization in many syntaxes, as well as a web application[3] providing an intuitive diagrammatic interface. Additional visual interfaces to FRED can be experienced in the Sheldon[4] framework.

FRED has a practical value to either general Web users or application developers. General Web users can appreciate the graph representation of a given sentence using the visualization tools provided, and semantics expert can analyze the RDF triples in more detail. More important, application developers can use the REST API for empowering applications using FRED capabilities. Developers of semantic technology applications can use FRED by automatically annotating text, by filtering FRED graphs with SPARQL, and by enriching their datasets with FRED graphs, and with the knowledge coming from linkable datasets. For example, the semantic subgraph in Figure 1 can be enriched with DBpedia data about the Black Hand, Sarajevo, and Franz Ferdinand, possibly filtering them with a relevance criterion.

The paper is organized as follows. Section 2 includes background and related work. Section 3 lists the current features of FRED. Section 4 describes the pipeline architecture of FRED. Section 5 includes technical details on the implementation and the API that we provide. Section 6 shows the quality, importance and impact of FRED, showing several tools built on top of it, and a discussion on its impact. Finally, Section 7 ends the paper with discussions and future directions.

2. Background

NLP and SW. The integration between Natural Language Processing (NLP) and Semantic Web under the hat of "semantic technologies" is progressing fast. Most work has been opportunistic: on the one hand exploiting NLP algorithms and applications (typically named-entity recognizers and sense taggers) to populate SW datasets or ontologies, or for creating NL query interfaces; on the other hand exploiting large SW datasets and ontologies (e.g. DBpedia, YAGO, Freebase, etc.) to improve NLP algorithms. For example, large text analytics and NLP projects such as Open Information Extraction (OIE, [10]), Alchemy API[5], and Never Ending Language Learning (NELL, [17]) perform grounding of extracted named entities in publicly available identities such as Wikipedia, DBpedia and Freebase.

The links between the two areas are becoming tighter, and clearer practices are barely needed. Standardization attempts are happening since a while with reference to linguistic resources (WordNet, FrameNet, and the growing linguistic linked open data cloud), and the recent proposal of Ontolex-Lemon by the Ontolex W3C Community Group[6] will possibly improve resource reuse. Recently, platforms such as Apache

[2] http://wit.istc.cnr.it/stlab-tools/fred/api
[3] http://wit.istc.cnr.it/stlab-tools/fred
[4] http://wit.istc.cnr.it/stlab-tools/sheldon
[5] http://www.alchemyapi.com
[6] http://www.w3.org/community/ontolex/wiki/Main_Page

Stanbol[7], NIF [19] and the NLP2RDF project, NERD [18], and FOX[8]) made it simpler to reuse NLP components as linked data.

Semantic interoperability issues. Interoperability efforts so far mainly concentrated on the direct transformation of NLP data models into RDF.

Classical work on ontology learning such as [6] takes the integration problem from a formal viewpoint, and uses linguistic features to extract occurrences of logical axioms, such as subclass of, disjointness, etc. Some work from NLP followed a similar direction [7]: NELL relation properties and ontology [21], and "formal semantics" applied to NL (e.g. [5],[22]). These works assume some axiomatic forms, and make the extraction process converge to that form.

This is good in principle, but the current state of the art does not really help with establishing clearcut criteria on how to convert NL extractions to RDF or OWL. Classical ontology learning mainly uncovers statistical regularities from large corpora, which could justify e.g. a subclass or disjointness axiom. Similarly, the NELL ontology[9] has been learnt by an algorithm that discovers binary relations and concepts. As reported in [34] the conversion of its data structure into RDF/OWL, makes it usable on SW.

NLP tools typically provide data without a standardized structure, and without really worrying of any SW or just formal reuse of those data: comparing for example the heterogeneous ways predicate/argument structures are represented in VerbNet, FrameNet, PropNet, OntoNotes, etc. In fact, the assumption is that any application would make an arbitrary reuse of those data. Although basic porting of the original data structures in RDF can be beneficial (cf. e.g. the LODifier method [4]), the SW needs homogeneous, self-connected graphs that integrate those data.

The (few) approaches from natural language formal semantics which output formal data structures are not easily interpretable into semantic web languages. For example, Discourse Representation Structure (DRS), as shown in the output of Boxer [5] is a first-order logic data structure that heavily uses *discourse referents* as variables to anchor the predicates into extensional interpretations, and a *boxing* representation that contextualizes the scope of logical (boolean, modal, inferential) operators. Both issues need non-trivial decisions on the side of RDF and OWL design: what variables should be accommodated in a SW representation, or ignored? What logical operators can be safely represented in the formal semantics supported by SW languages? What predicates should be represented, and in which form, in RDF or OWL?

3. FRED capabilities

FRED is a tool for automatically producing RDF/OWL ontologies and linked data from text. FRED formally represents, integrates, improves, and links the output of several NLP tools, as summarized in this section.

Deep parsing. The backbone deep semantic parsing is currently provided by Boxer [5], which uses a statistical parser (C&C) producing Combinatory Categorial Grammar trees, and thousands of heuristics that exploit existing lexical resources and gazetteers to generate representation structures according to Discourse Representation Theory (DRT) [20].

The basic NLP tasks performed by Boxer, and reused by FRED, include: (mostly) verbal event detection, semantic role labeling with VerbNet and FrameNet roles, first-order logic representation of predicate-argument structures, logical operators scoping (called *boxing*), modality detection, and tense representation.

Custom vocabularies have been derived from Boxer's data structures, and DRT/Boxing structures have been completely reengineered according to SW and linked data design practices. Events, role labeling, and boxing have been represented as typed n-ary logical patterns in RDF/OWL (`dul:Event` and `boxing:Situation`[10] are the two classes used for typing events and logical boxing respectively). E.g. in Turtle syntax:

```
:assassinate_1 a vndata:Assassinate_42010000 ;
  vnrole:Agent :Black_Hand ;
  vnrole:Patient :Franz_Ferdinand .
```

First-order, factual relations have been represented as OWL property assertions (basic triples), e.g.:

```
:assassinate_1 dul:hasQuality :Barbarously .
```

[7] http://stanbol.apache.org

[8] http://aksw.org/Projects/FOX.html

[9] http://rtw.ml.cmu.edu/rtw/kbbrowser/

[10] Besides the usual ones (rdf, rdfs, owl, dbpedia, schemaorg), the following prefixes are used here: fred: for http://www.ontologydesignpatterns.org/ont/fred/domain.owl\#, vnrole: for http://www.ontologydesignpatterns.org/ont/vn/abox/role/, dul: for http://www.ontologydesignpatterns.org/ont/dul/dul.owl\#, wn30: for http://www.w3.org/2006/03/wn/wn30/instances/

Modality and negation have been represented in a loose, RDF-oriented form instead of OWL or modal logic, because their NL semantics is unpredictable[11]. This solution also avoids the creation of blank nodes that create indirections in linked data querying. E.g.:

```
:assassinate_1 boxing:hasTruthValue boxing:False ;
  boxing:hasModality boxing:Possible .
```

Anyway, if a stronger semantics is desired for negation and modality, it is easy to refactor FRED graphs for that purpose (e.g. with a SPARQL construct query).

Compositional semantics, taxonomy and reification. NLP outputs usually do not contemplate SW design practices. FRED injects three such practices by further analyzing the output of NLP components. The first practice deals with coreferential predicates, e.g. from the term*programming language*, FRED derives a class representing a complex term, and its taxonomy:

```
:Programming_language rdfs:subClassOf
  :Language .
```

which is further submitted to disambiguation engines. The second practice deals with periphrastic properties such as `:survivorOf` (described later). The third practice is the reification of certain variables used as DRT discourse referents. This happens when a variable refers to something that has a role in the formal semantics of the sentence. For example, `cat_1` is the reification of the *x* variable in the first-order predication *Cat(x)* extracted from the sentence *The cat is on the mat*.

Data and vocabularies. Linguistic frames [25], ontology design patterns [12], linked open data and vocabularies are reused throughout the FRED's pipeline in order to resolve, align, or enrich data and ontologies produced by FRED. Used resources include: VerbNet[12], for disambiguation of verb-based events; WordNet-RDF[13]; OntoWordNet (OWN) 2012 [26], which is an OWL version of WordNet that extends OWN [15], and provides the alignment of classes to WordNet and DOLCE; DBpedia for the resolution and/or disambiguation of named entities; schema.org (among others) for typing the recognized named entities.

NER. Named Entity Recognition (NER) and Resolution (aka Entity Linking) is currently provided by TAGME [11], an algorithmic NER resolver to Wikipedia that heavily uses sentence and Wikipedia context to disambiguate named entities. For example:

```
:Franz_Ferdinand owl:sameAs
   dbpedia:Archduke_Franz_Ferdinand_of_Austria .
:Black_Hand owl:sameAs dbpedia:Black_Hand .
```

Since Wikipedia is also rich in "conceptual" entities, TAGME results to be also a precise word sense disambiguator, once its results are formally interpreted and contextually fine-tuned by FRED. For example, if the text segment annotated by TAGME is firstly annotated by FRED as an `owl:Class`, the resolved DBpedia entity is interpreted as an `owl:Class` as well; therefore, if an individual is typed by that class, it is typed by the DBpedia entity as well (since FRED applies a simple inheritance pattern here). This is also a way of contextualizing the semantics of DBpedia entities.

WSD. FRED produces RDF/OWL ontologies having classes (and related taxonomies) depending on the lexicon used in the text. In order to provide a public identity to such classes, FRED exploits Word-Sense Disambiguation (WSD) to resolve classes into WordNet or BabelNet [24]. FRED can use any WSD system, such as UKB [1] or Babelfy [23]. WSD also enables FRED to generate alignments to two top-level ontologies: WordNet supersenses and a subset of DOLCE+DnS Ultra Lite (DUL) classes. For example, from *programming language*, FRED produces the following alignment axioms[14]:

```
:Programming_language
  owl:equivalentClass
    wn30:synset-programming%20language-noun-1 ;
  rdfs:subClassOf
    dul:InformationEntity,
    wn30:supersense-noun_communication .
```

Textual annotation grounding. The Earmark vocabulary [28] and the NLP Interchange Format (NIF) [19] are employed by FRED in order to annotate text segments with the resources from its graphs. The integration of several NLP components is a real challenge, since text segmentation can be slightly different between components that in principle annotate the same text area. For example, certain text segments are annotated by FRED as periphrastic relations; e.g. from the sentence: *I am the only survivor of the expedition*, FRED annotates *survivor of* as `fred:survivorOf`, while other components may annotate *survivor* and

[11] For example, the combination of possibility and negation in the sentence processed as from Figure 1 – which expresses a counterfactual – cannot be rigorously expressed in OWL

[12] http://verbs.colorado.edu/~mpalmer/projects/verbnet.html

[13] http://www.w3.org/TR/wordnet-rdf/

[14] The prefix `wn30:` stands for http://www.w3.org/2006/03/wn/wn30/instances/

of separately. Earmark and NIF support "annotation rules" that help solving such cases.

Coreference Resolution. Coreference resolution is quite limited in Boxer, therefore FRED integrates CoreNLP[15]. Let us consider the following sentence: *If a farmer owns a donkey, he beats it.* The graph generated by FRED with no coreference resolution information is shown in Figure 2 (a) where the reader may notice the presence of the graph node *male_1* corresponding to the pronoun in the sentence *he* and the graph node *neuter_1* corresponding to the pronoun in the sentence *it*. CoreNLP integration allows FRED to resolve those pronouns, and to associate them to the correct entities, as shown in Figure 2.

Multilinguistic capability FRED takes as input a text in one of 48 different languages, and always provide the results with English identifiers for the graph elements. For this purpose, FRED uses the Bing Translation APIs[16]. If the chosen language is different from English, the tag <*BING_LANG:lang*> needs to precede the sentence, where *lang* indicates the code for the language[17]. For example, the sentence:

<*BING_LANG:it*>*Catania è una delle poche città in Italia ad offrire paesaggi tanto diversi concentrati in un solo sito.*[18]

is a valid Italian sentence to be processed. The Sheldon interface to FRED also supports automatic language recognition, therefore the language tag can be omitted.

4. FRED architecture

FRED's component architecture is depicted in Figure 3. It can be interpreted as a three-layer architecture.

The first layer takes care of *reengineering*, and involves external NLP components. Components are either tightly or loosely coupled (e.g. TAGME, accessed via REST). The core of the reengineering consists of data structure readers, which read component outputs, and pass them to refactoring components, possibly using Apache Stanbol[19] as an integrator.

The second layer takes care of *refactoring*, which has its core in a modular library of heuristical rules that receive the result of reengineering readers, and produce FRED's refactoring that is sent to the triplifier component, or to the production rule engine, which takes care of enhancing the extracted knowledge. Enhancing includes taxonomy composition from predicates, periphrastic relations (e.g. *survivorOf*), and the reification of discourse referent variables. The triplifier collects the enhanced knowledge, and applies configurations and unification rules. It then passes the triples to the formatter, which serializes RDF or visualizable graph data.

The third layer implements *integration*, *communication*, and *alignment*. Here formatted data are taken into account by K~ore, a software architecture which integrates FRED with NER and WSD, manages Linked Open Data queries and alignment to Semantic Web vocabularies, and provides a RESTful API. K~ore finally fine-tunes its output with the help of the SW fine-tuner that adjusts the logical assignments of RDF resources coming from different components, and delivers the final FRED's output based on the parameters selected from the REST API or the Web application.

5. Implementation of FRED

FRED is available as a demo web application, or as a REST service. The current output of FRED is either graphic or in any RDF encoding. Typically, FRED graphs are filtered or enriched according to the requirements of applied tasks, as shown e.g. in experiments of automatic typing of resources from Wikipedia definitions [26], aspect-based sentiment analysis [16], relevant event extraction [13][14], citational context understanding [8], etc.

FRED integration and communication component, K~ore, is designed by combining two architectural styles: the Component-based and the REST. It is implemented as a modular set of Java components. Each component is accessible via its own RESTful Web interface. From this viewpoint, all the features can be used via RESTful service calls. Components do not depend on each other, however they can be easily combined if needed. All components are implemented as

[15] http://nlp.stanford.edu/software/corenlp.shtml

[16] http://www.microsoft.com/web/post/using-the-free-bing-translation-apis

[17] check http://msdn.microsoft.com/en-us/library/hh456380.aspx for the list of language codes

[18] The English translation is "Catania is one of the few cities in Italy to offer such different landscapes concentrated in a single site."

[19] http://stanbol.apache.org

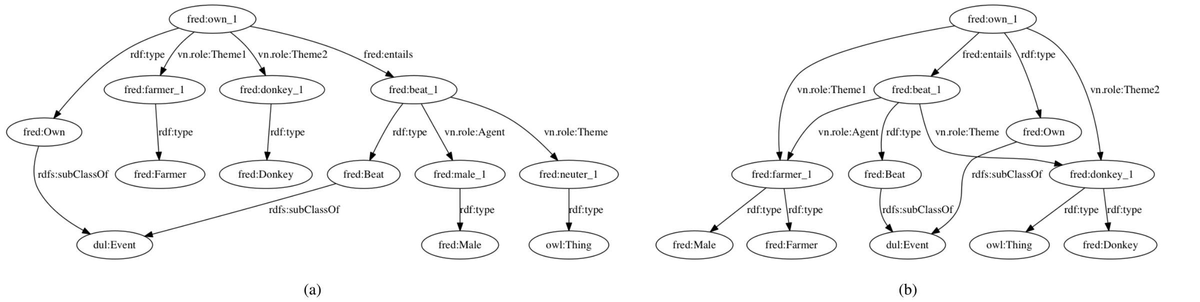


Fig. 2. FRED's graph for sentence *If a farmer owns a donkey, he beats it*. Without coreference resolution (a) and with coreference performed by CoreNLP (b).

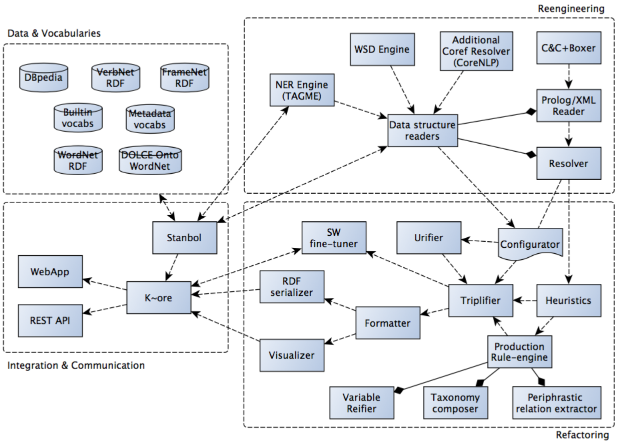


Fig. 3. FRED's component architecture.

OSGi [3] bundles, components and services. The OSGi implementation used by K~ore is Apache Felix[20]. In more detail, the Named-Entity Resolution functionality has been implemented by integrating in the architecture the bundles of the Apache Stanbol Enhancer[21], which has been configured by adding an enhancement engine based on Tagme[22] [11]. The Word-Sense Disambiguation has been obtained by implementing an OSGi wrapper for e.g. UKB [1] based on Apache Felix.

FRED is also accessible by means of a Python API, namely *fredlib*. It exposes features for retrieving FRED graphs from user-specified sentences, and managing them. More specifically, a simple Python function hides details related to the communication with the FRED service and returns to the user a FRED graph object that is easily manageable. FRED graph objects expose methods for retrieving useful information, including the set of individual and class nodes, equivalences and type information, categories of FRED nodes (events, situations, qualities, general concepts) and categories of edges (roles and non roles). *fredlib* supports *rdflib*[23] (for managing RDF graphs) and *networkx*[24] (for managing complex networks) libraries. It can be freely downloaded[25].

6. Quality, Importance, Impact

Since FRED is a semantic middleware, its quality is primarily assessed by evaluating the performance of the applications that depend on it. FRED has been suc-

[20] `http://felix.apache.org/`

[21] `http://stanbol.apache.org/`

[22] `http://tagme.di.unipi.it/`

[23] `http://code.google.com/p/rdflib/`

[24] `https://networkx.github.io/`

[25] `http://wit.istc.cnr.it/stlab-tools/fred/fredlib`

cessfully evaluated in the past for specific basic or application tasks [31,26,13,16,8,32,30,27]. In the following, we list some of them that have been formally evaluated.

Improvements on the frame detection task are reported in [31], where precision is comparable ($P = .75$) to the state-of-art tool, but FRED is one order of magnitude faster, and frame occurrences are formally represented. Recall was lower than the state of art tool, but the corpus used for the evaluation was the same as the one used to train that tool.

FRED has been used to implement *Sentilo* [16][32], a semantic sentiment analysis system that identifies opinion holders, detects topics, and scores opinions. The evaluations on a corpus of user-based hotel reviews, reported in the cited papers, show high accuracy of the system ($F_1 = .95$ for holder detection; $F_1 = .66$ for topic detection; $F_1 = .80$ for subtopic detection; .81 is the correlation with open-rating 5-star scores given for reviews). Sentilo[26] uses a set of heuristical rules to extract, from the FRED graph of a given sentence, information about holders of opinions (if any), its topics, and its opinion-expressing words (later used as opinion features). Besides, Sentilo enriches the RDF/OWL semantic representation of an opinion sentence with annotation triples based on an opinion model. Figure 4 shows the pipeline of Sentilo. Sentilo includes the following filtering and extensions to FRED:

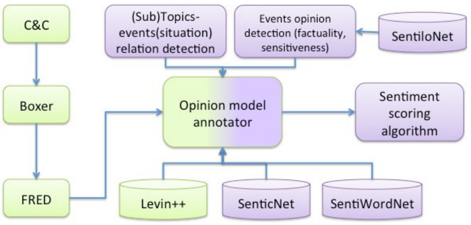

C&C
Boxer
FRED
(Sub)Topics-events(situation) relation detection
Events opinion detection (factuality, sensitiveness)
SentiloNet
Opinion model annotator
Sentiment scoring algorithm
Levin++
SenticNet
SentiWordNet

Fig. 4. Pipeline of Sentilo.

- an ontology for opinion sentences that reshapes FRED graphs;
- *SentiloNet*, a new lexical resource that enables the evaluation of opinions expressed by means of events;
- a novel scoring algorithm for opinion sentences;
- a high level graphical interface to show Sentilo results;

FRED has been used to develop *Legalo*[27][30], a novel approach for automatically typing links. Legalo firstly extracts natural language sentences (including links from a web page given as input), identifies any subject and object of a semantic relation (e.g. as suggested by a page link in Wikipedia, or from any sentence), and their lexicalizations in the associated sentence. Then, it passes each sentence to FRED, extracts from the resulting graph the subgraphs connecting the previously identified subjects and objects, and explores the path that connects subject and object in order to build a descriptive binary relation. Finally, it aligns the produced properties to existing properties from Semantic Web ontologies and vocabularies. The pipeline implemented by Legalo is shown in Figure 5. It extracts subgraphs from FRED graphs based on a set of graph patterns, and generates new binary relations between entities that are only indirectly connected in the original graph, so enriching it. A detailed evaluation on a corpus of Wikipedia pages is provided in [30], and shows high accuracy ($F_1 = .83$ for relations capturing the actual semantics of wikilinks).

FRED has been used to develop *Tipalo*[28] [26], a tool that automatically assigns types to DBpedia entities based on their definitions in natural language, provided by their corresponding Wikipedia pages. Tipalo firstly extracts definitions from Wikipedia pages through the definition extractor component. The definitions are parsed and represented in a logical form by FRED. Then, from the FRED graphs, Tipalo identifies the paths that provide typing information about the analyzed entity, and discards the rest. Once concepts expressing the types of an entity and their taxonomical relations have been identified, a WSD tool is used to gather their correct sense. The final step is to link the obtained concepts to other Semantic Web ontologies, in order to support shared interpretation and linked data enrichment. Figure 6 shows the pipeline of Tipalo. An evaluation is provided in [26], and it shows that Tipalo brings highly accurate entity typing ($F_1 = .92$ for the type selection, $F_1 = .75$ when WSD is added) with concepts extracted from the original definitions used in Wikipedia, so providing an alternative to DBpedia and YAGO [33].

FRED has also been used to develop *Citalo*[29] [8], a tool to automatically infer the function of citations by means of Semantic Web technologies and NLP tech-

[26] http://wit.istc.cnr.it/stlab-tools/sentilo

[27] http://wit.istc.cnr.it/stlab-tools/legalo

[28] http://wit.istc.cnr.it/stlab-tools/tipalo/

[29] http://wit.istc.cnr.it/stlab-tools/citalo/

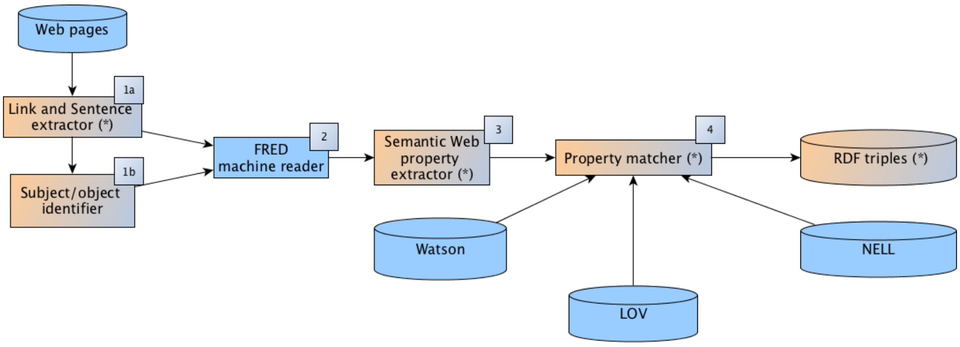

Web pages
1a
Link and Sentence extractor (*)
1b
Subject/object identifier
2
FRED machine reader
3
Semantic Web property extractor (*)
4
Property matcher (*)
RDF triples (*)
Watson
LOV
NELL

Fig. 5. Pipeline of Legalo. Numbers indicate the order of execution of a component in the pipeline.

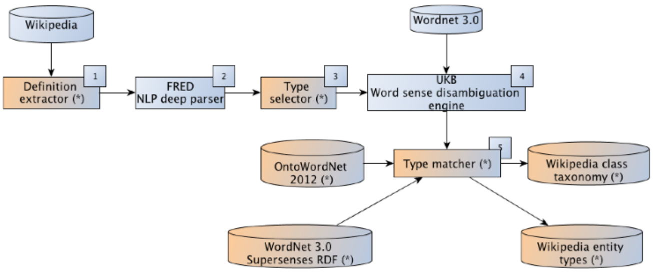


Fig. 6. Tipalo's pipeline. Numbers indicate the order of execution of a component in the pipeline.

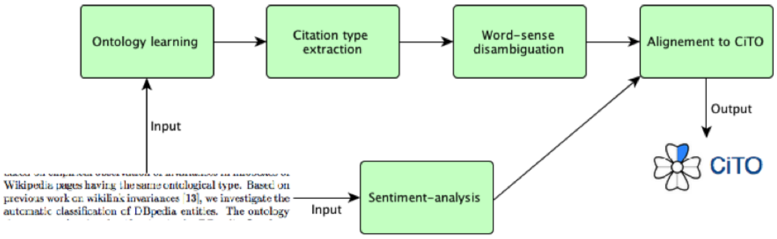

Ontology learning
Citation type extraction
Word-sense disambiguation
Alignement to CiTO
Input
Output
CiTO
Wikipedia pages having the same ontological type. Based on previous work on wikilink invariances [13], we investigate the automatic classification of DBpedia entities. The ontology
Input
Sentiment-analysis

Fig. 7. Citalo's pipeline.

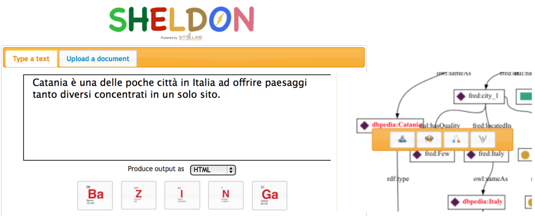
niques. It takes as input a sentence containing a reference to a bibliographic entity and the CiTO [29] ontology used to describe the nature of citations in scientific research articles, and infers the function of the citation. Citalo relies on FRED to extract ontological information from the input sentence and derive a logical representation of it. Candidate types for the citation are extracted by looking for patterns in the FRED result. A WSD algorithm is used to gather the sense of candidate types. The final step consists of assigning CiTO types to citations. Figure 7 shows the pipeline of Citalo. Some tests on a collection of documents has been carried out in [8] where strengths and weakness of the approach are described.

FRED has also been employed as a middleware for the development of a Semantic Holistic framework for LinkeD ONtology data (SHELDON)[30], a framework that provides semantic web capabilities and that resulted a finalist[31] at the Semantic Web Challenge held at ISWC2014. Given a sentence in any language, SHELDON provides several semantic functionalities (frame detection, topic extraction, named entity recognition, resolution and coreference, terminology extraction, sense tagging and disambiguation, taxonomy induction, semantic role labeling, type induction, senti-

[30] http://wit.istc.cnr.it/stlab-tools/sheldon

[31] http://challenge.semanticweb.org/2014/finalists.html

ment analysis, citation inference, relation and event extraction), visualization tools (which make use of the JavaScript infoVis Toolkit[32] and RelFinder[33]), and a knowledge enrichment component that extends machine reading to Semantic Web data. Figure 8 shows the main interface of SHELDON. A user can type a sentence in any language and decide which semantic task to perform.



Catania è una delle poche città in Italia ad offrire paesaggi tanto diversi concentrati in un solo sito.

Fig. 8. SHELDON front page (on the left), SHELDON's navigation toolbar for identified DBpedia entities (on the right).

The evaluation of applications depending on FRED shows good or high accuracy, hence proving its quality. Besides rigorous evaluation, FREDÕs quality is supported by evidence of its impact in the community. Firstly, a public forum in the software engineering community, stackoverflow.com, contains independent discussions about extending FRED's usage to other platforms (Python, C#)[34]. Another forum, answers.semanticweb.com, contains independent analyses and discussion of related works, revealing FREDÕs uniqueness in providing a solution for producing rich RDF datasets from text[35]. The Wikipedia entry for Knowledge Extraction[36] contains a list of knowledge extractors from text, and even looking only at the column for "extracted entities", FRED has the largest set. The only comparable entry is for a proprietary tool that does not include any demo available on the Web.

Finally, as hinted by its impact, the importance of FRED seems also to reside in its uniqueness in covering a large amount of linguistic semantic constructions in a native Semantic Web way. In [13], a detailed comparison of information extraction tools is presented. After manually converting their non-RDF output to RDF graphs, the study investigates what basic semantic tasks are actually covered by existing tools, and estimates their accuracy against a news text (producing a small gold standard consisting of 524 semantic triples). The results show that FRED is unique in covering – and in particular *integrating* – a full range of basic tasks, with high accuracy in all of them (NER $A = .84$, terminology extraction $F_1 = .87$, taxonomy induction $F_1 = .83$, relation extraction $F_1 = .76$, frame detection $F_1 =, 93$, and event detection $F_1 = .82$).

More literature shows the impact of FRED beyond semantic technology circles: a 2014 study about dealing with big data for statistics made by the United Nations Economic Commission for Europe[37] says that *"The knowledge extraction from unstructured text is still a hard task, though some preliminary automated tools are already available. For instance, the tool FRED (`http://wit.istc.cnr.it/stlab-tools/fred`) permits to extract an ontology from sentences in natural language."* A 2013 book [2] on big data computing says in the introduction: *"The technologies for this* [knowledge extraction] *are under intensive development currently, for example* `wit.istc.cnr.it/stlab-tools/fred` *(accessed October 8, 2012)".*

7. Conclusions

We presented FRED, a machine reader for extracting linked open data and ontologies from text. In the current landscape of attempts to integrate NLP and SW technologies and methods, FRED stands as the non-commercial tool having the largest coverage of formally defined tasks, and, to our knowledge, the largest coverage for the Semantic web specifically. We reported the accuracy of several tools developed on top of FRED that have been successfully adopted by the Semantic Web community. We also discussed the importance and impact of FRED. Ongoing work is concentrating on creating large repositories of FRED graphs, using typed named graphs and reconciliation techniques for the cases when the source texts are related for some reason, e.g. with news series, large texts, abstracts of categorized scientific articles, etc. The final goal is to implement a *knowledge extraction factory* that can be used to perform deep and formal annotation of large archives of documents, and to automatically produce formal relations between them. Another

[32] `http://philogb.github.io/jit/`

[33] `http://www.visualdataweb.org/relfinder.php`

[34] `http://tinyurl.com/qa2dyfj`, `http://tinyurl.com/o993scy`

[35] `http://tinyurl.com/n6pzpot`, `http://tinyurl.com/kb8564w`

[36] `http://en.wikipedia.org/wiki/Knowledge_extraction`

[37] `http://tinyurl.com/ml6ystn`

ongoing evolution of FRED is in the area of robot-human interaction, where FRED graphs extracted from natural language dialogues need to be interpreted with reference to physical environments.

References

[1] Eneko Agirre and Aitor Soroa. Personalizing pagerank for word sense disambiguation. In *EACL*, Athens, Greece, 2009. The Association for Computer Linguistics.

[2] Rajendra Akerkar. *Big data computing*. CRC Press, 2013.

[3] The OSGi Alliance. OSGi Service Platform Release 4 Version 4.2, Core Specification. Committee specification, Open Services Gateway initiative (OSGi), September 2009.

[4] Isabelle Augenstein, Sebastian Padó, and Sebastian Rudolph. Lodifier: Generating linked data from unstructured text. In *The Semantic Web: Research and Applications*, pages 210–224. Springer, 2012.

[5] Johan Bos. Wide-Coverage Semantic Analysis with Boxer. In Johan Bos and Rodolfo Delmonte, editors, *Semantics in Text Processing*, pages 277–286. College Publications, 2008.

[6] Philipp Cimiano and Johanna Völker. Text2onto - a framework for ontology learning and data-driven change discovery, 2005.

[7] Jesse Davis and Pedro Domingos. Deep transfer: A markov logic approach. *AI Magazine*, 32(1):51–53, 2011.

[8] Angelo Di Iorio, Andrea Giovanni Nuzzolese, and Silvio Peroni. Towards the automatic identification of the nature of citations. In *Proc. of 3rd Workshop on Semantic Publishing (SePublica 2013)*, pages 63–74, 2013.

[9] Oren Etzioni, Michele Banko, and Michael J. Cafarella. Machine reading. In *AAAI*, pages 1517–1519, 2006.

[10] Oren Etzioni, Anthony Fader, Janara Christensen, Stephen Soderland, and Mausam. Open information extraction: The second generation. In *IJCAI*, pages 3–10. IJCAI/AAAI, 2011.

[11] Paolo Ferragina and Ugo Scaiella. Tagme: On-the-fly annotation of short text fragments (by wikipedia entities). In *Proc. of the 19th ACM CIKM*, CIKM '10. ACM, 2010.

[12] A. Gangemi and V. Presutti. Ontology Design Patterns. In S. Staab and R. Studer, editors, *Handbook on Ontologies, 2nd Edition*. Springer Verlag, 2009.

[13] Aldo Gangemi. A comparison of knowledge extraction tools for the semantic web. In *Proc. of 10th ESWC 2013, Montpellier, France*. LNCS, Springer, 2013.

[14] Aldo Gangemi, Ehab Hassan, Valentina Presutti, and Diego Reforgiato Recupero. Fred as an event extraction tool. In *Proc. of DeRiVE 2013*, volume http://ceur-ws.org/Vol-1123/paper3.pdf. CEUR Workshop Proceedings, 2013.

[15] Aldo Gangemi, Roberto Navigli, and Paola Velardi. The OntoWordNet Project: Extension and Axiomatization of Conceptual Relations in WordNet. In *CoopIS/DOA/ODBASE*, pages 820–838, 2003.

[16] Aldo Gangemi, Valentina Presutti, and Diego Reforgiato Recupero. Frame-based detection of opinion holders and topics: a model and a tool. *IEEE Computational Intelligence*, 9(1), 2014.

[17] Matt Gardner, Partha Pratim Talukdar, Bryan Kisiel, and Tom Mitchell. Improving learning and inference in a large knowledge-base using latent syntactic cues. In *Proc. of the EMNLP 2013*, 2013.

[18] Giuseppe Rizzo, Raphaël Troncy, Sebastian Hellmann, and Martin Bruemmer. NERD meets NIF: Lifting NLP extraction results to the linked data cloud. In *LDOW, 5th Wks. on Linked Data on the Web, Lyon, France*, 04 2012.

[19] Sebastian Hellmann, Jens Lehmann, and Sören Auer. Linked-Data Aware URI Schemes for Referencing Text Fragments. In *EKAW 2012*, LNCS 7603. Springer, 2012.

[20] Hans Kamp. A theory of truth and semantic representation. In Jeroen A. G. Groenendijk, Teo M. V. Janssen, and Martin B. J. Stokhof, editors, *Formal Methods in the Study of Language*, volume 1, pages 277–322. Mathematisch Centrum, 1981.

[21] Thahir Mohamed, Estevam Hruschka, and Tom Mitchell. Discovering relations between noun categories. In *Proc. of EMNLP 2011*. ACL, 2011.

[22] Richard Moot and Christian Retoré. *The logic of categorial grammars: a deductive account of natural language syntax and semantics*. Springer, 2012.

[23] Andrea Moro, Alessandro Raganato, and Roberto Navigli. Entity linking meets word sense disambiguation: A unified approach. *Transactions of the Association for Computational Linguistics*, 2:231–244, 2014.

[24] Roberto Navigli and Simone Paolo Ponzetto. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250, 2012.

[25] A. G. Nuzzolese, A. Gangemi, and V. Presutti. Gathering Lexical Linked Data and Knowledge Patterns from FrameNet. In *K-CAP*, pages 41–48, Banff, Alberta, Canada, 2011.

[26] A. G. Nuzzolese, A. Gangemi, V. Presutti, P. Ciancarini, and A. Musetti. Automatic Typing of DBpedia Entities. In *Proc. of the International Semantic Web Conference (ISWC)*, Boston, MA, US, 2012.

[27] Andrea Giovanni Nuzzolese, Valentina Presutti, Aldo Gangemi, Alberto Musetti, and Paolo Ciancarini. Aemoo: Exploring knowledge on the web. In *Proc. of the 5th Annual ACM Web Science Conference*, pages 272–275. ACM, 2013.

[28] Silvio Peroni, Aldo Gangemi, and Fabio Vitali. Dealing with markup semantics. In *Proc. of the 7th International Conference on Semantic Systems*, pages 111–118. ACM, 2011.

[29] Silvio Peroni and David Shotton. Ontology paper: Fabio and cito: Ontologies for describing bibliographic resources and citations. *Web Semant.*, 17:33–43, December 2012.

[30] Valentina Presutti, Sergio Consoli, Andrea Giovanni Nuzzolese, Diego Reforgiato Recupero, Aldo Gangemi, Ines Bannour, and Haifa Zargayouna. Uncovering the semantics of wikipedia wikilinks. EKAW 2014, 2014.

[31] Valentina Presutti, Francesco Draicchio, and Aldo Gangemi. Knowledge extraction based on discourse representation theory and linguistic frames. In *EKAW*. Springer, 2012.

[32] Diego Reforgiato Recupero, Valentina Presutti, Sergio Consoli, Aldo Gangemi, and Andrea Nuzzolese. Sentilo: Frame-based sentiment analysis. *Cognitive Computation*, 2014.

[33] Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: A core of semantic knowledge. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, pages 697–706, New York, NY, USA, 2007. ACM.

[34] A. Zimmermann, C. Gravier, J. Subercaze, and Q. Cruzille. Nell2rdf: Read the Web, and turn it into RDF. In *2nd Int Workshop on Knowledge Discovery and Data Mining Meets Linked Open Data, 10th ESWC 2013, Montpellier, France*, 2013.