Aalborg Universitet

AALBORG UNIVERSITY
DENMARK

Addressing structural and linguistic heterogeneity in the Web

Gonzalez, Jacobo Rouces; de Melo, Gerard; Hose, Katja

Published in:
AI Communications

DOI (link to publication from Publisher):
10.3233/AIC-170745

Publication date:
2018

Document Version
Accepted author manuscript, peer reviewed version

Link to publication from Aalborg University

Citation for published version (APA):
Gonzalez, J. R., de Melo, G., & Hose, K. (2018). Addressing structural and linguistic heterogeneity in the Web. *AI Communications*, *31*(1), 3-18. https://doi.org/10.3233/AIC-170745

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal -

Take down policy
If you believe that this document breaches copyright please contact us at vbn@aub.aau.dk providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from vbn.aau.dk on: March 19, 2024

Aalborg Universitet

AALBORG UNIVERSITY
DENMARK

Addressing structural and linguistic heterogeneity in the Web

Gonzalez, Jacobo Rouces; de Melo, Gerard; Hose, Katja

Published in:
AI Communications

DOI (link to publication from Publisher):
10.3233/AIC-170745

Publication date:
2018

Document Version
Accepted author manuscript, peer reviewed version

Link to publication from Aalborg University

Citation for published version (APA):
Gonzalez, J. R., de Melo, G., & Hose, K. (2018). Addressing structural and linguistic heterogeneity in the Web. AI Communications, 31(1), 3-18. https://doi.org/10.3233/AIC-170745

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

? Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
? You may not further distribute the material or use it for any profit-making activity or commercial gain
? You may freely distribute the URL identifying the publication in the public portal ?

Take down policy
If you believe that this document breaches copyright please contact us at vbn@aub.aau.dk providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from vbn.aau.dk on: september 04, 2019

The final publication is available at IOS Press through http://dx.doi.org/10.3233/AIC-170745[1]

Addressing Structural and Linguistic Heterogeneity in the Web *

Jacobo Rouces [a], Gerard de Melo [b] and Katja Hose [a]

[a] *Department of Computer Science, Aalborg University*
E-mail: jacobo@rouces.org, khose@cs.aau.dk
[b] *Department of Computer Science, Rutgers University–New Brunswick*
E-mail: gdm@demelo.org

An increasing number of structured knowledge bases have become available on the Web, enabling many new forms of analyses and applications. However, the fact that the data is being published by different parties with different vocabularies and ontologies means that there is a high degree of heterogeneity and no common schema. At the same time, the abundance of different human languages across unstructured data presents a similar problem, because most text mining tools only cater to the English language. This paper presents solutions for these two kinds of heterogeneity. It introduces Klint, a Web-based system that automatically creates mappings to transform knowledge from heterogeneous sources into FrameBase, which is a broad linked data schema that enables the representation of a wide range of knowledge. With Klint, a user can review and edit the mappings with a streamlined interface, which in turn allows for human-level accuracy with minimum human effort. The paper further describes how FrameBase can be extended to support multilingual labels, which can aid in extending current tools for integrating English text into FrameBase knowledge.

Keywords: linked data, data integration, multilingual text mining

1. Introduction

The Web of Data includes a rich and increasing number of structured knowledge bases, and has enabled many new applications and forms of analyses. Such open knowledge bases often provide their data in RDF format [33], based on Subject-Predicate-Object triples. When coupled with the use of global identifiers in the form of Internationalized Resource Identifiers (IRIs), this facilitates the linking and merging of data from disparate sources into a single graph, usually referred to as the LOD (Linked Open Data) cloud. However, despite these advantages, different datasets may still model equivalent or overlapping information using different vocabularies and different triple patterns. This poses two important problems.

- **Querying data** can be cumbersome. In order to capture all pertinent knowledge, a structured query will have to consist of a disjunction of all possible semantic patterns occurring in the myriad of heterogeneous vocabularies used in the data.
- **Linking data** is typically carried out by creating triples with a predicate `owl:sameAs` connecting RDF terms deemed to have the same meaning [3, 14]. However, this may be infeasible when information is captured in structurally different forms across different datasets.

Beyond these, there is also the inevitable challenge that most data on the Web is in the form of natural language [16]. This can be regarded as a further, particularly challenging case of heterogeneity, either if text is considered in its most primitive form as strings of characters, or if it is modelled as a syntactic or semantic network as emitted by one of the state-of-the-art natural language processing libraries, such as CoreNLP [37].

Automatic integration could solve this challenge, either from fully structured and disambiguated heterogeneous sources, or from syntactic/semantic networks produced using existing text mining methods. However, it is often an AI-hard problem, especially since many applications may expect knowledge bases to have a high accuracy. This is because the accuracy is generally high in linked data KBs [22], owing to the fact that many KBs were built manually (e.g., folkso-

*The research leading to these results has received funding from the Danish Council for Independent Research (DFF) under grant agreement No. DFF-4093-00301, as well as the DARPA SocialSim program.

ISSN 0921-7126, IOS Press. All rights reserved

nomical approaches as for Wikidata) or using integration rules that exploit highly reliable markup and hyperlinking structures (Wikipedia infoboxes and taxonomical links in DBpedia [34] and YAGO [52]).

In this paper, we present two attempts at tackling these issues, both related to FrameBase, a dataset that has been able to integrate a wide range of knowledge from other sources.

Firstly, we describe Klint (Knowledge integrator), a Web-based system enabling semi-automatic schema integration. Given one or more existing RDF ontologies, Klint generates tentative integration rules for these ontologies by representing mappings in a unified schema. For this unified schema, Klint relies on FrameBase [43], a wide-coverage, highly expressive and extensible schema that can be used to represent and integrate [46] a wide spectrum of knowledge from many sources in a homogeneous and seamless way.[1] Simultaneously, Klint offers an agile and simple interface that enables the user to inspect and adapt the tentative integration rules, achieving the desired balance between precision and scalability.

Secondly, we introduce a method for adding multilingual support to FrameBase. There are already several systems to extract FrameBase-modelled knowledge from natural language [2, 11], but these are restricted to input sources given in the English language. Linked data is increasingly being embraced in domains such as government data, digital libraries, cultural heritage, and linguistics. As an increasing number of organizations around the world seek to adopt the principles of Linked data, it is important to also facilitate the interlinking of data captured in different natural languages. This is particularly important for multinational institutions such as within the European Union. We evaluate results obtained for the Swedish language, but the method is generalizable to other languages.

This paper is structured as follows. Section 2 discusses related work in this area. Section 3 introduces the FrameBase schema, on top of which the work described in this paper is based. Section 4 describes the details of our Klint system for semi-automatic integration of heterogeneous structured data. Section 5 presents the algorithms used to add multilingual support to FrameBase, and provides an evaluation of the results. Finally, Section 6 provides a conclusion.

[1] See `http://www.framebase.org/`.

2. Related work

In this section we present previous research that is related to the work described in this paper. Section 2.1 describes work in the field of ontology alignment. Section 2.2 describes existing tools for ontology visualization and edition, and for the creation of mappings between different ontologies. Section 2.3 describes existing work addressing the problem of linguistic heterogeneity and multilinguality in the context of ontologies and linked data.

2.1. Ontology Alignment

Integrating knowledge from different sources is a long-standing, ubiquitous problem [30]. In the context of linked data and ontology alignment [15, 21], there has been extensive work in tasks such as identifying equivalent classes from different ontologies, and in some cases also equivalent instances and properties [4, 35, 40, 51].

Since knowledge can be expressed in structurally different ways, one-to-one equivalences are not sufficient to exhaustively connect equivalent information across knowledge bases, and complex mappings become necessary. There has been some work addressing the problem of declaring and finding these mappings, either semi-automatically or fully automatically. The EDOAL (Expressive and Declarative Ontology Alignment Language) format [13] has been proposed to express complex relationships between properties, and complex correspondence patterns between ontologies have been described and classified in an ontology [48]. However, these works do not address the problem of identifying these correspondence patterns. The iMAP tool [20] searches a space of possible complex relationships between the values of entries in two knowledge bases, e.g., `room-price = room-rate * (1 + tax-rate)`. However, it is limited to certain types of attribute combinations. The S-Match tool [29] makes use of formal reasoning to prove possible matches between ontology classes, involving union and intersection operators, but it does not address complex matching of properties beyond this. The work of Ritze et al. [42] uses a rule-based approach to detect specific kinds of complex alignment patterns between entries in small ontologies.

FrameBase [43, 44] declares integration rules using SPARQL CONSTRUCT queries, which enables arbitrarily complex mappings between external knowledge bases and a hub schema based on frames (classes repre-

senting events or processes) and frame elements (properties of frames), extracted from FrameNet [25, 47]. Different methods have been developed to automatically create certain types of complex rules [44]. This will be explained in more depth in Section 3.

2.2. Graphical Tools for Addressing Structural Heterogeneity

There are several tools for visualizing linked data [7, 12, 28], but these do not provide facilities to create arbitrary mappings.

WebProtégé [54] is a widely used ontology engineering tool for RDF, with extensive support for OWL reasoning, but it is designed to facilitate creating individual ontologies, not mappings between them.

LodLive [9] allows users to browse RDF resources using a dynamic visual graph and to create 1-to-1 links between resources stored in different endpoints. However, it does not allow for creating the more complex kinds of mappings we consider in this work.

Fusion [1] provides facilities to create mappings that are more complex than 1-to-1 relations between entities, but it is still limited to mappings to a single relation in the source ontology, and hence does not support arbitrary mappings. More specifically, it is limited to one of the following two kinds.

- Mapping a single property in the source ontology to a path[2] in another, but not allowing arbitrary mappings between graphs. For instance, in GeoNames[3], the property `geo:parentFeature` connects cities with provinces and also provinces with countries. Hence, the pattern X `geo:parentFeature / geo:parentFeature` Y can be mapped to X `locatedIn` Y to connect a city with the country it is located in. The interface for this kind of mapping is graphical, but not based on graphs.
- Fusion also allows for mapping a single property in the source ontology to a complex pattern in another, which is defined in the imperative programming language Ruby.

FrameBase and Klint, on the contrary, allow complex mappings by means of arbitrarily complex but declarative graph patterns implemented in SPARQL.

[2] In the sense of property paths in SPARQL [31], i.e., a sequence of properties.

[3] `http://www.geonames.org/ontology/`

2.3. Linguistic Heterogeneity

There are knowledge bases that include multilingual labels, such as YAGO 3 [36], BabelNet [39], MENTA [17], and DBpedia [34]. A large part of the entities and relations in these ontologies are obtained from Wikipedia articles and their mutual hyperlinks. As part of this approach, multilingual labels can be obtained by exploiting the cross-lingual links in Wikipedia [18], which constitutes a fundamental part of these ontologies. However, these ontologies are not explicitly built with the purpose of knowledge integration.

In order to solve the problem of linguistic heterogeneity in the context of knowledge bases, there has been work on linking knowledge entries with lexical labels in different languages, using machine translation [26, 27], graph models [56], or a combination of dictionary and monolingual ontology mapping tools [53].

Our work differs from these two types of previous efforts in several ways. First, we do not use the cross-lingual links in Wikipedia, as done by YAGO 3 and DBpedia. Instead, we add multilingual labels to FrameBase by exploiting its connections to WordNet and FrameNet as well as the existence of high-quality multilingual versions of these linguistic resources. Second, we do not attempt to directly and reciprocally interlink multilingual knowledge bases, since FrameBase is meant as a central and unique hub for integrating knowledge, where multilinguality can be expressed by means of labels that will be annotated with Lemon [38], reflecting their language and morphosyntactic properties.

3. Brief Overview of FrameBase

FrameBase [43–46] provides a unified schema designed to integrate knowledge from heterogeneous datasets such as those in the linked data cloud. What makes FrameBase particularly well-suited for integrating heterogeneous knowledge is that it is highly expressive, both from a lexical perspective (containing thousands of lexical entries), but also structurally, because the frame-based representation it relies on can represent n-ary relations in an extensible way.

The backbone of FrameBase stems from FrameNet [25, 47], a linguistic resource that compiles frames to annotate how specific words, denoted as Lexical Units (LUs), evoke them when they appear in text. FrameBase extends its frame coverage using

synsets, which are sets of synonymous disambiguated words taken from another linguistic resource called WordNet [23]. However, it is possible to understand FrameBase without detailed knowledge of these resources.

The core elements of the FrameBase schema are classes that correspond to frames, which are conceptual structures representing situations, events, or processes of any kind. The frame classes have properties called Frame Elements (FEs), in the sense that each of these properties has a specific frame class as its `rdfs:domain`. For instance, a `frame:Forming_relationships.wedding.noun` frame class has FE properties such as `Partner_1`, `Partner_2`, `Means`, `Place`, and `Time`. A `frame:Building.construction.noun` frame class has FE properties such as `Agent` (the entity that performs the construction), `Components`, `Created_entity`, `Instrument`, `Manner`, `Place` and `Time`. These two frame classes are examples from the lowest part of the frame hierarchy in FrameBase, which contains specific terms, such as "wedding" and "construction". These terms possess very specific meanings, and they are also represented as lexical labels (using `rdfs:label`), although additional labels may exist. In the frame IRI, the terms are qualified with the name of upper, more general frames, which allows us to cope with polysemous words. In the examples above, the macroframes are "Forming relationships" and "Building". As an example of polysemy, the frame class `frame:Containing.house.verb` represents a situation in which a `Container` holds some `Contents` within its physical boundaries, whereas the frame class `frame:Provide_lodging.house.verb` represents situations in which a `Host` provides a temporary residence for a `Lodger`.

Figure 1 provides an example of how FrameBase represents the knowledge that two entities, John and Mary, married in 1964 (with the frame classes and FE properties depicted in blue).

In general, frame classes in FrameBase are organized into a hierarchy of three different kinds of frames: *microframes*, *miniframes*, and *macroframes*. Figure 2 shows some example frames to illustrate this hierarchy.

- Microframes have very specific meanings and are directly connected to sense-disambiguated terms. Microframes are divided into two categories.
  - LU-microframes, each associated with an LU in FrameNet. Some examples are illustrated as green nodes in Figure 2, such as `:Quitting_a_place.defection.noun`.
  - synset-microframes, each associated with a synset in WordNet. Some examples are illustrated as blue nodes in Figure 2, such as `:Quitting_a_place.00055315.desertion.n`.
- Miniframes cluster together microframes with near-equivalent meaning. Two examples are illustrated as pink nodes in Figure 2: `:Quitting_a_place.m.defect.verb` and `:Quitting_a_place.m.retreat.verb` (in order to be more easily interpretable, they are represented with one of the inheriting LUs, but the `.m.` component distinguishes the IRIs from microframes with the same LU).
- Macroframes, finally, constitute an upper level of general types of situations or events. For example, `:Quitting_a_place` in Figure 2.

The purpose of miniframes is offering an intermediate level of granularity, in which microframes representing equivalent concepts can be merged ("wedding", "spouse", and "marry" are under the same miniframe), thus reducing data sparsity. Still, a conflation of clearly different concepts is avoided, as would happen if macroframes were used ("wedding", "spouse", and "marry" are under the same macroframe as "divorce")

The FrameBase hierarchy contains around 130,000 frames in total, and supports efficient RDFS inference with a few extensions [10].

The FE properties are also imported from FrameNet. FE properties change across macroframes, but microframes and miniframes inheriting a given macroframe share the same FE properties. For instance, both microframes `frame:Building.assembly.noun` and `frame:Building.construct.verb`, which are subclasses of the macroframe `frame:Building`, share FE properties such as `fe:Building.Created_entity` (the entity that is created in the act of building), `fe:Building.Agent` (the agent that builds the created entity), and `fe:Building.Components` (the components that form the created entity).

The main purpose of FrameBase is being able to represent a wide range of knowledge in a homogeneous

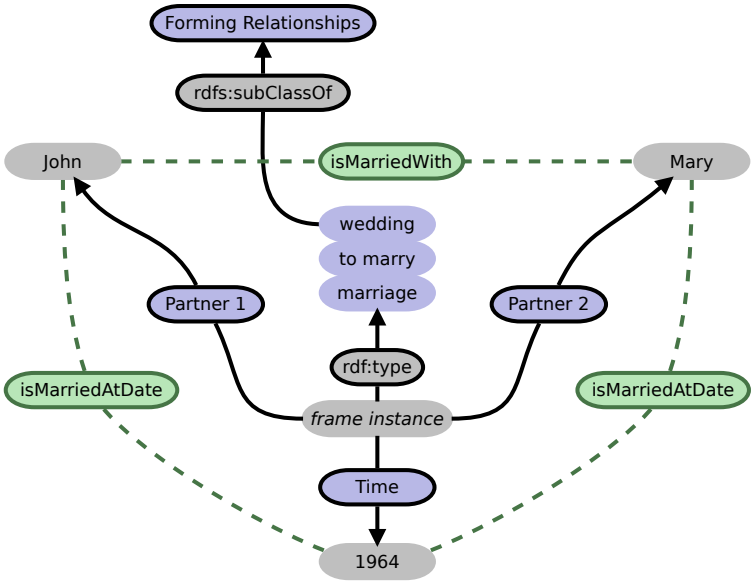


Fig. 1. Example of how FrameBase represents the knowledge that two entities, John and Mary, married in 1964. Blue entities constitute reified knowledge while green ones constitute dereified knowledge.

way. This should ease the difficulty associated with jointly querying data represented by an unbounded mix of different schemas that are heterogeneous not only at a lexical level but also structurally. This structural heterogeneity also prevents linking data by means of binary properties such as `owl:sameAs` (for instance, consider a knowledge base with a property `marriedWith` and another with a class `Marriage`).

Knowledge can directly be created under the FrameBase schema, or integrated from existing knowledge bases by means of integration rules (usually implemented as SPARQL CONSTRUCT queries), which allow what could be termed as "complex linked data", or "non-binary linked data".

The frame-based representation is the core of FrameBase and is chosen because it is very expressive and unambiguous, while striking a balance in terms of space complexity. One cannot unambiguously represent multiple marriages of a given person using `marriedWith` properties, unless some workarounds are used that entail a much higher triple overhead [43]. However, FrameBase additionally provides a second level of representation that is more concise, both in the knowledge base and in queries, in which only two FEs are specified for a frame. In this representation, the two FE properties, e.g. the two partners in a marriage, are directly connected via Direct Binary Predicates (DBPs). Figure 1 shows DBPs in green. The two kinds of representations that FrameBase offers are seamlessly connected via so-called reification and dereification rules. Each type is the converse of the other, and both can be expressed together as biconditional ReDer (Reification-Dereification) rules. Below we provide some examples. For instance, the first example rule converts between a triple (connecting a transformed entity with the cause for its transformation), and a pattern (with an entity representing a transformation event that is connected to the transformed entity and the cause). The latter representation is more verbose, but is necessary when additional information needs to be specified, such as the time of the transformation or the resulting entity.

```
PREFIX frame: <http://framebase.org/frame/>
PREFIX fe: <http://framebase.org/fe/>
PREFIX : <http://framebase.org/dbp/>
```

```
?S :Cause_change.isTransformedByCause ?O
  ↕
?R a frame:Cause_change.transform.verb ,
?R fe:Cause_change.Entity ?S ,
?R fe:Cause_change.Cause ?O .
```

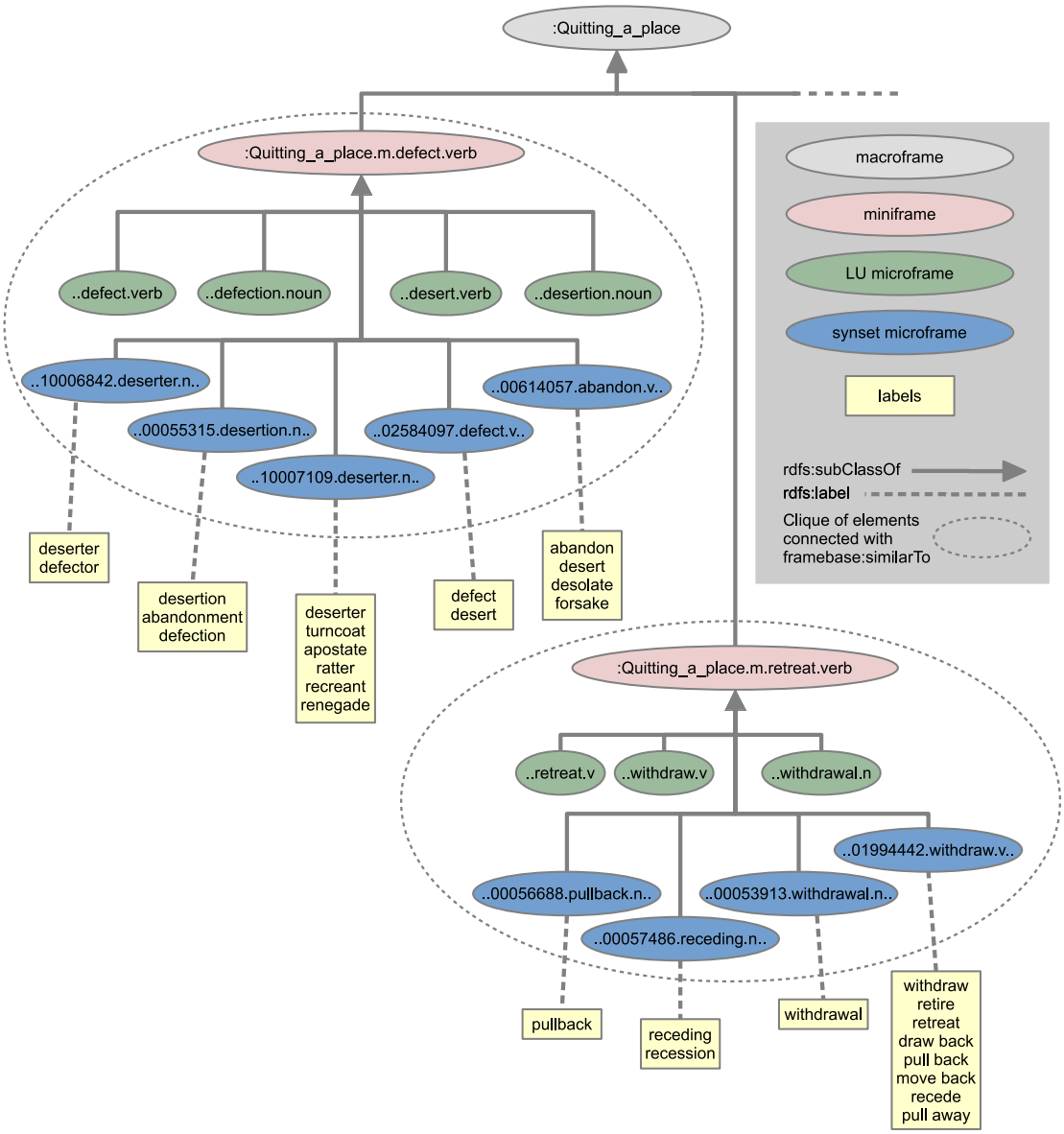


Fig. 2. Example illustrating the hierarchy of frames. The prefix before the colon is "`http://framebase.org/frame/`". Here, ".." represents further stemming of the IRIs in order to fit the text in the nodes.

```
?S :Cause_to_amalgamate.isUnitedBy ?O
↕
?R a frame:Cause_to_amalgamate.unite.verb ,
?R fe:Cause_to_amalgamate.Parts ?S ,
?R fe:Cause_to_amalgamate.Agent ?O .
```

```
?S :Fluidic_motion.flowsFromSource ?O
↕
?R a frame:Fluidic_motion.flow.verb ,
?R fe:Fluidic_motion.Fluid ?S ,
?R fe:Fluidic_motion.Source ?O .
```

Figure 3 provides a general overview of FrameBase. This paper also introduces a new version of FrameBase, with an extended set of synset-microframes and a more homogeneous structure in the IRIs. Previous versions of FrameBase (1.x) used identifiers for synset-microframes that included the name of the parent macroframe [43]. The parent macroframe was identified using an automatic method that was not able to map the entirety of WordNet Synsets into FrameBase. Some synsets should not even be mapped to an existing macroframe because this may not be available

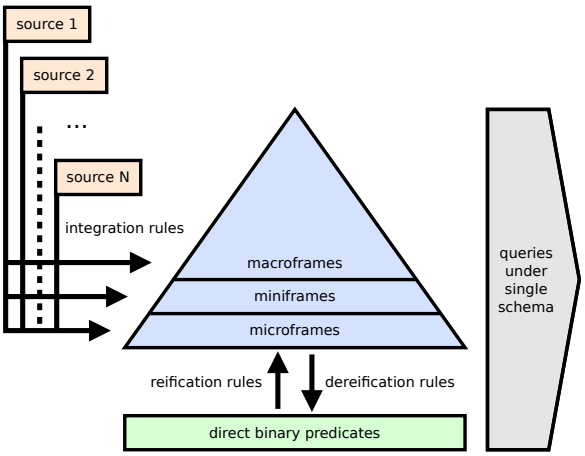


Fig. 3. Overview of the structure of FrameBase system (ontology and rules).

in the original FrameNet resource. However, including these orphan synset-microframes turned out to be advisable because even without a proper integration into the upper hierarchy (which can be added later), these synsets increase the expressivity of FrameBase. Hence, a new syntax has been defined for FrameBase 2.0, which removes parent frame information from synset-microframe IRIs, and allows orphan synset-microframes to be defined as children of the upper frame `frame:Top_frame` (which does not exist in FrameNet). Accordingly, the number of synset-microframes has increased from 6,418 to 117,659.

We have developed an integration engine that uses ReDer rules to map properties from source KBs into FrameBase, matching them with DBPs after a process of canonicalization [44, 46]. Figure 4 includes an example. In order to map classes from the source KBs, a Support Vector Machine model is trained to classify (Class,Frame) pairs based on a range of lexical similarity features.

Because of its ties to computational linguistics, FrameBase is not only appropriate for integrating heterogeneous structured knowledge, but also for integrating knowledge extracted from natural language. Currently, there are several third-party systems that extract FrameBase-modelled knowledge from natural language [2, 11], but these are restricted to source texts provided in the English language.

4. Klint

We will describe Klint from two different angles. First, Section 4.1 functionally describes the features of the application. Then, Section 4.2 provides non-functional details about the algorithm used for representing the graph. Finally, Section 4.3 illustrates the steps in Klint to edit a graph representing an integration rule.

4.1. Working Modes

Klint supports three working modes:

- The *Assisted Schema Integration* mode, which helps a user curate and create rules to integrate knowledge from external sources into FrameBase. It is explained in detail in Section 4.1.1.
- The *Visual Knowledge Building* mode, which assists a user in creating FrameBase knowledge directly. It is explained in detail in Section 4.1.2.
- The *Visual Query Building* mode, which assists a user in creating queries to retrieve existing Frames. It is explained in detail in Section 4.1.3.

4.1.1. Assisted Schema Integration Mode

Under the *Assisted Schema Integration Mode*, the user can integrate one or more entire knowledge bases (KBs) into FrameBase with reduced effort in comparison to an entirely manual or non-graphical approach. The input KBs can be loaded from an RDF file or a SPARQL endpoint. Other formats can also be used after pre-processing with a suitable RDF converter[4].

Integration Heuristics. Klint automatically creates complex integration rules for each element (i.e. beyond 1-to-1 equivalences between entities) in the source schema, using integration algorithms based on linguistic annotations in FrameBase [46]. These heuristics can create two kinds of integration rules:

- Property-Frame integration rules, where a property P in the source KB is mapped to a pattern consisting of an instantiated frame (that represents a situation or eventuality evoked by P). The frame is connected to the subject and object of P by two FE properties that represent "semantic roles" [24]. Figure 4 includes examples of these rules. These integration rules are created by matching the properties of the source KB with the FrameBase DBPs after a process of canonicalization [44, 46].
- Class-Frame integration rules, where a class is mapped to a frame instance, and some of its outgoing properties are mapped to FE properties of that frame, when there is textual overlap for all of them.

[4] `http://www.w3.org/wiki/ConverterToRdf`

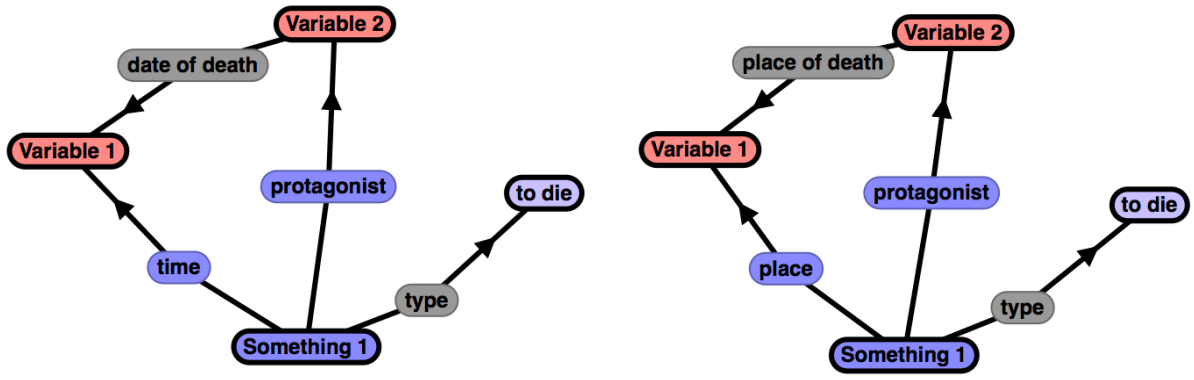


Fig. 4. Two rules automatically created by the integration engine, integrating different properties from Wikidata: "time of death" and "place of death". There is some knowledge overlap among them (the death event of type "to die" and the property "protagonist"), and also with the results from the rule in Figure 5. This is impossible to capture by simply linking the source schemas via equivalence or subsumption properties.

Graphical Interface. Each integration rule is represented as a graph with nodes and edges in the right pane of Klint's graphical interface (Figure 5). Users can navigate across different integration rules with the buttons at the top bar, making modifications in a given graph if necessary. The nodes can be of different types.

- **Grounded nodes** represent single entities, and can themselves be of three types:
  - **Source nodes** (green) represent resources from the source KB and connect two variable nodes.
  - **FrameBase nodes** (blue) represent FrameBase resources and also connect variable nodes. They provide the *translation* of the source pattern to FrameBase.
  - **Auxiliary nodes** (gray) represent resources from third-party KBs, usually representing common idioms or very specific entities.
- **Variable nodes** (presented in red) represent universally quantified variables over entities. They bind the pattern from the source KB to the integrated FrameBase pattern. The remaining nodes are classified according to the type of entity they represent.

The nodes are connected via directed edges representing triples. Since an RDF triple involves three elements, each triple is represented by two successive edges, one from the subject to the predicate and another from the predicate to the object.

In the *Assisted Schema Integration Mode*, both edges and any of the above-mentioned types of nodes can be added, deleted, and edited. When a node is selected, the node is highlighted and the left panel is activated, where the user can change its name and unique identifier (in RDF, this is an IRI). Klint automatically deduces which consecutive pairs of edges define triples.

Automatic Suggestions. When a new FrameBase node is created, its identifier is initially unspecified. Klint will try to use its context (the type of the neighboring nodes) and the integration engine to suggest possible values automatically. Specifically, if the node is a property emanating from a frame class, the FE properties of that frame class will be suggested, and if it is the subject of a FE property, the frame class to which this FE property belongs will be suggested. In case there are too many suggestions, the search function may be used instead.

4.1.2. Visual Knowledge Building

In this mode, the user is able to introduce FrameBase nodes as well as external nodes, but not variable nodes. Accordingly, the resulting graph then represents knowledge rather than a query, and can be exported in any of the common RDF formats. Unlike the *Assisted Schema Integration* mode, this is not meant to produce massive amounts of data from external structured sources, but rather serves as a source-neutral way of evaluating the expressivity of FrameBase when creating small examples of knowledge.

4.1.3. Visual Query Building

In this mode, the user can introduce FrameBase nodes and variable nodes, but not external nodes. The resulting graph represents any kind of query, with the purpose of retrieving existing knowledge under the

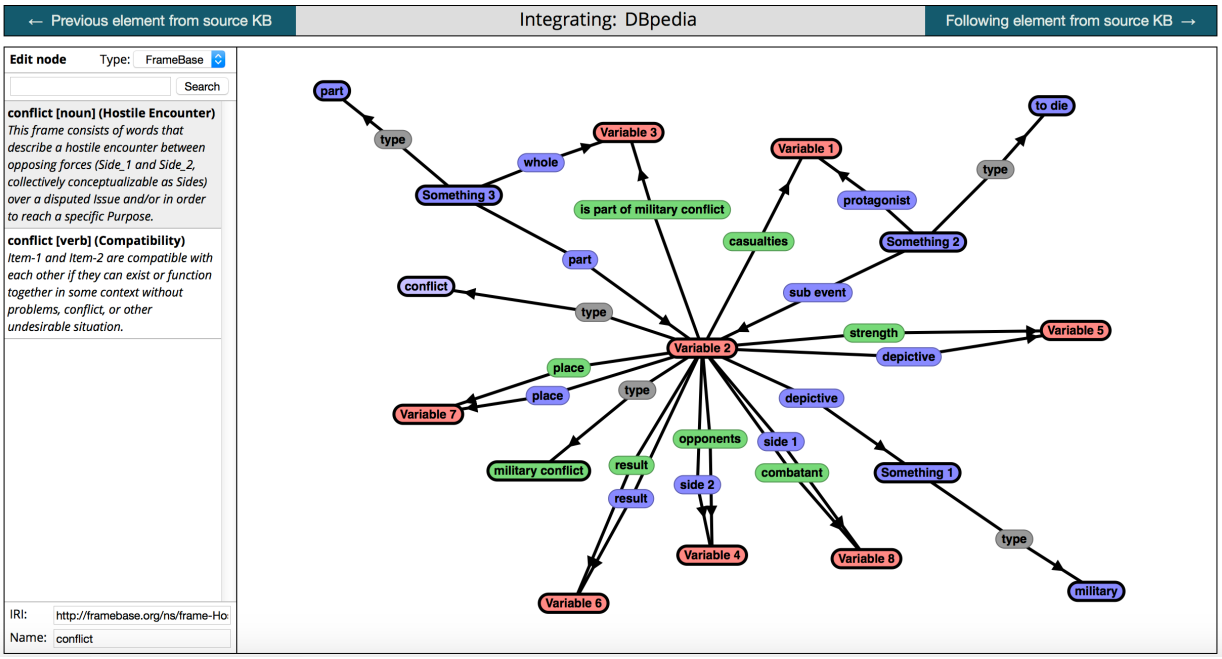


Fig. 5. Example of Klint's interface: integrating elements from DBpedia – Klint used the contextual and lexical information from the source elements to suggest two candidate values for the integrated type (selected node, "conflict"), for which the correct assigned value, `Hostile_encounter-conflict.n` was the first suggestion. The FrameBase properties were auto-inserted and some with high lexical overlap were automatically integrated as well. The complex structures that invoke some additional frames were created using the direct search function.

FrameBase schema, instead of an integration rule implemented specifically as a CONSTRUCT query, as in the case of the *Assisted Schema Integration* mode.

Specifically, the user may choose between the following different options:

- Obtain a SELECT SPARQL query, suited for a FrameBase KB, selecting all variables.
- Obtain a CONSTRUCT SPARQL query, suited for a FrameBase KB, extracting all the knowledge that follows a given pattern.
- Run the SELECT/CONSTRUCT SPARQL query directly, visualizing the results.

4.2. *Representation Layer*

RDF graphs are commonly regarded as directed labelled graphs, with each Subject-Predicate-Object triple as an edge between the subject and the object, and the predicate as its edge label. However, this is a simplification, as predicates can also be subjects or objects of some triples, and RDF graphs are truly bipartite graphs [32], where each triple is represented by two consecutive directed edges: subject–predicate and predicate–object ones. The graph representation in Klint maintains the bipartite model, so RDF is fully supported, but at the same time it maintains a data layer equivalent to a directed labelled graph (where each triple is an edge labelled with the predicate), which is the one used to provide an intuitive interface to the user. In order to make the presentation of this graph even more clear, Klint relies on a combination of visual clues: it uses physics simulation algorithms to maintain a similar orientation for edges of the same triple; it uses a distinctly different representation for subject and object nodes in relation to predicates; and it creates alias nodes for predicates, so if the same resource is used twice as a predicate in different triples, or as predicate in one and as a subject in another, then it is represented with two different nodes that are internally linked to the same resource. In order to create human-readable labels for nodes, Klint tries to find an explicit label for a node, and if it fails to find it, it uses a heuristic based on extracting the last part of the IRI, which is often human-readable.

When users want to make a modification in the graph, they can do so in a simple, visual way. Subject-Predicate-Object triples can be added or removed by adding or removing edges between nodes. The system automatically creates a new triple after subject-

predicate and predicate-object edges are created, sharing the predicate. Temporary links are shown for unfinished subject-predicate edges.

4.3. Editing an Integration Rule: Example

We next consider the steps one takes to edit a graph representing an integration rule so as to make it more precise. Figure 6 presents an integration rule that expresses that the DBpedia property `almaMater` can be represented in FrameBase by using an instance of the `Studying` frame with FE properties `Student` and `Institution`. While this, viewed as inference, is valid, it is not entirely complete, since the term and the property "alma mater" is only used for cases where the studies have been finished. This can be represented in different ways, but we will choose a relatively simple one for illustration purposes: adding a third FE property "Time" with a literal value "past".

While we have thus far not carried out any full-fledged user studies with untrained users, the manual construction of integration rules within the ePOOLICE EU project [45] showed that the use of a graphical interface of this sort would be advantageous.

5. Multilingual Support

We add multilingual support to FrameBase by adding multilingual labels to microframe classes, alongside the already existing ones in English. We consider this more appropriate than creating new microframe classes for terms in another language, since microframe classes are semantic elements that ought to be language-independent. While there may be connotational and denotational differences in nuance between translated terms, and there are terms that do not have a direct translation into other languages, the first phenomenon can be disregarded in most applications of knowledge representation and linked data, and the second phenomenon only pertains to a comparably small number of cases (and FrameNet as well as FrameBase allow for multi-word expressions, which can potentially alleviate this).

We do not consider multilingual labels of RDF terms such as FE properties, macroframes or FrameBase-specific classes and properties, because the lexical labels for these terms are not meant to be used in text mining.

5.1. Method Description

We implement our method for the case of Swedish, but it can be applied for another second language (which we will denote throughout the paper as $\mathscr{L}$). The method is a combination of two algorithms, which have different requirements and can work independently. This makes it more robust for the case that $\mathscr{L}$ does have partial coverage among computation lexicons—for instance, there is a WordNet extension but not a FrameNet one.

5.1.1. WordNet-based Method

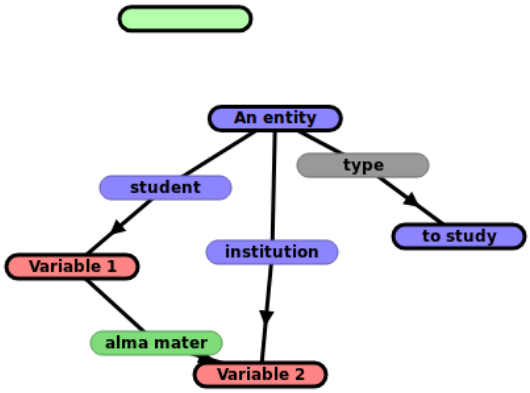
This method, described in Algorithm 1, depends on the availability of a mapping from Princeton WordNet synsets to some lexicon for the other language (we denote this mapping as $PwnMap()$). The lexicon can be a multilingual WordNet or another kind of lexicon. In our case, we use SALDO, a Swedish lexicon [6], for which there is a mapping consisting of 6,900 pairs of a WordNet word sense and a SALDO identifier[5]. SALDO identifiers are different from WordNet synsets in several ways. For instance, instead of being organized in a taxonomical hierarchy of "is a" relations, they are connected by means of a "non-classical" association relation that connects word senses with their primary and secondary descriptors. The relation between a word sense and its primary descriptor is the most common and also forms a tree, but unlike "is a", it connects a word sense with another one that is semantically and/or morphologically less complex, usually more frequent, stylistically more unmarked, and typically acquired earlier in first and second language acquisition [6]. Some examples are "fönster" (window) having "öppning" (opening) as primary descriptor (and "hus" or "house" as secondary descriptor), "operahus" (opera house) having "opera" as primary descriptor, and "dålig" (bad) having "bra" (good) as primary descriptor (and "motsats" or "opposite" as secondary descriptor).

This algorithm can be generalized for any language for which there is a multilingual WordNet version. Currently there are manually created open-source WordNets for over 30 languages, while many further languages are covered by automatically produced WordNets such as the Universal Wordnet [19].

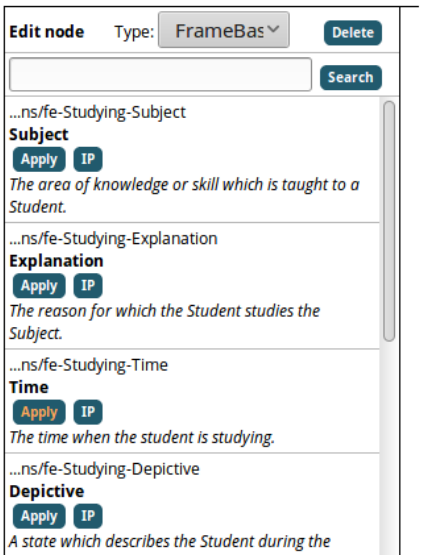
5.1.2. FrameNet-based Method

Algorithm 2 uses the following two resources.

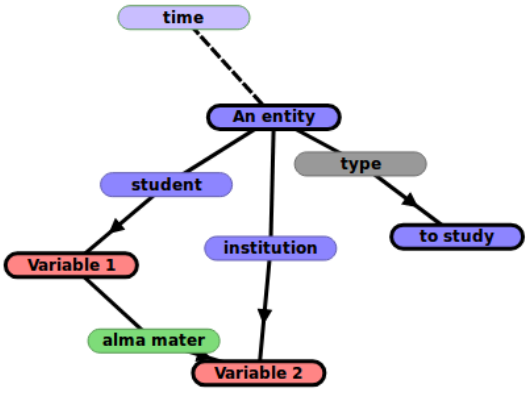
[5] `https://spraakbanken.gu.se/swe/resurs/wordnet-saldo`



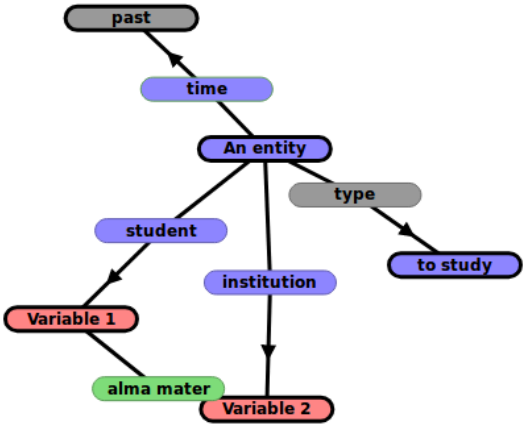
(a) The starting point is the initial integration rule in the lower part of the screen, which includes a frame instance labeled "An entity". After clicking the empty area in the upper space, a new node is created. Klint marks it green, assuming it belongs to a source KB.



(b) Once an edge has been created between the frame instance and the new node, Klint recognizes that this is meant to be a property whose domain is the type of the frame instance, i.e. the frame class, and therefore, when the new node is selected, it automatically proposes all the FE properties of that frame as possible values, including "Time".



(c) After selecting "Apply" for "Time", the new node is reassigned as a FrameBase node.



(d) After creating a literal node "past" and a second edge from "Time" to "Past", Klint recognizes the new triple and updates the rule.

Fig. 6.

Algorithm 1 WordNet-based Algorithm.

for all synset-microframe F in FrameBase **do**
  for all English-word-sense W in WordNet associated to the synset associated to F **do**
    if $PwnMap(W)$ exists **then**
      $\mathscr{L}$-word-sense $S \leftarrow PwnMap(W)$
      for all $\mathscr{L}$-lemma $L \in lemmas(S)$ **do**
        Add L as lexical label of M, with a language tag for $\mathscr{L}$.

Algorithm 2 FrameNet-based Algorithm.

for all microframe M in FrameBase **do**
  for all $L \in D_{en-\mathscr{L}}(posAnnotatedLabel_{en}(M))$ **do**
    if L is in $LU_{\mathscr{L}}(macroframeOf(M)$ **then**
      Add L as a label for M, with a language tag for $\mathscr{L}$.

- A function $LU_{\mathscr{L}}()$ that maps English Berkeley FrameNet frame names to sets of $\mathscr{L}$-language LUs. This can be obtained from any second-language FrameNet that uses frames from the English Berkeley FrameNet.
- A dictionary function $D_{en-\mathscr{L}}()$ that maps (lemma,POS) pairs between English and the second language $\mathscr{L}$, i.e. a POS-annotated but sense-oblivious translation table.

This algorithm relies on the premise that among the possible translations for the term associated with an English LU (the elements of $D(posAnnotatedLabel_{en}(M))$, those that are also present in the non-English FrameNet as terms associated with LUs of the same macroframe (i.e., belonging to $LU_{\mathscr{L}}(macroframeOf(M))$ are correct translations for the English LU. In other words, $\mathscr{L}$-language translations owing to unrelated senses (homonyms) of the English term are filtered out because they do not tend to appear as terms associated with LUs in the $\mathscr{L}$-language FrameNet. This assumption will be shown to hold true empirically in the results section.

We test this algorithm with Swedish in Section 5.2, but it can easily be extended to other languages.

- As mentioned above, $LU_{\mathscr{L}}()$ can be obtained from any second-language FrameNet that uses frames from the English Berkeley FrameNet. For our experiments in Swedish, we use SweFN++ [5]. Similar datasets for other languages are the German FrameNet from the SALSA project [8], the Spanish FrameNet [50] and the Japanese FrameNet [41]. There has also been work on automatic machine learning-based translations of FrameNet [55].
- Because $D_{en-\mathscr{L}}()$ does not need to map word senses, it can be obtained from any English–$\mathscr{L}$ dictionary that uses English senses different from WordNet or FrameNet, or makes no word sense distinction at all either for English or $\mathscr{L}$. For our experiments in Swedish, we obtain $D_{en-\mathscr{L}}()$ from DBnary [49], a dataset with multilingual lexical data extracted from Wiktionary. Wiktionary contains data for numerous further languages. The coverage for the results of both algorithms should be higher when applied to languages such as Spanish, German, French, or Chinese, which have a high number of speakers and consequently tend to be well covered in computational linguistics resources.

A particular example of a mistaken translation that is avoided by the algorithm is the following one. The LU-microframe `frame:Color.green.noun`, whose direct macroframe parent is `frame:Color`, has the label 'green', which has two Swedish translations in DBnary: 'grön' (green color) and 'miljövänlig' (green in the sense of environmentally friendly). Since the first is a LU in SweFN++ but the second is not, the first is chosen as the Swedish label for the LU-microframe, while the second one is filtered out.

5.2. Results and Evaluation

The results from Algorithm 1 covered 6,796 synset-microframes of a total of 117,659 in FrameBase 2.0, producing in total 7,331 Swedish labels. This does not

Table 1
Results of adding Swedish labels to FrameBase.

	synset-microframes	LU-microframes	labels	precision
Total in FrameBase	117,659	11,114	—	—
Algorithm 1	6,796	0	7,331	1 ‡
Algorithm 2	0	2,847	3,542	0,99

‡: Lossless use of manual mappings

Table 2
Examples of obtained Swedish annotations.

<http://framebase.org/frame/Motion.drift.verb>	rdfs:label	"driva"@sv
<http://framebase.org/frame/Corroding.rust.verb>	rdfs:label	"rosta"@sv
<http://framebase.org/frame/Achieving_first.discovery.noun>	rdfs:label	"upptäckt"@sv
<http://framebase.org/frame/Achieving_first.invent.verb>	rdfs:label	"uppfinna"@sv
<http://framebase.org/frame/Desirability.pathetic.adjective>	rdfs:label	"patetisk"@sv
<http://framebase.org/frame/Sufficiency.insufficient.adjective>	rdfs:label	"otillräcklig"@sv
<http://framebase.org/frame/Communicate_categorization.represent.verb>	rdfs:label	"föreställa"@sv
<http://framebase.org/frame/Being_located.located.adjective>	rdfs:label	"belägen"@sv
<http://framebase.org/frame/Hostile_encounter.war.noun>	rdfs:label	"krig"@sv
<http://framebase.org/frame/Rite.circumcision.noun>	rdfs:label	"omskärelse"@sv
<http://framebase.org/frame/Part_orientational.western.adjective>	rdfs:label	"västra"@sv
<http://framebase.org/frame/Accoutrements.goggles.noun>	rdfs:label	"glasögon"@sv
<http://framebase.org/frame/Experiencer_focus.abhor.verb>	rdfs:label	"hata"@sv
<http://framebase.org/frame/Experiencer_focus.abhor.verb>	rdfs:label	"avsky"@sv
<http://framebase.org/frame/Desirability.awful.adjective>	rdfs:label	"otäck"@sv
<http://framebase.org/frame/Likelihood.probable.adjective>	rdfs:label	"sannolik"@sv
<http://framebase.org/frame/Contingency.independent.adjective>	rdfs:label	"oberoende"@sv
<http://framebase.org/frame/Physical_artworks.photograph.noun>	rdfs:label	"fotografi"@sv
<http://framebase.org/frame/Intoxication.sober.adjective>	rdfs:label	"nykter"@sv
<http://framebase.org/frame/Containers.suitcase.noun>	rdfs:label	"resväska"@sv
<http://framebase.org/frame/Observable_body_parts.sole.noun>	rdfs:label	"fotsula"@sv
<http://framebase.org/frame/Mental_property.cynical.adjective>	rdfs:label	"cynisk"@sv
<http://framebase.org/frame/Word_relations.holonym.noun>	rdfs:label	"holonym"@sv
<http://framebase.org/frame/Stage_of_progress.sophisticated.adjective>	rdfs:label	"sofistikerad"@sv
<http://framebase.org/frame/Using.utilise.verb>	rdfs:label	"använda"@sv
<http://framebase.org/frame/Death.drown.verb>	rdfs:label	"drunkna"@sv
<http://framebase.org/frame/Successful_action.fail.verb>	rdfs:label	"fallera"@sv
<http://framebase.org/frame/Evaluative_comparison.comparable.adjective>	rdfs:label	"jämförbar"@sv
<http://framebase.org/frame/Food.cracker.noun>	rdfs:label	"kex"@sv
<http://framebase.org/frame/Calendric_unit.week.noun>	rdfs:label	"vecka"@sv
<http://framebase.org/frame/Attempt_suasion.discourage.verb>	rdfs:label	"avråda"@sv
<http://framebase.org/frame/Compliance.obey.verb>	rdfs:label	"lyda"@sv
<http://framebase.org/frame/Self_motion.jump.verb>	rdfs:label	"hoppa"@sv
<http://framebase.org/frame/Mental_property.naive.adjective>	rdfs:label	"naiv"@sv
<http://framebase.org/frame/Commitment.promise.verb>	rdfs:label	"lova"@sv
<http://framebase.org/frame/Food.egg.noun>	rdfs:label	"ägg"@sv
<http://framebase.org/frame/Judgment_communication.laud.verb>	rdfs:label	"prisa"@sv

need further evaluation for correctness because it relies on manual mappings.

The results from Algorithm 2 on LU-microframes produced 3,542 labels for 2,847 LU-microframes out of a total of 11,114. We evaluated 100 labels, of which 99 were correct. The labels were sampled randomly, with a uniform distribution, independently, without replacement. A label was judged correct if any possible English sentence was deemed possible so that (1) the English LU defining the LU-microframe was being used to evoke the macroframe, and (2) the translation of the sentence would include the Swedish label as translation of the English LU. The evaluation was jointly performed by two annotators, and it is available at `http://www.framebase.org`.

This evaluation yields a precision of 0.99, and the recall, under the simplifying assumption of one correct translation per LU, is 0.26. The somewhat low coverage is due to the low coverage of SweFn++ in relation to the Berkeley English FrameNet, but the high precision confirms the validity of the assumption made in Section 5. More details about the results are available in Table 1.

We also tested Algorithm 2 on synset-macroframes, and we used SweFn++ as gold standard for evaluation. In this case, it can only be applied to the 18,907 synset-microframes that have a parent other than `frame:Top_frame`. The obtained precision for a sample of 100 was rather low (0.38). This is due to two factors: synsets in WordNet are based on substantially more fine-grained sense distinctions in comparison with LUs in FrameNet, and SweFn++ is not complete (for instance, it has a constraint of one SALDO ID per WordNet synset). In order to alleviate this, we used a weighted synonymy graph included in SALDO, where SALDO IDs are weighted by similarity from 0 to 5. Expanding the gold standard with neighbors connected by a value higher or equal to 4 increased the precision to 67%, whereas using a threshold of 3 increased the precision to 93%. Table 2 lists a few correct examples of the resulting annotations.

Besides simple lexical annotations using `rdfs:label`, we also create morpho-syntactically rich annotations using Lemon [38].

6. Conclusion

We have presented two approaches to advance the integration of heterogeneous knowledge—both structured and non-structured—into FrameBase, which is a schema able to represent a wide range of knowledge using linguistic structures.

Klint is a Web-based framework that allows users to supervise an automatic integration of heterogeneous structured knowledge by providing a user-friendly graph-based interface that enables the reviewing and curation of complex integration rules produced by state-of-the-art integration algorithms. We have also demonstrated how FrameBase can be extended to support multilingual labels by reusing publicly available lexical resources, which has the potential to benefit text mining in languages other than English.

References

[1] Samur Araujo, Geert-Jan Houben, Daniel Schwabe, and Jan Hidders. Fusion–visually exploring and eliciting relationships in linked data. *The Semantic Web–ISWC 2010*, pages 1–15, 2010.

[2] Valerio Basile, Elena Cabrio, and Claudia Schon. Knews: Using logical and lexical semantics to extract knowledge from natural language. In *Proceedings of the 22nd European Conference on Artificial Intelligence (ECAI 2016)*, 2016.

[3] Christian Bizer, Tom Heath, and Tim Berners-Lee. Linked data–the story so far. *IJSWIS*, 5(3):1–22, 2009.

[4] Christoph Böhm, Gerard de Melo, Felix Naumann, and Gerhard Weikum. LINDA: Distributed Web-of-data-scale Entity Matching. In *Proceedings CIKM 2012*, CIKM '12, pages 2104–2108, 2012.

[5] Lars Borin, Dana Dannélls, Markus Forsberg, Maria Toporowska Gronostaj, and Dimitrios Kokkinakis. Swedish framenet++. *Swedish Language Technology Conference 2010*, 2010.

[6] Lars Borin, Markus Forsberg, and Lennart Lönngren. Saldo: a touch of yin to wordnet's yang. *Language resources and evaluation*, 47(4):1191–1211, 2013.

[7] Josep Maria Brunetti, Sören Auer, Roberto García, Jakub Klímek, and Martin Nečaský. Formal linked data visualization model. In *Proceedings of International Conference on Information Integration and Web-based Applications & Services*, IIWAS '13, New York, NY, USA, 2013. ACM.

[8] Aljoscha Burchardt, Katrin Erk, Anette Frank, Andrea Kowalski, Sebastian Padó, and Manfred Pinkal. The SALSA corpus: a German corpus resource for lexical semantics. In *Proceedings of LREC 2006*, pages 969–974, 2006.

[9] Diego Valerio Camarda, Silvia Mazzini, and Alessandro Antonuccio. Lodlive, exploring the web of data. In *Proceedings of the 8th International Conference on Semantic Systems*, I-SEMANTICS '12, pages 197–200, New York, NY, USA, 2012. ACM.

[10] Jeremy J. Carroll, Ian Dickinson, Chris Dollin, Dave Reynolds, Andy Seaborne, and Kevin Wilkinson. Jena: Implementing the semantic web recommendations. In *WWW '04*, 2004.

[11] Francesco Corcoglioniti, Marco Rospocher, and Alessio Palmero Aprosio. A 2-phase frame-based knowledge extraction framework. In *Proc. of ACM Symposium on Applied Computing (SAC'16)*, pages 354–361, 2016.

[12] Aba-Sah Dadzie and Matthew Rowe. Approaches to visualising linked data: A survey. *Semantic Web*, 2(2):89–124, 2011.

[13] Jérôme David, Jérôme Euzenat, François Scharffe, and Cássia Trojahn dos Santos. The Alignment API 4.0. *Semantic Web Journal*, 2(1):3–10, 2011.

[14] Gerard de Melo. Not quite the same: Identity constraints for the Web of Linked Data. In *Proceedings of AAAI 2013*, pages 1092–1098, Menlo Park, CA, USA, 2013. AAAI Press.

[15] Gerard de Melo, Fabian Suchanek, and Adam Pease. Integrating YAGO into the Suggested Upper Merged Ontology. In *Proceedings of the 20th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2008)*. IEEE Computer Society, Los Alamitos, CA, USA, 2008.

[16] Gerard de Melo and Gerhard Weikum. Language as a foundation of the Semantic Web. In *Proceedings of ISWC 2008*, volume 401 of *CEUR WS*, Karlsruhe, Germany, 2008. CEUR.

[17] Gerard de Melo and Gerhard Weikum. MENTA: Inducing multilingual taxonomies from Wikipedia. In *Proceedings of CIKM 2010*, pages 1099–1108. ACM, 2010.

[18] Gerard de Melo and Gerhard Weikum. Untangling the cross-lingual link structure of Wikipedia. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 844–853, Stroudsburg, PA, USA, July 2010. Association for Computational Linguistics.

[19] Gerard de Melo and Gerhard Weikum. UWN: A large multilingual lexical knowledge base. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, pages 151–156, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

[20] Robin Dhamankar, Yoonkyong Lee, AnHai Doan, Alon Halevy, and Pedro Domingos. iMAP: Discovering Complex Semantic Matches Between Database Schemas. In *SIGMOD 2004*, pages 383–394, 2004.

[21] Jérôme Euzenat and Pavel Shvaiko. *Ontology Matching*. Springer, 2007.

[22] Michael Färber, Frederic Bartscherer, Carsten Menne, and Achim Rettinger. Linked data quality of DBpedia, Freebase, OpenCyc, Wikidata, and YAGO. *Semantic Web*, (Preprint):1–53, 2016.

[23] Christiane Fellbaum, editor. *WordNet: An Electronic Lexical Database*. The MIT Press, 1998.

[24] Charles J Fillmore. The case for case reopened. *Syntax and semantics*, 8:59–82, 1977.

[25] Charles J Fillmore, Christopher R Johnson, and Miriam RL Petruck. Background to Framenet. *International journal of lexicography*, 16(3):235–250, 2003.

[26] Bo Fu, Rob Brennan, and Declan O'sullivan. Cross-lingual ontology mapping and its use on the multilingual semantic web. *MSW*, 571:13–20, 2010.

[27] Bo Fu, Rob Brennan, and Declan O'Sullivan. Cross-lingual ontology mapping–an investigation of the impact of machine translation. *The Semantic Web*, pages 1–15, 2009.

[28] Tong Ge, Yafang Wang, Gerard de Melo, and Haofeng Li. Visualizing and curating knowledge graphs over time and space. In *Proceedings of ACL 2016*. ACL, 2016.

[29] Fausto Giunchiglia, Pavel Shvaiko, and Mikalai Yatskevich. S-Match: an Algorithm and an Implementation of Semantic Matching. In *The Semantic Web: Research and Applications*, pages 61–75. Springer, 2004.

[30] Behzad Golshan, Alon Halevy, George Mihaila, and Wang-Chiew Tan. Data integration: After the teenage years. In *Proceedings of the 36th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, PODS '17, pages 101–106, New York, NY, USA, 2017. ACM.

[31] Steve Harris and Andy Seaborne. SPARQL 1.1 Query Language. W3C Recommendation, W3C Consortium, March 2013.

[32] Jonathan Hayes and Claudio Gutierrez. Bipartite graphs as intermediate model for RDF. In *ISWC 2004*, pages 47–61, 2004.

[33] Patrick Hayes and Peter Patel-Schneider. RDF 1.1 semantics. Technical report, W3C, 2014.

[34] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. DBpedia–a large-scale, multilingual knowledge base extracted from Wikipedia. *Semantic Web*, 6(2):167–195, 2015.

[35] Juanzi Li, Jie Tang, Yi Li, and Qiong Luo. RiMOM: A Dynamic Multistrategy Ontology Alignment Framework. *IEEE TKDE*, 21(8):1218–1232, 2009.

[36] Farzaneh Mahdisoltani, Joanna Biega, and Fabian Suchanek. Yago3: A knowledge base from multilingual wikipedias. In *7th Biennial Conference on Innovative Data Systems Research*. CIDR Conference, 2014.

[37] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60, 2014.

[38] John McCrae, Dennis Spohr, and Philipp Cimiano. Linking lexical resources and ontologies on the semantic web with lemon. In *The semantic web: research and applications*, pages 245–259. Springer Berlin Heidelberg, 2011.

[39] 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.

[40] Axel-Cyrille Ngonga Ngomo and Sören Auer. LIMES: a time-efficient approach for large-scale link discovery on the web of data. IJCAI'11, pages 2312–2317. AAAI Press, 2011.

[41] Kyoko Ohara. Semantic annotations in japanese framenet: Comparing frames in japanese and english. In *Proceedings of LREC 2012*, Istanbul, Turkey, May 2012. European Language Resources Association (ELRA).

[42] Dominique Ritze, Christian Meilicke, Ondrej Sváb-Zamazal, and Heiner Stuckenschmidt. A Pattern-based Ontology Matching Approach for Detecting Complex Correspondences. In *OM'10*, 2008.

[43] Jacobo Rouces, Gerard de Melo, and Katja Hose. FrameBase: Representing N-ary Relations using Semantic Frames. In *Proceedings of the 12th Extended Semantic Web Conference, ESWC*, 2015.

[44] Jacobo Rouces, Gerard de Melo, and Katja Hose. Heuristics for Connecting Heterogeneous Knowledge via FrameBase. In *ESWC'16*, 2015.

[45] Jacobo Rouces, Gerard de Melo, and Katja Hose. Representing Specialized Events with FrameBase. In *DeRiVE '15*, 2015.

[46] Jacobo Rouces, Gerard de Melo, and Katja Hose. Complex Schema Mapping and Linking Data: Beyond Binary Predicates. In *LDOW'16*, 2016.

[47] Josef Ruppenhofer, Michael Ellsworth, Miriam R.L. Petruck, Christopher R. Johnson, and Jan Scheffczyk. *FrameNet II: Extended Theory and Practice*. ICSI, 2006.

[48] François Scharffe and Dieter Fensel. Correspondence patterns for ontology alignment. EKAW '08, pages 83–92, Berlin, Heidelberg, 2008. Springer-Verlag.

[49] Gilles Sérasset. Dbnary: Wiktionary as a lemon-based multilingual lexical resource in rdf. *Semantic Web*, 6(4):355–361, 2015.

[50] Carlos Subirats. Spanish FrameNet: A frame-semantic analysis of the Spanish lexicon. In Hans Boas, editor, *Multilingual FrameNets in Computational Lexicography: Methods and Applications*, pages 135–162. Mouton de Gruyter, Berlin/New York, 2009.

[51] Fabian M. Suchanek, Serge Abiteboul, and Pierre Senellart. PARIS: Probabilistic Alignment of Relations, Instances, and Schema. *Proc. VLDB Endow.*, 5(3):157–168, November 2011.

[52] Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. YAGO: A Large Ontology from Wikipedia and WordNet. *Web Semantics: Science, Services and Agents on the World Wide Web* , 6(3):203 – 217, 2008.

[53] Cássia Trojahn, Paulo Quaresma, and Renata Vieira. A framework for multilingual ontology mapping. In *Proceedings of LREC 2008*. ACM, 2008.

[54] Tania Tudorache, Csongor Nyulas, Natalya F Noy, and Mark A Musen. Webprotégé: A collaborative ontology editor and knowledge acquisition tool for the web. *Semantic web*, 4(1):89–99, 2013.

[55] Oliver Čulo and Gerard de Melo. Source-Path-Goal: Investigating the cross-linguistic potential of frame-semantic text analysis. *it - Information Technology*, 54, 2012.

[56] Zhichun Wang, Juanzi Li, Zhigang Wang, and Jie Tang. Cross-lingual knowledge linking across wiki knowledge bases. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12, pages 459–468, New York, NY, USA, 2012. ACM.