A Frame-Based Approach for Integrating Heterogeneous Knowledge Sources

Gonzalez, Jacobo Rouces

DOI (link to publication from Publisher):
10.5278/vbn.phd.engsci.00102

Publication date:
2016

Document Version
Publisher's PDF, also known as Version of record

A FRAME-BASED APPROACH FOR INTEGRATING HETEROGENEOUS KNOWLEDGE SOURCES

BY
JACOBO ROUCES

DISSERTATION SUBMITTED 2016

AALBORG UNIVERSITY
DENMARK

A Frame-Based Approach for Integrating Heterogeneous Knowledge Sources

Ph.D. Dissertation

Jacobo Rouces

Dissertation submitted April 18, 2016

Dissertation submitted: April 18, 2016

PhD supervisor: Assoc. Prof. Henrik Legind Larsen
Dept. of Electronic Systems, Aalborg University

Assistant PhD supervisor: Assoc. Prof. Katja Hose
Dept. of Computer Science, Aalborg University

PhD committee: Associate Professor Daniel Ortiz-Arroyo (chairman)
Aalborg University, Denmark

Professor Pierre Nugues
Lund University, Sweden

Professor Anders Søgaard
University of Copenhagen, Denmark

PhD Series: Faculty of Engineering and Science, Aalborg University

ISSN (online): 2246-1248
ISBN (online): 978-87-7112-557-3

Published by:
Aalborg University Press
Skjernvej 4A, 2nd floor
DK – 9220 Aalborg Ø
Phone: +45 99407140
aauf@forlag.aau.dk
forlag.aau.dk

Printed in Denmark by Rosendahls, 2016

Abstract

Large knowledge bases are increasingly available in the web, linked to each other so that their information can be combined and joined and they can be queried as a single knowledge base or navigated by software agents. The resulting global knowledge base is usually referred to as linked data, linked open data, linked open data cloud, or the semantic web.

Commonly used standards for these knowledge bases specify a simple and homogeneous syntax based on triples that can be interpreted, under some simplifications, as defining a labelled directed graph, where the nodes and the edges are identified with internationalized resource identifiers that use domains owned by the stakeholder that publishes each graph. This makes easier the aggregation of data, which can be reduced to a union of triples.

However, the use of different modelling patterns in different graphs requires any structured query over the global graph to combine all possible modelling patterns that may exist in it, including all possible nodes or edges that are synonymous, which is impractical. This semantic heterogeneity also prevents certain pieces of equivalent knowledge to be linked, because what is represented by one node in a graph may be represented implicitly, as some sort of graph pattern, in another graph using a different modelling pattern, making impossible the linking by means of simple equivalence edges. Furthermore, certain existing modelling patterns are too verbose and inefficient, while others lack expressiveness and fail to capture essential connections in the data.

This thesis describes the creation of FrameBase, a system that reuses theory and resources from linguistics and cognitive science to provide different connected layers of representation that combine the expressiveness of some modelling patterns with the conciseness of others, while at the same time providing a common basic vocabulary that can be extended by stakeholders. The FrameBase multilayered system of modelling patterns model allows representing a wide range of knowledge in a way that allows truly seamless integration and querying of data.

The thesis also introduces different methods to integrate knowledge from external knowledge bases, and eventually from any source of structured data.

The methods range from manual to automatic, with different semi-automatic approaches being developed. Automatic methods exploit the ties of FrameBase with linguistics to create complex mappings with the names of elements in the external knowledge bases. The manual approach is streamlined with a web application that allows a user to build mappings in a simple graphical manner. Two different semi-automatic approaches are implemented: one enhancing automatic methods with simple heuristics specific to each knowledge base, and another enhancing the manual method by re-using the automatic and semi-automatic methods as part of a suggestion and search engine in the web application. The results of all automatic methods are evaluated with human annotators.

Additionally, FrameBase's ties to linguistics also provide potential methods to interface with natural language, either for the purpose of text mining or question answering. The thesis discusses these potential methods at the end, hinting at possible lines of future work.

Resumé

Store videnbaser er i stigende grad offentliggjorte på webbet, knyttet sammen således at deres viden kan forenes og de kan forespørges på en gang eller navigeres ved softwareagenter. Den resulterende globale videnbase er normalt omtalt som *linked data*, *linked open data*, *linked open data cloud*, eller det semantiske web.

Almindeligt brugte standarder for disse videnbaser angiver en simpel, homogen syntaks baseret på triples der under nogle forenklinger kan fortolkes som en mærket orienteret graf, hvor knude og kanter er identificeret med IRIs (Internationalized Resource Identifiers), der bruger domæner ejet af de enkelte udgivere. Det gør det lettere at aggregere viden ved at sammenføje triplerne.

Imidlertid kræver brugen af forskellige modelleringsmønstre i forskellige grafer, at en struktureret forespørgsel over den globale graf kombinerer alle mulige modelleringsmønstre, herunder alle mulige knuder og kanter som er synonymer, hvilket er upraktisk. Denne semantiske uensartethed forhindrer desuden visse stykker af tilsvarende viden at blive tilkoblede, fordi det, som er repræsenteret ved en knude i en graf under en bestemt modelleringsmønster, kan repræsenteres implicit som en slags grafmønster i en anden graf under et anden modelleringsmønster, hvilket gør sammenkoblingen ved brug af simple ækvivalenskanter umuligt. Endvidere er visse eksisterende modelleringsmønstre for detaljerede og ineffektive, mens andre mangler udtryksfuldhed og undlader vigtige forbindelser i dataene.

Denne afhandling beskriver oprettelsen af FrameBase, et system der anvender teori og ressourcer fra lingvistik og kognitionsforskning til at give forskellige forbundne repræsentationslag, der kombinerer udtryksfuldhed af nogle modelleringsmønstre med koncisheden af andre, mens de på samme tid giver et fælles, grundlæggende ordforråd der kan udvides af udgivere og brugerne. FrameBase's flerlagede system af modelleringsmønstre tillader at repræsentere en bred vifte af viden på en måde, som tillader trinløs integrering af såvel viden som forespørgsler til videnbasen.

Afhandlingen introducerer desuden forskellige metoder til at integrere viden fra eksterne videnbaser og endelig fra enhver kilde af strukturerede data. Metoderne spænder fra manuelle til automatiske, og forskellige semi-

automatiske tilgange bliver udviklet. Automatiske metoder udnytter FrameBase's forbindelse til lingvistik for at skabe komplekse mapninger til navnene af elementer i de eksterne videnbaser. Den manuelle tilgang er effektiviseret med en webapplikation, der tillader en bruger at skabe mapninger på en simpel, grafiske måde. To forskellige semi-automatiske tilgang er implementeret: en, som udvider automatiske metoder med simple heuristikker knyttet til hver videnbase, og en anden, som udvider den manuelle metode med at genbruge de automatiske og semi-automatiske metoder for at implementere anbefalings- og søgefunktioner i webapplikationen. Resultaterne for automatiske metoder evalueres af mennesker.

Desuden giver Framebase's bånd til lingvistik mulighed for at forbinde til naturligt sprog, enten med henblik på tekstudvinding (*text mining*) eller forespørgselsbesvarelse (*question answering*). Afhandlingen diskuterer desuden potentielle metoder til dette og antyder mulige linjer for det fremtidige arbejde.

Contents

Thesis Details

Thesis Title:	A Frame-Based Approach for Integrating Heterogeneous Knowledge Sources
Ph.D. Student:	Jacobo Rouces González
Supervisors:	Assoc. Prof. Henrik Legind Larsen, Dept. of Electronic Systems, Aalborg University Assoc. Prof. Katja Hose, Dept. of Computer Science, Aalborg University

The main body of this thesis consist of the following papers.

- J. Rouces. "Enhancing Recall in Semantic Querying", In *Proc. 12th Scandinavian Conference for Artificial Intelligence*. 2013.
- J. Rouces, G. De Melo, and K. Hose, "FrameBase: Representing N-ary Relations using Semantic Frames", In *Proc. 12th Extended Semantic Web Conference (ESWC)*, 2015.
- J. Rouces, G. De Melo, and K. Hose, "Representing Specialized Events with FrameBase", In *Proc. 4th International Workshop on Detection, Representation, and Exploitation of Events in the Semantic Web (DeRiVE)*, 2015.
- J. Rouces, G. De Melo, and K. Hose, "Heuristics for Connecting Heterogeneous Knowledge". In *Proc. 13th Extended Semantic Web Conference (ESWC)*, 2016.
- J. Rouces, G. De Melo, and K. Hose, "Complex Schema Mapping and Linking Data: Beyond Binary Predicates", In *Proc. Workshop on Linked Data on the Web (LDOW), co-located with WWW*, 2016.
- J. Rouces, G. De Melo, and K. Hose, "Integrating Heterogeneous Knowledge with FrameBase", **Submitted** to: *Semantic Web Journal (SWJ)*, 2016.
- J. Rouces, G. De Melo, and K. Hose. Klint: Assisting Integration of Heterogeneous Knowledge. Demonstration paper in: *Proc. 25th International Joint Conference on Artificial Intelligence (IJCAI)*, 2016.

In addition to the enclosed papers, the following publications have also been made.

[1] A. Gerdes, H. L. Larsen, and J. Rouces, "Issues of Security and Informational Privacy in relation to an Environmental Scanning System for Fighting Organized Crime", In *Proc. 10th International Conference on Flexible Query Answering Systems (FQAS)*, 2013.

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement No. FP7-SEC-2012-312651 (ePOOLICE project).
This thesis has been submitted for assessment in partial fulfillment of the PhD degree. The thesis is based on the submitted or published scientific papers which are listed above. Parts of the papers are used directly or indirectly in the extended summary of the thesis. As part of the assessment, co-author statements have been made available to the assessment committee and are also available at the Faculty. The thesis is not in its present form acceptable for open publication but only in limited and closed circulation as copyright may not be ensured.

Part I

Overview

Chapter 1

Introduction

An increasing number of knowledge bases (KBs) are being published in the web by different kinds of stakeholders. Many of these KBs use standards that make it easier to link elements from each other, so that they can be treated and queried as a single global KB. The advantage of this is that it should allow data across different KBs to be added and combined so that the global KB can provide information that the individual KBs, if were disconnected, could not provide. The data in the global linked KB is usually referred to as *Linked (Open) Data* (LOD) [3], and the global (usually distributed) KB is referred to as the LOD cloud. The *semantic web* is the projected goal of the LOD cloud comprising the main backend of the web.

The most prevalent standards recommended and used for this purpose are RDF (Resource Description Framework) [7] and SPARQL (SPARQL Protocol and RDF Query Language) [12]. RDF represents data as a set of (subject, predicate, object) triples like (Robb, marries, Talisa), (Robb, hasFather, Eddard), (Eddard, isbornAtDate, 263 AC). However, except for strings (called "literals") and a few other exceptions, RDF uses Internationalized Resource Identifiers (IRIs) to identify entities denoted in any of the positions of a triple. This allows that different stakeholders can coin identifiers using the domains they own, avoiding name collisions, which serves very well the purpose of the LOD cloud. The names "subject", "predicate" and "object" arise from the usual grammatical role of the English names associated to each element (Bran, builds, The Wall), though sometimes this is not the case (The Wall, builder, Bran). The set of triples constituting a KB can be interpreted, in a somewhat simplified fashion, as a directed labelled graph, where the subject and the object are nodes connected by a directed labelled edge identified by the predicate. This is a simplification because in RDF, the predicate of a triple can also serve as subject or object in other triples, and therefore the graph model is more complex, but in many cases the simplification is enough.

The RDF format provides a very simple and homogeneous syntax that combined with the global identifiers favours the aggregation of data. Provided the individual KBs include the links to other KBs, the LOD cloud is simply defined by the union of the sets of triples of each KB (some RDF serialization formats require renaming locally scoped identifiers called "anonymous nodes" to avoid naming collisions, but this is rather straightforward). Currently, the LOD cloud contains over 30 billion triples spread over 295 KBs [1].

However, different modelling decisions make graphs that contain the same or overlapping information to be completely different, not only lexically – which would imply a sort of graph isomorphism that could be solved by equivalence relations between nodes– but in a structural sense too. These modelling decisions define different *modelling patterns* that, in turn, are usually encoded as different *schemas*. Schemas are sets of triples that carry additional logical semantics that allow to infer new triples from the existing ones.

(set-theoretical of classes and properties with classes defined ***

This is similar to how, in software engineering, a given set of functional specifications can be implemented in an object-oriented programming language using a different set of classes and methods, attending to different design patterns.

In this work, we analyze different kinds of modelling patterns that currently co-exist when representing N-ary relations in the LOD, and their problems, both intrinsic and arising when combined in the LOD (c.f. Paper D). Graphs representing these different models can be found in Figure 1.1.

- The pattern in Figure 1.1a is very basic and just connects pairw-wise the arguments of the n-ary relation. If one regards every triple as representing an underlying n-ary relation with only two arguments filled, it could be said that this pattern takes place in every KB in the LOD. It lacks the expressive power to connect more than two arguments of the same n-ary relation.

- The pattern in Figure 1.1b is used in the YAGO ontologies [14] and attempts to solve the above problem by using a mechanism called RDF reification, but it incurs in significant overhead that is superlinear to the number of elements in the relation, and its semantics are also problematic.

- The pattern in Figure 1.1c attempts to improve the pattern above [19], but still carries some of the same problems.

- The pattern in Figure 1.1d is an event-centric pattern used frequently in specific parts of many KBs (i.e. Freebase [5]), usually to represent public events by means of a reduced ad-hoc vocabulary.

- Other ad-hoc solutions can be found, for instance encoding the value of the third, fourth, etc. argument in the IRI of a property connecting the

[1] `http://lod-cloud.net/state/`

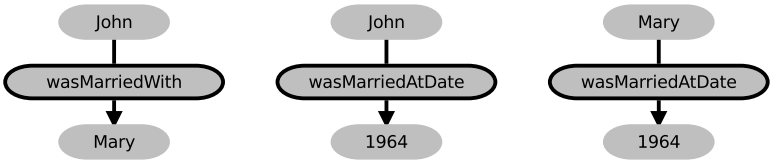
first two, for instance `John marriesMaryAtDate 1964`. This reduces the overhead of the patterns from Figures 1.1b and 1.1c but at the cost of breaking the RDF standard by creating ad-hoc semantics encoded within the IRIs, which would require extra processing and in the long run produce incompatibilities and defeat the purpose of the RDF standard of having a simple, homogeneous standard.

A detailed version of this analysis can be found in Paper D.

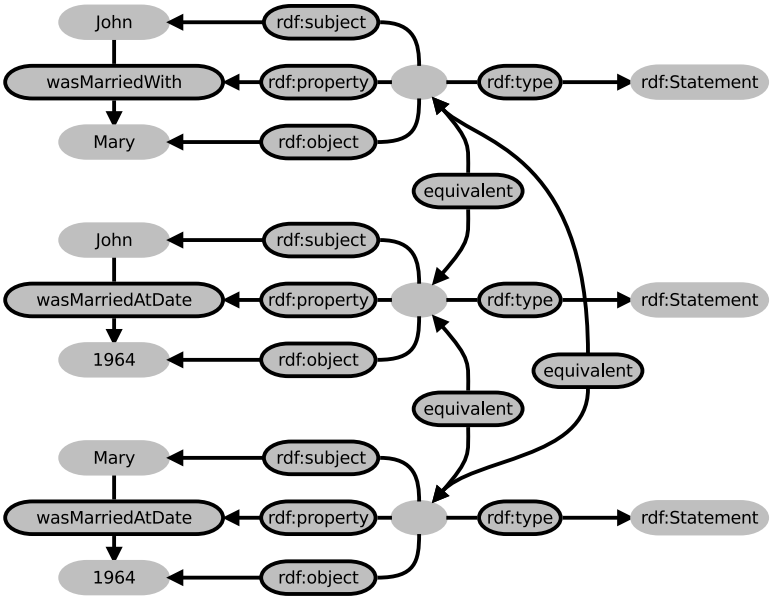
This sort of semantic heterogeneity in the LOD cloud has several detrimental effects:

- When the LOD cloud is queried as a whole, the query needs to include all the possible alternate terms for a given concept, from all the different vocabularies used in the different linked KBs, as aliases exist even for the most general and well-known concepts. Even worse, it has to include all the possible structural alternatives (graph patterns) used in the different KBs. This is impractical both because of the complexity acquired by the query and because it requires the user to keep track of all these alternatives.
- When the data from different KBs are linked, an entity under one model may easily not have a corresponding entity in another, because it is represented implicitly, usually by means of a graph pattern residing in the data. For instance, there is no equivalent entity to `wasMarriedWith` in Fig. 1.1a in either Figs. 1.1b, 1.1c or 1.1d. However, most current efforts and techniques for linking data focus on linking pairs of individual entities in different graphs by means of equivalence or subsumption relations.
- Although some of the KBs can work as de-facto hubs (currently DBpedia [4] tends to be used as such), there is no KB specifically designed to work as semantic hub. DBpedia is mostly based on information extracted from infoboxes in Wikipedia and therefore its vocabulary is skewed towards the kind of information collected in these boxes, and lacks a global approach towards semantics that resources in linguistics may have.

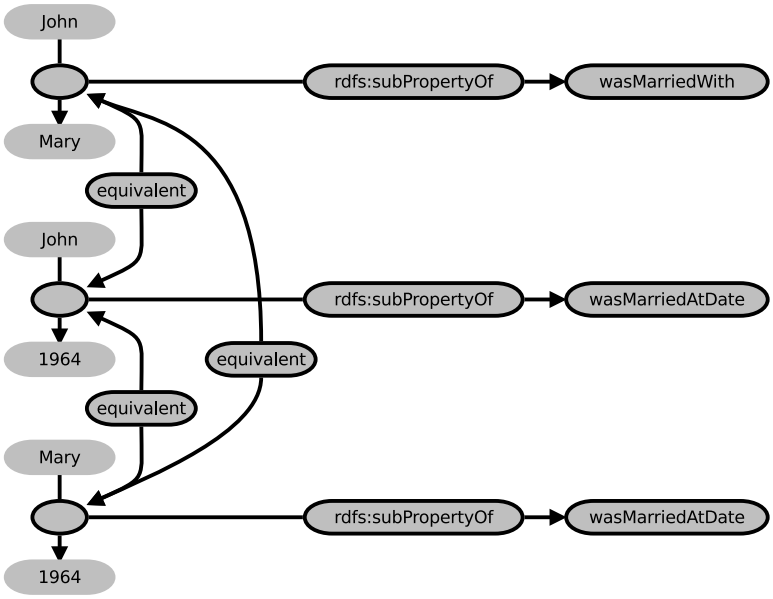
It is important to note that the problem of semantic heterogeneity is not unique to LOD. It can be found when trying to integrate other kinds of structured knowledge: for instance when two companies merge they may want to merge their internal relational databases too, and similar problems arise. The LOD standards try to address some issues arising when structured data produced by different stakeholders is combined or linked, but as it will be shown below, they only succeed partially. Therefore, the problem of semantic heterogeneity does not exist *because of* the semantic web standards, but *despite* them. The reason why semantic heterogeneity is an important problem in the LOD and why the analysis and the products in this work focus on the perspective of the LOD is that the prime objective of the LOD is linking and



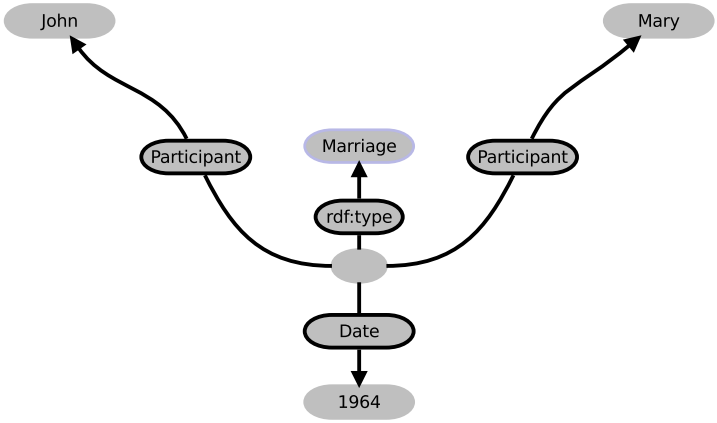
(a) Basic triple pattern.



(b) Pattern used in the YAGO* KB [14].



(c) Singleton property pattern proposed to improve the YAGO model [19].



(d) Event-centric pattern used frequently in specific parts of many KBs (i.e. Freebase [5]), usually to represent public events by means of a reduced ad-hoc vocabulary.

Fig. 1.1: The same information represented using different modelling patterns found in different KBs in the LOD.

combining data, and therefore the problem of semantic heterogeneity is more pervasive. Many private databases are never linked or merged with other databases and therefore semantic heterogeneity never manifests itself, except in a transient manner in the mind of a user or new database administrator that has to adapt from one model or schema to another.

A complete review of the state of the art can be found in the papers in Part II, specially in Papers D and E.

This work describes the construction of FrameBase, a system composed of a multi-layered KB with a schema that allows to represent a wide range of knowledge.

The more expressive but more verbose layer is denoted as the reified layer. It consists of classes, representing frames, which can be events, situations, processes of a very general kind. It also contains *Frame Element* properties that specify qualities about frame instances: agents participating in different ways, time, place, cause, consequence, instrument, etc. The frames are organized in a rich hierarchy of macroframes, cluster-microframes, and synset- and LU-microframes, in order from more general to more specific. Synsets and LUs (Lexical Units) are concepts imported from WordNet [11] and FrameNet [2] respectively, which are resources from computational linguistics. FrameNet constitutes the backbone of FrameBase and is a compilation of such frames and FEs to annotate the semantics of natural language. WordNet is a computational lexicon that includes word senses grouped by synonymy and other semantic relations. Both synsets and LUs are closely related to sense-disambiguated words and therefore they are used to produce the most specific frames, whereas cluster-microframes and macroframes represent groups of near-synonymous or related concepts. Figure 1.2 shows an example of how FrameBase represents knowledge, and Figure E.1 illustrates a sample of the hierarchy.

The less expressive but more concise layer is denoted as the dereified layer, and is formed by Direct Binary Predicates (DBPs) that connect directly the objects of specific pairs of FEs.

The two different layers provide a trade-off between expressiveness and efficiency, and are connected by inference rules, called Reification-Dereification (ReDer) rules.

Data from external KBs in the LOD cloud can be imported using *integration rules*, which can create FrameBase instance data from the instance fata of the external KBs. This work also describes the creation of these rules in manual, semi-automatic and automatic ways, exploiting the linguistic aspects of FrameBase inherited from FrameNet. The results for automatic and semi-automatic methods are evaluated. Examples are also provided of how the resulting FrameBase instance data can be queried. Figure 1.4 provides a general overview of the dataflow in the FrameBase system.

The relation of FrameBase to linguistics provides additional potential for

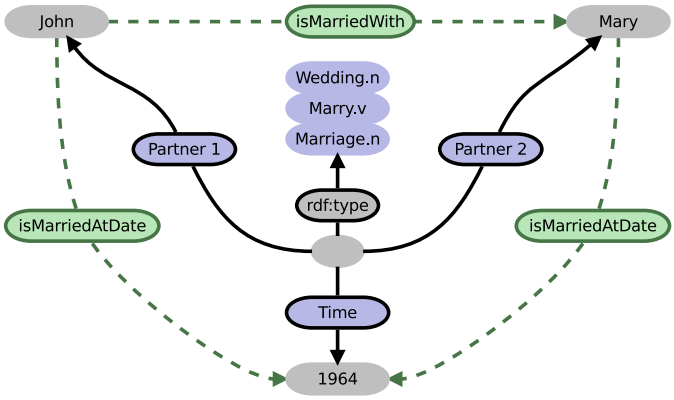


Fig. 1.2: Information from Figure 1.1 represented under the FrameBase model, which combines expressiveness with conciseness by combining different representation layers (reified in blue, dereified in green).

natural language processing tasks such as text mining and question answering, which will be discussed in Section 4.

The rest of Part I is structured as follows. Chapter 2 summarizes the content of each of the publications included in this thesis, and it adds complementary information. Chapter 3 provides the conclusion of this thesis and outlines future research. Part II contains the complete publications.

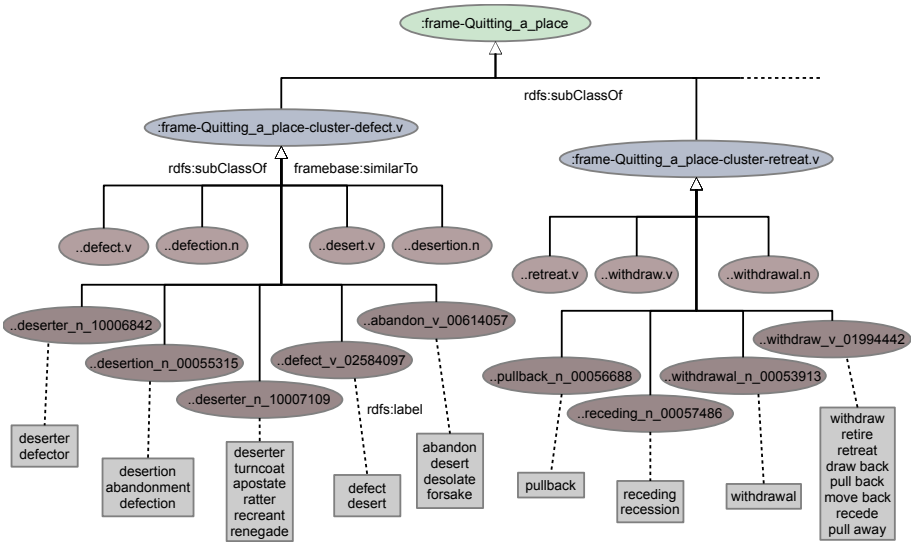


Fig. 1.3: Sample of the frame hierarchy in FrameBase.

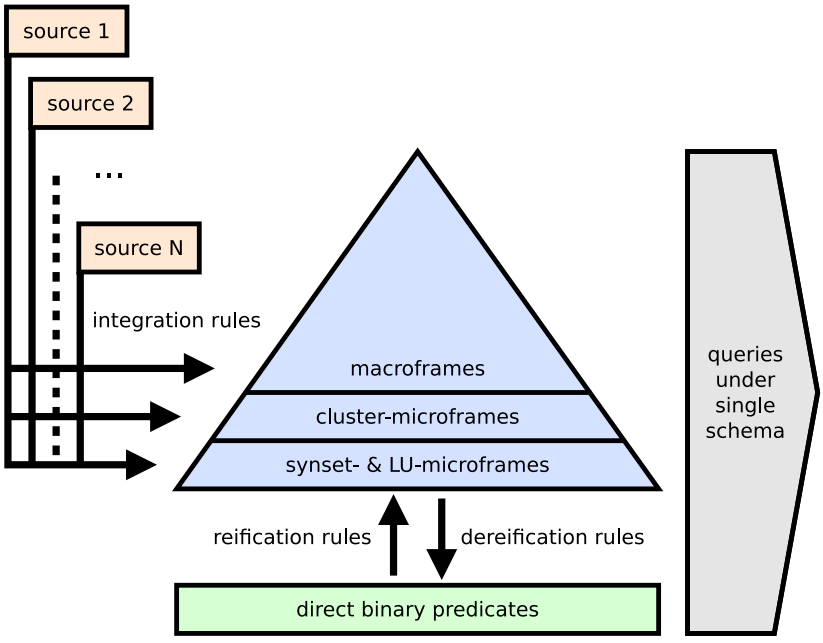


Fig. 1.4: Overview of the data flow of FrameBase.

Chapter 2

Summary of Contributions

This chapter summarizes and complements the content of each of the publications included in this thesis. Section 1 lists the publications ordered by submission date, summarizing their content and specifying the eventual overlap between them. Section 2 provides supplementary information about some of the papers. The reading of this information is optional and requires the previous reading of the papers.

1 Publications

Paper A J. Rouces. "Enhancing Recall in Semantic Querying", In *Proc. 12th Scandinavian Conference for Artificial Intelligence (SCAI)*. 2013.

In this paper, different types or "patterns" of semantic heterogeneity in the LOD are identified, and their consequences upon structured querying are analysed: a query following one pattern will not retrieve results with identical or overlapping semantics that use another pattern, therefore reducing recall. A very general overview of how these heterogeneity issues could be addressed is also included.

Issues 1.1. (Single Property or Event Attachment), 1.2. (Event Modeling: Different Approaches) and 1.4. (Numerous Overlapping Vocabularies), together with the proposed approach outlined for solving them, provides the motivation and foundation for the FrameBase system developed in the following papers.

Paper B J. Rouces, G. De Melo, and K. Hose, "FrameBase: Representing N-ary Relations using Semantic Frames", In *Proc. 12th Extended Semantic Web Conference (ESWC)*, 2015.

In this paper, a more detailed analysis of different representation models for N-ary relations in the LOD is provided (extending the discussion of points 1.1 and 1.2 in Paper A). Additionally, it identifies heterogeneity as a problem

not only when querying linked data but also for the very enterprise of linking data, since certain kinds of heterogeneity require linking different kinds of patterns, whereas most current established methods for linking data focus on linking individual entities by means of binary properties.

Then, the paper describes how FrameBase system is built reusing existing resources from linguistics and cognitive science (FrameNet [2] and WordNet [11]). The resulting FrameBase system consists of a core schema made of frame classes organized in a rich hierarchy, associated properties called Frame Elements (FEs), Direct Binary Properties (DBPs) connecting pairs of objects of the FE properties for a given frame, and Reification-Dereification (ReDer) rules connecting the DBPs with the frame-FE patterns.

The paper illustrates how the resulting FrameBase model is at least as expressive as the other models discussed, and at the same time it is at least as space-efficient. In other words: it is as expressive as the most expressive but inefficient models analyzed, and as efficient as the most efficient but inexpressive models analysed. It also reduces heterogeneity by providing a core set of common concepts that can be combined or extended, and it provides potential connections with natural language.

Finally, the accuracy of the resulting system is analyzed, and the paper provides a few examples of manually-built integration rules importing knowledge from other knowledge bases into FrameBase, and of how the integrated knowledge can be queried.

Paper D, submitted to the Semantic Web Journal, is an extension of this paper. Therefore, reading both papers is not necessary, and reading Paper D instead of this one is recommended because of the additional material, including figures and examples.

Paper C J. Rouces, G. De Melo, and K. Hose, "Representing Specialized Events with FrameBase", In *Proc. 4th International Workshop on Detection, Representation, and Exploitation of Events in the Semantic Web (DeRiVE)*, 2015.

This paper discusses the use of FrameBase to represent events. Since frames can be viewed as subsuming events (or interpreted as a very general concept of events), the application is straightforward. The paper provides examples of integrating event data from different sources and types, proving the expressiveness of FrameBase. It also discusses the different ways in which integration rules can be complex. This is an attempt to create an alphabet of atomic transformations that applied to a straightforward integration rule (one made of a set of binary correspondences) can create an arbitrarily complex integration rule. Such catalogue is built with the purpose of contributing towards the extremely difficult task of automatically creating arbitrarily complex integration rules.

Paper D J. Rouces, G. De Melo, and K. Hose, "Integrating Heterogeneous Knowledge with FrameBase", **Submitted** to: *Semantic Web Journal (SWJ)*, 2016.

This paper has been submitted to a conference follow-up special issue of the Semantic Web journal, and therefore is an extension of Paper B. It also contains some examples from Paper C. Therefore, it is recommended its reading over Paper B. The additions in relation to Paper B are summarized in Section 2.1.

Paper E J. Rouces, G. De Melo, and K. Hose, "Heuristics for Connecting Heterogeneous Knowledge". In *Proc. 13th Extended Semantic Web Conference (ESWC)*, 2016.

This paper describes methods for automatically creating integration rules, combining a support vector machine with heuristics tailored at idiosyncrasies of certain source knowledge bases.

Paper F J. Rouces, G. De Melo, and K. Hose, "Complex Schema Mapping and Linking Data: Beyond Binary Predicates", In *Proc. Workshop on Linked Data on the Web (LDOW), co-located with WWW*, 2016.

This paper describes in depth a method for automatically creating a certain type of integration rules called property-frame rules. It does so by re-using DBPs and ReDer rules in FrameBase, canonicalizing properties from external knowledge bases, and mapping them to DBPs using a custom similarity measure. While property-frame integration rules do not solve alone all the integration needs to map knowledge between FrameBase and external knowledge bases, or between external knowledge bases through FrameBase, they provide an essential building block that ontology alignment systems producing binary equivalence links cannot represent.

Paper G J. Rouces, G. De Melo, and K. Hose. Klint: Assisting Integration of Heterogeneous Knowledge. Demonstration paper in: *Proc. 25th International Joint Conference on Artificial Intelligence (IJCAI)*, 2016.

The levels of virtually perfect accuracy required in most knowledge bases is not possible complex integration rules are built automatically, especially if the patterns involved have a high or unbounded complexity. Hence, a human in the loop is necessary for achieving these levels of accuracy. Therefore, we develop Klint (Knowledge Integrator), a web-based application that by means of a graphical interface leverages the algorithms developed for creating integration rules, using them as a suggestion engine that can assist a human user and reduce his involvement.

Additional features of Klint are described in Section 2.2.

2 Supplementary Information

2.1 Additional content of Paper D

The additions in relation to Paper B are the following:

In the section "State of the Art", section 2.3 has been added, which includes a detailed analysis and comparison with the role system in schema.org. This

model is also included in the triple overhead analysis (Table 2). Furthermore, details to the description of reasoning in the same analysis have been added.

In the section "System Overview", it is described how the schema has been extended with reified cluster microframes and a new property for which the transitive closure is computed. Examples, a diagram of the schema (Figure 1), the linking with other LOD datasets (Lexvo.org [9] and the Princeton RDF Wordnet [17]) and further clarifications have been added.

10,270 new Direct Binary Predicates and Reification-Dereification rules have been added based on nouns, for which the head verbs have been extracted with a novel method that is described in Section 5.2. More examples of ReDer rules have been added to the paper. Additionally, new rules have been added for cases where FrameNet's annotations were insufficient, by extracting statistics from the cases where the annotations were sufficient. Linguistically rich annotations to all Direct Binary Predicates have been added, using the Lemon model [18].

2.2 Additional Features in Klint

Klint supports additional features not describes in Paper **??**.

Visual Query and Knowledge Building Klint supports a *Visual Knowledge and Query Building* mode that allows the user to create knowledge and queries under the FrameBase schema, in a visual and simple way. The way it works is identical to *Assisted Schema Integration* explained in the paper, but creating a graph from scratch without elements of the source KB. There are two possible sub-modes: Visual Knowledge Building and Visual Query Building.

• **Visual Knowledge Building** . When the user introduces only FrameBase and, optionally, external nodes, the resulting knowledge 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 it is rather a source-neutral way to test the expressiveness of FrameBase creating small examples of knowledge.

• **Visual Query Building** . When the user adds also some variable node, then the system will allow him to export the resulting knowledge pattern as a SPARQL query, and optionally run it against the knowledge integrated with the available integration rules. The user can choose between 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 that pattern.
- Run the SELECT/CONSTRUCT SPARQL query directly and visualizing the results.

Representation Algorithm RDF graphs are usually represented as directed labelled graphs with each subject-predicate-object triple as an edge between the subject and the object, with the property as the label. However, this is a simplification, as predicates can also be subjects or objects of some triples, and RDF graphs are truly bipartite graphs [13] where each triple is represented by two consecutive directed edges: subject-property and property-object. The graph representation in Klint maintains the bipartite model, so RDF is fully supported, but at the same time it maintains a presentation similar to the directed labelled graph, which is visually more intuitive for the user. It uses 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 predicates in different triples, or as predicate in one and subject in another, then it is represented with two different nodes that are internally linked to the same resource. Klint has several heuristics for creating human-readable node labels, which are combined with labels extracted from the imported schema and a public LOD cache.

When users want to make a modification in the graph, they can do so in a visual and simple 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 a subject-predicate and a predicate-object edges are created sharing the predicate. Temporary links are shown for unfinished subject-predicate edges.

Chapter 3

Concluding Remarks

1 Conclusion

This thesis has presented the construction of FrameBase, a knowledge representation system that based on resources from linguistics, can represent a wide range of knowledge in an unambiguous and efficient way, providing an upper layer of knowledge that can be extended by interested stakeholders. The thesis has also introduced methods for integrating data from other structured sources, either automatically or semi-automatically.

FrameBase is especially suited for integrating data from the Linked Open Data cloud. This was chosen because due to the nature of LOD datasets, which are published by independent stakeholders and meant to be linked to each other, the LOD cloud is the prime example of heterogeneous data that must be integrated. However, the problems, methods and techniques presented in this thesis apply also to other kinds of datasets that need to be integrated, for instance relational databases that should be integrated after the merge of two companies. The methods and techniques here introduced could be applied to cases like this by either changing the implementation formats of the ontology and rules, or pre-processing the datasets at hand with one of the many existing systems to convert structured data to RDF[1].

2 Future Work

Besides the abovementioned process of extending or adapted FrameBase to cope with non-RDF data sources, which could be somehow be integrated into a the FrameBase system (but also be left to each user, who may have particular needs), different research lines have been identified with potential to significantly break the state of the art building upon the work described in this thesis. These are enumerated next.

[1] `https://www.w3.org/wiki/ConverterToRdf`

Interfacing with natural language . Due to its use of linguistic resources for ontological purposes, FrameBase has a big potential for text mining and other natural language related tasks, such as question answering. Three distinct strategies have been identified that could serve to extract FrameBase structured knowledge from natural language, either for creating grounded knowledge or queries. These strategies could be combined.

- Reusing Semantic Role Labelling (SRL) systems for FrameNet such as SEMAFOR [8]. Direct text-to-ontology systems such as FRED [20] or Pikes [6], or an extractor of events to the LODE (Linking Open Descriptions of Events) ontology [10] already make use of SRL systems and could be adapted or extended.

- DBPs can be matched against running text in a similar fashion as in Paper F, which would provide additional means towards relation extraction. Alternatively, they could be matched to clauses extracted from already existing clause mining systems, such as OpenIE [1] This would constitute on its own a specialized SRL system with high accuracy but restricted to two FEs, and would re-use much of the work described in F (both the implementation and the future work). The restriction to 2 FEs could be surmounted by performing a later step of merging frame instances under certain criteria of similarity and closeness in the text (the closeness measure could be further refined by using anaphora resolution). The extracted frames could likewise be combined with others extracted from existing SRL systems.

- The FrameBase schema could be extended with PropBank [15], which is a linguistic resource similar to FrameNet, but closer to natural language syntax. PropBank uses a more reduced set of generic roles than FrameNet's FEs, and it does not declare the abstract frames behind FrameNet, limiting itself to word senses. FrameNet was chosen for the backbone of FrameBase for its hierarchy and being in general semantically richer; however, integrating PropBank into FrameBase may provide additional advantages, such as increasing the accuracy of the SRL system [16].

Text mining methods could be reused for question answering with relatively few adaptations. The simplest strategy would be mapping wh-words to SPARQL variables. Due to current SRL systems having far from perfect accuracy, it would be advisable using a mixed approach combining unstructured search in order to fill the gaps (i.e., make for the imperfect recall of the SRL system) and provide a weighted alternative to the obtained structured query (i.e., make for the imperfect precision). At the same time, even though semantic role labeling is still challenging, semantics is one of the largest research areas in natural language processing now and thus FrameBase can easily benefit from progress made in this area in the future.

Extending integration rules . Creating integration rules automatically is an extremely difficult task, illustrated by the low inter-annotator agreement rates obtained even for rules whose complexity is bounded (i.e., they have a fixed structure like property-frame and class-frame rules, c.f. Paper E). For rules with unbounded complexity (involving arbitrary patterns), the task is even more complex and inter-annotator agreements is likely to drop to extremely low levels. Despite this, there are potentially ways to create arbitrarily complex rules exploiting FrameBase's connection to natural language. Two main strategies seem possible, which are not completely disjoint but differ in that one is more driven by FrameBase and natural language generation and the other is driven by the source KBs and relies on natural language processing. Both of them produce rules connecting a property from the source KB with a complex pattern in FrameBase, which could be combined with other rules created with one-to-one mappings produced by existing ontology alignment systems.

- **FrameBase driven.** This involves extending the approach in Paper F, creating very complex ReDer rules whose DBPs could also be matched with external properties. These DBPs could have for instance a "(VP <VBZ> (NP <NP_1> (PP <IN> <NP_2>)))" structure, like for instance "developsUnderstandingOfContent" or "startsDemolitionOfBuilding" (but other more complex structures would be possible too). As explained in Section 4 (Future Work) in Paper F, this involves two frame instances (one evoked by VBZ and the other by NP_1), and some challenges:
  - Syntactically correct but semantically nonsensical combinations should be filtered out (e.g. "procrastinationDrunkByQuadruplicity"). This could be done based on example sentences in FrameNet.
  - If the frames evoked by the VBZ and NP_1 are not annotated in the same sentence, the correct pair of frames should be chosen from the pair of lexical units (VBZ,NP_1), and the correct FE connecting both should be chosen too.

  An advantage of this approach is that it provides richer ReDer rules in FrameBase, but the disadvantage is that being driven by FrameBase, it may have poor recall for real-life datasets, both because of its reliance on FrameNet example sentences and FrameNet's non-specialized vocabulary. The latter problem could be significantly reduced by also updating the similarity function between DBPs and source properties, to account to hypernymy and synonymy relations that would allow capturing very specific concepts in source KBs for which hypernyms can be found in FrameBase (for instance: "increasesSpeedOfProcess" and "catalyzesChemicalReaction").
- **Source-data driven.** This would involve parsing predicate names with a SRL system, in a similar way as explained in the previous research line

"Interfacing with natural language". However, SRL systems are also constrained by their reliance on example annotated sentences for training. In any case, an advantage of this approach is that if FrameBase is extended with PropBank, SRL systems for this could be used as well, as it was also proposed in the prevopus research line.

Implementing virtual querying . So far the integration rules for integrating source KBs into FrameBase have been implemented as SPARQL CONSTRUCT queries that applied over the sources' data, which can be used to materialize the integrated knowledge.

This allows for efficient evaluation of queries on the integrated knowledge base. To account for updates, the SPARQL CONSTRUCT queries have to be re-run periodically. An alternative implementation would be using virtual querying: using the integration rules to provide FrameBase-adapted virtual views of the source KBs. This would allow re-using existing SPARQL endpoints from the different sources and enable access to the most recent version of the source data. On the other hand, this introduces a dependence on the availability of the SPARQL endpoints hosting the source data, as well as the need for federated query processing techniques to compute the results.

A similar strategy could be used for ReDer rules, either alone on top of materialized FrameBase instance data, or in conjunction with the virtual views of external sources described above. The case of ReDer rules is relatively simpler for the several reasons. First, they are currently definite clauses, which allows using existing reasoners (for example, the ReDer rules have been implemented also in the Jena rule language), and at the same time they are currently not expected to be chained (there are only two levels of reification and they are disjoint). Additionally, they do not require federation unless the FrameBase instance was purposely distributed over different triplestores.

References

[1] G. Angeli, M. J. Johnson Premkumar, and C. D. Manning, "Leveraging linguistic structure for open domain information extraction," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Beijing, China: Association for Computational Linguistics, July 2015, pp. 344–354. [Online]. Available: http://www.aclweb.org/anthology/P15-1034

[2] C. F. Baker, C. J. Fillmore, and J. B. Lowe, "The Berkeley FrameNet Project," in *Proceedings of the 17th international conference on Computational linguistics – Volume 1*, ser. ICCL '98, 1998, pp. 86–90.

[3] C. Bizer, T. Heath, and T. Berners-Lee, "Linked data–the story so far," *IJSWIS*, vol. 5, no. 3, pp. 1–22, 2009.

[4] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann, "DBpedia-A crystallization point for the Web of Data," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 7, no. 3, pp. 154–165, 2009.

[5] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, "Freebase: A Collaboratively Created Graph Database for Structuring Human Knowledge," in *SIGMOD'08*, 2008, pp. 1247–1250.

[6] F. Corcoglioniti, M. Rospocher, and A. P. Aprosio, "A 2-phase frame-based knowledge extraction framework," in *Proc. of ACM Symposium on Applied Computing (SAC'16)*, 2016, pp. 354–361.

[7] R. Cyganiak, D. Wood, and M. Lanthaler, "RDF 1.1 Concepts and Abstract Syntax," W3C Consortium, W3C Recommendation, Feb. 2014.

[8] D. Das, D. Chen, A. F. T. Martins, N. Schneider, and N. A. Smith, "Frame-Semantic Parsing," *Computational Linguistics*, vol. 40, no. 1, pp. 9–56, 2014.

[9] G. de Melo and G. Weikum, "Language as a foundation of the Semantic Web," in *Proceedings of the Poster and Demonstration Session at the 7th International Semantic Web Conference (ISWC 2008)*, ser. CEUR WS, C. Bizer and A. Joshi, Eds., vol. 401. Karlsruhe, Germany: CEUR, 2008.

[10] P. Exner and P. Nugues, "Using semantic role labeling to extract events from Wikipedia," in *DeRiVE'11*, 2011.

[11] C. Fellbaum, *WordNet: An Electronic Lexical Database*. MIT Press, 1998.

[12] S. Harris and A. Seaborne, "SPARQL 1.1 Query Language," W3C Consortium, W3C Recommendation, Mar. 2013.

[13] J. Hayes and C. Gutierrez, "Bipartite graphs as intermediate model for RDF," ser. ISWC 2004, 2004, pp. 47–61.

[14] J. Hoffart, F. M. Suchanek, K. Berberich, and G. Weikum, "YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia," *Artificial Intelligence* , vol. 194, no. 0, pp. 28–61, 2013.

[15] P. Kingsbury and M. Palmer, "From TreeBank to PropBank." ser. LREC '02, 2002.

[16] M. Kshirsagar, S. Thomson, N. Schneider, J. G. Carbonell, N. A. Smith, and C. Dyer, "Frame-semantic role labeling with heterogeneous annotations," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 2: Short Papers*, 2015, pp. 218–224. [Online]. Available: http://aclweb.org/anthology/P/P15/P15-2036.pdf

[17] J. McCrae, C. Fellbaum, and P. Cimiano, "Publishing and Linking WordNet using lemon and RDF," in *Proceedings of the 3rd Workshop on Linked Data in Linguistics*, 2014.

[18] J. McCrae, D. Spohr, and P. Cimiano, "Linking lexical resources and ontologies on the semantic web with lemon," in *The semantic web: research and applications*. Springer Berlin Heidelberg, 2011, pp. 245–259.

[19] V. Nguyen, O. Bodenreider, and A. Sheth, "Don't Like RDF Reification?: Making Statements About Statements Using Singleton Property," ser. WWW '14, 2014.

[20] V. Presutti, F. Draicchio, and A. Gangemi, "Knowledge Extraction Based on Discourse Representation Theory and Linguistic Frames," in *Knowledge Engineering and Knowledge Management*, ser. Lecture Notes in Computer Science, A. Teije, J. Völker, S. Handschuh, H. Stuckenschmidt, M. d'Acquin, A. Nikolov, N. Aussenac-Gilles, and N. Hernandez, Eds. Springer Berlin Heidelberg, 2012, vol. 7603, pp. 114–129.

Part II

Papers

Paper A

Enhancing Recall in Semantic Querying

Jacobo Rouces

The paper has been published in the
Proceedings of the 12th Scandinavian Conference for Artificial Intelligence (SCAI),
2013

The layout has been revised.

RDF and SPARQL are currently state-of-the-art W3C standards to respectively represent and query structured information, especially when information from different sources must be federated. However, there are various reasons for which the same knowledge can be modeled in RDF graphs that are both lexically and structurally different, which we will introduce in the next section. As RDF graphs from different sources are expected to be linked, the modeling heterogeneities will make the federated graph become sparser and inconsistent. This is detrimental to the recall of SPARQL queries, as the query graph will be built following one particular modeling choice that may not be consistently used across the reachable parts of the federated graph.

1 Same Meaning but Different Graph

Of the problems we will identify, some are more general than others, but even those that are not inherent to RDF and other semantic web technologies, have become more problematic under this new paradigm. Here, as opposed to traditional centralized knowledge management systems, information is no longer shaped by a closed and well defined model that both editors and users know and are expected to follow, as it will cointain knowledge from different sources.

1.1 Single Property or Event Attachment

When additional information needs to be assigned to the event implied by a property, the property needs to be reified into a new entity that represents the event, so the additional information can be assigned to the event as additional properties. This reification can also be done without additional information, and is related to the neo-Davidsonian form [1, 600f.], though it is denoted as "representation of n-ary relations" in much of the RDF literature. The problem is illustrated in [2], with the following example:

`A_Einstein wonPrize NobelPrize`	`A_Einstein`	`winner`	`AEinstWonNP1921`
	`NobelPrize`	`prize`	`AEinstWonNP1921`
	`1921`	`time`	`AEinstWonNP1921`

As a solution, [2] proposes choosing a primary pair for each n-ary relation and appending the rest to this by means of RDF reification. However, a simple and universal way to choose this primary pair for any n-ary relation seems not to be straightforward. Also, using RDF reification, the properties are attached to the triple that contains the property, not the property itself, which is not the same. Another possible solution would be always enforcing the use of property reification[1]. This has the disadvantage of some overhead of triples to express non-qualified relations that could just be expressed with a single triple, but

[1] We use "property reification" to denote the process of reifying a property/predicate and "RDF reification" for reifying a triple.

languages based on thematic roles, such as Universal Networking Language, as commented in section 1.2, take this approach. Another option is including some flexibility in the retrieval, so both reified and non-reified relations are retrieved together. These options are discussed later in this document. This problem can also occur when the property is `rdf:type`.

1.2 Event Modeling: Different Approaches

When we model an event by reifying a property, like described in section 1.1, there are different modelling options. We enumerate them below.

a) Like in the example in [2], using some rather specific properties like winner and prize. The property time, in contrast, is quite generic. The event can belong to an independent specific class (e.g. To-win) or a generic one (e.g. Event). We think the first approach is not recommendable because it is redundant. Another approach is making the event belong to specific classes defined as the domains and ranges of the specific properties, so the redundance is controlled and there is no need to keep two separate vocabularies.

b) Making the event belong to a specific class (To-win), and using strictly generic event-wise thematic roles (also called semantic roles [1, 704f.]), such as agent, patient, product, place, time, etc., as properties. This also avoids unnecesary redundancy between events and properties, but the burden of the vocabulary complexity is moved towards the events. It is the approach taken in Universal Networking Language (UNL) [3], a knowledge representation language used as interlingua for machine translation.

1.3 Single Entity or Complex Structure

Similarly to properties in section 1.2, an entity can be replaced with a complex structure that contains additional information, or the same information better structured. Two examples of this can be found in the RDF primer[2]. A similar case can happen with the use of `rdf:value`[3].

1.4 Numerous Overlapping Vocabularies

There are many different topic-specific vocabularies for RDF, with semantically overlapping terms. Alligning them with properties such as `owl:sameAs` is a problematic task which grows to an unmanageable size at the web scale. Manual methods are time- and resource-consuming and automatic methods are error-prone. When two equivalent resources are not properly linked by `owl:sameAs`, the whole graph becomes semantically sparser, and a query using one resource will not retrieve solutions that use the semantically equivalent one, thus reducing recall.

[2] `http://www.w3.org/TR/2004/REC-rdf-primer-20040210/#structuredproperties`
[3] `http://www.w3.org/TR/2004/REC-rdf-primer-20040210/#rdfvalue`

1.5 Instance Property or Subclass Property

OWL, which is based on description logics, forces a dichotomy between classes and instances. However, the choice of whether something is modeled as a class or as an instance (i.e. whether it belongs to the Abox or the Tbox in description logics) may depend on the context, like the assumptions the ontologist makes regarding the purpose and the granularity of the ontology.

For example, a particular car model can be considered an instance in an ontology set up to describe car models by their properties (manufacturer, year, body style, etc.). At the same time, in an ontology concerning production of cars by the manufacturer, it can be modeled as a class, possibly a subclass of the same class(es) it was an instance of in the previous case, so the instances are actual car-objects with a serial number. If both ontologies were to be linked, a fundamental problem would arise: a resource would simultaneously be a subclass and an instance of another resource. This, besides requiring higher-order logics (which OWL 2 can handle but in a limited way) is lexically inconsistent.

2 Proposed Approach

From the perspective of retrieving the information, solving the heterogeneities explained above would increase recall without significantly reducing precision, in the sense that relevant facts that before were excluded from the query results for using a different modeling choice than the query, would then be retrieved. In order to do this, we intend to combine two strategies:

At modeling stage, by declaring a general-purpose vocabulary, together with usage rules and patterns, that reduce as much as possible the number of different RDF graphs that relate to the same specification. This vocabulary could be based on Wordnet, for which a direct translation to RDF already exists [4]. Wordnet provides sense disambiguation, so polysemy would not be a problem. This would not contravene the open nature of the semantic web, as everyone could link vocabularies containing specific entities of their own responsibility (e.g., a company listing prices for their products). Also, for very specific terms or senses that are not present in Wordnet, specific linked vocabularies could still be used, in a similar fashion to how natural language can be extended with topic-specific jargons. However, a big part of the resources present in popular vocabularies of the semantic web, like Dublin Core or FOAF, are equivalent to certain synsets in Wordnet, and the same can be expected for similar future topic-specific vocabularies. Therefore, one of the most obvious benefits of this approach would be reducing the high amount of URI aliases that can be found among the many specific-purpose, but partially overlapping, existing RDF vocabularies, thus addressing the point 1.4.

Using Wordnet can also provide an additional advantage. Being related more directly to the English lexicon, it may reduce the learning curve for

human users, especially compared to learning many topic-specific heterogeneous vocabularies and their mutual links. However, some word types – like prepositions, and compound terms in general – are not part of Wordnet, so its use as a general vocabulary is not straighforward.

The point 1.5 could be circumvented by using the linguistic copula and dropping class modeling, but this means dropping RDFS/OWL logical inference too. The point 1.2 could be addressed by sticking to one of the two non-redudant options, preferably the one based on thematic roles, as it seems a more common approach.

Regarding the implementation of thematic roles on top of RDF, there are already other proposals, like SEM or LODE [4], but they seem limited in extent. A much more comprehensive set of thematic roles can be found in UNL (Universal Networking Language) [3], that provide the ability to express most of the meaning of any natural language utterance, and could be imported to RDF. In [3], it is proposed that UNL might be of use in the context of the semantic web, though RDF and the issues associated to linking heterogeneous datasets are not mentioned. Using an approach like UNL would also allow using *only* thematic roles as properties, which is equivalent to performing a comprehensive property reification, and it would solve the first point in 1.1. However, this enforcement is not realistic when linking data, as the existing corpus of knowledge in RDF would not comply. A more realistic approach would then be complementing the language with a flexible query system which retrieves both reified and unreified properties. We discuss this next.

At querying stage, by adding some flexibility that matches different graphs that have the same or similar semantics. There have been several proposals for flexible SPARQL [5], relaxing queries by replacing concepts with superconcepts, based on a background ontology, but this only matches a very specific kind of ontology-backed isomorphic graphs, and does not cover the modeling cases explained above. Therefore, a structural approach, similar to the one mentioned in [5], would be needed. Combining lexical and structural flexibility, the cases introduced before can be addressed by means of equivalence rules.

References

[1] D. Jurafsky and J. H. Martin, *Speech and Language Processing*, 2nd ed. Pearson Prentice Hall, 2009.

[2] F. M. Suchanek, G. Kasneci, and G. Weikum, "YAGO: A Core of Semantic Knowledge," in *WWW'07*, 2007, pp. 697–706.

[3] J. Cardeñosa, C. Gallardo, and L. Iraola, "Interlinguas: A classical Approach for the Semantic Web. A practical case," ser. MICAI '06, 2006, pp. 932–942.

[4] W. R. Van Hage, V. Malaisé, R. Segers, L. Hollink, and G. Schreiber, "Design and use of the Simple Event Model (SEM)," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 9, no. 2, pp. 128–136, 2011.

[5] O. Corby, R. Dieng-Kuntz, F. Gandon, and C. Faron-Zucker, "Searching the semantic web: Approximate query processing based on ontologies," *Intelligent Systems, IEEE*, vol. 21, no. 1, pp. 20–27, 2006.

Paper B

FrameBase: Representing N-ary Relations Using Semantic Frames

Jacobo Rouces, Gerard De Melo, Katja Hose

The paper has been published in the
Proceedings of the 12th Extended Semantic Web Conference (ESWC), 2015

The layout has been revised.

Abstract

Large-scale knowledge graphs such as those in the Linked Data cloud are typically represented as subject-predicate-object triples. However, many facts about the world involve more than two entities. While n-ary relations can be converted to triples in a number of ways, unfortunately, the structurally different choices made in different knowledge sources significantly impede our ability to connect them. They also make it impossible to query the data concisely and without prior knowledge of each individual source. We present FrameBase, a wide-coverage knowledge-base schema that uses linguistic frames to seamlessly represent and query n-ary relations from other knowledge bases, at different levels of granularity connected by logical entailment. It also opens possibilities to draw on natural language processing techniques for querying and data mining.

1 Introduction

Over the past few years, large-scale knowledge bases (KBs) have grown to play an important role on the Web. Many institutions rely on Linked Data principles to publish their data using Semantic Web standards [1]. These KBs are mostly based on simple subject-predicate-object (SPO) triples, as defined by the RDF model [2]. Such triples are convenient to process and can be visualized as entity networks with labeled edges.

Whereas triple representations work straightforwardly for relations involving two entities, many interesting facts relate more than just two participants – a problem that has gained renewed attention in several recent papers [3, 4] as well as in the current W3C proposal to add roles to schema.org [5]. For a birth event, for instance, one may wish to capture not just the time but also the location and parents. For an actress starring in a movie, the name of the portrayed character may be relevant. Such facts naturally correspond to n-ary relations. In order to capture them as triples, several different representation schemes have been proposed. Table D.1 shows some possibilities of expressing that an entity John was married in 1964, some of which also include additional information such as the name of the bride. We will discuss these representations in more detail later in Sect. 2.

As the example shows, this sort of semantic heterogeneity leads to significant data integration challenges. One KB might use a simple binary property between two entities, whereas another may instead choose a more complex representation that accommodates additional arguments. The representations can easily be so at odds with each other that no particular mapping between entities could bridge the differences. There are entities at each side that have no counterpart at the other. This leads to several challenging problems:

1. When **linking data**, there are currently no mechanisms to connect KBs with different modeling choices. Predicates exist to link equivalent classes,

instances, or properties, but not for connecting the different patterns, as explained above. Existing work on ontology and KB alignment [6] is limited to finding aliases.

2. When **querying**, the query must be built in a way that fits the particular modeling choices made for the respective KB. Otherwise, the recall may be as low as zero [7]. Even worse, when we don't have a single coherent KB but a set of different KBs, there is no simple query (as could be formulated on a single given schema) that will have a high recall across all KBs.
3. When **natural language interfaces** to KBs are queried, state-of-the-art systems typically attempt to map verbs and predicate phrases to RDF predicates [8]. This approach, however, cannot be applied when the KB fails to provide a compatible binary relation.

Direct Binary Relation

```
John  marriedOnDate 1964 .
```

RDF Reification

```
John marries        Mary .
s    type           Statement .
s    subject        John .
s    property       marries .
s    object         Mary .
s    time           1964 .
```

Subproperties

```
p    subPropertyOf  Marriage .
John p              Mary .
p    time           1964 .
```

Neo-Davidsonian (Specific Roles)

```
e type   Marriage .
e groom  John .
e bride  Mary .
e time   1964 .
```

Neo-Davidsonian (General Roles)

```
e type   Marriage .
e agent  John .
e agent  Mary .
e time   1964 .
```

Table B.1: Triple Representations of n-ary Relations

FrameBase. To address these problems, we have created FrameBase, a broad-coverage schema that can homogeneously integrate other KBs and has strong connections to natural language. It overcomes the above-mentioned forms of heterogeneity – by sticking to a specific modeling choice general enough to subsume the others (neo-Davidsonian representation) – together with a large vocabulary for events and roles. This vocabulary is reusable and based on an extensible hierarchy. We also develop a mechanism to convert back and forth between the new representation and direct binary relations, using a vocabulary of binary relations automatically generated from linguistic annotations. These are more concise and can be used when only two arguments are relevant.

This paper is structured as follows. After analyzing the state of the art in Sect. 2, an overview of FrameBase is given in Sect. 3. Section 4 explains how we construct the FrameBase schema, while Sect. 5 presents our representation

	All triples	Core	Linking event	Reasoning
RDF Reification	$(n+4)k$	$(n+3)k$	$+k(k-1)$	1 definite clause
Subproperties	$(n+2)k$	$(n+1)k$	$+k(k-1)$	RDFS
Neo-Davidsonian	$1+n+k$	$1+n$	$+0$	>1 def. clauses

Table B.2: Triple Overhead. *n is the number of participants in an event, and k the number of pairs that are relevant to be linked by direct binary relations. The first column indicates the total number of triples that can be materialized. The second column excludes direct binary relationships, which can be inferred unambiguously by the inference system in the last column. In the case of RDF reification, this inference could be accomplished by a rule creating the triple from its reification triples. In the case of neo-Davidsonian representation, we use rules of a different form (described later in Sect. 5). In both cases, each rule is a definite clause, i.e. a disjunction of logical atoms with only one negated, which is the consequent when the clause is written as an implication. The third column indicates the number of triples needed to connect entities that represent the same event, which is a phenomenon that arises when using RDF reification or subproperties.*

conversion mechanism. Section 3 provides a qualitative evaluation, and Sect. 3 concludes the paper with an outlook to future work.

2 State of the Art

In this section, we review related work and conduct a thorough analysis of existing approaches for modeling n-ary relations, which are synthesized in Table D.1. In Table D.2, we provide a detailed comparison of their space efficiency, which has consequences with regards to their applicability for large-scale KBs.

2.1 Direct Binary Relations

A common way to represent n-ary facts is to simply decompose them directly into binary relations between two participants [9]. But in doing so, important information may be lost. For instance, given a triple with property `wasMarriedOnDate` and two triples with `gotMarriedTo`, we cannot be sure to which marriage the given time span applies.

2.2 RDF Reification

The RDF standard proposes RDF reification [2], which introduces a new identifier (IRI) for a statement and then describes the original RDF statement using three new triples with `subject`, `predicate`, and `object` properties. Subsequently, arbitrary properties of the statement can be captured by adding further triples about it.

In the different versions of YAGO [10], RDF reification is used to attach additional information to the event represented by the original RDF triple (evoked by its property) – as in the *RDF Reification* example in Table D.1. This has the advantage that both the original triple as well as the reified triple can

be present in the KB and queries that do not require the additional information can still use the original binary relation directly. However, this also has several drawbacks:

- Formally, the event represented by a triple and the triple as a statement are different entities with different properties. For instance, an institution may endorse the triple as a statement without endorsing the marriage. Using RDF reification, both are represented by the same RDF resource identifier, which conceptually is meant to be unambiguous. This is a potential source of confusion and inconsistency.
- The number of triples increases by a factor of 4. For each triple `S P O`, one has to add `T a rdf:Statement`, `T rdf:subject S`, `T rdf:predicate P`, and `T rdf:object O`. These do not add any new information themselves but are merely a prerequisite for then being able to extend the original binary relation to an n-ary relation by subsequently adding more triples with `T` as subject.
- The advantage of being able to include the original non-reified triple only applies for the primary binary relation, and not for the other $\frac{n(n-1)}{2} - 1$ ones that can be formed (not counting inverses). Some of these may be rare or irrelevant, but others may be important and are indeed used in YAGO (e.g. `bornAtPlace`, `bornOnDate`).
- The choice of the primary pair of entities and their binary relation (John and Mary in Table D.1) is arbitrary, and a third party willing to query the KB cannot replicate the choice independently. If their choice is different, they will not obtain any results. A possible solution, which is actually implemented in YAGO2s, is to include the triples for the other pairs and reify them, too, but this adds yet another factor of overhead, besides data redundancy that would complicate updates.
- When two or more different events share the same values for the primary pair of arguments, they will share the same triple, but require separate reifications, producing non-unique triple identifiers. For example, if there are two flight connections between Paris and London with different airlines, the triple `Paris isConnectedTo London` will be reified twice, with two different triple identifiers.

If the triplestore implementation makes use of quads[1], the 4-fold overhead can be avoided (though the underlying storage needs a new column), but the other disadvantages still remain. Quad-based singleton named graphs [2] could be used instead of RDF reification, the problems being the same.

2.3 Subproperties

A recent proposal [4] aims to solve some of the issues with RDF reification by instead declaring a subproperty of the original property in the primary pair,

[1] `http://www.w3.org/TR/n-quads/`

and using this subproperty as the subject for the other arguments of the n-ary relation. This is shown in the *Subproperties* example in Table D.1.

While the approach enables us to use RDFS reasoning to obtain the triple with the parent property that relates two of the participants, and also reduces the overhead of RDF reification, it still suffers from the problems mentioned above related to the existence of a primary pair. For one, the non-reified binary relationships for the other pairs cannot be inferred from that subproperty.

2.4 Neo-Davidsonian Representations

Another approach, and the one that we will adapt for FrameBase, is to make use of so-called neo-Davidsonian representations [11, p. 600f.]. This means that we first define an entity that represents the event or situation (also referred to as a *frame*) underlying the n-ary relation. Then, this entity is connected to each of the n arguments by means of a property describing the *semantic role* [3, 12].

Note that the process of converting from the binary representation to the neo-Davidsonian one is also called reification, but this is different from *RDF reification* as discussed earlier. In RDF reification, an entity is defined that stands for a whole triple so that additional triples can be used to describe the reified triple as a unit that represents a statement. However, in the context of event semantics, reification is used to denote the process by which an entity is defined that refers to the event, process, situation, or more generally, frame, evoked by a property or binary relation. Having done this, additional information about it can then easily be added. Both kinds of reification have in common that a new entity is defined to refer to something that before was not explicitly represented by an entity in the KB, but in one case it is a RDF statement while in the other it is an event.

Advantages. Table D.2 compares the neo-Davidsonian approach to the alternatives. These require a lot more triples when several direct binary relations need to be included. In the worst case, $k = \frac{n(n-1)}{2}$ despite discounting reciprocal relations, but even if not all of these relations are relevant, connecting all agents and possibly patients to all other elements would be relevant, which would easily satisfy $k > n$.

Semantic Heterogeneity. Unfortunately, there are different ways of using the neo-Davidsonian approach, with different levels of granularity for the events and the semantic roles, from a very small set of abstract generic ones [13] to more specific ones [14].

The Simple Event Model (SEM) Ontology [15] falls within the category of neo-Davidsonian representation with general roles (see Table D.1). It defines four very general entities, *Event*, *Actor*, *Place*, and *Time*. It also establishes a framework for creating more specific ones by extending these, but it does not provide these extensions, nor ways to integrate existing KBs in a way that would solve the problem of semantic heterogeneity. Similarly, LODE (Linking

Open Descriptions of Events) [13] specifies only very general concepts such as the four just mentioned.

Freebase [14] is a KB built both from tapping on existing structured sources and via collaborative editing. Although it uses its own formalisms, there are official and third-party translations to RDF. Freebase makes use of so-called *mediators* (also called *compound value types*, CVTs) as a way to merge multiple values into a single value, similar to a `struct` datatype in C. There are around 1,870 composite value types in Freebase (1,036 with more than one instance) and around 14 million composite value instances. While CVTs do not represent frames or events per se, from a structural perspective, they can be regarded as isomorphic to a neo-Davidsonian representation with specific roles (see Table D.1). However, Freebase places a number of restrictions on CVTs. For instance, they cannot be nested, and there is no hierarchy or network of them that would for example relate a purchasing event to a getting event.

There is ongoing work to add the modeling of semantic roles to schema.org [5]. Schema.org is an effort sponsored by Google, Yahoo, and Microsoft to establish common standards for semantic markup in Web pages. Currently, the new roles pattern proposal is just a proposed model without a proper role inventory, and schema.org merely targets a small restricted number of domains.

FrameNet [16, 17] is a well-known resource in natural language processing (NLP) that defines over 1,000 *frames* with participants (so-called *frame elements*). For example, the verb *to buy* and the noun *acquisition* are assumed to evoke a commercial transaction frame, with frame elements for the seller, the buyer, the goods, and so on.

Previous work has proposed general patterns for using FrameNet in knowledge representation [18] and converted FrameNet to RDF [19], proposing a way to generate schemas from FrameNet. Similarly, the FRED system [20] for building semantic representations from natural language can be configured to use FrameNet.

3 System Overview

As we have seen, there are a number of different representations used in KBs. In this paper, we use the linguistic resources FrameNet [16] and WordNet [21] to fully develop an extensive schema for large-scale knowledge representation and integration. The schema is composed of an expressive neo-Davidsonian level that draws on a large common inventory of frames, together with a more concise level of direct binary relations, which is connected to the former by means of inference rules.

3.1 FrameNet-based Representation

The use of FrameNet is motivated by the following considerations.

- FrameNet has a long history and aims at descriptions of arbitrary natural language. It thus provides a relatively large and growing inventory of frames and roles, with a broad coverage of numerous different domains.
- FrameNet comes with a large collection of English sentences annotated with frame and frame element labels. This data led to the task of automatic *semantic role labeling* (SRL) [22] of text, now one of the standard tasks in NLP. This strong connection to natural language facilitates question answering and related tasks.
- While FrameNet's lexicon and annotations cover the English language, its frame inventory is abstract enough to be adopted for languages as different as Spanish and Japanese [23]. This also makes it much more suitable as a basis for knowledge representation than language-specific syntax-oriented SRL resources such as PropBank [24].
- FrameNet provides an reasonable level of granularity for the phenomena that humans care to describe. From a theoretical perspective, there is no universally appropriate single level of reification. Any frame element might be reified on its own, and any two elements of a frame could be connected directly by a predicate. Using FrameNet, we strike a well-motivated balance, at a point that is granular enough to constitute a model for natural language semantics. As we will explain in Sect. 5, we also provide a second level of representation, less expressive but more concise, based on the direct binary predicates between frame elements.

3.2 Overview

For creating the FrameBase schema using FrameNet, we take the following steps, which will be further explained in Sect. 4.

a) **FrameNet–WordNet Mapping.** First, we create a high-precision mapping between FrameNet and another well-known lexical resource called WordNet [21], which will be used to enrich the lexical coverage and relations of the FrameBase schema.
b) **Schema Induction.** We use FrameNet, WordNet, and the mapping to create an RDFS schema for FrameBase that has very wide coverage and is extensible. The schema exploits semantic relations from these components (e.g., synonymy, hyponymy, and perspectivization) to transform the original resources for our lightweight RDFS model.
c) **Automatic Reification–Dereification Mechanism.** We create reification–dereification rules in the form of definite clauses that allow the KB to be queried independently of whether a frame is reified or not, and that may also be used to reduce overhead in the KB.

4 FrameBase Schema Creation

4.1 FrameNet–WordNet Mapping

While FrameNet [16, 17] is the largest high-quality inventory of semantic frame descriptions and their participants, WordNet [21] is the most well-known resource capturing meanings of words in a lexical network, covering for example nouns and named entities missing in FrameNet. WordNet, for instance, serves as the backbone of YAGO's ontology. We propose a novel way of mapping the two resources, which later enables us to integrate both of them into our schema.

WordNet contains synsets, which are sets of sense-disambiguated synonymous words with a given part of speech (POS), such as noun or verb. FrameNet contains lexical units (LUs), which are also POS-annotated words associated to frames. Because of the semantics of the containing frame, lexical units are also disambiguated to a certain extent, though not with the same granularity as in WordNet. Our objective is to map synsets and lexical units with the same meaning, so we can later use this to enrich our FrameNet-based schema with relations and annotations from WordNet.

We choose to map each lexical unit to one and only one synset. While there are some lexical units that could be mapped to more than one synset, this will favor precision, which is desirable for the purpose of obtaining a clean knowledge base. The only cases where this model would be detrimental to precision are those where lexical units do not have any associated synset, but these are few and most can easily be avoided by omitting lexical units with parts of speech not covered in WordNet, such as prepositions.

Our choice allows us to model the mapping as a function $S(l|a,b)$ from lexical units to synsets as in (D.1). S_l stands for the synsets that have the same lexical label and POS as the lexical unit l, μ_L and μ_G are the lexical and gloss (definition) overlap, respectively, f yields the corpus frequency of the synset, and a and b are parameters for a linear combination (the third parameter can be omitted because of the argmax function).

$$S(l|a,b) = \operatorname*{argmax}_{s \in S_l} \mu_\mathrm{L}(l,s) + a \cdot \mu_\mathrm{G}(l,s) + b \cdot f(s) \tag{B.1}$$

The lexical overlap μ_L of a lexical unit l and a synset s is the size of the intersection between the POS-annotated words from the lexical units in the same frame as l and the POS-annotated words in s and its neighborhood. We define the neighborhood as the synsets connected by a selection of lexical and semantic pointers such as "See also", "Similar to", "Antonym", "Attribute" and "Derivationally related". This expansion is useful to reduce sparsity and better match the sets with those generated for the lexical units, which due to the different semantics of frames and synsets, may already include these related words.

The gloss overlap μ_G is the size of the intersection between the set of words in the definition of the lexical unit and the gloss of the synset. For preprocessing these, we rely on the CoreNLP library [25] to clean XML tags, tokenize, POS-label, and lemmatize the text, and we filter out all words except nouns and verbs.

We trained a and b with a greedy search over several randomized seeds, obtaining optimal values $a = 5, b = 0.13$.

4.2 Schema Induction

We model frames as classes whose instances are the particular events. The frame elements of each frame are properties whose domain is that frame. We create a class hierarchy of frames as follows.

1. **General Frames:** FrameNet's frame inheritance and perspectivization relations are modeled as class subsumption between frames, by means of two specific properties that inherit from `rdfs:subClassOf`, so that both remain distinguishable but contribute to the hierarchy and allow RDFS inference. We additionally declared a top frame for the hierarchy. Inheritance between frame element properties is modeled with a direct subproperty relation.
   Thus, under this model, an instance of the *Commerce_sell* frame with a certain *Commerce_sell-Buyer* x, is also an instance of the *Giving* frame and x is the *Giving-Recipient*, because the first frame inherits from the latter. Likewise, it is also an instance of *Transfer* and x is the *Transfer-Recipient*, because *Giving* is a perspective on *Transfer*.
2. **Leaf Nodes:** Since FrameNet's original frame inventory is coarse-grained and different lexical units like *construction* and *to glue* evoke the same frame, we generate what has occasionally been called a *microframe* model: We transform FrameNet such that every lexical unit is treated as evoking its own separate fine-grained frame, which is made a subclass of the more coarse-grained original FrameNet frame.
3. **Intermediate Nodes** The microframe nodes are very fine-grained, e.g. distinguishing *buy* from *acquire*, while some original frames from FrameNet are very coarse-grained, as mentioned above. For instance, various kinship relationships such as *mother*, *sister-in-law*, etc. are lumped together. This wide range of lexical units may stand in various lexical-semantic relationships without these being indicated, including synonymy, antonymy, or nominalization. The only characteristic they have in common is that, by definition, they evoke a similar kind of situation. Overall, neither the fine-grained nor the coarse-grained levels are ideal for knowledge representation purposes. We address this by providing a novel intermediate level composed of *synset-microframes* that group equivalent *LU-microframes* together. For this, we generate a set of directly equivalent synset-microframes for each LU-microframe, and we declare `owl:equivalentClass` predicates between

these pairs. This is the only predicate we use that needs inference beyond pure RDFS, but we also include a pair of reciprocal `rdfs:subClassOf`, which is semantically equivalent and leaves the possibility of using any out-of-the-box RDFS inference engine. The clusters are thus defined as the resulting equivalence classes over the set of all microframes.
These clusters are built in several steps. First, for a given LU, we get the corresponding synsets from the FrameNet–WordNet mapping in Sect. 4.1. In the case of our mapping, the set has no more than one element, but in the general case it could have more. Then, we expand that set by adding all other synsets related by lexical relations reflecting cross-POS morphological transformations: "Derivationally related", "Derived from Adjective", "Participle" and "Pertainym". In general, these lexical relations do not necessarily imply any close semantics (e.g., *create/make – creature/animal*), but when restricted to synsets all tied to the same FrameNet frame, such cases are normally factored out. The goal of using the lexical relations is linking cross-POS LUs that evoke the same specific situation with a different syntactic form, such as nominalizations (*produce–production*), non-finite verb forms (*produce–produced*), adjectivization, or adverbization.

We also use names, definitions and glosses in FrameNet and WordNet to create text annotations for our schema. We attach lexical forms with `rdfs:label` and definitions and glosses from FrameNet and WordNet with `rdfs:comment`.

5 Automatic Reification–Dereification Mechanism

While frames are convenient for representational purposes, users wishing to query the knowledge base benefit from binary predicates between pairs of frame elements. For example, for a birth event, binary predicates like `bornInPlace` and `bornOnDate` can facilitate querying by offering a more compact and simple representation.

We thus present a novel mechanism to seamlessly convert between frame representations and DBPs. This mechanism can also allow us to avoid materializing frame instances when only two frame elements are needed.

We generate *dereification rules* of the following form:

```
?s BinaryPredicate ?o ← ?f a Frame, ?f FE1 ?s, ?f FE2 ?o
```

Additionally, for each dereification rule there is a converse reification rule so that one can go back from binary predicates to the frame representation. Each direct binary predicate (DBP) has only one set of possible frame and frame elements associated, and therefore chaining reification and dereification rules is an idempotent operation.

We build the reification–dereification rules automatically using the annotations of English sentences given for different LUs in FrameNet, namely

the grammatical functions (GFs) and phrase types (PTs) [17] associated with different frame elements in the example sentences of each lexical unit.

For verb-based microframes, FrameNet provides three kinds of GF labels: External Argument (Ext), Object (Obj), and Dependent (Dep). Some of the PT labels that can be found are N, NP, Obj, PPinterrog [17]. We create dereified binary predicates for the pairs of frame elements whose syntactic annotations for some sentence satisfy the creation rules below, using the GF and grammatical PT labels. We list the creation rules below, and add some examples of reification-dereification rules associated to the DBPs created by some of them. The postfixes "-s" and "-o" indicate the data associated to the FEs that fill the first and second arguments of the DBP, or equivalently, the subject and the object of the resulting RDF triple.

- Create DBP with name "ConjugateThirdPersSing(LU)" if

  (GF-s equals Ext) & (GF-o equals Obj) &

  (PT-o in { N, NP, Obj, PPinterrog, Sinterrog, QUO, Sfin, Sub, VPing })

  Examples of obtained resulting DBPs and reification-dereification rules:

```
?S :dereif-Forming_relationships-divorces ?O
          ⎧ ?R a :frame-Forming_relationships-divorce.v ,
      ↔   ⎨ ?R :fe-Forming_relationships-Partner_1 ?S ,
          ⎩ ?R :fe-Forming_relationships-Partner_2 ?O .
?S :dereif-Win_prize-wins ?O
          ⎧ ?R a :frame-Win_prize-win.v ,
      ↔   ⎨ ?R :fe-Win_prize-Competitor ?S ,
          ⎩ ?R :fe-Win_prize-Prize ?O .
```

- Create DBP with name "is ConjugatePastParticiple(LU) by" if

  (GF-s equals Obj) & (GF-o equals Subj) &

  (PT-o in { N, NP, Obj, PPinterrog, Sinterrog, QUO, Sfin, Sub, VPing })

- Create DBP with name "ConjugateThirdPersSing(LU) Prep" if

  (GF-s equals Ext) & (GF-o equals Dep) & (PT-o equals PP(Prep))

  Examples of obtained resulting DBPs and reification-dereification rules:

```
?S :dereif-Creating-createsFrom ?O
          ⎧ ?R a :frame-Creating-create.v ,
      ↔   ⎨ ?R :fe-Creating-Creator ?S ,
          ⎩ ?R :fe-Creating-Components ?O .
?S :dereif-Win_prize-winsAt ?O
          ⎧ ?R a :frame-Win_prize-win.v ,
      ↔   ⎨ ?R :fe-Win_prize-Competitor ?S ,
          ⎩ ?R :fe-Win_prize-Venue ?O .
```

For some FEs in this and the next rule, we assign a specific preposition, like "at" for `Time` *and "in" for* `Place`*. For example:*

```
?S :dereif-Destroying-destroysAtTime ?O
          ⎧ ?R a :frame-Destroying-destroy.v ,
      ↔   ⎨ ?R :fe-Destroying-Cause ?S ,
          ⎩ ?R :fe-Destroying-Time ?O .
?S :dereif-Intentionally_create-establishesInPlace ?O
          ⎧ ?R a :frame-Intentionally_create-establish.v ,
      ↔   ⎨ ?R :fe-Intentionally_create-Creator ?S ,
          ⎩ ?R :fe-Intentionally_create-Place ?O .
```

- Create DBP with name "`is` ConjugatePastParticiple(LU) Prep" if

  (GF-s equals Obj) & (GF-o equals Dep) & (PT-o equals PP(Prep))

By using the grammatical subject as subject of the triple, we avoid rules defining certain kinds of DBPs that would be rarely useful, like those connecting the time and place, or the place and the cause.

There is no explicit syntactic annotation in FrameNet to indicate if the example sentences are in passive form. We used two different heuristics for detecting this. One draws on the POS annotations available in FrameNet, and decides that a sentence is in passive iff the target (LU) verb is conjugated as a past participle, and there is a conjugated form of the verb *to be* in a prior position, without another verb in between. The other heuristic uses the Stanford Parser [26]. Both heuristics make type I and II mistakes differently, so we discarded the cases where they disagree, and for the ones that they agree that they are passive, we created the rules inverting the Ext/Obj GFs.

We restrict ourselves to verb-based microframes, because the process above is more difficult and error-prone with nouns. However, the synset-microframe clustering of our schema already makes many of the morphosemantic variations of a verb, including nominalizations, logically equivalent.

With the rules obtained with the process above, the same DBP can be associated to different pairs of frame elements in a given LU-microframe, owing to different senses or syntactic frames for a given verb (for example the transitive and intransitive frames for *smuggle*). This would conflate different senses, and if the reification and the dereification directions of the rules were chained, it would logically entail different pairs of frame elements, which would not be sound. Furthermore, a given pair of frame elements can also produce different DBPs. To achieve the idempotency mentioned earlier, we use the Kuhn–Munkres algorithm to obtain a one-to-one assignment, using as weights the number of annotated example sentences for a DBP and a pair of frame elements, because the patterns with more example sentences are usually

more intuitive. The cubic complexity of the algorithm is not a concern because each frame leads to a separate graph on which we can operate independently.

We have implemented the reification-dereification rules as SPARQL CONSTRUCT queries, due to SPARQL's prominence as a standard query language for KBs. These can be used to materialize the DBPs into the KB. Other options would be possible, such as using a general-purpose inference engine that can handle propositional clauses, like the Rubrik reasoner in Jena [27].

6 Evaluation

We now evaluate the quality of the results and show some example queries.

6.1 FrameNet–WordNet Alignment

To evaluate the created schema, we first compared our FrameNet–WordNet mapping to the MapNet gold standard [28]. MapNet uses older versions of FrameNet and WordNet, so that we had to apply mappings from WordNet 1.6 to 3.0 [29], removing those with a confidence lower than one. For mapping FrameNet 1.3 to 1.5, we removed the few LUs that are not contained in the new version. Table D.3 compares the results against state-of-the-art approaches and the scores that they report on the MapNet gold standard. As expected, our approach achieves high precision, while still maintaining good recall. We use 5-fold cross-validation for our results.

	Prec	Rec	F1	Acc
SVM Polyn. kernel 1 [28]	0.761	0.613	0.679	—
SVM Polyn. kernel 2 [28]	0.794	0.569	0.663	—
SSI-Dijkstra [30]	0.78	0.63	0.69	—
SSI-Dijkstra+ [30]	0.76	0.74	0.75	—
Neighborhoods [31]	—	—	—	0.772
Our mapping	0.789	0.709	0.746	0.864

Table B.3: Comparison of our FrameNet–WordNet mapping to state-of-the-art approaches in terms of precision, recall, F1, and accuracy

6.2 Schema Induction

The FrameBase schema is based on FrameNet and WordNet and our mappings between the two resources. It provides 19,376 frames, including 11,939 LU-microframes and 6,418 synset-microframes, all with lexical labels. A total of 18,357 microframes are clustered into 8,145 logical clusters, which are the sets of microframes whose elements are linked by a logical equivalence relation. The size of the schema is 250,407 triples.

We have obtained an average precision of $87.55\% \pm 6.18\%$ with a 95% Wilson confidence interval. The evaluation showed a small change of nuance

for 31.15% ± 9.38% of the correct pairs – most of these are caused by our choice to use semantic pointers such as "Similar to", which could be removed if we desire very fine-grained distinctions of microframes. The precision has been calculated from a random sample of 100 intra-cluster pairs that have been independently annotated by two of the authors. We have obtained the linear weighted Cohen's Kappa over the three-valued combination of the two variables with which we annotate each cluster pair, obtaining a value of 0.23 over a maximum of 0.87. We obtained the scores with a random annotator.

In addition to the number of frames, the FrameBase schema provides a vocabulary of frame elements that goes well beyond the knowledge currently included in most KBs, in particular beyond time and location. This additional knowledge is routinely conveyed in natural language, and we believe that using a schema that provides for it paves the way to include it in KBs, either manually or automatically.

6.3 Reification–Derefication Rules

We also provide 14,930 reification–dereification rules for the same number of direct binary predicates, with both human-readable IRIs and lexical labels. We obtained an average precision of 86.59% ± 6.41%, and 76.13% ± 8.65% of the correct rules were found easily readable. We consider a rule to be not easily readable if the name of the direct binary predicate contains a frame element whose meaning is not obvious for a layman reader, or if it contains a preposition that is appropriate for some but not all possible objects, or it is not appropriate for the frame element in the name. For this evaluation, we followed the same annotation methodology as for the intra-cluster pairs, obtaining a Cohen's kappa of 0.39 over a maximum of 0.54.

6.4 Knowledge Base Integration and Querying

Knowledge from other KBs such as Freebase can be integrated using *integration rules*. These rules can also be implemented as SPARQL CONSTRUCT queries. The two examples below were created manually.

```
CONSTRUCT {
  _:e a framebase:frame-People_by_jurisdiction-citizen.n .
  _:e framebase:fe-People_by_jurisdiction-Person ?person .
  _:e framebase:fe-People_by_jurisdiction-Jurisdiction ?country .
} WHERE {
  ?person freebase:people.person.nationality ?country .
}
```

```
CONSTRUCT {

  _:e a framebase:frame-Leadership-leader.n .
  _:e framebase:fe-Leadership-Leader ?o1 .
  _:e framebase:fe-Leadership-Governed ?o2 .
  _:e framebase:fe-Leadership-Role ?o3 .
  _:e framebase:fe-Leadership-Type ?o4 .
  _:timePeriod a framebase:frame-Timespan-period.n .
  _:timePeriod framebase:fe-Timespan-Start ?o5 .
  _:timePeriod framebase:fe-Timespan-End ?o6 .
} WHERE {
  ?cvti a freebase:organization.leadership .
  OPTIONAL { ?cvti freebase:organization.leadership.person ?o1 .}
  OPTIONAL { ?cvti ...:organization.leadership.organization ?o2 .}
  OPTIONAL { ?cvti freebase:organization.leadership.role ?o3 .}
  OPTIONAL { ?cvti freebase:organization.leadership.title ?o4 .}
  OPTIONAL { ?cvti freebase:organization.leadership.from ?o5 .}
  OPTIONAL { ?cvti freebase:organization.leadership.to ?o6 .}
}
```

FrameBase facilitates novel forms of queries. The following query, for instance, uses reified patterns to find the heads of the World Bank. Note that the clusters implemented in RDFS allow searching for the noun *head* (from the leadership frame), although the integration rule above only produced an instance of `fmbs:frame-Leadership-leader.n`. The results in Table D.4 show example instances seamlessly integrated into our FrameBase schema from both Freebase (rows 1–3, extracted from the second example integration rule above) and YAGO2s (rows 4–5, extracted with a similar integration rule made for YAGO2s).

```
SELECT DISTINCT ?leader ?role WHERE {
  ?lumfi a fmbs:frame-Leadership-head.n .
  ?lumfi fmbs:fe-Leadership-Governed ?worldBank.
  ?lumfi fmbs:fe-Leadership-Leader ?leader .
  VALUES ?worldBank {yago:World_Bank freebase:m.02vk52z}
  OPTIONAL{ ?lumfi fmbs:fe-Leadership-Role ?role }
}
```

Alternatively, a direct binary predicate from the dereification rules can be used to obtain the same non-optional results, as illustrated in the query below. Either *leads* or *heads* can be used because the LU-microframes for these verbs are in the same cluster as the nouns *leader* and *head*, and there is a dereification rule between the *Leader* and *Governed* frame elements for both.

```
SELECT DISTINCT ?leader WHERE {
  ?leader fmbs:dereif-Leadership-heads ?worldBank.
  VALUES ?worldBank {yago:World_Bank freebase:m.02vk52z}
}
```

FrameBase can also be applied with natural language processing tools for question answering and data mining. For example, given the question "Who has been the head of the World_Bank", the SRL tool SEMAFOR [32] successfully extracts the frame *Leadership* with lexical unit *head.noun* and frame elements *Governed* and *Leader*. Based on this, and after a named entity disambiguator like AIDA [33] matches World_Bank to the entities in the KBs, the structured query can easily be built. Moreover, the same procedure can also be used to integrate new knowledge from a text into the KB, like FRED [20] does.

7 Conclusion

FrameBase is a novel approach for connecting knowledge from different heterogeneous sources to decades of work from the NLP community. Events can be described in very different ways across different knowledge bases. Our framework not only provides an efficient model to describe n-ary relations, but also integrates and transforms FrameNet and WordNet to yield a broad-coverage inventory of frames. Additionally, linguistic annotations in FrameNet such as the ones used to create the reification–dereification rules can also be used to generate natural language, for instance, for summarizing a portion of a KB for non-technical users.

Regarding future lines of work, we are currently completing the integration of the instance data from YAGO2s and Freebase into the FrameBase schema, using integration rules such as the examples in Sect. 7.4, but automatically generated. This will lead to the first large-scale FrameNet-based KB. Given FrameBase's close connection to natural language, we also intend to study methods for better adapting semantic role labeling tools to question

?leader	?role
fb:m/0h_ds2s 'Caroline Anstey'	fb:m/04t64n 'Managing Director'
fb:m/0d_dq5 'Mahmoud Mohieldin'	fb:m/04t64n 'Managing Director'
fb:m/047cdkk 'Sri Mulyani Indrawati'	fb:m/01yc02 'Chief Operating Officer'
yago:Jim_Yong_Kim	-
yago:Robert_Zoellick	-

Table B.4: Results from the query

answering [32]. We are also investigating the ways that FrameBase enables for querying multiple KBs simultaneously with on-the-fly data integration.

Please refer to `http://framebase.org` for information on using FrameBase.

References

[1] C. Bizer, T. Heath, and T. Berners-Lee, "Linked data–the story so far," *IJSWIS*, vol. 5, no. 3, pp. 1–22, 2009.

[2] P. Hayes and P. Patel-Schneider, "RDF 1.1 semantics," W3C, Tech. Rep., 2014, http://www.w3.org/TR/rdf11-mt/.

[3] A. Gangemi and V. Presutti, "A Multi-dimensional Comparison of Ontology Design Patterns for Representing n-ary Relations," ser. SOFSEM '13, P. Emde Boas, F. Groen, G. Italiano, J. Nawrocki, and H. Sack, Eds., 2013.

[4] V. Nguyen, O. Bodenreider, and A. Sheth, "Don't Like RDF Reification?: Making Statements About Statements Using Singleton Property," ser. WWW '14, 2014.

[5] "Roles in Schema.org," W3C Consortium, Tech. Rep., 2014, https://www.w3.org/wiki/WebSchemas/RolesPattern.

[6] C. Böhm, G. de Melo, F. Naumann, and G. Weikum, "LINDA: Distributed Web-of-data-scale Entity Matching," in *CIKM'12*, 2012, pp. 2104–2108.

[7] J. Rouces, "Enhancing Recall in Semantic Querying," ser. SCAI '13, vol. 257, 2013, p. 291.

[8] M. Yahya, K. Berberich, S. Elbassuoni, M. Ramanath, V. Tresp, and G. Weikum, "Natural language questions for the web of data," ser. EMNLP-CoNLL '12, 2012. [Online]. Available: http://www.aclweb.org/anthology/D12-1035

[9] L. Del Corro and R. Gemulla, "Clausie: Clause-based open information extraction," ser. WWW '13, 2013.

[10] J. Hoffart, F. M. Suchanek, K. Berberich, and G. Weikum, "YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia," *Artificial Intelligence* , vol. 194, no. 0, pp. 28–61, 2013.

[11] D. Jurafsky and J. H. Martin, *Speech and Language Processing*, 2nd ed. Pearson Prentice Hall, 2009.

[12] N. Noy and A. Rector, "Defining N-ary Relations on the Semantic Web," W3C Consortium, W3C Working Group Note, April 2006, http://www.w3.org/TR/swbp-n-aryRelations/.

[13] R. Shaw, R. Troncy, and L. Hardman, "LODE: Linking Open Descriptions of Events," in *ASWC '09*, ser. Lecture Notes in Computer Science, 2009, pp. 153–167.

[14] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, "Freebase: A Collaboratively Created Graph Database for Structuring Human Knowledge," in *SIGMOD'08*, 2008, pp. 1247–1250.

[15] W. R. Van Hage, V. Malaisé, R. Segers, L. Hollink, and G. Schreiber, "Design and use of the Simple Event Model (SEM)," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 9, no. 2, pp. 128–136, 2011.

[16] C. J. Fillmore, C. R. Johnson, and M. R. Petruck, "Background to Framenet," *International Journal of Lexicography*, vol. 16, no. 3, pp. 235–250, 2003.

[17] J. Ruppenhofer, M. Ellsworth, M. R. Petruck, C. R. Johnson, and J. Scheffczyk, *FrameNet II: Extended Theory and Practice*. ICSI, 2006.

[18] A. Gangemi and V. Presutti, "Towards a pattern science for the semantic web," *Semantic Web*, vol. 1, no. 1, pp. 61–68, 2010.

[19] A. G. Nuzzolese, A. Gangemi, and V. Presutti, "Gathering lexical linked data and knowledge patterns from FrameNet," ser. K-CAP '11, 2011, pp. 41–48.

[20] V. Presutti, F. Draicchio, and A. Gangemi, "Knowledge Extraction Based on Discourse Representation Theory and Linguistic Frames," in *Knowledge Engineering and Knowledge Management*, ser. Lecture Notes in Computer Science, A. Teije, J. Völker, S. Handschuh, H. Stuckenschmidt, M. d'Acquin, A. Nikolov, N. Aussenac-Gilles, and N. Hernandez, Eds. Springer Berlin Heidelberg, 2012, vol. 7603, pp. 114–129.

[21] C. Fellbaum, *WordNet: An Electronic Lexical Database*. MIT Press, 1998.

[22] D. Gildea and D. Jurafsky, "Automatic labeling of semantic roles," *Computational linguistics*, vol. 28, no. 3, pp. 245–288, 2002.

[23] C. Subirats, "Spanish FrameNet: A frame-semantic analysis of the Spanish lexicon," in *Multilingual FrameNets in Computational Lexicography: Methods and Applications*. Mouton de Gruyter, 2009.

[24] P. Kingsbury and M. Palmer, "From TreeBank to PropBank." ser. LREC '02, 2002.

[25] K. Toutanova, D. Klein, C. D. Manning, and Y. Singer, "Feature-rich part-of-speech tagging with a cyclic dependency network," ser. HTL-NAACL '03, 2003.

[26] D. Klein and C. D. Manning, "Accurate unlexicalized parsing," in *ACL'03*, 2003, pp. 423–430.

[27] J. J. Carroll, I. Dickinson, C. Dollin, D. Reynolds, A. Seaborne, and K. Wilkinson, "Jena: Implementing the Semantic Web Recommendations," in *WWW'04*, 2004, pp. 74–83.

[28] S. Tonelli and D. Pighin, "New Features for FrameNet: WordNet Mapping," ser. CoNLL '09, 2009, pp. 219–227.

[29] J. Daudé, L. Padró, and G. Rigau, "Mapping wordnets using structural information." ser. ACL, 2000.

[30] E. Laparra, G. Rigau, and M. Cuadros, "Exploring the integration of WordNet and FrameNet," in *GWC'10*, 2010.

[31] O. Ferrández, M. Ellsworth, R. Munoz, and C. F. Baker, "Aligning FrameNet and WordNet based on Semantic Neighborhoods," ser. LREC '10, 2010.

[32] D. Das, D. Chen, A. F. T. Martins, N. Schneider, and N. A. Smith, "Frame-Semantic Parsing," *Computational Linguistics*, vol. 40, no. 1, pp. 9–56, 2014.

[33] J. Hoffart, M. A. Yosef, I. Bordino, H. Fürstenau, M. Pinkal, M. Spaniol, B. Taneva, S. Thater, and G. Weikum, "Robust Disambiguation of Named Entities in Text," ser. EMNLP '11, 2011, pp. 782–792.

Paper C

Representing Specialized Events with FrameBase

Jacobo Rouces, Gerard De Melo, Katja Hose

The paper has been published in the
Proceedings of the 4th International Workshop on Detection, Representation, and Exploitation of Events in the Semantic Web (DeRiVE), 2015

The layout has been revised.

Abstract

Events of various sorts make up an important subset of the entities relevant not only in knowledge representation but also in natural language processing and numerous other fields and tasks. How to represent these in a homogeneous yet expressive, extensive, and extensible way remains a challenge. In this paper, we propose an approach based on FrameBase, a broad RDFS-based schema consisting of frames and roles. The concept of a frame, which is a very general one, can be considered as subsuming existing definitions of events. This ensures a broad coverage and a uniform representation of various kinds of events, thus bearing the potential to serve as a unified event model. We show how FrameBase can represent events from several different sources and domains. These include events from a specific taxonomy related to organized crime, events captured using schema.org, and events from DBpedia.

1 Introduction

The surge of research on large-scale knowledge bases in recent years has largely been driven by the availability of new sources of information about entities. While structured data about millions of places, people, or companies are very valuable, there have been comparably few new results on capturing events of various sorts. Most existing event-oriented ontologies have introduced only a few abstract classes of events, and typical knowledge bases tend to describe just a small number of specific types of events.

Often, however, there is a need to talk about a broad range of very specific sorts of events. For instance, one might want to distinguish battles from both gunfights and from wars, and capture the class-specific details of such events. We adopt a broad notion of events here. This includes the prototypical cases, e.g. local happenings such as concerts, gatherings, or competitions, and world events such as those reported in the news. It also encompasses the more general abstract definition of events, for instance as "happenings in the real world" [1], which would include, e.g., the birth of a person or a commercial transaction between two people. Clearly, such events make up an important aspect of the world that is relevant in knowledge representation, natural language processing, and numerous other fields and tasks. Occasionally, the term *eventuality* is used to denote a broader notion of events that explicitly includes states, e.g. two people knowing each other.

In this paper, we address this challenge of representing many different notions of events under a common schema, from the very prototypical cases to the very abstract, in a way that has both a broad coverage yet supplies sufficient detail to model event-specific properties. For this, we present a new approach for representing event information that is based on FrameBase [2], a broad RDFS-based schema made of frames and their roles. FrameBase provides a predefined vocabulary with event-specific properties for thousands

of different kinds of events. For instance, FrameBase's schema accounts for the fact that a battle takes place in a certain time and place and normally involves two parties. For this, the schema draws on two linguistic resources, FrameNet [3] and WordNet [4]. As these describe important fragments of the English lexicon, their coverage is quite substantial. Additionally, as we illustrate later on, FrameBase can be easily extended.

In the following, we prove the suitability of FrameBase for representing different kinds of events by creating rules that integrate instances from different domains:

- A taxonomy of event classes relating to organized crime from the EU FP7 project ePOOLICE[1]. In the project, the event classes in the taxonomy are used as types of entities that are extracted from documents crawled from the web, as part of a strategic early-warning system. The taxonomy was originally captured using the Conceptual Graphs formalism [5]. We use and integrate the event taxonomy as it is, without ad-hoc modifications to the schema.
- The subclasses and properties of the "Event" class in schema.org, which "provides a collection of schemas that webmasters can use to markup HTML pages in ways recognized by major search providers, and that can also be used for structured data interoperability" [6].
- The subclasses and properties of the "Event" class in DBpedia [7], which are extracted from the infoboxes in Wikipedia.
- We conclude with a more general overview of how salient aspects of events [1] can be mapped into FrameBase.

This paper is structured as follows. After describing previous approaches and research in Section 2, a brief overview of the FrameBase schema is given in Section 3. Section 2 then shows how we can rely on the FrameBase schema to represent events from several different sources and domains. Finally, Section 3 provides concluding remarks and describes avenues for future research and applications of our work.

2 Related Work

Considering their importance and unique characteristics, events have been included in numerous upper-level ontologies and vocabularies. In [1], existing event models are reviewed, but these define very broad abstract categorizations or meta-models. Only few example specializations or vocabularies for narrow domains exist, and their overall size is relatively small.

[1] `https://www.epoolice.eu/`

For instance, the Simple Event Model (SEM) Ontology [8] introduces the four types *Event*, *Actor*, *Place*, and *Time*. While it provides a mechanism to create more specific ones by extending these, it does not actually define any specific kinds of events itself. Similarly, the LODE (Linking Open Descriptions of Events) model [9] provides very general concepts, such as the four just mentioned. The event model E [10] proposes a generic structure for the definition of events, but a specific vocabulary is provided only for the domain of media events with sensor data. The Event Ontology [11] defines a single event class, for which time, place, agents, factors, products, and meronymic relations can be specified, and the domain of focus is music events. Likewise, the Context Ontology (CONON) is limited to the domain of pervasive computing environments [12].

FrameBase's schema instead aims at a broader coverage of many domains by building on natural language resources. Previous work has made use of natural language processing techniques to extract events from text. For instance, one study [13] relies on semantic role labelling (SRL) in conjunction with VerbNet to collect events from text and convert them to the LODE vocabulary mentioned above. Another system [14] extracts events both from text and from semi-structured data. We believe that such automatic extraction methods would benefit from being able to use a standardized wide-coverage representation schema for their output.

3 The FrameBase Schema

The FrameBase schema [2] consists of classes representing frames, and properties representing frame elements. A *frame* describes any kind of situation, state or action, in which several elements, participants (agents, patients, etc.) or properties are involved. Examples include commercial exchanges, marriages, or the act of stomping. The *frame elements* refer to the participants or properties that are involved in a particular frame instance. Common general frame elements include those of agent, patient, time, and location, but not all frames involve these. Frame elements are sometimes also referred to as semantic roles, roles, or theta roles, especially when they are very general.

The frames and the frame elements in FrameBase are organized in hierarchies of classes (based on subclass relationships) and of properties (based on subproperty relationships), respectively. There are three kinds of frames in FrameBase: LU-microframes, synset-microframes and non-lexical frames. Non-lexical frames are very general and are situated in the upper part of the hierarchy. LU-microframes (lexical unit microframes) descend from non-lexical frames, but are much more specific by being associated with the meaning of particular words (the lexical units). They come from FrameNet [15, 16]. Synset-microframes allow an intermediate level of granularity connecting synonymous LU-microframes, e.g. for *marriage* and *matrimony*. These are based

on WordNet [4], and thus also have allowed us to extend the coverage of FrameBase beyond that of FrameNet. In the field of linguistics, frames are said to be evoked by words: for example, both the verb *to create* and the noun *creation* evoke the `Creation` frame.

FrameBase additionally provides direct binary predicates to directly connect certain values for elements of a given frame. For example, in a creation event, the agent and the place are directly connected via the `establishesInPlace` relation. This enables more concise queries and representations when only two elements are involved in a particular frame. The frame patterns and the direct binary predicates are logically connected by means of definite clauses that can be used with different kinds of inference systems.

For interoperability with existing resources, FrameBase relies on the standard RDF model [17], which has become a common choice for representing knowledge. This is particularly true in the context of the Linked Data [18], a large Web of datasets referring to and reusing each other's elements. The RDF model uses subject-predicate-object triples to represent statements. Each triple can also be seen as an edge in a directed labelled entity-relationship graph. SPARQL [19] is the standard query language for RDF, which is what we use in order to integrate other event representations into FrameBase.

Event frames are specific kinds of frames, subsuming a range of different notions of events, from the very abstract (e.g., "a natural abstraction of happenings in the real world" [1]) to notions with a notably narrower scope, such as that of widely-known events [14]. Frame elements correspond to what are referred to as *aspects* in the event literature [1]. However, frames can also be more general, and include what the event model E categorizes separately as entities [10]. For example, FrameNet, from which FrameBase is derived, includes a frame `People` that is evoked by lexical units (LUs) such as the noun *man*, and with frame elements such as `Age` and `Origin`.

We believe that the advantage of FrameBase over the existing event models lies on the fact that while extensible as the others, it already provides a broad-coverage vocabulary out of the box in order to bootstrap widespread adoption. Besides, its connection to natural language provides potential advantages, like interfacing with text for question answering or text mining.

FrameBase includes, from FrameNet, an Event frame, which inherits from the `Change of state scenario` frame, and includes a relatively rich hierarchy below for events like creation and destruction events (including more specific ones such as births and deaths), and some others. However, not every event must necessarily fall below this event frame, nor does doing so preclude it from being mapped to other frames that represent other conceptualizations for events, or reflect other perspectives of the frame that stress different aspects than the eventive one. Therefore, the representation of events in FrameBase is not confined to the Event frame and its subframes. We will see examples of

this in the next section.

4 Integrating Events

In the first subsections of this section, we present manually built rules for integrating events from three different sources into FrameBase. Later, we add further explanations about these rules and discuss the complexity of the integration rules, and the challenges they present, in particular when they are to be established automatically.

4.1 Representing Events about Organized Crime

The following list of integration rules shows, for each instance of an event class in the organized crime conceptual graph (in bold), the corresponding representation in RDF that it would have in FrameBase. In particular, the main event instance is represented by the anonymous node `_:e`. The default prefix indicates elements that already existed in the core FrameBase schema created from FrameNet and WordNet.

- **Event** `_:e a :frame-Event-event.n`
  - **Act** `_:e a :frame-Intentionally_act-act.n`
    - **Arrest** `_:e a :frame-Arrest-arrest.n`
      - **Drug Possession Arrest** `_:e a :frame-Arrest-arrest.n . _:e :fe-Arrest-Offense _:e2 . _:e2 a :frame-Offenses-possession.n`
      - **Human Trafficking Arrest** `_:e a :frame-Arrest-arrest.n . _:e :fe-Arrest-Offense _:e2 . _:e2 a :frame-Commerce_scenario-trafficker.n . _:e2 :fe-Commerce_scenario-Goods :frame-People-human.n`
      - **Metal Theft Arrest** `_:e a :frame-Arrest-arrest.n . _:e :fe-Arrest-Offense _:e2 . _:e2 a :frame-Theft-theft.n . _:e2 :fe-Theft-Goods :frame-Substance-metal.n . _:e2 a :frame-Offenses-theft.n`
    - **Buy** `_:e a :frame-Commerce_buy-buy.v`
    - **Crime** `_:e a :frame-Committing_crime-crime.n`
      - **Illegal Drug Use** `_:e a :frame-Ingest_substance-use.v`
        - **Consume** `_:e a :frame-Ingestion-consume.v`
        - **Inhale** `_:e a :frame-Ingest_substance-sniff.v`
        - **Inject** `_:e a :frame-Ingest_substance-inject.v`
        - **Possession** `_:e a :frame-Offenses-possession.n`
        - **Smoke** `_:e a :frame-Ingest_substance-smoke.v`
      - **Organised Crime** `_:e a fbe:frame-Organization-criminal%20organization.n`

- **Theft** `_:e a :frame-Theft-theft.n .`
  `_:e a :frame-Offenses-theft.n`
  - **Metal Theft** `_:e a :frame-Theft-theft.n .`
    `_:e :fe-Theft-Goods :frame-Substance-metal.n .`
    `_:e a :frame-Offenses-theft.n`
- **Trafficking** `_:e a :frame-Commerce_scenario-trafficker.n`
  - **Drug Trafficking** `_:e a :frame-Commerce_scenario-trafficker.n .`
    `_:e :fe-Commerce_scenario-Goods :frame-Intoxicants-drug.n`
  - **Human Trafficking** `_:e a :frame-Commerce_scenario-trafficker.n .`
    `_:e :fe-Commerce_scenario-Goods :frame-People-human.n`
- **Seizure** `_:e a :frame-Taking-seizure.n`
  - **Drug Seizure** `_:e a :frame-Taking-seizure.n .`
    `_:e :fe-Taking-Theme :frame-Intoxicants-drug.n`
- **Sell** `_:e a :frame-Commerce_sell-sell.v`
- **Transaction** `_:e a :frame-Commercial_transaction-transaction.n`
  - **Crime Transaction** `_:e a :frame-Commercial_transaction-transaction.n .`
    `_:e a :frame-Committing_crime-crime.n`
    - **Drug Trafficking Transaction**

```
_:e a :frame-Commercial_transaction-transaction.n .
_:e a :frame-Committing_crime-crime.n .
_:e :fe-Commercial_transaction-Goods :frame-Intoxicants-drug.n
```

    - **Human Trafficking Transaction**

```
_:e a :frame-Commercial_transaction-transaction.n .
_:e a :frame-Committing_crime-crime.n .
_:e :fe-Commercial_transaction-Goods :frame-People-human.n
```

    - **Metal Theft Transaction**

```
_:e a :frame-Commercial_transaction-transaction.n .
_:e a :frame-Committing_crime-crime.n .
_:e :fe-Commercial_transaction-Goods :frame-Substance-metal.n
```

The hierarchy in the original ontology is not necessarily consistent with the hierarchy in FrameBase. Only in certain cases does a superclass relationship between two elements of the source also exist between the two elements' respective translations to FrameBase. Therefore, for each translation of an original class of event, the translations of the parents in the original ontology can be added to the set of instances (ABox) in FrameBase, and this will provide additional knowledge that would not always be inferred by the FrameBase schema alone.

We minimize the need for declaring new frames and frame elements for specialized domains by making use of the compositionality of most specialized terms, creating complex structures that combine the semantics of simpler, basic elements. For instance, the translation for the event of type "Drug Possession Arrest" declares an event of type arrest, and specifies that it is about drug possession by assigning drug possession (`Offenses-possession.n`) as the offence.

Owing to this flexibility, we merely needed to mint one single new entity that had not existed in the core FrameBase schema (the microframe `Organization-criminal%20organization.n`, with the prefix `fbe:` denoting that this is an extension). This exemplifies the potential of FrameBase to represent events from relatively specialized domains, but at the same time the capacity to be extended to fill any possible gaps.

For representing timelines, the frame `Individual_history-history.n` can be used. Each timeline can be represented with one instance of that frame. This instance can be linked with the frame element `Individual_history--Domain` to the topic, which is preferably an entity (or alternatively, a literal or an anonymous node or dummy entity named with a literal). The instance can also be linked with the frame element `Individual_history-Event` to each of the elements in the timeline. Additional frame elements are available in FrameBase, originating from FrameNet, for expressing participants, total duration, etc.

Then, complex queries such as retrieving all events in a given timeline between two given dates, can be built in SPARQL. Similarly, sub-events can be represented with the property path: `^:fe-Part_whole-Part/:fe--Part_whole-Whole`.

4.2 Representing Events from DBpedia.org

We now turn to the `Event` class in DBpedia, and its subclasses, showing how these can be integrated into FrameBase. The integration is implemented using SPARQL CONSTRUCT rules because DBpedia is already in RDF. We only add a couple of subclasses, but most of the properties belong to the parent `Event` class itself.

Top event

```
CONSTRUCT {
  ?e a :frame-Event-event.n .
  ?e :fe-Event-Time _:timePeriod .
    _:timePeriod a fbe:frame-Timespan-period.n ;
      fbe:fe-Timespan-Start ?o1 ; fbe:fe-Timespan-End ?o2 .
  _:e2 a :frame-Relative_time-preceding.a ;
    :fe-Relative_time-Landmark_occasion ?e ;
    :fe-Relative_time-Focal_occasion ?o3 .
  _:e3 a :frame-Relative_time-following.a ;
    :fe-Relative_time-Landmark_occasion ?o3 ;
    :fe-Relative_time-Focal_occasion ?e .
  _:e4 a :frame-Relative_time-following.a ;
    :fe-Relative_time-Landmark_occasion ?e ;
    :fe-Relative_time-Focal_occasion ?o4 .
  _:e5 a :frame-Relative_time-preceding.a ;
    :fe-Relative_time-Landmark_occasion ?o4 ;
    :fe-Relative_time-Focal_occasion ?e .
  ?e :fe-Event-Reason ?o5 .
  ?e a :frame-Social_event-meeting.n ;
```

```
    :fe-Social_event-Attendee ?o8 .
} WHERE {
  ?e a dbpedia-owl:Event .
  OPTIONAL{?e dbpedia-owl:startDate ?o1}
  OPTIONAL{?e dbpedia-owl:endDate ?o2}
  OPTIONAL{?e dbpedia-owl:previousEvent ?o3}
  OPTIONAL{?e dbpedia-owl:followingEvent|dbpedia-owl:nextEvent ?o4}
  OPTIONAL{?e dbpedia-owl:causedBy ?o5}
  OPTIONAL{?e dbpedia-owl:duration ?o6}
  OPTIONAL{?e dbpedia-owl:numberOfPeopleAttending ?o7} #Omitted
  OPTIONAL{?e dbpedia-owl:participant ?o8}
}
```

For sub-classes of dbpedia-owl:Event

```
CONSTRUCT {
  ?e a :frame-Social_event-meeting.n .
} WHERE {?e a dbpedia-owl:SocietalEvent}
```

For sub-classes of dbpedia-owl:SocietalEvent

```
CONSTRUCT {
  ?e a :frame-Project-project.n .
  ?e :fe-Project-Activity dbpedia:Space_exploration .
} WHERE {?e a dbpedia-owl:SpaceMission}
```

For sub-classes of dbpedia-owl:SocietalEvent

```
CONSTRUCT {
  ?e a fbe:frame-Social_event-convention.n .
} WHERE {?e a dbpedia-owl:Convention}
```

Out of the 9 properties of the class `Event`, the only omitted one was `numberOfPeopleAttending`, because the class `Event` is too general for it, as it has subclasses such as `NaturalEvent` (`SolarEclipse`) and `PersonalEvent` (`Birth`, etc.). The `SocietalEvent` class appears more appropriate for this.

4.3 Representing Events from schema.org

Finally, we present the translation of the Event class in schema.org. Again, SPARQL CONSTRUCT rules are used because schema.org can be expressed using RDFa, and SPARQL offers a standard way of representing knowledge graph transformations. Due to space restrictions, we omit the subclasses here, but these have very few genuine properties, and therefore the specialization is relatively simple. Besides, the taxonomy of schema.org events has some inconsistency issues that makes its use complex: the Event class is defined as capturing events such as concerts, lectures, and festivals, with properties such as "typical age range", but there are sub-events such as UserInteraction and UserPlusOnes that actually represent a more general kind of events.

```
CONSTRUCT {
  ?e a :frame-Social_event-meeting.n .
```

```
  ?e :fe-Social_event-Time _:timePeriod .
    _:timePeriod a fbe:frame-Timespan-period.n ;
      fbe:fe-Timespan-Start ?Osta ; fbe:fe-Timespan-End ?Oend .
  ?e :fe-Social_event-Duration ?Odur . ?e :fe-Social_event-Place ?Oloc .
  ?e :fe-Social_event-Attendee ?Oatt . ?e :fe-Social_event-Host ?Oorg .
  ?e :fe-Social_event-Occasion ?Osup . ?Osub :fe-Social_event-Occasion ?e .
  ?Ooff a :frame-Offering-offer.v ;
    :fe-Offering-Theme ?e .
  ?e a :frame-Performing_arts-performance.n ;
    :fe-Performing_arts-Performer ?Oper ;
    :fe-Performing_arts-Performance ?Owor .
  _: a :frame-Recording-record.v ;
    :fe-Recording-Phenomenon ?e ;
    :fe-Recording-Medium ?Orec .
} WHERE {
  ?e a sch:Event .
  # Unambiguous translation
  OPTIONAL{?e sch:startDate ?Osta}     OPTIONAL{?e sch:endDate ?Oend}
  OPTIONAL{?e sch:duration ?Odur}      OPTIONAL{?e sch:location ?Oloc}
  OPTIONAL{?e sch:attendee ?Oatt}      OPTIONAL{?e sch:organizer ?Oorg}
  OPTIONAL{?e sch:superEvent ?Osup}    OPTIONAL{?e sch:subEvent ?Osub}
  OPTIONAL{?e sch:offers ?Ooff}        OPTIONAL{?e sch:performer ?Oper}
  OPTIONAL{?e sch:workPerformed ?Owor} OPTIONAL{?e sch:recordedIn ?Orec}
  # Ambiguous translation
  OPTIONAL{?e sch:doorTime ?Odoo}
  # No translation
  OPTIONAL{?e sch:eventStatus ?Oeve}
  OPTIONAL{?e sch:typicalAgeRange ?Otyp}
  OPTIONAL{?e sch:previousStartDate ?Opre}

}
```

The only extension of the FrameBase schema used here was the frame `:frame-Timespan-period.n` with the start and end frame elements, used to denote periods of time. This, however, is not an ad-hoc extension motivated by a particular need of only one source, but a very general one. Out of the 16 properties of the `Event` class, 12 were translated without loss of meaning. One was translated with partial loss of meaning (`doorTime`, translated as a generic start time) and 3 of them were not translated. Whether these can be integrated too, by means of more complex structures, is something we are investigating.

4.4 Mapping Event Aspects to Frame Elements

The survey by Scherp and Mezaris [1] proposes a classification of salient aspects of events. We use this classification to show in a more general way how event aspects can relate to frame elements in the FrameNet-based schema of FrameBase.

- **Time** and **Space**: When applicable, frames include frame elements `Time` and `Place`.

- **Participation**: The classification defines this as "participation of objects in event, where objects can be any living as well as non-living things and include people, buildings, and other even intangible objects like the roles a person plays in a specific situation" [1]. FrameBase provides a large inventory of more specific roles to capture such participants. Often, these correspond to what are sometimes called the proto-agent and proto-patient roles, whose realization in FrameBase depends on the frame. Some examples are `:fe-Commerce_buy-Buyer`, `:fe-Destroying-Destroyer` and `:fe-Destroying-Undergoer`, which are subproperties of `:fe-Getting-Recipient`, `:fe-Transitive_action-Agent` and `:fe-Transitive_action-Patient`, respectively.

- Relations between events.

  - **Mereology**: The relation between two events, when one is part of another. Some frames will have a frame element that will fill this role, like `:fe-Social_event-Occasion` in the example of the Event class in schema.org. In other cases, an additional frame instance of type `:frame-Part_whole` can be used.
  - **Causality**: One event is the cause of another. Some frames will have a frame element that will fill this role, like `:fe-Event-Reason` in the example for the Event class in DBpedia. In other cases, an additional frame instance of type `:frame-Causation` can be used.
  - **Correlation**: When "two (or more) events have a common cause, but this common cause cannot be explained". If we can assume there is a common cause as in the definition, then the causal relationships can be represented with two instances of `:frame-Causation` connecting with an anonymous node for the unknown cause.

- **Documentation**: Events can be "documented using some media like photos or videos captured during the event". This relation is between an event and such documentation. It can be expressed connecting the events by an additional frame of type `:frame-Recording-document.v`, `:frame-Recording-record.v`, and `:frame-Recording-register.v`, or some extension if needed.

- **Interpretation**: This aspect aims at capturing "subjectivity that may exist on the other aspects of events". This is a very broad category that may include different phenomena. The perspectivization relation in FrameNet [16] connects frames representing objective events with frames describing them from a particular perspective. For instance, `:frame-Commerce_Sell` and `:frame-Commerce_Buy` are perspectivizations of `:frame-Commerce_Scenario`. In other cases, an additional frame instance of a pertinent type can be used, for instance `:frame-Becoming_aware`.

4.5 Complex Transformations

Most of the integration rules we have described follow a pattern which involves an event *class* in the source being translated as a frame class, and each of their outgoing properties being mapped to individual frame elements. However, there are multiple ways in which the rules can differ from this basic pattern.

1. Sometimes, a class integration rule may need to instantiate multiple frames rather than just a single one. We distinguish two main types of this phenomenon.

    a) The instantiated frame instances may be connected by frame elements. Examples of this include the frame `:frame-Timespan-period.n` created to represent time periods, and the subframes of `Relative_time` to express precedence between events (all in the example for `dbpedia-owl:Event`). The same applies when a frame element is used to specify a frame beyond the lexical unit (see the rule for `dbpedia:Space_exploration`).

    b) Several frames can also be evoked separately, without the instances being directly connected by any frame element. When these frames describe different perspectives of the same event, there is the possibility that FrameNet links them by means of *perspectivization*, and therefore FrameBase can infer one from another. For example, classes `:frame-Commerce_buy-buy.v` and `:frame-Commerce_sell-sell.v`, which are used for classes `Buy` and Sell in the organized crime taxonomy, are both perspectivizations of `:frame-Commerce_goods-transfer`. In this case, inference is possible because RDFS subclass and subproperty properties are used in FrameBase to reflect the perspectivization relation between frame classes and frame elements respectively. Another example are `:frame-Receive_visitor_scenario` and `:frame-Visit_host`, which are perspectives of `:frame-Visitor_and_host`. However, in other cases one cannot rely on existing inference. For instance, see how the rule to translate `Event` from schema.org, besides frames `Event-event.n` and `Timespan-period.n`, also instantiates `Performing_arts-performance.n`, `Recording-record.v` and `Offering-offer.v` when certain properties are present.

2. Another possible source of complexity is that frame elements can be inverted. In this case, the integration rules need to invert the order of the arguments, like in the second appearance of `:fe-Social_event-Occasion` in the integration rule for the class `Event` in schema.org.

3. Oftentimes, a *property* (rather than a class) in the source can be translated as evoking a frame on its own. In this case, the two involved entities become

connected to the new frame by means of frame elements. This would be the case for a property like `fightAgainst`, which might evoke an event or frame of type armed conflict, about which additional information could be added. None of the examples we have covered above are of this kind, because we use sources that explicitly represent, or *reify*, events. In other sources, however, this phenomenon appears quite frequently.

Arbitrary combinations of these phenomena are possible (e.g. the rule integrating the Event class from schema.org). Overall, this makes automatic generation of the integration rules a very hard task, because it generates so many free variables that any attempt to train a system would face extreme sparsity. In some cases, it may thus make sense to sacrifice some recall, developing a system that only covers simpler transformations.

4.6 Representational Flexibility

Finally, another potential challenge for data integration is that even when a homogeneous schema such as FrameBase is used, certain kinds of knowledge can still be expressed in multiple possible ways.

- One example is that there are several ways of narrowing down the meaning of a frame instance. One is creating a new sub-microframe associated with a new lexical unit. Another one is assigning a value to a frame element (see example for `SpaceMission`), as mentioned above. This may lead to divergent choices of representation even within the core part of the schema that comes from FrameNet.
- Another example of this is when a frame element needs to be reified, i.e. represented as a frame instance, to express something additional about it (as would be the case of the property `previousStartDate` in schema.org), or when there is no direct frame element available and creating it would lead to a combinatorial explosion in the size of the schema. An example of the latter is the difference between our proposal for using the frame `Part_whole` for expressing sub-event relations, and how we used the frame element `Occasion` for the frame `Social_event`, but this is a particularity of that frame. Again, this may lead to an incoherent representations in the knowledge base. One potential way of addressing this would be extending the reification–dereification mechanism of FrameBase [2].

5 Conclusion

We have shown how events from specialized domains can be represented with the FrameBase schema under a unified model, integrating events in the prototypical sense with more general kinds of events in the sense of abstract happenings or situations. This model has proven to have a high degree of

coverage because it needed just few extensions to accommodate the integrated knowledge, and we have illustrated how these extensions can be performed when needed. We have also discussed the various challenges and problems one faces when the integration rules from disparate structured sources of event information are to be built automatically.

Extremely specialized domains, such as quantum physics, may produce lower coverage and need more extensions, although in some cases the creators of FrameNet have also been involved in projects that led to the inclusion of specific scientific and technical domains.

The integration rules that we produce can be used in the future as gold standards for training and testing automatic methods for creating rules from other schemas. We are currently performing research on these methods to integrate further sources such as YAGO2s, Freebase, and Wikidata.

Please refer to `http://framebase.org` for information on using FrameBase and the integration rules.

References

[1] A. Scherp and V. Mezaris, "Survey on modeling and indexing events in multimedia," *Multimedia Tools and Applications*, vol. 70, no. 1, pp. 7–23, 2014.

[2] J. Rouces, G. de Melo, and K. Hose, "FrameBase: Representing N-ary Relations Using Semantic Frames," in *ESWC'15*, 2015.

[3] C. F. Baker, C. J. Fillmore, and J. B. Lowe, "The Berkeley FrameNet Project," in *Proceedings of the 17th international conference on Computational linguistics – Volume 1*, ser. ICCL '98, 1998, pp. 86–90.

[4] C. Fellbaum, *WordNet: An Electronic Lexical Database*. MIT Press, 1998.

[5] J. F. Sowa, "Conceptual graphs," in *In Handbook of Knowledge Representation*. Elsevier, 2008, pp. 213–237.

[6] "Schema.org," http://schema.org.

[7] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann, "DBpedia-A crystallization point for the Web of Data," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 7, no. 3, pp. 154–165, 2009.

[8] W. R. Van Hage, V. Malaisé, R. Segers, L. Hollink, and G. Schreiber, "Design and use of the Simple Event Model (SEM)," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 9, no. 2, pp. 128–136, 2011.

[9] R. Shaw, R. Troncy, and L. Hardman, "LODE: Linking Open Descriptions of Events," in *ASWC '09*, ser. Lecture Notes in Computer Science, 2009, pp. 153–167.

[10] A. Scherp, S. Agaram, and R. Jain, "Event-centric media management," in *Electronic Imaging 2008*. International Society for Optics and Photonics, 2008, pp. 68 200C–68 200C.

[11] Y. Raimond and S. Abdallah, "The Event Ontology," Tech. Rep., Oct. 2007, http://motools.sf.net/event.

[12] X. H. Wang, D. Q. Zhang, T. Gu, and H. K. Pung, "Ontology based context modeling and reasoning using owl," in *Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second IEEE Annual Conference on*. Ieee, 2004, pp. 18–22.

[13] P. Exner and P. Nugues, "Using semantic role labeling to extract events from Wikipedia," in *DeRiVE'11*, 2011.

[14] E. Kuzey and G. Weikum, "Extraction of temporal facts and events from wikipedia," in *Proceedings of the 2nd Temporal Web Analytics Workshop*. ACM, 2012, pp. 25–32.

[15] C. J. Fillmore, C. R. Johnson, and M. R. Petruck, "Background to Framenet," *International Journal of Lexicography*, vol. 16, no. 3, pp. 235–250, 2003.

[16] J. Ruppenhofer, M. Ellsworth, M. R. Petruck, C. R. Johnson, and J. Scheffczyk, *FrameNet II: Extended Theory and Practice*. ICSI, 2006.

[17] P. Hayes and P. Patel-Schneider, "RDF 1.1 semantics," W3C, Tech. Rep., 2014, http://www.w3.org/TR/rdf11-mt/.

[18] C. Bizer, T. Heath, and T. Berners-Lee, "Linked data–the story so far," *IJSWIS*, vol. 5, no. 3, pp. 1–22, 2009.

[19] S. Harris and A. Seaborne, "SPARQL 1.1 Query Language," W3C Consortium, W3C Recommendation, Mar. 2013.

Paper D

Integrating Heterogeneous Knowledge with FrameBase

Jacobo Rouces, Gerard De Melo, Katja Hose

The paper has been submitted to the
Semantic Web Journal in November 2015 (Tracking number: 1239-2451)

The layout has been revised.

Abstract

Large-scale knowledge graphs such as those in the Linked Data cloud are typically represented as subject-predicate-object triples. However, many facts about the world involve more than two entities. While n-ary relations can be converted to triples in a number of ways, unfortunately, the structurally different choices made in different knowledge sources significantly impede our ability to connect them. They also increase semantic heterogeneity, making it impossible to query the data concisely and without prior knowledge of each individual source. This article presents FrameBase, a wide-coverage knowledge-base schema that uses linguistic frames to represent and query n-ary relations from other knowledge bases, providing also different levels of granularity connected by logical entailment. This altogether provides for flexible and expressive seamless semantic integration from heterogeneous sources. It also opens possibilities to draw on natural language processing techniques for querying and data mining.

1 Introduction

Over the past few years, large-scale knowledge bases (KBs) have grown to play an important role on the Web. Many institutions rely on Linked Data principles to publish their data using Semantic Web standards [1]. These KBs are mostly based on simple subject-predicate-object (SPO) triples, as defined by the RDF model [2]. Such triples are convenient to process and can be visualized as entity networks with labeled edges.

Commercial search engines exploit them to provide direct answers to user queries, and IBM's Watson question answering system [3], which defeated human champions of the Jeopardy! quiz show, used them to find and to rule out answer candidates.

Whereas triple representations work straightforwardly for relations involving two entities, many interesting facts relate more than just two participants – a problem that has gained renewed attention in several recent papers [4, 5] as well as in the current W3C proposal to add roles to schema.org [6]. For a birth event, for instance, one may wish to capture not just the time but also the location and parents. For an actress starring in a movie, the name of the portrayed character may be relevant. Such facts naturally correspond to n-ary relations. In order to capture them as triples, several different representation schemes have been proposed. Table D.1 shows some possibilities of expressing that an entity John was married in 1964, some of which also include additional information such as the name of the bride. These representations will be discussed in more detail later in section 2.

As the example shows, this sort of semantic heterogeneity leads to significant data integration challenges. One KB might use a simple binary property between two entities, whereas another may instead choose a more complex representation that accommodates additional arguments. The representations

can easily be so at odds with each other that no particular mapping between entities could bridge the differences. There are entities at each side that have no counterpart at the other. This leads to several challenging problems:

1. When **linking data**, there are currently no mechanisms to connect KBs with different modeling choices. Predicates exist to link equivalent classes, instances, or properties, but not for connecting the different patterns, as explained above. Existing work on ontology and KB alignment [7] is limited to finding aliases.
2. When **querying**, the query must be built in a way that fits the particular modeling choices made for the respective KB. Otherwise, the recall may be as low as zero [8]. Even worse, for the case of a set of different KBs instead of a single coherent KB, there is no simple query (as could be formulated on a single given schema) that will have a high recall across all KBs.
3. When **natural language interfaces** to KBs are queried, state-of-the-art systems typically attempt to map verbs and predicate phrases to RDF predicates [9]. This approach, however, cannot be applied when the KB fails to provide a compatible binary relation.

FrameBase. These problems are addressed by FrameBase [10, 11], a broad-coverage schema that can homogeneously integrate other KBs and has strong connections to natural language. It overcomes the above-mentioned forms of heterogeneity – by sticking to a specific modeling choice general enough to subsume the others (neo-Davidsonian representation) – together with a large vocabulary for events and roles. This vocabulary is reusable and based on an extensible hierarchy. FrameBase also provides a mechanism to convert back and forth between the new representation and direct binary relations, using a vocabulary of binary relations automatically generated from linguistic annotations. These are more concise and can be used when only two arguments are relevant.

This paper is structured as follows. Section 2 reviews related work and conducts a thorough analysis of existing approaches for modeling n-ary relations and their space efficiency. Then, an overview of FrameBase is given in section 3. Section 4 explains how the FrameBase schema is constructed, including rules to convert between different levels of reification. Section 2 presents methods to integrate knowledge from external KBs into FrameBase. Section 3 provides a qualitative evaluation, and section 3 concludes the paper with an outlook to future work.

2 State of the Art

Different approaches for modeling n-ary relations exist, which are summarized in Table D.1. Table D.2, provides a novel comparison of their space efficiency,

Table D.1: Triple Representations of n-ary Relations

Direct Binary Relation

```
John        wasMarriedOnDate    1964 .
```

RDF Reification

```
John        marries             Mary .
s           type                Statement .
s           subject             John .
s           property            marries .
s           object              Mary .
s           time                1964 .
```

Subproperties

```
p           subPropertyOf       Marriage .
John        p                   Mary .
p           time                1964 .
```

Neo-Davidsonian (Specific Roles)

```
e           type                Marriage .
e           groom               John .
e           bride               Mary .
e           time                1964 .
```

Neo-Davidsonian (General Roles)

```
e           type                Marriage .
e           agent               John .
e           agent               Mary .
e           time                1964 .
```

which has consequences with regards to their applicability for large-scale KBs. Each approach will be discussed in detail in the following subsections.

2.1 Direct Binary Relations

A common way to represent n-ary facts is to simply decompose them directly into binary relations between two participants [12]. But in doing so, important information may be lost. For instance, given a triple with property `wasMarriedOnDate` and two triples with `gotMarriedTo`, we cannot be sure to which marriage the given time span applies.

2.2 RDF Reification

The RDF standard proposes RDF reification [2], which introduces a new identifier (IRI) for a statement and then describes the original RDF statement

	All triples	Core	Linking event	Reif. Inf.	Dereif. Inf.
RDF Reification	$(n+4)k$	$(n+3)k$	$k(k-1)$	$4k$ dc	k dc
Subproperties	$(n+2)k$	$(n+1)k$	$k(k-1)$	$2k$ dc	1 gdc / RDFS
Schema.org Roles	$(n+3)k$	$(n+2)k$	$k(k-1)$	$3k$ dc	k dc
Neo-Davidsonian	$1+n+k$	$1+n$	0	$3k$ dc	k dc

Table D.2: Triple Overhead. *n is the number of participants in an event, and $k <= \frac{n(n-1)}{2}$ the number of pairs that are relevant to be linked by direct binary relations. "All triples" indicates the total number of triples that can be materialized. "Core" excludes the k direct binary relations, which can always be retrieved with some sort of inference. "Linking event" indicates the number of triples needed to connect entities that represent the same event (aliases), which is something that is not required with Neo-Davidsonian representation, because it can use a single one. "Reification Reasoning" indicates the inference system required to obtain the representation in "All triples" or "Core" from the k direct binary relations. "Dereification Reasoning" indicates the inference system required to obtain the k direct binary relations or the representation in "All triples" from the representation in "Core". "x dc" means that x definite clauses are required for each event; "1 gdc / RDFS" means that one global definite clause would be enough (providing for subproperty closure, which is part of RDFS inference). Definite clauses are a kind of rules which can be expressed as a disjunction of logical atoms with only one negated, which is the consequent when it is written as an implication (rule). In this context, the atoms are of the form triple(subject,predicate,object). In section 5, we will describe more in detail these rules for the case of FrameBase.*

using three new triples with `subject`, `predicate`, and `object` properties. Subsequently, arbitrary properties of the statement can be captured by adding further triples about it.

In the different versions of YAGO [13–15], RDF-reification[1] is used to attach additional information to the event represented by the original RDF triple (evoked by its property) – as in the *RDF-Reification* example in Table D.1. This has the advantage that both the original triple as well as the RDF-reified triple can be present in the KB and queries that do not require the additional information can still use the original binary relation directly. However, this also has several drawbacks:

- Formally, the event represented by a triple and the triple as a statement are different entities with different properties. For instance, an institution may endorse the triple as a statement without endorsing the marriage. Using RDF-reification, both are represented by the same RDF resource identifier, which conceptually is meant to be unambiguous. This is a potential source of confusion and inconsistency.
- The number of triples increases by a factor of 4. For each triple `S P O`, one has to add `T a rdf:Statement`, `T rdf:subject S`, `T rdf:predicate P`, and `T rdf:object O`. These do not add any new information themselves

[1] We will use the term *RDF-reification* because the term reification has other meanings, one of which will be heavily used later in the paper.

but are merely a prerequisite for then being able to extend the original binary relation to an n-ary relation by subsequently adding more triples with `T` as subject.
- The advantage of being able to include the original non-RDF-reified triple only applies for the primary binary relation, and not for the other $\frac{n(n-1)}{2} - 1$ ones that can be formed (not counting inverses). Some of these may be rare or irrelevant, but others may be important and are indeed used in YAGO (e.g. `bornAtPlace`, `bornOnDate`).
- The choice of the primary pair of entities and their binary relation (John and Mary in Table D.1) is arbitrary, and a third party willing to query the KB cannot replicate the choice independently. If their choice is different, they will not obtain any results. A possible solution, which is actually implemented in YAGO, is to include the triples for the other pairs and reify them, too, but this adds yet another factor of overhead, besides data redundancy that would complicate updates.
- When two or more different events share the same values for the primary pair of arguments, they will share the same triple, but require separate RDF-reifications, producing non-unique triple identifiers. For example, if there are two flight connections between Paris and London with different airlines, the triple `Paris isConnectedTo London` will be RDF-reified twice, with two different triple identifiers.

If the triplestore implementation makes use of quads[2], the 4-fold overhead can be avoided (though the underlying storage needs a new column), but the other disadvantages still remain. Quad-based singleton named graphs [2] could be used instead of RDF-reification, the problems being the same.

2.3 Subproperties

A recent proposal [5] aims to solve some of the issues with RDF-reification by instead declaring a subproperty of the original property in the primary pair, and using this subproperty as the subject for the other arguments of the n-ary relation. This is shown in the *Subproperties* example in Table D.1.

While the approach enables us to use RDFS reasoning to obtain the triple with the parent property that relates two of the participants, and also reduces the overhead of RDF-reification, it still suffers from the problems mentioned above related to the existence of a primary pair. For one, the non-RDF-reified binary relationships for the other pairs cannot be inferred from that subproperty.

2.4 Schema.org's "Roles"

Schema.org is an effort sponsored by Google, Yahoo, and Microsoft to establish common standards for semantic markup in Web pages. It offers a method to

[2] `http://www.w3.org/TR/n-quads/`

qualify additional information to a binary predicate [6], which in practice is equivalent to representing the n-ary relation arising from adding arguments to the binary relation underlying the binary predicate. It works by substituting the object of the binary predicate with a fresh instance of a class *Role*[3] (or a more specific sub-class with its own properties), and appending to this role instance the old object by means of the same binary predicate, alongside other properties such as time, instrument, etc.

For example `:SanFrancisco49ers schema:athlete :JoeMontana` would be converted to:

```
:SanFrancisco49ers schemaorg:athlete _:x
_:x a schemaorg:Role .
_:x schemaorg:athlete :JoeMontana .
_:x schemaorg:startDate "1979" .
```

This transformation offers certain level of compatibility between the simple pattern with the direct binary predicate and the complex pattern, because the binary predicate is preserved in the complex pattern, with the same subject. However, the object changes, and therefore the simple pattern as such is not truly preserved after the transformation. Besides, the definition or "contract" of the direct binary predicate is broken in the complex pattern. For example, `schemaorg:athlete` has SportsTeam and Person as domain and range respectively, and the semantics is that the object is a person that plays in the team denoted by the subject. However, none of the two usages in the complex pattern follow this: one has SportsTeam and Role as domain and range, and the other has Role and Person.

An example of how this conflation can lead to problems can be fully appreciated with non-transitive predicates. In the case the predicate was `somekb:fatherOf`, someone's children would become his grandchildren after the transformation.

Furthermore, the complex pattern produced by this method, given a direct binary predicate between two entities and a further qualifying value (like time in the example), is not equivalent to the one produced by another binary predicate between one of these entities and the qualifying value. This produces a similar effect of redundancy than in the method using RDF-reification.

2.5 Neo-Davidsonian Representations

Another approach, and the one that FrameBase will adapt, is to make use of so-called neo-Davidsonian representations [18, p. 600f.]. This means that we first define an entity that represents the event or situation (also referred to as

[3]Schema.org's use of the term "role" differs from its standard use in linguistics, which are qualifying properties such as agent and patient [16]. This definition has also been adopted in ontologies, for instance `CaseRole` in the SUMO ontology [17].

a *frame*) underlying the n-ary relation. Then, this entity is connected to each of the n arguments by means of a property describing the *semantic role* [4, 19].

The process of converting from the binary representation to the neo-Davidsonian one is called reification, but this is different from *RDF-reification* as discussed earlier. In RDF-reification, an entity is defined that stands for a whole triple so that additional triples can be used to describe the reified triple as a unit that represents a statement. However, in the context of event semantics, reification is used to denote the process by which an entity is defined that refers to the event, process, situation, or more generally, frame, evoked by a property or binary relation. Having done this, additional information about it can then easily be added. Both kinds have in common that a new entity is defined to refer to something that before was not explicitly represented by an entity in the KB, but in one case it is an RDF statement while in the other it is an event.

Advantages. Table D.2 compares the neo-Davidsonian approach to the alternatives. These require a lot more triples when several direct binary relations need to be included. In the worst case, $k = \frac{n(n-1)}{2}$ despite discounting reciprocal relations, but even if not all of these relations are relevant, connecting all agents and possibly patients to all other elements would be relevant, which would easily satisfy $k > n$.

Semantic Heterogeneity. Unfortunately, there are different ways of using the neo-Davidsonian approach, with different levels of granularity for the events and the semantic roles, from a very small set of abstract generic ones [20] to more specific ones [21].

The Simple Event Model (SEM) Ontology [22] falls within the category of neo-Davidsonian representation with general roles (see Table D.1). It defines four very general entities, *Event*, *Actor*, *Place*, and *Time*. It also establishes a framework for creating more specific ones by extending these, but it does not provide these extensions, nor ways to integrate existing KBs in a way that would solve the problem of semantic heterogeneity. Similarly, LODE (Linking Open Descriptions of Events) [20] specifies only very general concepts such as the four just mentioned.

Freebase [21] is a KB built both from tapping on existing structured sources and via collaborative editing. Although it uses its own formalisms, there are official and third-party translations to RDF. Freebase makes use of so-called *mediators* (also called *compound value types*, CVTs) as a way to merge multiple values into a single value, similar to a `struct` datatype in C. There are around 1,870 composite value types in Freebase (1,036 with more than one instance) and around 14 million composite value instances. While CVTs do not represent frames or events per se, from a structural perspective, they can be regarded as isomorphic to a neo-Davidsonian representation with specific roles (see Table D.1). However, Freebase places a number of restrictions on CVTs. For

instance, they cannot be nested, and there is no hierarchy or network of them that would for example relate a purchasing event to a getting event.

FrameNet [23, 24] is a well-known resource in natural language processing (NLP) that defines over 1,000 *frames* with participants (so-called *frame elements*). For example, the verb *to buy* and the noun *acquisition* are assumed to evoke a commercial transaction frame, with frame elements for the seller, the buyer, the goods, and so on.

Previous work has proposed general patterns for using FrameNet in knowledge representation [25] and converted FrameNet to RDF [26], proposing a way to generate schemas from FrameNet. Similarly, the FRED system [27] for building semantic representations from natural language can be configured to use FrameNet.

3 System Overview

As seen in the previous section, there are a number of different representations used in KBs. FrameBase will use the linguistic resources FrameNet [23] and WordNet [28] to fully develop an extensive schema for large-scale knowledge representation and integration. The schema is composed of an expressive neo-Davidsonian level that draws on a large common inventory of frames, together with a more concise level of direct binary relations, which is connected to the former by means of inference rules.

3.1 FrameNet-based Representation

The use of FrameNet is motivated by the following considerations.

- FrameNet has a long history and aims at descriptions of arbitrary natural language. It thus provides a relatively large and growing inventory of frames and roles, with a broad coverage of numerous different domains.
- FrameNet comes with a large collection of English sentences annotated with frame and frame element labels. This data led to the task of automatic *semantic role labeling* (SRL) [29] of text, now one of the standard tasks in NLP. This strong connection to natural language facilitates question answering and related tasks.
- While FrameNet's lexicon and annotations cover the English language, its frame inventory is abstract enough to be adopted for languages as different as Spanish and Japanese [30]. This also makes it much more suitable as a basis for knowledge representation than language-specific syntax-oriented SRL resources such as PropBank [31].
- FrameNet provides a reasonable level of granularity for the phenomena that humans care to describe. From a theoretical perspective, there is no universally appropriate single level of reification. Any frame element might be reified on its own, and any two elements of a frame could be connected

directly by a predicate. Using FrameNet strikes a well-motivated balance, at a point that is granular enough to constitute a model for natural language semantics. As section 5 will explain, a second level of representation will be provided as well, which will be based on the direct binary predicates between frame elements, and therefore less expressive but more concise.

3.2 Overview

For creating the FrameBase schema using FrameNet, the following steps have been taken, which will be further explained in section 4.

a) **FrameNet–WordNet Mapping.** First, a high-precision mapping is created between FrameNet and another well-known lexical resource called WordNet [28], which will be used to enrich the lexical coverage and relations of the FrameBase schema.
b) **Schema Induction.** FrameNet, WordNet, and the mapping are used to create an RDFS schema for FrameBase that has very wide coverage and is extensible. The schema exploits semantic relations from these components (e.g., synonymy, hyponymy, and perspectivization) to transform the original resources into FrameBase's lightweight RDFS model.
c) **Automatic Reification–Dereification Mechanism.** Reification–dereification rules are created, in the form of definite clauses that allow the KB to be queried independently either using reified frames or dereified direct binary predicates, and that may also be used to reduce overhead in the KB.

4 FrameBase Schema Creation

Before external KBs can be integrated, the FrameBase schema must be created. This process involves creating an initial mapping between FrameNet and WordNet (section 4.1), the use of these resources and mappings to create the FrameBase core schema (Section 4.2). It also involves the creation of reification–dereification rules to enable the use of direct binary predicates (section 5).

4.1 FrameNet–WordNet Mapping

While FrameNet [23, 24] is the largest high-quality inventory of semantic frame descriptions and their participants, WordNet [28] is the most well-known resource capturing meanings of words in a lexical network, covering for example nouns and named entities missing in FrameNet. WordNet, for instance, serves as the backbone of YAGO's ontology. This section proposes a novel way of mapping the two resources, which later enables us to integrate both of them into FrameBase's schema.

WordNet contains synsets, which are sets of sense-disambiguated synonymous words with a given part of speech (POS), such as noun or verb.

FrameNet contains lexical units (LUs), which are also POS-annotated words associated to frames. Because of the semantics of the containing frame, LUs are also disambiguated to a certain extent, though not with the same granularity as in WordNet. The objective at hand is to map synsets and LUs with the same meaning, so this can be later used to enrich FrameBase's FrameNet-based schema with relations and annotations from WordNet.

More specifically, the objective is to map each LU to one and only one synset. While there are some LUs that could be mapped to more than one synset, this will favor precision, which is desirable for the purpose of obtaining a clean knowledge base. The only cases where this model would be detrimental to precision are those where LUs do not have any associated synset, but these are few and most can easily be avoided by omitting LUs with parts of speech not covered in WordNet, such as prepositions.

This choice allows to model the mapping as a function $S(l|a,b)$ from LUs to synsets as in (D.1). S_l stands for the synsets that have the same lexical label and POS as the LU l, μ_{L} and μ_{G} are the lexical and gloss (definition) overlap, respectively, f yields the corpus frequency of the synset, and a and b are parameters for a linear combination (the third parameter can be omitted because of the argmax function).

$$S(l|a,b) = \underset{s \in S_l}{\operatorname{argmax}}\ \mu_{\mathrm{L}}(l,s) + a \cdot \mu_{\mathrm{G}}(l,s) + b \cdot f(s) \tag{D.1}$$

The lexical overlap μ_{L} of a LU l and a synset s is the size of the intersection between the POS-annotated words from the LUs in the same frame as l and the POS-annotated words in s and its neighborhood. The neighborhood is defined as the synsets connected by a selection of lexical and semantic pointers such as "See also", "Similar to", "Antonym", "Attribute" and "Derivationally related". This expansion is useful to reduce sparsity and better match the sets with those generated for the LUs, which due to the different semantics of frames and synsets, may already include these related words.

The gloss overlap μ_{G} is the size of the intersection between the set of words in the definition of the LU and the gloss of the synset. CoreNLP library [32] is used to clean XML tags, tokenize, POS-label, and lemmatize the text, and all words except nouns and verbs are filtered out.

Parameters a and b are trained with a greedy search over several randomized seeds, obtaining optimal values $a = 5, b = 0.13$.

4.2 Schema Induction

In FrameBase, frames are modeled as classes whose instances are the particular events. The frame elements of each frame are properties whose domain is that frame. The class hierarchy of frames is created as follows.

1. **General Frames:** FrameNet's frame inheritance and perspectivization relations are modeled as class subsumption between frames, by means of two

specific properties that inherit from `rdfs:subClassOf`, so that both remain distinguishable but contribute to the hierarchy and allow RDFS inference. Additionally, a top frame is declared for the hierarchy. Inheritance between frame element properties is modeled with a direct subproperty relation. Semantic types are sometimes provided as ranges in FrameNet, but their current coverage is limited, and therefore have been left out of FrameBase. Under this model, an instance of the *Commerce_sell* frame with a certain *Commerce_sell-Buyer* x, is also an instance of the *Giving* frame and x is the *Giving-Recipient*, because the first frame inherits from the latter. Likewise, it is also an instance of *Transfer* and x is the *Transfer-Recipient*, because *Giving* is a perspective on *Transfer*.

2. **Leaf Nodes:** Since FrameNet's original frame inventory is coarse-grained and different LUs like *construction* and *to glue* evoke the same frame, more specific frames associated to each LU are employed. In other words, every LU is treated as evoking its own separate fine-grained frame, denoted as *LU-microframe*, which is made a subclass of the more coarse-grained original FrameNet frame. In addition, another type of microframes, denoted as *synset-microframes*, are created from the synsets in WordNet 3.0.
3. **Intermediate Nodes:** The LU-microframes resulting from the process above are very fine-grained. There are distinct LUs for *buy* from *acquire*. This is a problem for knowledge representation because it increases sparsity. At the same time, some original frames from FrameNet are very coarse-grained, as mentioned above, so they cannot be used. For instance, various kinship relationships such as *mother*, *sister-in-law*, etc. are lumped together. This wide range of LUs may stand in various lexical-semantic relationships without these being indicated, including synonymy, antonymy, or nominalization. The only characteristic they have in common is that, by definition, they evoke a similar kind of situation. Overall, neither the fine-grained nor the coarse-grained levels are ideal for knowledge representation purposes.
   This is addressed by providing a novel intermediate level composed of *cluster-microframes* that group equivalent LU-microframes and synset-microframes together, solving the problem described above, and integrating synset-microframes into a single backbone.
   The clusters are generated in the following way. First, for each LU-microframe, the corresponding synsets from the FrameNet–WordNet mapping are retrieved. In the case of the mapping in section 4.1, the set has no more than one element, but in the general case it could have more. Then, that set is expanded by adding all other synsets related by lexical relations reflecting cross-POS morphological transformations: "Derivationally related", "Derived from Adjective", "Participle" and "Pertainym". In general, these lexical relations do not necessarily imply any close semantics (e.g., *create/make – creature/animal*), but when restricted to synsets all tied to the same FrameNet frame, such cases are normally factored out. Therefore, the

set is reduced to those synsets that also belong to another set produced from a sibling LU from the same frame. The goal of using the lexical relations is linking cross-POS LU-microframes that evoke the same specific situation with a different syntactic form, such as nominalizations (*produce–production*), non-finite verb forms (*produce–produced*), adjectivization, or adverbization. Next, the LU-microframe is connected with the synset-microframes from the set of synsets, using the property `framebase:isSimilarTo`, which is declared to be transitive and symmetric in OWL.
After the process is run for all LU-microframes and the transitive closure of `framebase:isSimilarTo` is materialized, each cluster is represented by a clique of `framebase:isSimilarTo`. Finally, the intermediate cluster-microframes are reified[4] and declared superframes of the members of the cluster, and subframes of their previously immediate superframe. The cluster-microframe is also connected by `framebase:isSimilarTo` to the subframes. An example of two sibling cluster-microframes with all their members can be appreciated in Figure E.1.
The use of the property `framebase:isSimilarTo` allows to have a direct connection between members of the cluster. It may also be convenient in contexts where a user wants to reduce sparsity by completely merging all members of each cluster. In this case, he can do it as simply as declaring `framebase:isSimilarTo` to be a subproperty of `rdfs:subClassOf` and enable RDFS inference. By virtue of the already materialized inverses of `framebase:isSimilarTo`, every instance of a member of the cluster, including the cluster-microframe, will become an instance of the others. Alternatively, `owl:equivalentClass` can be used.

Names, definitions and glosses in FrameNet and WordNet are also used to create text annotations for our schema. Lexical forms are attached with `rdfs:label` and definitions and glosses from FrameNet and WordNet are attached with `rdfs:comment`. Additional linguistically rich annotations are added using Lemon [33].

Following the best practices in the Linked Open Data community, we link synset-microframes to URIs in the canonical RDF translation of WordNet [34]. We also provide links to word-sense URIs in lexvo.org, a KB that connects information about languages, words, characters, and other human language-related entities [35]. This allows FrameBase to be transitively connected to other KBs in the Linked Open Data web, as well as provide multilingual support.

[4]This is yet another different but related use of the term *reification*. In general, reification means the process of making something real, and in the context of knowledge bases, can be used whenever a new entity is created for something that was only implicitly represented before, generally as a function of pre-existing entities.

example for schema.pdf example for schema.pdf

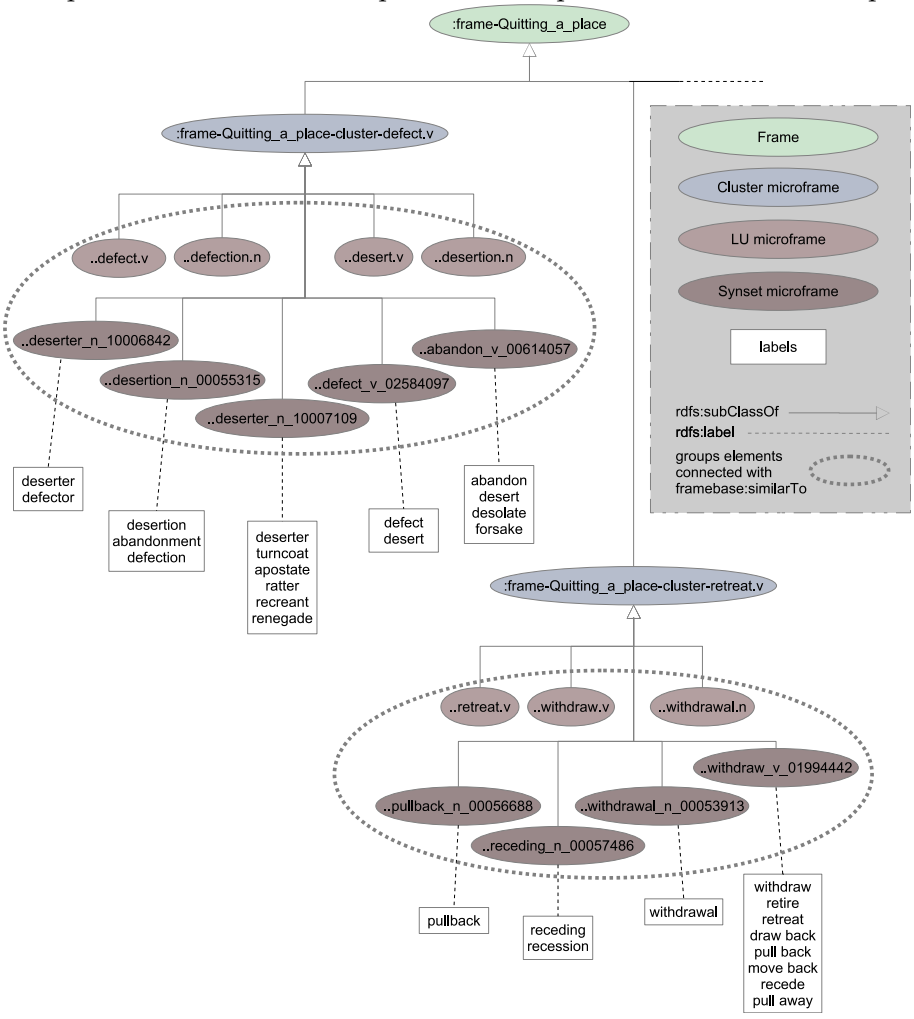


Fig. D.1: Example of some microframes and labels under the general frame class `:frame-Quitting_a_place`. The initial part of the names of classes is common and has been ommitted.

5 Automatic Reification–Dereification Mechanism

While frames are convenient for representational purposes, users wishing to query the knowledge base benefit from binary predicates between pairs of frame elements. For example, for a birth event, binary predicates like `bornInPlace` and `bornOnDate` can facilitate querying by offering a more compact and simple representation.

Thus, FrameBase presents a novel mechanism to seamlessly convert between frame representations and DBPs. This mechanism can also allow us to avoid materializing frame instances when only two frame elements are needed.

5.1 Structure of ReDer rules

The *dereification rules* have the form expressed in Figure D.2. Additionally, for each dereification rule there is a converse reification rule so that one can go back from binary predicates to the frame representation. Each DBP (direct binary predicate) has only one set of possible frame and frame elements associated, and therefore chaining reification and dereification rules is an idempotent operation. We term the pair of a reification rule and its converse dereification rule as a ReDer (reification-dereification) rule. An example of a ReDer rule is provided in Figure D.3.

```
?s  <DIRECT_BINARY_PREDICATE>  ?o
↕
<FRAME_INSTANCE>  a  <FRAME_CLASS> ,
<FRAME_INSTANCE>  <FRAME_ELEMENT-S>  ?s ,
<FRAME_INSTANCE>  <FRAME_ELEMENT-O>  ?o .
```

Fig. D.2: The general pattern of a dereification rule.

```
?s  :dbp-Statement-writesAboutTopic  ?o
↕
?F  a  :frame-Statement-write.v ,
?F  :fe-Statement-Speaker  ?s ,
?F  :fe-Statement-Topic  ?o .
```

Fig. D.3: The general pattern of a dereification rule.

The ReDer rules can be implemented in different ways.

- As SPARQL CONSTRUCT queries, due to SPARQL's prominence as a standard query language for KBs. These can be used to materialize the DBPs into the KB.
- As clauses with triples as atoms to be fed in general-purpose inference engines, with or without materialization. For example, ReDer rules have also been implemented as rules for the Rubrik reasoner in Jena [36].

Besides the plain `rdfs:label` and `rdfs:comment` annotations, we annotate the DBPs using Lemon [33], which allows syntactically rich annotations that describe the internal structure and external syntactic frame of their labels. Instead of using the automatic generator, that uses automatic tokenization, parsing, etc, we use our knowledge of the structure of the different possible labels for DBPs to create perfect annotations. Similarly, we also use Lemon for annotating microframes.

5.2 Creation of ReDer rules

The ReDer rules are automatically built using the annotations of English sentences given for different LUs in FrameNet, namely the grammatical function (GFs) and phrase types (PTs) [24]. Each instance of an example sentence annotated by a frame is accompanied by the GF and PT associated to each of the FEs of that frame filled in that sentence.

For verb-based LUs, FrameNet provides three kinds of GF labels: External Argument (Ext), Object (Obj), and Dependent (Dep). Some of the PT labels that can be found are N, NP, Obj, PPinterrog [24]. Dereified binary predicates and reification-dereification rules are created for the pairs of frame elements whose syntactic annotations for some sentence satisfy the creation rules below, using the GF and PT labels.

As for the general reification-dereification rule pattern in Figure D.2, the postfixes "-s" and "-o" indicate the data associated to the FEs that fill the first and second arguments of the DBP, or equivalently, the subject and the object of the resulting RDF triple. The creation of the DBP implies a creation of a dereification rule following the pattern in Figure D.2, with `<FRAME_CLASS>` defined by the LU, and `<FRAME_CLASS>` left as a free variable. The corresponding reification rule is built similarly, but assigning an anonymous node or a skolem constant to `<FRAME_CLASS>`.

Creation Rule 1: Verb Agent-Patient
Create DBP with name
"ConjugateThirdPersonSingular(LU)"
if
IsVerb(LU) and PT-o in {N, NP, Obj, PPinterrog, Sinterrog, QUO, Sfin, Sub, VPing} and ((GF-s==Ext and GF-o==Obj and not IsPassivePosHeuristic(LU) and not IsPassiveDepHeuristic(LU)) or (GF-s==Obj and GF-o==Ext and IsPassivePosHeuristic(LU) and IsPassiveDepHeuristic(LU)))

Examples of obtained DBPs and reification-dereification rules:

```
?S :dbp-Forming_relationships-divorces ?O
↕
?R a :frame-Forming_(...)-divorce.v ,
?R :fe-Forming_relationships-Partner_1 ?S ,
?R :fe-Forming_relationships-Partner_2 ?O .
```

```
?S :dbp-Win_prize-wins ?O
↕
?R a :frame-Win_prize-win.v ,
?R :fe-Win_prize-Competitor ?S ,
?R :fe-Win_prize-Prize ?O .
```

Creation Rule 2: Verb Patient-Agent
Create DBP with name
"is ConjugatePastParticiple(LU) by"
if
IsVerb(LU) and PT-o in {N, NP, Obj, PPinterrog, Sinterrog, QUO, Sfin, Sub, VPing} and ((GF-s==Obj and GF-o==Ext and not IsPassivePosHeuristic(LU) and not IsPassiveDepHeuristic(LU)) or (GF-s==Ext and GF-o==Obj and IsPassivePosHeuristic(LU) and IsPassiveDepHeuristic(LU)))

Examples of obtained DBPs and reification-dereification rules:

```
?S :dbp-Filling-isLoadedBy ?O
↕
?R a :frame-Filling-load.v ,
?R :fe-Filling-Goal ?S ,
?R :fe-Filling-Agent ?O .
```

```
?S :dbp-Kidnapping-isKidnapedBy ?O
↕
?R a :frame-Kidnapping-kidnap.v ,
?R :fe-Kidnapping-Victim ?S ,
?R :fe-Kidnapping-Perpetrator ?O .
```

Creation Rule 3: Verb Agent-Complement
Create DBP with name
"ConjugateThirdPersonSingular(LU) Prep FrameElement-o"
if
IsVerb(LU) and PT-o==PP[Prep] and ((GF-s==Ext and GF-o==Dep and not IsPassivePosHeuristic(LU) and not IsPassiveDepHeuristic(LU)) or (GF-s==Obj and GF-o==Dep and IsPassivePosHeuristic(LU) and IsPassiveDepHeuristic(LU)))

Examples of obtained DBPs and reification-dereification rules:

```
?S :dbp-Creating-createsFromComponents ?O
```
↕
```
?R a :frame-Creating-create.v ,
?R :fe-Creating-Creator ?S ,
?R :fe-Creating-Components ?O .
```

```
?S :dbp-Win_prize-winsAtVenue ?O
```
↕
```
?R a :frame-Win_prize-win.v ,
?R :fe-Win_prize-Competitor ?S ,
?R :fe-Win_prize-Venue ?O .
```

Creation Rule 4: Verb Patient-Complement
Create DBP with name
"is ConjugatePastParticiple(LU) Prep FrameElement-o"
if
IsVerb(LU) and PT-o==PP[Prep] and ((GF-s==Obj and GF-o==Dep and not IsPassivePosHeuristic(LU) and not IsPassiveDepHeuristic(LU)) or (GF-s==Ext and GF-o==Dep and IsPassivePosHeuristic(LU) and IsPassiveDepHeuristic(LU)))

Examples of obtained DBPs and reification-dereification rules:

```
?S :dbp-Destroying-isDestroyedByMeans ?O
↕
?R a :frame-Destroying-destroy.v ,
?R :fe-Destroying-Undergoer ?S ,
?R :fe-Destroying-Means ?O .
```

```
?S :dbp-Beat_opponent-isDefeatedByWinner ?O
↕
?R a :frame-Beat_opponent-defeat.v ,
?R :fe-Beat_opponent-Loser ?S ,
?R :fe-Beat_opponent-Winner ?O .
```

Using only agent and patient as subject of the triple avoids rules defining certain kinds of DBPs that would be rarely useful, like those connecting the time and place, or the place and the cause.

There is no explicit syntactic annotation in FrameNet to indicate if the verb LUs are evoked in passive form. Therefore, two different heuristics for detecting this. One (IsPassivePosHeuristic(LU)) draws on the POS annotations available in FrameNet, and decides that the target (LU) verb is in passive iff it appears as a past participle, and the verb *to be*, in any form, is in a prior position, without another verb in between. The other heuristic (IsPassiveDepHeuristic(LU)) uses the Stanford dependency parser [37], determining that the target (LU) verb is in passive iff it is the source of any of the dependencies nsubjpass, csubjpass or auxpass. Both heuristics make type I and II mistakes differently, so the cases where they disagree were discarded, and in the ones where they agree that they there is passive form, the rules are

created inverting the Ext and Obj GFs.

For noun-based LU-microframes, a verb is needed that takes the noun as argument, normally as direct object. Across RDF vocabularies and ontologies, this verb is sometimes made implicit in human-readable IRIs. For example `skos:hasTopConcept` includes "has" explicitly, while `skos:topConceptOf` includes "is" implicitly. In FrameBase, the modeling choice has been to always make it explicit both in the IRI and the lexical annotations, in order to avoid ambiguity and prevent incorrect use. The verbs have been conjugated in third person of singular.

For each noun LU in an annotation, the head verb has been extracted by parsing the example annotated sentences with the Stanford dependency parser and searching the paths of dependencies indicated in the creation rules 5 and 6[5]. For brevity, the paths are annotated with the notation of SPARQL property paths, but this is not part of any query.

Creation rule 5 contains several possible dependency paths.

- (LU `^dobj` HeadVerb) matches HeadVerb="*make*" and LU="*comment*" for the sentence "*I have decided not to make any further comment concerning the change of ball during the lunch interval at Lord 's on Sunday*".
- (LU `cop` HeadVerb) matches HeadVerb="*is*" and LU="*maiden name*" for the sentence "*The maiden name of one of his wives (probably the second) was Watt*".
- (LU `^nsubj/cop` HeadVerb) matches HeadVerb="*is*" and LU="*cause*" for the sentence "*The short-term cause of overriding local significance were the droughts and crop failures in 1920 and 1921*".
- (LU `^prep_*/cop` HeadVerb) matches HeadVerb="*is*" and LU="*cause*" for the sentence "'Well-meaning ignorance is one of the biggest causes of animal suffering in this country (...)'.
- (LU `^prep_*/^dobj` HeadVerb) matches HeadVerb="*give*" and LU="*thought*" for the sentence "*I have given a great deal of thought as to how much I should actually tell you about this period and what just to leave to your imagination*".

Creation rule 6 fires cases with phrasal verbs, where the head verb must be extracted with a particle.

- (LU `^prep_`VerbParticle HeadVerb) matches HeadVerb="*go*", VerbParticle="*on*" and LU="*tour*" for the sentence "*Something else I shall miss by going on this dratted tour with Gwen !*".

[5]We use collapsed CC-processed dependencies, version 3.2.0)

Creation Rule 5: Verb Noun
Create DBP with name
"ConjugateThirdPersonSingular(HeadVerb) LU Prep Frame-Element-o"
if
IsNoun(LU) and PT-o==PP[Prep] and GF-s==Ext and GF-o==Dep and (LU ^dobj HeadVerb or LU cop HeadVerb or LU ^nsubj/cop HeadVerb or LU ^prep_*/cop HeadVerb or LU ^prep_*/^dobj HeadVerb)

Examples of obtained DBPs and reification-dereification rules:

```
(...)-makesInferenceFromEvidence ?O
```
↕
```
?R a :frame-Coming_to_believe-inference.n ,
?R :fe-Coming_to_believe-Cognizer ?S ,
?R :fe-Coming_to_believe-Evidence ?O .
```

```
?S :dbp-Arriving-makesEntranceByMeans ?O
```
↕
```
?R a :frame-Arriving-entrance.n ,
?R :fe-Arriving-Theme ?S ,
?R :fe-Arriving-Means ?O .
```

Creation Rule 6: Verb Particle Noun
Create DBP with name
"ConjugateThirdPersonSingular(HeadVerb) VerbParticle LU Prep Frame-Element-o"
if
IsNoun(LU) and PT-o==PP[Prep] and GF-s==Ext and GF-o==Dep and (LU ^prep_VerbParticle HeadVerb)

Examples of obtained DBPs and reification-dereification rules:

```
(...)-worksTowardsUnderstandingAboutTopic ?O
↕
?R a :frame-Awareness-understanding.n ,
?R :fe-Awareness-Cognizer ?S ,
?R :fe-Awareness-Topic ?O .
```

```
(...)-goesIntoDiscussionWithInterlocutor2 ?O
↕
?R a :frame-Discussion-discussion.n ,
?R :fe-Discussion-Interlocutor_1 ?S ,
?R :fe-Discussion-Interlocutor_2 ?O .
```

In the cases where the FRAME-ELEMENT-O is included in the DBP but the FrameNet annotations cannot provide a suitable preceding preposition PREP, we use statistics from the cases where such preposition can be obtained, and we choose, if available, the most common preposition associated to the name of that FE across all frames.

With the rules obtained with the process above, the same DBP can be associated to different pairs of frame elements in a given LU-microframe, owing to different senses or syntactic frames for a given verb (for example the transitive and intransitive frames for *smuggle*). This would conflate different senses, and if the reification and the dereification directions of the rules were chained, it would logically entail different pairs of frame elements, which would not be sound. Furthermore, a given pair of frame elements can also produce different DBPs. To achieve the idempotency mentioned earlier, the Kuhn–Munkres algorithm is used in order to obtain a one-to-one assignment, using as weights the additive inverse of the number of annotated example sentences for a DBP and a pair of frame elements, because the patterns with more example sentences are usually more intuitive. The cubic complexity of the algorithm is not a concern because each frame leads to a separate graph which can be handled independently.

6 Integration

Knowledge from other KBs such as Freebase can be integrated *integration rules* with two graph patterns as antecedent and consequent sharing some variables. When there is a variable substitution that, applied to the antecedent, makes it a subset of the source KB, then the consequent after the same transformation can be added to the FrameBase instance data (A-Box in the jargon of description logics). When the sources are in RDF, the integration rules can be implemented as SPARQL CONSTRUCT queries. Otherwise, an off-the-shelf RDF converter[6] can be applied to pre-process the source.

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

The SPARQL examples in this and the next sections use the following prefixes.

```
PREFIX :      <http://framebase.org/ns/>
PREFIX freeb: <http://rdf.freebase.com/ns/>
PREFIX dbr:   <http://dbpedia.org/resource/>
PREFIX sch:   <http://schema.org/>
```

In the first subsection of this section, some example integration rules are presented for integrating events from different sources into FrameBase. Later, a discussion about the complexity of integration rules in general and the challenges they present is added.

6.1 Example Integration Rules

The two example integration rules above integrate knowledge from Freebase. They follow a relatively simple pattern, the first reifying a property from the source KB into a frame in FrameBase (Property–Frame), and the second translating a class from the source KB into a frame in FrameBase, and the outgoing properties into FE properties (Class–Frame).

```
CONSTRUCT {
  _:f a :frame-People_by_jurisdiction-citizen.n .
  _:f :fe-People_by_jurisdiction-Person ?person .
  _:f :fe-People_by_jurisdiction-Jurisdiction ?country .
} WHERE {
  ?person freeb:people.person.nationality ?country .
}

CONSTRUCT {
  _:f a :frame-Leadership-leader.n .
  _:f :fe-Leadership-Leader ?o1 .
  _:f :fe-Leadership-Governed ?o2 .
  _:f :fe-Leadership-Role ?o3 .
  _:f :fe-Leadership-Type ?o4 .
  _:timePeriod a :frame-Timespan-period.n .
  _:timePeriod :fe-Timespan-Start ?o5 .
  _:timePeriod :fe-Timespan-End ?o6 .
} WHERE {
  ?cvti a freeb:organization.leadership .
  OPTIONAL { ?cvti
    freeb:organization.leadership.person ?o1 .}
  OPTIONAL { ?cvti
    ...organization.leadership.organization ?o2 .}
  OPTIONAL { ?cvti
    freeb:organization.leadership.role ?o3 .}
  OPTIONAL { ?cvti
```

```
    freeb:organization.leadership.title ?o4 .}
  OPTIONAL { ?cvti
    freeb:organization.leadership.from ?o5 .}
  OPTIONAL { ?cvti
    freeb:organization.leadership.to ?o6 .} }
```

The next example pertains the Event class in DBpedia.

```
CONSTRUCT {
  ?f a :frame-Event-event.n .
  #
  ?f :fe-Event-Time _:timePeriod .
    _:timePeriod a :frame-Timespan-period.n ;
      fbe:fe-Timespan-Start ?o1 ;
      fbe:fe-Timespan-End ?o2 .
  #
  _:af2 a :frame-Relative_time-preceding.a ;
    :fe-Relative_time-Landmark_occasion ?f ;
    :fe-Relative_time-Focal_occasion ?o3 .
  #
  _:af3 a :frame-Relative_time-following.a ;
    :fe-Relative_time-Landmark_occasion ?o3 ;
    :fe-Relative_time-Focal_occasion ?f .
  #
  _:af4 a :frame-Relative_time-following.a ;
    :fe-Relative_time-Landmark_occasion ?f ;
    :fe-Relative_time-Focal_occasion ?o4 .
  #
  _:af5 a :frame-Relative_time-preceding.a ;
    :fe-Relative_time-Landmark_occasion ?o4 ;
    :fe-Relative_time-Focal_occasion ?f .
  #
  _:af6 a :frame-Relative_time-following.a ;
    :fe-Relative_time-Landmark_occasion ?f ;
    :fe-Relative_time-Focal_occasion ?o5 .
  #
  _:af7 a :frame-Relative_time-preceding.a ;
    :fe-Relative_time-Landmark_occasion ?o5 ;
    :fe-Relative_time-Focal_occasion ?f .
  #
  ?f :fe-Event-Reason ?o6 .
  #
  _:af8 a :frame-Dimension-length.n ;
    :fe-Dimension-Object ?f ;
    :fe-Dimension-Measurement ?o7 .
  #
```

```
  ?f a :frame-Social_event-meeting.n ;
    :fe-Social_event-Attendee ?o9 ;
    :fe-Social_event-Duration ?o7 .
  #
} WHERE {
  ?f a dbr:Event .
  OPTIONAL{?f dbr:startDate ?o1}
  OPTIONAL{?f dbr:endDate ?o2}
  OPTIONAL{?f dbr:previousEvent ?o3}
  OPTIONAL{?f dbr:followingEvent ?o4}
  OPTIONAL{?f dbr:nextEvent ?o5}
  OPTIONAL{?f dbr:causedBy ?o6}
  OPTIONAL{?f dbr:duration ?o7}
  OPTIONAL{ #Omitted
    ?f dbr:numberOfPeopleAttending ?o8}
  OPTIONAL{?f dbr:participant ?o9}
}
```

From the 9 properties of the class `Event`, `numberOfPeopleAttending` was omitted because the class `Event` is too general for it, as it has subclasses such as `PersonalEvent` (`Birth`, etc.) and `SocietalEvent`, that appear more appropriate for this. The remaining 8 properties were integrated, but even though the example shares the same basic structure as the Class–Frame rule provided for Freebase, it includes additional complex patterns in the consequent.

The `dbr:Event` class has several subclasses which can also be translated. However, the hierarchy in the original ontology is not necessarily consistent with the hierarchy in FrameBase. Only in certain cases does a subsumption relationship between two entities of the source also exist between the two entities' respective translations to FrameBase. Therefore, for each translation of an element in the source KB, the translations of more general elements can be added, and this will provide additional knowledge that would not always be inferred by the FrameBase schema alone.

For example, using RDFS inference, the substitutions for `?f` that fire the rule below will also fire the one for `dbr:Event`, because `dbr:SocietalEvent` is a subclass of `dbr:Event`. This rule is very short because all of the outgoing properties belong to the parent `Event` class itself.

```
CONSTRUCT {
  ?f a :frame-Social_event-meeting.n .
} WHERE {
  ?f a dbr:SocietalEvent
}
```

Similarly, the substitutions for `?f` that fire the rest of the examples from DBpedia below, will also fire the ones for `dbr:SocietalEvent` and

dbr:Event, because the classes captured in the antecedent are subclasses of dbr:SocietalEvent.

```
CONSTRUCT {
  ?f a :frame-Project-project.n .
  ?f :fe-Project-Activity dbr:Space_exploration .
} WHERE {
  ?f a dbr:SpaceMission
}
```

In the rule above, we minimize the need for declaring new frames and frame elements for specialized domains by making use of the compositionality of most specialized terms, creating complex structures that combine the semantics of simpler, basic elements. For instance, the translation for the type dbr:SpaceMission declares a frame of type Project-project.n, and specifies that it is about space exploration by assigning dbrl:SpaceMission as the value for the Project-Activity FE.

```
CONSTRUCT {
  ?f a fbe:frame-Social_event-convention.n .
} WHERE {
  ?f a dbr:Convention
}

CONSTRUCT {
  ?f a :frame-Change_of_leadership-election.n .
} WHERE {
  ?f a dbr:Election .
}

CONSTRUCT {
  ?f a :frame-Social_event-festival.n .
  ?f :fe-Social_event-Attendee ?o3 .
  ?f :fe-Social_event-Descriptor dbr:Film .
  ?f a :frame-Competition-competition.n .
  ?f :fe-Competition-Participant_1 ?o3 .
  ?f :fe-Competition-Competition dbr:Film .
  _:af1 a :frame-Ordinal_numbers-first.a .
  _:af1 :fe-Ordinal_numbers-Item ?o1 .
  _:af1 :fe-Ordinal_numbers-Comparison_set ?f .
  _:af1 :fe-Ordinal_numbers-Comparison_set dbr:Film .
  _:af2 a :frame-Ordinal_numbers-last.a .
  _:af2 :fe-Ordinal_numbers-Item ?o2 .
  _:af2 :fe-Ordinal_numbers-Comparison_set ?f .
  _:af2 :fe-Ordinal_numbers-Comparison_set dbr:Film .
} WHERE {
  ?f a dbr:FilmFestival .
```

```
  OPTIONAL{?f dbr:closingFilm ?o1}
  OPTIONAL{?f dbr:openingFilm ?o2}
  OPTIONAL{?f dbr:film ?o3}
}

CONSTRUCT {
  ?f a :frame-Hostile_encounter-hostility.n .
  _:af1 a :frame-Death-die.v .
  _:af1 :fe-Death-Sub_event ?f .
  _:af1 :fe-Death-Protagonist ?o1 .
  ?f :fe-Hostile_encounter-Side_1 ?o2 .
  _:af3 a :frame-Part_whole-part.n .
  _:af3 :fe-Part_whole-Part ?f .
  _:af3 :fe-Part_whole-Whole ?o3 .
  ?f :fe-Hostile_encounter-Place ?o4 .
  ?f :fe-Hostile_encounter-Result ?o5 .
  ?f :fe-Hostile_encounter-Depictive ?o6 .
  ?f :fe-Hostile_encounter-Side_2 ?o7 .
} WHERE {
  ?f a dbr:MilitaryConflict .
  OPTIONAL{?f dbr:casualties ?o1}
  OPTIONAL{?f dbr:combatant ?o2}
  OPTIONAL{?f dbr:isPartOfMilitaryConflict ?o3}
  OPTIONAL{?f dbr:place ?o4}
  OPTIONAL{?f dbr:result ?o5}
  OPTIONAL{?f dbr:strength ?o6}
  OPTIONAL{?f dbr:opponents ?o7}
}
```

We also present the translation of the class `Event` in schema.org. This provides an example of integration. Due to space restrictions, we omit the subclasses here, but these have very few genuine properties, and therefore the specialization is relatively simple. Besides, the taxonomy of schema.org events has some inconsistency issues that makes its use complex: the Event class is defined as capturing events such as concerts, lectures, and festivals, with properties such as "typical age range", but there are sub-events such as `UserInteraction` and `UserPlusOnes` that actually represent a more general kind of events.

```
CONSTRUCT {
  ?f a :frame-Social_event-meeting.n .
  ?f a :frame-Event-event.n .
  #
  ?f :fe-Social_event-Time _:timePeriod .
    _:timePeriod a fbe:frame-Timespan-period.n ;
      fbe:fe-Timespan-Start ?Osta ;
      fbe:fe-Timespan-End ?Oend .
```

```
  ?f :fe-Event-Time _:timePeriod .
  #
  ?f :fe-Social_event-Duration ?Odur .
  ?f :fe-Event-Duration ?Odur .
  #
  ?f :fe-Social_event-Place ?Oloc .
  ?f :fe-Event-Place ?Oloc .
  #
  ?f :fe-Social_event-Attendee ?Oatt .
  ?f :fe-Social_event-Host ?Oorg .
  #
  ?f :fe-Social_event-Occasion ?Osup .
  ?Osub :fe-Social_event-Occasion ?f .
  #
  ?Ooff a :frame-Offering-offer.v ;
    :fe-Offering-Theme ?f .
  #
  ?f a :frame-Performing_arts-performance.n ;
    :fe-Performing_arts-Performer ?Oper ;
    :fe-Performing_arts-Performance ?Owor .
  #
  _:af1 a :frame-Recording-record.v ;
    :fe-Recording-Phenomenon ?f ;
    :fe-Recording-Medium ?Orec .
  #
  ?f :fe-Social_event-Descriptor ?Oeve .
  #
  _:af2 a Change_event_time-postpone.v ;
    Change_event_time-Event ?f;
    Change_event_time-Landmark_time ?Opre.
  #
  _:af a :frame-Typicality-normal.a .
  _:af :fe-Typicality-Entity _:af2 .
  _:af2 :frame-Age-age.n .
  _:af2 :fe-Age-Age ?Otyp .
} WHERE {
  ?f a sch:Event .
  OPTIONAL{?f sch:startDate ?Osta}
  OPTIONAL{?f sch:endDate ?Oend}
  OPTIONAL{?f sch:duration ?Odur}
  OPTIONAL{?f sch:location ?Oloc}
  OPTIONAL{?f sch:attendee ?Oatt}
  OPTIONAL{?f sch:organizer ?Oorg}
  OPTIONAL{?f sch:superEvent ?Osup}
  OPTIONAL{?f sch:subEvent ?Osub}
  OPTIONAL{?f sch:offers ?Ooff}
```

```
  OPTIONAL{?f sch:performer ?Oper}
  OPTIONAL{?f sch:workPerformed ?Owor}
  OPTIONAL{?f sch:recordedIn ?Orec}
  OPTIONAL{?f sch:eventStatus ?Oeve}
  OPTIONAL{?f sch:previousStartDate ?Opre}
  OPTIONAL{?f sch:typicalAgeRange ?Otyp}
  # No translation
  OPTIONAL{?f sch:doorTime ?Odoo}
}
```

The only extension of the FrameBase schema used for these examples was the frame `:frame-Timespan-period.n` with the start and end frame elements, used to denote periods of time. This, however, is not an ad-hoc extension motivated by a particular need of only one source, but a very general one. Of the 16 properties of the `Event` class, only one (`sch:doorTime`, with an official gloss "The time admission will commence"), was not integrated. The remaining 15 were integrated.

6.2 Complex Transformations

Most of the integration rules we have described follow a pattern which involves an event *class* in the source being translated as a frame class, and each of their outgoing properties being mapped to individual frame elements. However, there are multiple ways in which the rules can differ from this basic pattern.

1. Sometimes, a class integration rule may need to instantiate multiple frames rather than just a single one. We distinguish two main types of this phenomenon.

   a) The instantiated frame instances may be connected by frame elements. Examples of this include the frame `:frame-Timespan-period.n` created to represent time periods, and the subframes of `Relative_time` to express precedence between events (all in the example for `dbr:Event`). The same applies when a frame element is used to specify a frame beyond the lexical unit (see the rule for `dbr:Space_exploration`).

   b) Several frames can also be evoked separately, without the instances being directly connected by any frame element. When these frames describe different perspectives of the same event, there is the possibility that FrameNet links them by means of *perspectivization*, and therefore FrameBase can infer one from another. For example, classes `:frame-Commerce_buy-buy.v` and `:frame-Commerce_sell-sell.v`, which are used for classes `Buy` and Sell in the organized crime taxonomy, are both perspectivizations of `:frame-Commerce_goods-transfer`. In this case, inference is possible because RDFS subclass and subproperty properties are

used in FrameBase to reflect the perspectivization relation between frame classes and frame elements respectively. Another example are `:frame-Receive_visitor_scenario` and `:frame-Visit_host`, which are perspectives of `:frame-Visitor_and_host`. However, in other cases one cannot rely on existing inference. For instance, see how the rule to translate `Event` from schema.org, besides frames `Event-event.n` and `Timespan-period.n`, also instantiates `Performing_arts-performance.n`, `Recording-record.v` and `Offering-offer.v` when certain properties are present.

2. Another possible source of complexity is that frame elements can be inverted. In this case, the integration rules need to invert the order of the arguments, like in the second appearance of `:fe-Social_event-Occasion` in the integration rule for the class `Event` in schema.org.

3. Oftentimes, a *property* (rather than a class) in the source can be translated as evoking a frame on its own. In this case, the two involved entities become connected to the new frame by means of frame elements. This phenomenon can also appear on its own: an example of this is the first integration rule example, for `freeb:people.person.nationality`.

Arbitrary combinations of these phenomena are possible (e.g. the rule integrating the Event class from schema.org). Overall, this makes automatic generation of the integration rules a very hard task, because it generates so many free variables that any attempt to train a system would face extreme sparsity. In some cases, it may thus make sense to sacrifice some recall, developing a system that only covers simpler transformations.

6.3 Representational Flexibility

Finally, another potential challenge for data integration is that even when a homogeneous schema such as FrameBase is used, certain kinds of knowledge can still be expressed in multiple possible ways.

- One example is that there are several ways of narrowing down the meaning of a frame instance. One is creating a new sub-microframe associated with a new lexical unit. Another one is assigning a value to a frame element (see example for `SpaceMission`), as mentioned above. This may lead to divergent choices of representation even within the core part of the schema that comes from FrameNet.

- Another example of this is when a frame element needs to be reified, i.e. represented as a frame instance, to express something additional about it (as would be the case of the property `previousStartDate` in schema.org), or when there is no direct frame element available and creating it would lead to a combinatorial explosion in the size of the schema. An example

of the latter is the difference between our proposal for using the frame `Part_whole` for expressing sub-event relations, and how we used the frame element `Occasion` for the frame `Social_event`, but this is a particularity of that frame. Again, this may lead to an incoherent representations in the knowledge base. One potential way of addressing this would be extending the reification–dereification mechanism of FrameBase.

7 Evaluation

This section evaluates the quality of the results and show some example queries.

7.1 FrameNet–WordNet Alignment

To evaluate the created schema, the created FrameNet–WordNet mapping has been compared to the MapNet gold standard [38]. MapNet uses older versions of FrameNet and WordNet, so mappings from WordNet 1.6 to 3.0 [39] had to be applied, removing those with a confidence lower than one, and the few LUs of FrameNet 1.3 that are not contained in FrameNet 1.5 were discarded. Table D.3 compares the results against state-of-the-art approaches and the scores that they report on the MapNet gold standard. As desired, the approach described in section 4 achieves high precision, while still maintaining good recall. 5-fold cross-validation was used for obtaining the results.

	Prec	Rec	F1	Acc
SVM Polynomial kernel 1 [38]	0.761	0.613	0.679	—
SVM Polynomial kernel 2 [38]	0.794	0.569	0.663	—
SSI-Dijkstra [40]	0.78	0.63	0.69	—
SSI-Dijkstra+ [40]	0.76	0.74	0.75	—
Neighborhoods [41]	—	—	—	0.772
FrameBase's mapping	0.789	0.709	0.746	0.864

Table D.3: Comparison of FrameBase's FrameNet–WordNet mapping to state-of-the-art approaches in terms of precision, recall, F1, and accuracy.

It may be relevant to note that there is in practice an upper bound to precision scores in tasks like this, because of the subjective component of any gold standard. The creators of the gold standard [38] report "0.90 as Cohen's Kappa computed over 192 LU-synset pairs for the same mapping task" by [42]. More generally, [43] maintains that "both people and automatic systems, when asked to assign tokens in a text to the appropriate senses in dictionaries, find the task difficult and do not agree among themselves".

7.2 Schema Induction

The FrameBase schema is based on FrameNet and WordNet and the mapping created between the two resources. It provides 19,376 frames, including 11,939

LU-microframes and 6,418 synset-microframes, all with lexical labels. A total of 18,357 microframes are clustered into 8,145 logical clusters, which are the sets of microframes whose elements are linked by a logical equivalence relation. The size of the schema is 250,407 triples.

An average precision of 87.55% $\pm$ 6.18% with a 95% Wilson confidence interval has been obtained. The evaluation showed a small change of nuance for 31.15% $\pm$ 9.38% of the correct pairs – most of these are caused by the choice to use semantic pointers such as "Similar to", which could be removed if very fine-grained distinctions of microframes were desired. The precision has been calculated from a random sample of 100 intra-cluster pairs that have been independently annotated by two of the authors. The linear weighted Cohen's Kappa over the three-valued combination of the two variables with which are annotated for each cluster pair, has a value of 0.23 over a maximum of 0.87. The scores were obtained with a random annotator.

In addition to the number of frames, the FrameBase schema provides a vocabulary of frame elements that goes well beyond the knowledge currently included in most KBs, in particular beyond time and location. This additional knowledge is routinely conveyed in natural language, and it seems likely that using a schema that provides for it paves the way to include it in KBs, either manually or automatically.

7.3 Reification–Derefication Rules

Additionally, reification–dereification rules are provided, with the same number of direct binary predicates, with both human-readable IRIs and lexical labels. 14,930 are verb-based and 10,270 are noun-based. The obtained average precision for verb-based rules is 96.22% $\pm$ 3.22%, and 80.43% $\pm$ 7.61% of the correct rules were found easily readable. For noun-based rules, the scores are 87.5% $\pm$ 6.41% and 91.91% $\pm$ 6.28%. A rule is considered to be not easily readable if the name of the direct binary predicate contains a frame element whose meaning is not obvious for a layman reader, or if it contains a preposition that is appropriate for some but not all possible objects, or it is not appropriate for the frame element in the name. For this evaluation, the same annotation methodology as for the intra-cluster pairs was followed, obtaining a Cohen's kappa of 0.39 over a maximum of 0.54.

7.4 Querying

FrameBase facilitates novel forms of queries. The following query, for instance, uses reified patterns to find the heads of the World Bank. Note that the clusters implemented in RDFS allow searching for the noun *head* (from the leadership frame), although the integration rule above only produced an instance of `fmbs:frame-Leadership-leader.n`. The results in Table D.4 show example instances seamlessly integrated into the FrameBase schema from both Freebase (rows 1–3, extracted from the second example integration rule

above) and YAGO2s (rows 4–5, extracted with a similar integration rule made for YAGO2s).

```
SELECT DISTINCT
?leader ?leaderLabel ?role ?roleLabel
WHERE {
  ?lumfi a :frame-Leadership-head.n .
  ?lumfi :fe-Leadership-Governed ?worldBank.
  ?lumfi :fe-Leadership-Leader ?leader .
  ?leader rdfs:label ?leaderLabel .
  VALUES ?worldBank {
    yago:World_Bank freeb:m.02vk52z
  }
  OPTIONAL{
    ?lumfi :fe-Leadership-Role ?role .
    ?role rdfs:label roleLabel .
  }
}
```

Alternatively, a direct binary predicate from the dereification rules can be used to obtain the same non-optional results, as illustrated in the query below. Either *leads* or *heads* can be used because the LU-microframes for these verbs are in the same cluster as the nouns *leader* and *head*, and there is a dereification rule between the *Leader* and *Governed* frame elements for both.

```
SELECT DISTINCT ?leader WHERE {
  ?leader :dereif-Leadership-heads ?worldBank .
  VALUES ?worldBank {
    yago:World_Bank freeb:m.02vk52z
  }
}
```

FrameBase can also be applied with natural language processing tools for question answering and data mining. For example, given the question “Who has

?leader	?role
fb:m/0h_ds2s ‘Caroline Anstey’	fb:m/04t64n ‘Managing Director’
fb:m/0d_dq5 ‘Mahmoud Mohieldin’	fb:m/04t64n ‘Managing Director’
fb:m/047cdkk ‘Sri Mulyani Indrawati’	fb:m/01yc02 ‘Chief Operating Officer’
yago:Jim_Yong_Kim	-
yago:Robert_Zoellick	-

Table D.4: Results from the query

been the head of the World_Bank", the SRL tool SEMAFOR [44] successfully extracts the frame *Leadership* with lexical unit *head.noun* and frame elements *Governed* and *Leader*. Based on this, and after a named entity disambiguator like AIDA [45] matches World_Bank to the entities in the KBs, the structured query can easily be built. Moreover, the same procedure can also be used to integrate new knowledge from a text into the KB, like FRED [27] does.

8 Conclusion

FrameBase is a novel approach for connecting knowledge from different heterogeneous sources to decades of work from the NLP community. Events can be described in very different ways across different knowledge bases. Our framework not only provides an efficient model to describe n-ary relations, but also integrates and transforms FrameNet and WordNet to yield a broad-coverage inventory of frames. Additionally, linguistic annotations in FrameNet such as the ones used to create the reification–dereification rules can also be used to generate natural language, for instance, for summarizing a portion of a KB for non-technical users.

In our future work we will continue our efforts to integrate arbitrary knowledge with frame structures by automatically generating integration rules such as the examples in section 2, for arbitrary knowledge bases. Given FrameBase's close connection to natural language, we also intend to study methods for better adapting semantic role labeling tools to question answering [44].

The state-of-the art FrameNet SRL system is Google-internal [46], but the CMU system is close [44]. Using FrameBase, would automatically benefit from the rapid advances in NLP.

Details and more information about FrameBase are available at `http://framebase.org`. FrameBase data is freely available under a Creative Commons Attribution 4.0 International license (CC-BY 4.0).

References

[1] C. Bizer, T. Heath, and T. Berners-Lee, "Linked data–the story so far," *IJSWIS*, vol. 5, no. 3, pp. 1–22, 2009.

[2] P. Hayes and P. Patel-Schneider, "RDF 1.1 semantics," W3C, Tech. Rep., 2014, http://www.w3.org/TR/rdf11-mt/.

[3] D. A. Ferrucci, E. W. Brown, J. Chu-Carroll, J. Fan, D. Gondek, A. Kalyanpur, A. Lally, J. W. Murdock, E. Nyberg, J. M. Prager, N. Schlaefer, and C. A. Welty, "Building Watson: An Overview of the DeepQA Project," *AI Magazine*, vol. 31, no. 3, 2010.

[4] A. Gangemi and V. Presutti, "A Multi-dimensional Comparison of Ontology Design Patterns for Representing n-ary Relations," ser. SOFSEM '13, P. Emde Boas, F. Groen, G. Italiano, J. Nawrocki, and H. Sack, Eds., 2013.

[5] V. Nguyen, O. Bodenreider, and A. Sheth, "Don't Like RDF Reification?: Making Statements About Statements Using Singleton Property," ser. WWW '14, 2014.

[6] "Roles in Schema.org," W3C Consortium, Tech. Rep., 2014, https://www.w3.org/wiki/WebSchemas/RolesPattern.

[7] C. Böhm, G. de Melo, F. Naumann, and G. Weikum, "LINDA: Distributed Web-of-data-scale Entity Matching," in *CIKM'12*, 2012, pp. 2104–2108.

[8] J. Rouces, "Enhancing Recall in Semantic Querying," ser. SCAI '13, vol. 257, 2013, p. 291.

[9] M. Yahya, K. Berberich, S. Elbassuoni, M. Ramanath, V. Tresp, and G. Weikum, "Natural language questions for the web of data," ser. EMNLP-CoNLL '12, 2012. [Online]. Available: http://www.aclweb.org/anthology/D12-1035

[10] J. Rouces, G. de Melo, and K. Hose, "FrameBase: Representing N-ary Relations Using Semantic Frames," in *ESWC'15*, 2015.

[11] ——, "Representing Specialized Events with FrameBase," in *DeRiVE'15*, 2015.

[12] L. Del Corro and R. Gemulla, "Clausie: Clause-based open information extraction," ser. WWW '13, 2013.

[13] F. M. Suchanek, G. Kasneci, and G. Weikum, "YAGO: A Large Ontology from Wikipedia and WordNet," *Web Semantics: Science, Services and Agents on the World Wide Web* , vol. 6, no. 3, pp. 203 – 217, 2008, <ce:title>World Wide Web Conference 2007Semantic Web Track</ce:title>.

[14] J. Hoffart, F. M. Suchanek, K. Berberich, and G. Weikum, "YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia," *Artificial Intelligence* , vol. 194, no. 0, pp. 28–61, 2013.

[15] F. M. Suchanek, J. Hoffart, E. Kuzey, and E. Lewis-Kelham, "YAGO2s: Modular High-Quality Information Extraction with an Application to Flight Planning," in *BTW*, 2013, pp. 515–518.

[16] W. Frawley, *Linguistic semantics*. Hillsdale, 1992.

[17] A. C. Schalley and D. Zaefferer, *Ontolinguistics: How Ontological Status Shapes the Linguistic Coding of Concepts*. Walter de Gruyter, 2007, vol. 176.

[18] D. Jurafsky and J. H. Martin, *Speech and Language Processing*, 2nd ed. Pearson Prentice Hall, 2009.

[19] N. Noy and A. Rector, "Defining N-ary Relations on the Semantic Web," W3C Consortium, W3C Working Group Note, April 2006, http://www.w3.org/TR/swbp-n-aryRelations/.

[20] R. Shaw, R. Troncy, and L. Hardman, "LODE: Linking Open Descriptions of Events," in *ASWC '09*, ser. Lecture Notes in Computer Science, 2009, pp. 153–167.

[21] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, "Freebase: A Collaboratively Created Graph Database for Structuring Human Knowledge," in *SIGMOD'08*, 2008, pp. 1247–1250.

[22] W. R. Van Hage, V. Malaisé, R. Segers, L. Hollink, and G. Schreiber, "Design and use of the Simple Event Model (SEM)," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 9, no. 2, pp. 128–136, 2011.

[23] C. J. Fillmore, C. R. Johnson, and M. R. Petruck, "Background to Framenet," *International Journal of Lexicography*, vol. 16, no. 3, pp. 235–250, 2003.

[24] J. Ruppenhofer, M. Ellsworth, M. R. Petruck, C. R. Johnson, and J. Scheffczyk, *FrameNet II: Extended Theory and Practice*. ICSI, 2006.

[25] A. Gangemi and V. Presutti, "Towards a pattern science for the semantic web," *Semantic Web*, vol. 1, no. 1, pp. 61–68, 2010.

[26] A. G. Nuzzolese, A. Gangemi, and V. Presutti, "Gathering lexical linked data and knowledge patterns from FrameNet," ser. K-CAP '11, 2011, pp. 41–48.

[27] V. Presutti, F. Draicchio, and A. Gangemi, "Knowledge Extraction Based on Discourse Representation Theory and Linguistic Frames," in *Knowledge Engineering and Knowledge Management*, ser. Lecture Notes in Computer Science, A. Teije, J. Völker, S. Handschuh, H. Stuckenschmidt, M. d'Acquin, A. Nikolov, N. Aussenac-Gilles, and N. Hernandez, Eds. Springer Berlin Heidelberg, 2012, vol. 7603, pp. 114–129.

[28] C. Fellbaum, *WordNet: An Electronic Lexical Database*. MIT Press, 1998.

[29] D. Gildea and D. Jurafsky, "Automatic labeling of semantic roles," *Computational linguistics*, vol. 28, no. 3, pp. 245–288, 2002.

[30] C. Subirats, "Spanish FrameNet: A frame-semantic analysis of the Spanish lexicon," in *Multilingual FrameNets in Computational Lexicography: Methods and Applications*. Mouton de Gruyter, 2009.

[31] P. Kingsbury and M. Palmer, "From TreeBank to PropBank." ser. LREC '02, 2002.

[32] K. Toutanova, D. Klein, C. D. Manning, and Y. Singer, "Feature-rich part-of-speech tagging with a cyclic dependency network," ser. HTL-NAACL '03, 2003.

[33] J. McCrae, D. Spohr, and P. Cimiano, "Linking lexical resources and ontologies on the semantic web with lemon," in *The semantic web: research and applications*. Springer Berlin Heidelberg, 2011, pp. 245–259.

[34] J. McCrae, C. Fellbaum, and P. Cimiano, "Publishing and Linking WordNet using lemon and RDF," in *Proceedings of the 3rd Workshop on Linked Data in Linguistics*, 2014.

[35] G. de Melo and G. Weikum, "Language as a foundation of the Semantic Web," in *Proceedings of the Poster and Demonstration Session at the 7th International Semantic Web Conference (ISWC 2008)*, ser. CEUR WS, C. Bizer and A. Joshi, Eds., vol. 401. Karlsruhe, Germany: CEUR, 2008.

[36] J. J. Carroll, I. Dickinson, C. Dollin, D. Reynolds, A. Seaborne, and K. Wilkinson, "Jena: Implementing the Semantic Web Recommendations," in *WWW'04*, 2004, pp. 74–83.

[37] D. Klein and C. D. Manning, "Accurate unlexicalized parsing," in *ACL'03*, 2003, pp. 423–430.

[38] S. Tonelli and D. Pighin, "New Features for FrameNet: WordNet Mapping," ser. CoNLL '09, 2009, pp. 219–227.

[39] J. Daudé, L. Padró, and G. Rigau, "Mapping wordnets using structural information." ser. ACL, 2000.

[40] E. Laparra, G. Rigau, and M. Cuadros, "Exploring the integration of WordNet and FrameNet," in *GWC'10*, 2010.

[41] O. Ferrández, M. Ellsworth, R. Munoz, and C. F. Baker, "Aligning FrameNet and WordNet based on Semantic Neighborhoods," ser. LREC '10, 2010.

[42] D. De Cao, D. Croce, and R. Basili, "Extensive Evaluation of a FrameNet-WordNet mapping resource." ser. LREC, 2010.

[43] C. Fellbaum and C. F. Baker, "Can WordNet and FrameNet be Made "Interoperable"?" ser. ICGL '08, 2008.

[44] D. Das, D. Chen, A. F. T. Martins, N. Schneider, and N. A. Smith, "Frame-Semantic Parsing," *Computational Linguistics*, vol. 40, no. 1, pp. 9–56, 2014.

[45] J. Hoffart, M. A. Yosef, I. Bordino, H. Fürstenau, M. Pinkal, M. Spaniol, B. Taneva, S. Thater, and G. Weikum, "Robust Disambiguation of Named Entities in Text," ser. EMNLP '11, 2011, pp. 782–792.

[46] K. M. Hermann, D. Das, J. Weston, and K. Ganchev, "Semantic Frame Identification with Distributed Word Representations," in *ACL'14*, Jun. 2014. [Online]. Available: http://research.google.com/pubs/pub42245.html

Paper E

Heuristics for Connecting Heterogeneous Knowledge via FrameBase

Jacobo Rouces, Gerard De Melo, Katja Hose

The paper has been accepted for publication in the
Proceedings of the 13th Extended Semantic Web Conference (ESWC), 2016

The layout has been revised.

Abstract

With recent advances in information extraction techniques, various large-scale knowledge bases covering a broad range of knowledge have become publicly available. As no single knowledge base covers all information, many applications require access to integrated knowledge from multiple knowledge bases. Achieving this, however, is challenging due to differences in knowledge representation. To address this problem, this paper proposes to use linguistic frames as a common representation and maps heterogeneous knowledge bases to the FrameBase schema, which is formed by a large inventory of these frames. We develop several methods to create complex mappings from external knowledge bases to this schema, using text similarity measures, machine learning, and different heuristics. We test them with different widely used large-scale knowledge bases, YAGO2s, Freebase and WikiData. The resulting integrated knowledge can then be queried in a homogeneous way.

1 Introduction

In the past decades, numerous large-scale knowledge bases (KBs) have become available and are now essential both in research and in the commercial world, e.g., for IBM's Jeopardy!-winning question answering system Watson [1] and for Google's Knowledge Graph-driven search results. The Web of Linked Data has grown to the point that the numerous different KBs that have been published can no longer easily be visualized in a single cloud image.

Since numerous stakeholders are publishing separate KBs focusing on different domains and sources, a given application often needs to combine knowledge from multiple KBs. Hence, there is a clear need for methods to *integrate* such knowledge. A substantial body of work has aimed to address this problem by automatically aligning individual entries across KBs, both at the schema level [2] and at the level of entity instances [3]. These methods often produce a list of binary links using properties such as `owl:sameAs`. Unfortunately, different KBs often model the world in quite distinct ways. Despite the adoption of standards such as the use of subject-predicate-object triples in RDF [4], the same piece of information can be represented in ways such that a one-to-one alignment is no longer possible.

Consider, for instance, a marriage between two people. The YAGO KB [5] captures this using a binary predicate (`isMarriedTo`) between two persons. The Freebase KB [6], in contrast, relies on a special entity called a mediator or Compound Value Type (CVT) to describe the marriage, as well as several subject-predicate-object triples to list properties of the marriage, such as involved people, time, location, etc. In cases like this, which are not uncommon, neither `owl:sameAs`, `rdfs:subClassOf`, `owl:equivalentProperty`, nor any other individual property or binary relation can fully express the complex n-ary relationships between these resources.

In this paper, we propose to address this problem by integrating heterogeneous data into the FrameBase schema [7], which consists of a large inventory of *frames* that homogeneously represent n-ary relations. Frame structures are used in linguistics to describe the meaning of a sentence as scenarios with multiple participants and properties filling specific semantic roles. A marriage frame involves two partners, a time and a place, among other things. This is similar to Freebase's CVTs. However, in contrast to the few hundreds of CVTs in Freebase, FrameBase uses a larger number of frames ($\sim$20,000) organized in a dense hierarchy [8].

While FrameBase offers a flexible system for representing knowledge from existing knowledge sources [7, 9], there has not been any research showing how to automatically or semi-automatically integrate heterogeneous knowledge under its schema. In this paper, we develop a generic algorithm to create complex integration rules from external KBs into this schema. These rules go beyond existing alignment mechanisms designed for binary mappings between elements of different KBs. In our experiments, we show results on three particularly heterogeneous sources: Freebase [6] and WikiData [10] are KBs with an especially large schema. YAGO2s [5], in contrast, uses only a small number of properties, but relies heavily on reification to describe phenomena such as time and locations.

2 Related Work

Connecting knowledge sources is a long-standing problem. At the level of individual records in databases, this has variously been addressed as record linkage, entity resolution, and data de-duplication [11]. In KBs, this roughly corresponds to the problems of ontology alignment, data linking [12], and instance matching [3].

For KBs, there has been substantial work on ontology alignment [2] to identify matching classes from different sources, and in some cases also instances and properties across different sources [13]. A closely related task is that of canonicalizing or reconciliating knowledge from open information extraction [14, 15], which focuses on aligning names of entities and predicates by clustering synonymous entries. To achieve this, the knowledge extracted from each text source has to be reconciled, sometimes using complex graph matching algorithms [15]. But as the same extraction tool is used for each text source, the resulting graphs are constructed in similar ways and therefore follow a common model. Hence, the applied techniques for reconciliation are different from the ones necessary to reconcile ontologies created by completely independent parties and tools.

Only very little work has considered scenarios in which the same type of ontological knowledge is modeled in entirely different ways. In these cases, alignment by means of binary properties such as equivalence or subsumption

is no longer sufficient, because a KB may not have a direct counterpart for an element of another KB. The EDOAL (Expressive and Declarative Ontology Alignment Language) format [16] has been proposed to express complex relationships between properties. It defines a way to describe complex correspondences but it does not address how to create them. Similarly, complex correspondence patterns between ontologies – or ontologies and databases – have been described and classified in an ontology [17]. However, this approach does not provide any method to create the correspondence patterns, neither fully nor semi-automatically. The iMAP tool [18] explores a search space of possible complex relationships between the values of entries in two databases, e.g., `room-price = room-rate * (1 + tax-rate)`, but these are limited to specific types of attribute combinations. The S-Match tool [19] uses formal ontological reasoning to prove possible matches between ontology classes, involving union and intersection operators, but does not address complex matching of properties beyond this. Ritze et al. [20] use a rule-based approach to detect specific kinds of complex alignment patterns between entries in small ontologies.

Unlike previous work, the approach presented in this paper does not focus on matching individual entities but provides techniques to match knowledge that can also be expressed with complex patterns involving multiple entities.

3 Frames for Data Integration

FrameBase [7] relies on the concept of linguistic frames as provided by FrameNet [8]. Such frames represent events or situations with characteristics denoted as Frame Elements (FEs). As FrameNet's original purpose is semantic annotation of natural language, many frames have associated Lexical Units (LUs), i.e., terms that, when appearing in a text, may evoke a frame, which may be connected via FEs to some other parts of the text.

FrameBase represents the information about "*John's 7-year marriage to Mary*" by creating an entity *e* that is an instance of FrameNet's `Personal_relationship` frame (or a more specific one for marriages, as we describe later on). Relevant FEs such as the marriage partners and the duration are then captured by adding triples with *e* as subject. For instance, properties `Partner_1` and `Partner_2` connect *e* to entities representing John and Mary, respectively, while the property `Duration` is used for the time their marriage lasted.

FrameBase thus repurposes FrameNet frames, originally intended to represent natural language semantics, for knowledge representation with subject-predicate-object triples, using what is also called *neo-Davidsonian representation*: One first introduces an entity *e* that is an instance of a frame class, and hence represents a particular event or situation. This entity is then connected to

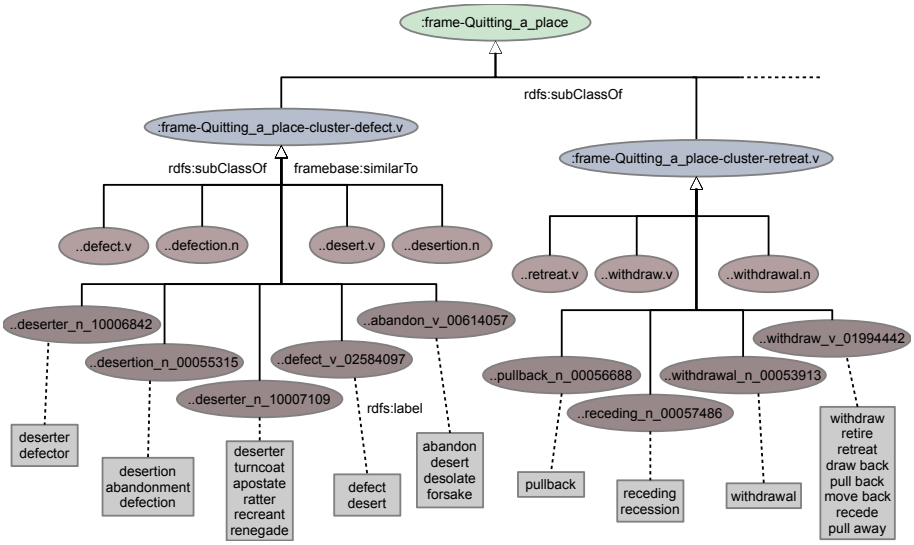


Fig. E.1: Example of a hierarchy with a macroframe `:frame-Quitting_a_place`, two cluster-microframes that are direct subclasses of the macroframe, and several LU- and synset-microframes that are direct subclasses of the cluster-microframe. All the microframes under a given synset-microframe are also connected via the symmetric property `framebase:similarTo` (for clarity, the transitive closure is omitted). The synset-microframes also have labels extracted from WordNet. The microframe identifiers have a shared prefix that has been abbreviated.

other entities (for example other frame instances, literals, or named entities) by means of properties representing the frame elements.

To adapt FrameNet for knowledge representation, FrameBase extends the inventory of frames defined by FrameNet in a hierarchy consisting of the following levels (Figure E.1):

- *Macroframes* are very coarse-grained and correspond to regular frames in FrameNet. The `Personal_relationship` frame class, for example, subsumes `spouse`, `marriage`, `girlfriend`, and `divorced`.
- *Microframes* inherit the general semantics and FE properties from their parent macroframes. They can be classified into 3 types:
  1. *LU-micoframes* are based on a frame's LUs and are represented as subframes in FrameNet, and therefore as subclasses in FrameBase. `Personal_relationship-married.a` and `Personal_relationship-divorced.a`, for example, are subclasses of `Personal_relationship`.
  2. *Synset-microframes* are created for synsets (sense-disambiguated synonymous words) in WordNet [21] that LUs can be mapped to. For instance, the two LU-microframes `Personal_relationship-suitor.n` and ... `Personal_relationship-court.v` are connected to each other by means of synset-microframes.

3. *Cluster-microframes* are created to cluster sets of LU- and synset-microframes with similar meaning. `Personal_relationship`, for instance, clusters (encoded as subclasses) `Personal_relationship-married.a`, `Personal_relationship-divorced.a`, `Personal_relationship-suitor.n`, and `Personal_relationship-court.v`.

To enable more efficient querying without involving frame instances, FrameBase also provides Direct Binary Predicates (DBPs) that directly connect pairs of FEs. For instance, the two partners involved in a marriage are directly connected by a triple with `marriedTo` as property. The schema provides both reification and dereification (ReDer) rules to convert knowledge between the two representations (frame and DBPs). Two example ReDer rules are presented in Figure E.2.

```
?S :...isSplitIntoParts ?O
↕
?R a :frame-Separating-split.v ,
?R :fe-Separating-Whole_1 ?S ,
?R :fe-Separating-Parts_2 ?O .

?S :dbp-Motion-movesInCarrier ?O
↕
?R a :frame-Motion-move.v ,
?R :fe-Motion-Theme_1 ?S ,
?R :fe-Motion-Carrier_2 ?O .
```

Fig. E.2: Two example ReDer rules. The direct binary predicate is the property in the dereified pattern, on the top. The reified pattern is at the bottom.

Overall, the FrameBase RDFS schema currently contains 19,376 frames, including 11,939 frames for specific lexical units and 6,418 frames for WordNet's sets of synonyms. In addition to ReDer rules, the schema uses efficient RDFS+ inference (RDFS extended with a transitive, symmetrical, and reciprocal property used to link elements of a cluster).

4 Knowledge Base Integration

We now outline our approach for integrating heterogeneous knowledge bases using the FrameBase schema. Although the techniques can be applied to a wide range of KBs, we focus in particular on YAGO2s [5], Freebase [6], and WikiData [10].

Our integration algorithm produces integration rules describing how to transform knowledge from a KB into FrameBase. These rules do not connect individual instances but are defined at the schema level and therefore resemble Global-As-View mappings in relational database systems [22]. Formally speaking, the produced integration rules can be expressed in first-order logic – with

Algorithm 1 FrameBase Integration Algorithm.

Require: K ▷ input knowledge base
$R \leftarrow \emptyset$ ▷ set of SPARQL CONSTRUCT rules
for all classes C in K **do** ▷ create class-frame rules
  for all frames $F \in \text{mappings}_{\text{C}-\text{F}}(C)$ **do**
    $M \leftarrow \emptyset$ ▷ property mappings
    for all properties P such that $\exists\, s, o : \langle s\ P\ o \rangle \in K$ **and** $s \in C$ **do**
      for all $E \in \text{mappings}_{\text{PF}-\text{E}}(P, F)$: E is not in R **do**
        $M \leftarrow M\ \cup (P, E)$
    $R \leftarrow R \cup \text{ClassFrameRule}(C, F, M)$
for all properties P in K **do** ▷ create core property-frame rules
  if the domain of P is not `rdf:Statement` **then**
    for all $(F, E_s, E_o) \in \text{mappings}_{\text{P}-\text{FEE}}(P)$ **do**
      $R \leftarrow R \cup \text{PropertyFrameRule}(P, F, E_s, E_o)$
for all properties P' in K **do** ▷ extend property-frame rules
  if the domain(P')=`rdf:Statement` **then**
    for all properties P in K satisfying $\langle$`P ^rdf:property/P' y`$\rangle$ **do**
      for all property-frame rules r in R **do**
        if r matches $\text{PropertyFrameRule}(P, F, E_s, E_o)$ **then**
          for all frame elements $E_{P'} \in \text{mappings}_{\text{PF}-\text{E}}(P', F)$ **do**
            $\text{Extend}(r, P', E_{P'})$
return R ▷ final set of integration rules

triples represented as 3-ary predicates (Figure E.3). Nevertheless, we implement these rules using SPARQL CONSTRUCT queries [23] because SPARQL is a widely supported standard for KBs available in RDF format. Non-RDF KBs can also be integrated by either using an alternative rule formalism or invoking off-the-shelf or custom-purpose RDF converters[1].

Algorithm 1 sketches our approach, which relies on three mapping functions that are discussed in Section 4.3 and three rule instantiation functions given in Figure E.3. The mapping functions relate entities from the source KB with entities from FrameBase into which they can be translated, but they do not provide the structure of the integration rules. The structure is specified by the instantiation functions, which take elements from the source KB and FrameBase, and return structured integration rules.

The instantiation functions are used to create two kinds of integration rules: (i) class-frame rules, which convert classes and properties from the original KB into similar elements in FrameBase (Section 4.1) and (ii) property-frame rules, which convert properties from the source KB into frames (Section 4.2).

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

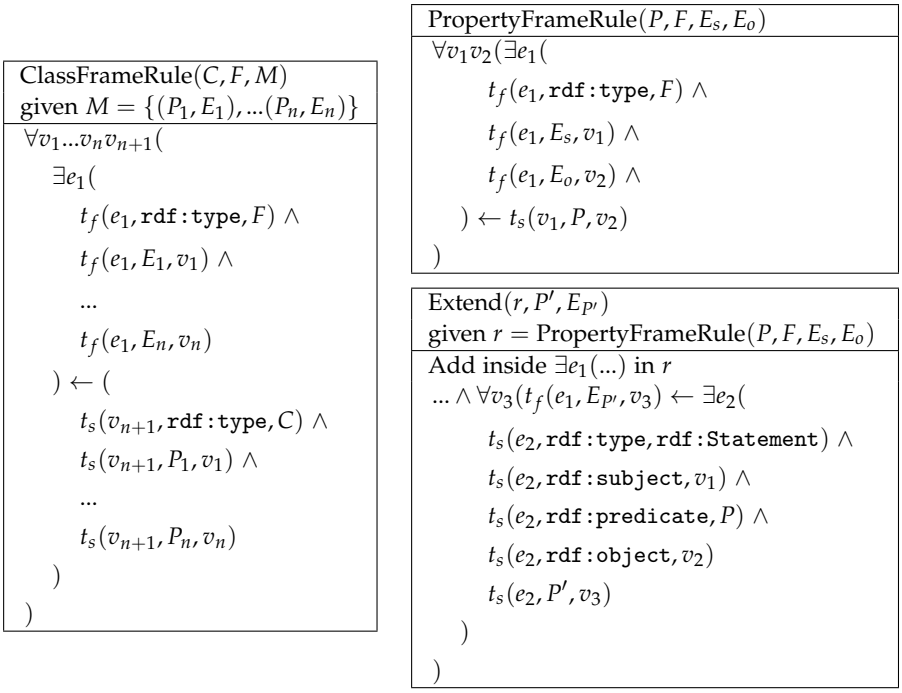


Fig. E.3: Instantiation functions for the integration rules used by Algorithm 1. $t_s(s, p, o)$ stands for a triple in a source KB and $t_f(s, p, o)$ for a triple in FrameBase. v_i and e_i are variables (universally and existentially quantified, respectively) over entities in the source KB.

4.1 Class-Frame Rules

The process of creating class-frame rules starts in line 2 in Algorithm 1, relying on mapping functions $\text{mappings}_{\text{C}-\text{F}}$ and $\text{mappings}_{\text{PF}-\text{E}}$. Class-frame rules are produced by the rule instantiation function ClassFrameRule(C, F, M) from Figure E.3. They convert a class C into a frame F that represents an event, situation or state of affairs, given $M = \{(P_1, E_1), ...(P_n, E_n)\}$ maping properties P_i for C to frame elements E_i of F. Figure E.4 provides an example of a class-frame rule automatically generated for integrating Freebase.

4.2 Property-Frame Rules

In general, the purpose of a property-frame rule is to translate a property in a source KB as an instance of a frame with at least two properties. These rules are built in two steps.

Creation of core property-frame rules. The process of creating core property-frame rules starts in line 9 in Algorithm 1, relying on the mapping function $\text{mappings}_{\text{P}-\text{FEE}}$. Core property-frame rules are produced by the instantiation function PropertyFrameRule(P, F, E_s, E_o) from Figure E.3. Each RDF triple in

```
CONSTRUCT {
  _:e a :frame-Win_prize-win.v    ; :fe-Win_prize-Time ?y
  ; :fe-Win_prize-Prize ?a        ; :fe-Win_prize-Competitor ?aw
  ; :fe-Win_prize-Explanation ?hf ; :fe-Win_prize-Competition ?c
  ; :fe-Win_prize-Rank ?al        ; :fe-Win_prize-Event_description ?ed .
} WHERE {
  ?m a fb:award.award_honor .
  OPTIONAL { ?m fb:award.award_honor.year ?y }
   ...honor.award ?a }                ...honor.award_winner ?aw }
   ...honor.honored_for ?hf }         ...honor.ceremony ?c }
   ...honor.achievement_level ?al }   ...honor.notes_description ?ed } }
```

Fig. E.4: Class-Frame rule, automatically generated rule for integrating Freebase.

the source KB matching pattern `?x` P `?y`, is transformed into a frame instance of type F with two frame-element properties E_s and E_o whose values are `?x` and `?y`, respectively. Figure E.5 provides an example of a core property-frame rule automatically generated for integrating Wikidata.

```
#SOURCE_PROPERTY_NAME='depicts'
#SOURCE_PROPERTY_DESCR='depicted person, place, object or event'
CONSTRUCT {
 _:r a :frame-Communicate_categorization-depict.v .
 _:r :fe-Communicate_categorization-Speaker ?S .
 _:r :fe-Communicate_categorization-Item ?O .
} WHERE { ?S <http://www.wikidata.org/entity/P180> ?O }
```

Fig. E.5: Property-Frame rule, automatically generated for integrating Wikidata.

Extending core property-frame rules to capture RDF reification. Additional clauses may be added by Algorithm 1 in the loop starting in line 13. This process relies on the mapping function $\text{mappings}_{\text{PF}-\text{E}}$. It uses the instantiation function $\text{Extend}(r, P', E)$ from Figure E.3, which takes a property-frame rule $r = \text{PropertyFrameRule}(P, F, E_s, E_o)$ as argument and returns an extended version of it to capture knowledge attached to triples by means of RDF reification [7]. KBs such as YAGO use this to represent n-ary relationships, but the FrameBase model is more efficient for this purpose. Figure E.6 provides an example of an extended property-frame rule generated for integrating YAGO.

4.3 Mapping Functions

The mapping functions use an automatic general technique meant to be used with big and dynamic source KBs, extended with heuristics that apply for common patterns across large source KBs or cover most small source KBs.

```
CONSTRUCT { _:event a :frame-Ride_vehicle-flight.n          core
  ; :fe-Ride_vehicle-Source ?s ; :fe-Ride_vehicle-Goal ?o   core
  ; :fe-Ride_vehicle-Vehicle ?objTransp .                   extension
} WHERE { ?s yago:isConnectedTo ?o .                        core
  OPTIONAL { ?sid rdf:type rdf:Statement .                  extension
    ?sid rdf:subject ?s ; rdf:object ?o                     extension
      ; rdf:predicate yago:isConnectedTo .                  extension
    OPTIONAL { ?sid yago:byTransport ?objTransp }}}         extension
```

Fig. E.6: Extended property-frame rule, generated for integrating YAGO.

P-FEE Mapping Function Given property P from the source KB, $\text{mappings}_{\text{P-FEE}}(P)$ returns 3-tuples of a frame F, and frame element properties E_s, E_o associated with P. Informally, it means that property `P` from the source KB can be substituted with a path $\hat{}E_s/E_o$ in FrameBase.

General Method. For the general variant of $\text{mappings}_{\text{P-FEE}}(P, F)$, we exploit the fact that the direct binary predicates built into FrameBase, which allow us to directly connect two frame elements, are directly mappable to external properties that should evoke a frame and two frame elements. Since the direct binary predicates were created with labels that follow the prevailing conventions in other LOD KBs [7], we can use a text similarity measure to find equivalent direct binary predicates, and for those found, use the frame and FEs in the associated reification rule. For example, if a property in a source KB is named "is split in", it turns out to be similar to the direct binary property "is split into parts" from the first example in Figure E.2, which can be used to create an integration rule that translates that source KB property into the reified pattern of FrameBase's ReDer rule.

To compare direct binary predicates with external ones, the text similarity we use is cosine distance of bag-of-words vectors. We split predicate names into tokens using capitalization, use proper lemmatization (with Stanford CoreNLP 3.6.0 [24]) instead of stemming, and do not filter stop-words, since in this case certain closed-set parts of speech such as prepositions are very important. The use of this measure significantly improved the results compared to using ADW [25], arguably because the latter is not tuned for our kind of text. Besides, our method was much faster.

For each external KB property, we run the similarity function against all existing DBPs in FrameBase, and we take the best candidate if it has a score higher than a threshold of 0.8. The threshold value was chosen empirically to balance precision and recall.

Additional Heuristics. Our system admits manually crafted heuristics to be added to $\text{mappings}_{\text{P-FEE}}(P, F)$. When one of the heuristics fire, they take preference over the general method. The vast majority of datasets in the Linked Open Data cloud rely on very small hand-crafted ontologies and vocabularies.

In this case, relying on the heuristics is particularly useful, because they can cover most of the elements of the source KB. In particular, we do this for YAGO2s, which is not a small ontology per se (it has a rather big class hierarchy and millions of instances) but uses just 77 different non-metadata properties. The heuristics can be expressed using an RDF ontology that is loaded by the system at startup.

PF-E Mapping Function Given a property P from the source KB and a frame F associated with P, $\text{mappings}_{\text{PF}-\text{E}}(P, F)$ returns frame element properties E with domain F, and associated with P. Informally, this means that property `P` from the source KB can be substituted with property `E` in FrameBase.

General Method. The implementation of $\text{mappings}_{\text{PF}-\text{E}}(P, F)$ computes the text similarity between the name of P concatenated with the name of its range, and the names of the FEs whose domain is F, using the ADW similarity measure [25]. It chooses the candidate with the maximum score for each FE. Note that our algorithm only considers these mappings in restricted settings, e.g. when a frame F has already been chosen. This greatly reduces the set of candidates in practice and enables this approach to deliver good results.

Additional Heuristics. To the general method, we add a heuristic that increases similarity to 1 if the following condition is met: $\text{endsWith}(\text{P}, \text{X}) \wedge \text{endsWith}(\text{FE}, \text{Y})$. The possible values required for X and Y can also be loaded from the heuristic ontology. For Freebase, we use the following two pairs: $(X, Y) \in \{(\texttt{from}, \texttt{time}), (\texttt{place}, \texttt{place})\}$. For YAGO, 4 pairs are required: $(X, Y) \in \{(\texttt{happenedIn}, \texttt{place}), (\texttt{happenedOnDate}, \texttt{time}), (\texttt{endedOnDate}, \texttt{time}), (\texttt{startedOnDate}, \texttt{time})\}$.

C-F Mapping Function Given a class C from the source KB, $\text{mappings}_{\text{C}-\text{F}}(C)$ returns frames F associated with C. Informally, this means that class C from the source KB can be substituted with class F in FrameBase.

General Method. We let $F(C)$ denote a candidate set of relevant frames F. In order to filter out noisy and incomplete parts of the source KB, $\text{mappings}_{\text{C}-\text{F}}(C)$ returns $\emptyset$ for classes from the origin KB that do not have at least 10 instances and at least 3 outgoing properties with text annotations. Otherwise, $F(C)$ is defined to include all LU-microframes with non-zero lexical overlap (some word in common in the text labels) between C's name and the set of text labels for the synonymous frames from the cluster that F belongs to. Clusters of synonymous frames are formed by LU-microframes that are deemed equivalent via links through synset-microframes. To disambiguate and choose the best frame F among all candidates $F(C)$, we train (and later test, c.f. Section 5.1) logit and SVM classifiers over (C, F) pairs of this form, taken from a gold standard. The (C, F) pairs are considered true when there is a class-frame rule

in the gold standard with C in the antecedent and F in the consequent, and false otherwise. Then, for each source KB item, we choose the frame whose pair has the highest score. Although this entails an implicit assumption of functionality, in practice, this results in very significant gains in precision. As input to the model, we use the following four features:

1. The lexical overlap between (i) C's name and (ii) the lexical labels of the cluster of synonymous LU-microframes for F.

2. The lexical overlap between (i) the syntactic head of C's name determined iteratively using the Collins Algorithm [26] and (ii) the lexical labels of the cluster of synonymous LU-microframes for f.

3. The lexical overlap between the descriptions (the longer text labels sometimes identified as comments).

4. If C is a class, the lexical overlap between the union of labels and descriptions of the outgoing properties, upweighting the labels by a factor of 10. When available, the labels and descriptions of the ranges are added too.

For all features, we lemmatize and filter out stop words (closed word classes) and use TF-IDF to compute the feature values (although the second feature is boolean in practice).

In Section 5.1, we test this method using a gold standard manually created for Freebase [6], which is a typical case of a large, open-ended schema, where a fully automatic approach becomes more necessary.

Additional Heuristics. A high-accuracy heuristic can be applied for those source KBs that are linked to WordNet, leveraging that FrameBase includes a significant part of WordNet synset as synset-microframes, which are linked to FrameNet-based LU-microframes.

The heuristic works as follows. If a given source KB class C is associated with a WordNet synset, the synset-microframe based on this synset is looked up in FrameBase. If found, this is the match, and if it is not found, a class C' is selected that is the next most specific WordNet-based parent of C. That is, $C' \supset C \wedge (C'' \supset C \rightarrow C'' = C')$. Now a synset-microframe is searched for C'. If it is not found, the process is repeated until a match is found, or a maximum number of steps is reached (e.g., 6), in order to avoid overly general rules. With this method, a sound rule can still be created, even if it loses some specificity, and it accounts for the fact that not all synsets are mapped in FrameBase.

This heuristic is particularly relevant for YAGO2s, whose upper class hierarchy is based on WordNet nouns, which makes the mapping obvious. However, it also applies to any other KB for which a mapping to WordNet exists, even if this is an external or a-posteriori one. Since WordNet is a very commonly used linguistic resource, this is reasonably common in LOD KBs.

5 Evaluation

5.1 Integration Rules Created

In this section we present examples of creating integration rules with Algorithm 1 for the test cases of YAGO2s, Freebase (2014-09-21 version), and WikiData (2015-09-28 version).

Creation of Class-Frame Integration Rules

Freebase. To evaluate the results of this method on an arbitrary KB, we produced a manual gold standard consisting of 31 classes and 141 external properties from Freebase [6], paired with their candidate frames or FE properties, respectively. The gold standard is available at `http://framebase.org/data`. Using two independent annotators, we obtained a Cohen's kappa (inter-annotator agreement) $k = 0.69$ for class to macroframe mappings, and $k = 0.38$ for property to frame element mappings. The second is lower because it accumulates the errors from the first, which illustrates how difficult it is to create a gold standard for structured knowledge integration. The classes were randomly chosen from Freebase, disregarding classes whose candidate set did not include a valid match in FrameBase. Freebase was chosen for testing this method because it features Compound Value Types (CVTs), which have a similar role to frames, but we are also able to map some non-CVT classes. Out of a total of 155 outgoing properties for the randomly chosen Freebase classes, 141 could successfully manually be matched to frame elements in the gold standard.

Table E.1: Evaluation of mapping external classes to FrameBase classes.

	B-1	B-2	Our Method	
			Logit	SVM
Recall	0.21	0.60	0.50	0.77
Precision	0.12	0.15	0.88	0.77
F1	0.15	0.24	0.63	0.77

Table E.2: Evaluation of mapping external properties to FrameBase properties.

Metric	Score
Precision	0.81
Recall	0.30
Accuracy	0.36

Table E.1 shows the results for automatic class mappings, averaging over 10 random training/test partitions of ratio 2:1. We compare three different methods.

- Baseline 1 (B-1) takes the frame class with maximum lexical overlap in names (as in feature 1 of our method) and for which the candidate set $F(x)$ consists of all FrameBase classes (which is a sort of metric that can be configured with the Link Specification language in Silk [12], a state-of-the-art ontology alignment system). However ontology alignment systems alone cannot produce complex mappings.

- Baseline 2 (B-2) uses the same measure as above, but applying the candidate set $F(C)$ chosen in our method, described in Section 4.3.
- Our method described in Section 4.3, using a logistic regression (logit) classifier and the functional assumption in conjunction with a fixed acceptance threshold of $p > 0.5$, where p is the probability obtained from the logit.
- Our method described in Section 4.3, using a support vector machine (SVM) with radial kernel, selecting, for each source KB class, the candidate frame whose score is highest, given by the distance to the frontier (functional assumption).

Table E.2 provides the results obtained by our method for properties, averaging over 10 random training/test partitions of the ground truth data, each of ratio 2:1. Precision and recall are calculated with respect to the gold standard. We obtain higher precision with the logit method because we use the output probabilities to apply a condition that filters out false positives at the cost of a lower recall. Both classifier-based methods outperform the baseline.

Note that in general, word sense disambiguation is considered a hard and yet unsolved problem in natural language processing. This is particular relevant when matching properties that come with little or no metadata. For example, the Freebase classes *education.academic_post* and *base.banned.exiled* must be mapped to the *Employing* and *Residence-reside.v* frames, respectively, for which there is no obvious lexical connection. The same applies when mapping, for example, Freebase properties *education.academic_post.institution* and *geography.river.length* to frame elements *Employing-Employer* and *Natural_features-Descriptor*, respectively. A complete high-precision integration of Freebase into another knowledge base thus requires a larger community effort with additional manual revisions. Our system can be used to automatically propose suggestions to speed up this process.

YAGO. 450 class-frame integration rules were automatically created for YAGO2s. The results are given in Table E.3. It shows how the number of matches decreases as n increases and the WordNet-based heuristic for $\text{mappings}_{\text{C-F}}(C)$ moves up the WordNet hierarchy. For $n > 6$ the results are negligible. The ratio of correctly matched entities is 0.789, which is equivalent to the precision of the WordNet-FrameNet mapping used for creating the schema [7] – via clustering of near-equivalent microframes, which uses other links in FrameNet and WordNet that are annotated by experts and therefore expected to be nearly error-free. Figure E.7 provides an example of a class-frame integration rule created for YAGO2s.

Property-Frame Integration Rules

State-of-the-art ontology alignment systems cannot produce something comparable to property-frame integration rules because the binary links produced

```
CONSTRUCT { ?s a :frame-Change_of_leadership-revolt.n ;
            :fe-Change_of_leadership-Place ?o .
} WHERE {   ?s a/rdfs:subClassOf yago:wordnet_rebellion_100962129 .
            OPTIONAL { ?s yago:happenedIn ?o }  }
```

Fig. E.7: Class-Frame rule, automatically generated for integrating YAGO2s.

Table E.3: Number of created class-frame rules for YAGO2s. Matches(n) denotes the number of matches obtained for n being the maximum number of generalization steps. For each column, the left side shows the number of created rules and the right side the number of triples in YAGO2s matching these rules. `endedOnDate` has no significant occurrence in YAGO2s and was therefore omitted.

	`happenedIn`		`happenedOnDate`		`startedOnDate`	
	Rules	Triples	Rules	Triples	Rules	Triples
Matches(0)	38	11,149	86	16,836	4	13
Matches(1)	25	944	83	3,579	5	5
Matches(2)	24	469	58	14,329	1	1
Matches(3)	15	1,232	39	2,315	1	2
Matches(4)	9	540	30	986	0	0
Matches(5)	5	42	14	121	0	0
Matches(6)	2	2	11	39	0	0
All matches	118	14,378	321	38,205	11	21
No match	42	633	148	13,195	0	0
Total	160	15,011	469	51,400	11	21
% Match	73%	95%	68%	74%	100%	100%

by these systems (equality, subsumption, etc.) cannot reflect the complex 4-ary nature of property-frame integration rules.

WikiData. We use the general method to automatically extract property-frame rules from WikiData. We evaluate it on YAGO2s, as we can re-use the manually created FrameBase mappings for YAGO2s (described below) as a ground truth. Evaluating the results directly, we obtain a precision of 0.80, and using the YAGO integration rules as ground truth we obtain a recall of 0.21. Figure E.5 shows an example of a rule extracted from WikiData.

YAGO. Using the RDF ontology with manually specified heuristics mentioned in Section 4.3, 62 out of the 77 non-metadata properties in YAGO2s (i.e., 81%) could be perfectly integrated into FrameBase using simple property-frame rules.

6 Conclusion

In this paper, we have shown that knowledge base heterogeneity is a problem that goes beyond just the use of different identifiers that need to be aligned.

We provide a general analysis of declarative constructs – integration rules – that can also achieve kinds of mappings other than basic entity alignments. We further show that FrameBase is able to incorporate multiple broad-coverage knowledge sources, despite their structural heterogeneity, opening up the possibility for it to serve as a hub for semantic integration of other KBs.

We also provide practical methods to produce these rules, combining general methods with heuristics. The quality of the output is certainly not perfect, but while traditional ontology alignment is already a difficult task, complex mappings have combinatorially more possible candidates and are thus much harder. Our results constitute a first step towards a more comprehensive linking of knowledge.

The total size of the instance data obtained from these source KBs is 40,411,393 statements, which renders it the largest collection of facts linked to FrameNet.

All FrameBase data (schema, ReDer rules, integration rules, instance data, and gold standards) is published under a Creative Commons CC–BY 4.0 International license at `http://framebase.org`.

References

[1] A. Kalyanpur *et al.*, "Structured data and inference in DeepQA," *IBM Journal of Research and Development*, vol. 56, no. 3.4, pp. 10:1–10:14, 2012.

[2] J. Euzenat and P. Shvaiko, *Ontology Matching*. Springer, 2007.

[3] Z. Dragisic *et al.*, "Results of the Ontology Alignment Evaluation Initiative 2014," in *OM'14*, 2014, pp. 61–104.

[4] P. Hayes, "RDF Semantics," W3C Consortium, Tech. Rep., 2004, http://www.w3.org/TR/2004/REC-rdf-mt-20040210/.

[5] F. M. Suchanek, J. Hoffart, E. Kuzey, and E. Lewis-Kelham, "YAGO2s: Modular High-Quality Information Extraction with an Application to Flight Planning," in *BTW*, 2013, pp. 515–518.

[6] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, "Freebase: A Collaboratively Created Graph Database for Structuring Human Knowledge," in *SIGMOD'08*, 2008, pp. 1247–1250.

[7] J. Rouces, G. de Melo, and K. Hose, "FrameBase: Representing N-ary Relations Using Semantic Frames," in *ESWC'15*, 2015.

[8] C. J. Fillmore, C. R. Johnson, and M. R. Petruck, "Background to Framenet," *International Journal of Lexicography*, vol. 16, no. 3, pp. 235–250, 2003.

[9] J. Rouces, G. de Melo, and K. Hose, "Representing Specialized Events with FrameBase," in *DeRiVE'15*, 2015.

[10] F. Erxleben *et al.*, "Introducing wikidata to the linked data web," in *The Semantic Web – ISWC 2014*. Springer International Publishing, 2014, pp. 50–65.

[11] A. K. Elmagarmid, P. G. Ipeirotis, and V. S. Verykios, "Duplicate Record Detection: A Survey," *IEEE TKDE*, vol. 19, no. 1, pp. 1–16, 2007.

[12] J. Volz, C. Bizer, M. Gaedke, and G. Kobilarov, "Discovering and Maintaining Links on the Web of Data," in *ISWC'09*, 2009.

[13] F. M. Suchanek, S. Abiteboul, and P. Senellart, "PARIS: Probabilistic Alignment of Relations, Instances, and Schema," *PVLDB*, vol. 5, no. 3, pp. 157–168, 2011.

[14] L. Galárraga, G. Heitz, K. Murphy, and F. M. Suchanek, "Canonicalizing Open Knowledge Bases," in *CIKM'14*, 2014, pp. 1679–1688.

[15] M. Mongiovì, D. R. Recupero, A. Gangemi, V. Presutti, A. G. Nuzzolese, and S. Consoli, "Semantic Reconciliation of Knowledge Extracted from Text Through a Novel Machine Reader," in *K-CAP 2015*, 2015, pp. 25:1–25:4.

[16] J. David, J. Euzenat, F. Scharffe, and C. Trojahn dos Santos, "The Alignment API 4.0," *Semantic Web Journal*, vol. 2, no. 1, pp. 3–10, 2011.

[17] F. Scharffe and D. Fensel, "Correspondence patterns for ontology alignment," in *Proceedings of the 16th International Conference on Knowledge Engineering: Practice and Patterns*, ser. EKAW '08. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 83–92.

[18] R. Dhamankar, Y. Lee, A. Doan, A. Halevy, and P. Domingos, "iMAP: Discovering Complex Semantic Matches Between Database Schemas," in *SIGMOD 2004*, 2004, pp. 383–394.

[19] F. Giunchiglia, P. Shvaiko, and M. Yatskevich, "S-Match: an Algorithm and an Implementation of Semantic Matching," in *The Semantic Web: Research and Applications*. Springer, 2004, pp. 61–75.

[20] D. Ritze, C. Meilicke, O. Sváb-Zamazal, and H. Stuckenschmidt, "A Pattern-based Ontology Matching Approach for Detecting Complex Correspondences," in *OM'10*, 2008.

[21] C. Fellbaum, *WordNet: An Electronic Lexical Database*. MIT Press, 1998.

[22] A. Doan, A. Y. Halevy, and Z. G. Ives, *Principles of Data Integration*. Morgan Kaufmann, 2012. [Online]. Available: http://research.cs.wisc.edu/dibook/

[23] S. Harris and A. Seaborne, "SPARQL 1.1 Query Language," W3C Consortium, W3C Recommendation, Mar. 2013.

[24] D. Klein and C. D. Manning, "Accurate unlexicalized parsing," in *ACL'03*, 2003, pp. 423–430.

[25] M. T. Pilehvar, D. Jurgens, and R. Navigli, "Align, Disambiguate and Walk: A Unified Approach for Measuring Semantic Similarity," in *ACL'13*, 2013, pp. 1341–1351.

[26] M. Collins, "Head-Driven Statistical Models for Natural Language Parsing," *Computational Linguistics*, vol. 29, no. 4, pp. 589–637, 2003.

Paper F

Complex Schema Mapping and Linking Data: Beyond Binary Predicates

Jacobo Rouces, Gerard De Melo, Katja Hose

The paper has been published in the
Proceedings of the 9th Proc. Workshop on Linked Data on the Web (LDOW), 2016

The layout has been revised.

Abstract

Currently, datasets in the Linked Open Data (LOD) cloud are mostly connected by properties such as `owl:sameAs`*,* `rdfs:subClassOf`*, or* `owl:equivalentProperty`*. These properties either link pairs of entities that are equivalent or express some other binary relationship such as subsumption. In many cases, however, this is not sufficient to link all types of equivalent knowledge. Often, a relationship exists between an entity in one dataset and what is represented by a complex pattern in another, or between two complex patterns. In this paper, we present a method for linking datasets that is expressive enough to support these cases. It consists of integration rules between arbitrary datasets and a mediated schema. We also present and evaluate a method to create these integration rules automatically.*

1 Introduction

The most common way in which datasets in the Linked Open Data (LOD) cloud are currently connected to each other is by means of triples expressing simple one-to-one relationships. The most well-established property is `owl:sameAs`, which indicates equivalence between entities. Others, such as `owl:equivalentClass` and `owl:equivalentProperty` describe equivalences for classes and properties. Previous work has exposed widespread cases of misuse of the `owl:sameAs` property and proposed alternatives expressing different forms of near-identity [1, 2]. Additionally, properties such as `rdfs:subClassOf` and `rdfs:subPropertyOf` denote subsumption between classes and properties. Still, all of these properties have in common that they are binary predicates. Thus, they always link two individual items.

Knowledge in different datasets can, however, be related in other ways than via a direct one-to-one relationship between a pair of entities. Often, a relation may exist between an entity in one dataset and what is captured using a complex pattern in another, or between two complex patterns. For example, when using binary predicates, an example class such as `BirthEvent` in one dataset cannot simply be linked to a property such as `bornOnDate` or `bornInPlace` from another schema or dataset. Yet, these two sources clearly capture the same sort of knowledge, especially if the class `BirthEvent` is the domain of properties such as `personBorn`, `date`, and `place`. The first-order logic expression in Figure F.1 formalizes one of these more complex relations. Even more complex patterns are possible, as illustrated in Figure F.2. In this example, the complex pattern covers more information than captured by the simpler pattern using the property `stopConstructionOf`.

In contrast to binary relationships, related work has only paid minimal attention to complex patterns. The EDOAL format [3] and an ontology of correspondence patterns [4] have been created as a way to express and categorize complex correspondences between ontologies. These methods, however, do

not address the actual integration, i.e., the method to establish these relationships. The iMAP system [5] explores a space of possible complex relationships between the values of entries in two relational databases, for instance `address = concat(city,state)`. Ritze et al. [6] use a rule-based approach for detecting specific kinds of complex alignment patterns between entities in small ontologies.

In this paper, we generalize to a more general method for linking datasets through a mediated schema, that can support cases such as the above-mentioned examples. In addition, we propose an automatic approach to create some complex integration rules, which can be combined with existing 1-to-1 links. We use a mediated schema as a hub because the resulting star topology reduces the complexity of the overall linking from quadratic to linear with respect to the number of datasets. More specifically, we use FrameBase [7–9], a rich schema based on linguistic frames, as the mediated schema, because it is highly expressive and possesses the metadata and structures that enable automatic creation of mappings. Additionally, it has a strong connection to natural language.

As SPARQL has become a common standard with the necessary expressiveness to support logical rules, we implement schema mappings and integration rules as SPARQL construct queries (Figures F.1 and F.2). However, the system can easily be adapted to other formalisms and implementations.

$$
\begin{aligned}
&\forall v_1 v_2 (\\
&\quad \exists e_1 (\\
&\qquad t_f(e_1, a, BirthEvent) \wedge \\
&\qquad t_f(e_1, subject, v_1) \wedge \\
&\qquad t_f(e_1, date, v_2) \wedge \\
&\quad) \\
&\quad \leftrightarrow \\
&\quad t_s(v_1, bornOnDate, v_2) \\
&)
\end{aligned}
$$

Fig. F.1: Complex relation between two schemas, expressed in first-order logic

2 Complex Integration Rules

In order to connect complex patterns across different data sources, we develop an automatic method to produce integration rules that convert information

$$
\begin{aligned}
&\forall v_1 v_2 \ (\\
&\quad \exists e_1 \ (\\
&\qquad t_f(e_1, a, Construction) \ \wedge \\
&\qquad t_f(e_1, createdEntity, v_1) \ \wedge \\
&\qquad t_f(e_2, a, StopProcess) \ \wedge \\
&\qquad t_f(e_2, cause, v_2) \ \wedge \\
&\quad) \\
&\quad \leftrightarrow \\
&\quad t_s(v_2, stopsConstructionOf, v_1) \\
&)
\end{aligned}
$$

Fig. F.2: Very complex relation between two schemas, expressed in first-order logic.

from arbitrary sources to information expressed using the FrameBase schema. This method consists of three operations.

1. **Creating Candidate Properties in FrameBase:** We first identify complex patterns within FrameBase that might match properties from other sources. For each of these complex patterns, we define a new candidate property as a shorthand form, with a concise human-readable text label. All of this is done automatically by exploiting the linguistic annotations available in FrameBase. The result is a large set of matching candidates in FrameBase.

2. **Processing Candidate Properties in the Source Datasets:** We canonicalize properties in the source datasets by extending their names.

3. **Matching:** We match the (refined) names of the properties from the source dataset with the names of the binary candidate properties prepared for FrameBase. When a sufficiently high match is encountered, we produce an integration rule that connects the source property with the complex triple pattern.

2.1 Creating Candidate Properties in FrameBase

The first step is to identify complex patterns in FrameBase to which other data sources could potentially be matched. For each of these complex patterns, we define a simple new binary candidate properties that serves as a shorthand form between two variables present in the complex pattern. In FrameBase terminology, these new properties are called Direct Binary Predicates (DBPs) and their relationship to the original complex patterns in FrameBase is expressed via Reification–Dereification (ReDer) rules [7]. As a result, we obtain a large

candidate set of binary predicates to which properties from other datasets can be matched.

In order to detect relevant complex patterns in FrameBase, we exploit its ties to natural language.

Binary Predicates Based on Verbs and Nouns For relationships typically expressed using verbs, we have already equipped FrameBase with a set of direct binary predicates and corresponding reification-dereification rules [7]. Figure F.3 illustrates the structure of such a ReDer rule, while Figure F.4 provides an example of a DBP with the verb "destroyed" as the syntactic head.

```
?s  <DIRECT_BINARY_PREDICATE>  ?o
↕
<FRAME_INSTANCE>  a  <FRAME_CLASS> ,
<FRAME_INSTANCE>  <FRAME_ELEMENT-S>  ?s ,
<FRAME_INSTANCE>  <FRAME_ELEMENT-O>  ?o .
```

Fig. F.3: The general pattern of a simple dereification rule

```
?s dbp-Destroying-destroyedByMeans ?o
↕
f type frame-Destroying-destroy.v ,
f fe-Destroying-Undergoer ?s ,
f fe-Destroying-Means ?o .
```

Fig. F.4: Example of a simple dereification rule

Our previous work has already produced some DBPs based on verbs [7] (such as in the example in Figure F.4), and nouns [9], such as `isCauseOfEffect`.

However, in all of these cases, the reified side of the Reification–Dereification rules (the one at the bottom in the examples) was restricted to a specific pattern using three lines, such as the ones in Figures F.3 and F.4.

Binary Predicates Based on Adjectives To these noun and verb-based DBPs, we now add new binary predicates based on adjectives, using a generalization to more complex patterns such as the one illustrated in Figure F.2. One can view these as *very complex patterns*, as they are more involved than the ones considered previously in Figure F.3.

FrameNet [10], a database of frames used to annotate the semantics of natural language, forms the backbone of frames in FrameBase. In FrameNet, different frames represent different kinds of events or situations with participants, called Frame Elements (FEs). Frames also have Lexical Units (LUs), which are words and terms that are associated to that frame, and may evoke that frame when appear in a text. Example texts are *semantically parsed* by annotating places where a LU is evoking a frame, and neighboring words or phrases are the values of some of the FEs belonging to that frame. FrameBase

Creation Rule: Copula+Adjective
Create DBP with name "is LU PREP FE-o" *if*
IsADJECTIVE(LU) AND phrase-type-o==PP[PREP] AND grammatical-function-s==Ext AND grammatical-function-o==Dep

Fig. F.5: Method for creating new adjective-based DBPs

```
?s dbp-Sound_level-isLoudToDegree ?o
↕
f type frame-Sound_level-loud.a ,
f fe-Sound_level-Entity ?s ,
f fe-Sound_level-Degree ?o .
```

Fig. F.6: Example of an adjective-based dereification rule using the copula "to be"

represents frames and LUs as classes that can be instantiated and FEs as properties whose frame is their domain.

Figure F.5 summarizes how to create adjective-based DBPs using FrameNet's example annotations of English sentences. We define two target types of FEs: FE-s, the FE that should connect to the subject of the triple whose property is the DBP; and FE-o, the FE that should connect to the object. Figure F.3 shows how these two FEs are used in a ReDer rule. For an adjective in a sentence that is annotated with a frame, we need to check whether the text annotation also contains two FEs that fulfill the following conditions. First, the phrase type of FE-o needs to be a prepositional phrase (PP) with preposition PREP. Second, the grammatical function of FE-s needs to be that of a subject (Ext). And third, the grammatical function of FE-o needs to be that of a dependent (Dep). Figure F.6 presents an example of a DBP created with this method.

Although most occurrences of adjectives in the FrameNet annotations involve the verb "to be", pseudo-copulas "to seem" and "to become" can also be combined with any adjective. Therefore, we generate all possible DBPs with these three copulas for all adjectives. For "to be" there are no additional semantics (Figure F.6). The pseudo-copulas, however, carry additional semantics, which are expressed in a more complex pattern with an additional frame instance (Figures F.7 and F.8). Figure F.7 presents an example using the `Becoming` frame and the FE `fe-Becoming-Final_state` instead of `fe-Becoming-Final_category` (in FrameNet the former is used with adjectives and adjective phrases, while the latter is used with nouns and noun phrases). Figure F.8 shows an example for the `Appearance` frame.

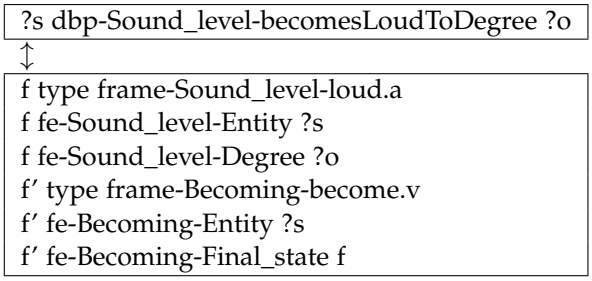


Fig. F.7: Example of an adjective-based dereification rule using the pseudo-copula "to become"

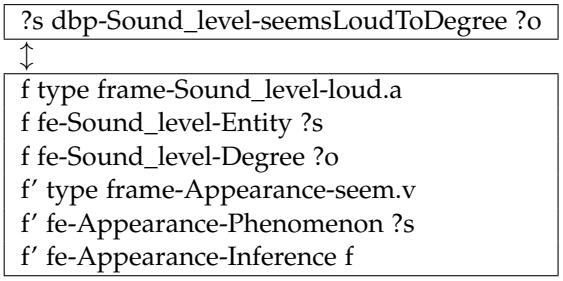


Fig. F.8: Example of an adjective-based dereification rule using the pseudo-copula "to seem"

2.2 Processing Candidate Properties in the Source Datasets

Having prepared a set of candidate predicates, each standing for a complex pattern in FrameBase, we now turn to the source datasets that we wish to connect. For a given source dataset, we process all its properties. Property names are often composed of a single word that is highly polysemous. This is particularly true when the verbs "to be" or "to have" are omitted, which unfortunately is very often the case. For example, many datasets use property names such as `address` instead of `hasAddress`, or `father` instead of `isFatherOf`.

Our approach consists of the following six steps.

1 If the name p of a property is a past participle, it can be extended with the prefix "is" (without postfix "of").

2 If the name p of a property is a noun or a noun phrase, and a range is declared for the property, let X be a set containing p's name and the hypernyms of all its word senses (obtained from WordNet [11]). If for any element x in X, p is a substring of x or x is a substring of p, then p can be extended with the prefix "has".

3 The same rule as above, but using the domain instead of the range, which allows p to be extended with the prefix "is" and postfix "of".

4 If the property is symmetric, we can introduce extensions both with "has" and with "is"+ . . . +"of".

5 For every property p corresponding to the pattern "is X of", an inverse property can be created of the form "has X".

6 For every property p corresponding to the pattern "has X", an inverse property can be created of the form "is X of".

This process resembles a sort of canonicalization of entity names [12], but in our case for properties. Note that steps 4–5 can also be carried out on the DBPs with identical results.

This canonicalization has independent value beyond its use for matching as in this paper; especially when the canonicalization, as in our case, does not merely make the names conform to a given pattern but also less ambiguous as well as easier to understand by humans.

2.3 Matching

The final step is to match properties across datasets. We focus on creating matches between direct binary predicates in FrameBase and the refined property names of other sources.

In order to find matches, we use bag-of-words cosine similarity measures that are optimized for the task at hand. We tokenize the names, but do not use stemming, since we want to increase specificity. We also do not perform stopword removal, because, unlike in the typical use case of matching large documents, common words such as prepositions can be relevant in this context (consider "run for" versus "run against").

Each source dataset property is compared to each DBP using a weighted combination of measures.

$$w_1 \cos(v_1^{\text{SDP}}, v_1^{\text{DBP}}) + w_2 \cos(v_2^{\text{SDP}}, v_2^{\text{DBP}}) + w_3 c_1 + w_4 c_2$$

- $\cos(v_1^{\text{SDP}}, v_1^{\text{DBP}})$ is the cosine between the vector for the name of the source dataset property v_1^{SDP} and the vector for the DBP's name v_1^{DBP}. For DBPs, we remove the "frame-element-object (FE-o)" name [7] because these do not occur very frequently. For instance, "original" is ommitted for "is copy of original".

- $\cos(v_2^{\text{SDP}}, v_2^{\text{DBP}})$ is the cosine between vectors with additional terms describing the properties' semantics. v_2^{SDP} includes terms from the name of the property, plus from the domain and the range if available. v_2^{DBP} includes the terms from the DBP's name, plus the FE-o, the FE-s, and the name and description of the associated frame as well as all its superframes.

- c_1 has value 1 if the frame element FE-o is classified as *Core FE* if FrameNet, which means that it instantiates a conceptually necessary component of a frame. These kinds of frames are more likely to appear. The value is also 1 if the FE is about Time or Place, because this information is also frequent in datasets.
- c_2 is the same for FE-s.

The DBP with the highest score is chosen, if this is higher than a threshold T. The vector of weights w is set to $w = (0.7, 0.1, 0.1, 0.1)$ so that the three last elements favor the closest match whenever there is a tie for $\cos(v_1^{\text{SDP}}, v_1^{\text{DBP}})$, which can happen between two DBPs that only differ by the FE-o name. $\cos(v_2^{\text{SDP}}, v_2^{\text{DBP}})$ is computationally more heavy and, for reasons of efficiency, it is only evaluated when $\cos(v_1^{\text{SDP}}, v_1^{\text{DBP}})$ is higher than Tw_1. The value of the global threshold is set at $T = w_1$ so $\cos(v_1^{\text{SDP}}, v_1^{\text{DBP}}) = 1$ is enough to fire a rule.

3 Results

We test our method on DBpedia [13]. We canonicalized 1,608 DBpedia properties and evaluated a random sample of 40, out of which 32 turned out to be correct. Of the 8 that were incorrect, 2 were also incorrect in their original DBpedia form, resulting in a true precision of 85%. Some examples are presented in Table F.1.

Table F.1: Example canonicalized properties.

source property IRI	
source property name	**canonicalization**
`http://dbpedia.org/property/currentlyRunBy`	
currently run by	is currently run by
`http://dbpedia.org/ontology/goldenRaspberryAward`	
golden raspberry award	has golden raspberry award
`http://dbpedia.org/ontology/statistic`	
statistic	is statistic of
`http://dbpedia.org/ontology/linkTitle`	
link title	has link title
`http://dbpedia.org/ontology/firstLeader`	
first leader	has first leader

We obtained a total of 315 integration rules (some examples below). We evaluated a random sample of 40, of which 29 were valid and symmetric,

1 was valid but mapped to a generalization of the meaning, 8 were wrong originating from a correct (yet sometimes with incomplete name) property, and 2 were wrong but also incorrect in DBpedia. The resulting precision for valid rules was 79%. Below we reproduce some obtained integration rules.

```
CONSTRUCT {
 _:r a :frame-Appearance-smell.v .
 _:r :fe-Appearance-Phenomenon ?S .
 _:r :fe-Appearance-Characterization ?O .
} WHERE {
 ?S <http://dbpedia.org/property/smellsLike> ?O .
}
```

```
CONSTRUCT {
 _:r a :frame-Residence-reside.v .
 _:r :fe-Residence-Resident ?S .
 _:r :fe-Residence-Location ?O .
} WHERE {
 ?S <http://dbpedia.org/property/residesIn> ?O .
}
```

```
CONSTRUCT {
 _:r a :frame-Experiencer_focus-dislike.v .
 _:r :fe-Experiencer_focus-Experiencer ?S .
 _:r :fe-Experiencer_focus-Content ?O .
} WHERE {
 ?S <http://dbpedia.org/property/dislikes> ?O .
}
```

```
CONSTRUCT {
 _:r a :frame-Possession-own.v .
 _:r :fe-Possession-Owner ?S .
 _:r :fe-Possession-Possession ?O .
} WHERE {
 ?S <http://dbpedia.org/ontology/owns> ?O .
}
```

This is an example of a wrong rule.

```
CONSTRUCT {
 _:r a :frame-Education_teaching-school.v .
 _:r :fe-Education_teaching-Student ?S .
 _:r :fe-Education_teaching-Skill ?O .
} WHERE {
 ?S <http://dbpedia.org/property/schooledAt> ?O .
}
```

?s dbp-Awareness-developsUnderstandingOfContent ?o

↕

f type frame-Progress-develop.v
f fe-Progress-Entity ?s
f fe-Progress-Post_state f′
f′ type frame-Awareness-understanding.n
f′ fe-Awareness-Cognizer ?s
f′ fe-Awareness-Content ?o

Fig. F.9: Example of a very complex noun-based ReDer rule

4 Future Work

We are currently working on creating very complex patterns for certain DBPs whose syntactic head is a noun for which the governing verb (the verb whose object is the noun) adds semantics in a similar way as the pseudo-copulas in Section 2.1. Figure F.9 shows an example of a very complex noun-based ReDer rule. In this case, it is not possible to work with all possible combinations of governing verbs as we did with the copulas, because many verbs will not make sense (compare "develop understanding" with "run understanding"). Therefore, we must use the governing verb from FrameNet's example sentences. Because many verbs can be associated with different frames, the right frame must be chosen on the reified side of the ReDer rule. Due to the high number of possible verbs that could be governing nouns in the example sentences, an automatic disambiguation method is necessary. Likewise, an automatic selection of the FE connecting the frames for the noun and the governing verb is necessary, e.g., `Post_state` in Figure F.9.

We are also working on creating integration rules to express very complex reification patterns for certain linguistic patterns in the property name. For instance, Figure F.10 shows an example expressing amounts for property names satisfying the regular expression `(has )?number of (.*)`, which is relatively common among LOD datasets mined from tables with statistics. The recall of this method can be increased if the canonicalization is also extended to complete these patterns in case parts of them are omitted. For instance, for the example about amounts given above, the prefix "has number of" could be added to those properties whose name is a countable noun or a noun phrase, and whose range is a positive integer (in LOD datasets typically implemented as literals with datatypes `xsd:nonNegativeInteger`).

Finally, we are also working on combining all these rules with other types of rules that map entities of the same type (classes with classes, properties with properties), and can be built re-using existing `owl:sameAs` ontology alignment systems. This combination will allow arbitrarily complex mappings, not only between the external datasets and FrameBase, but transitively between the external datasets.

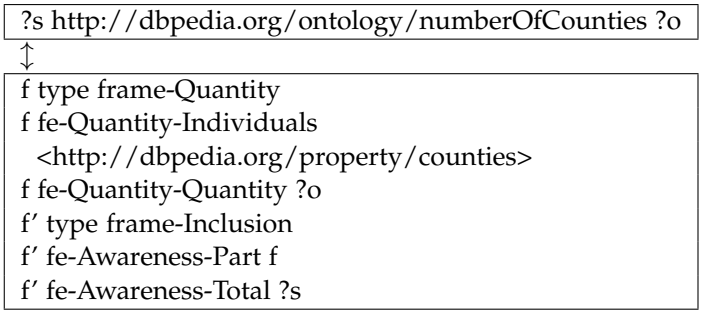


Fig. F.10: Example of a very complex integration rule to express amounts

5 Conclusion

In this paper, we have shown the importance of establishing complex mappings between linked open datasets, transcending the space of binary relationships that can be captured using simple links of type `owl:sameAs`, `rdfs:subClassOf`, or `rdfs:subPropertyOf`. We have shown schema-level methods to create these complex mappings, using a star-based topology with a wide schema as a central hub, and exploiting its connections to computational linguistics. As part of this process, we have also provided heuristics to extend, disambiguate, and canonicalize the names of properties in the source datasets. We have evaluated our approach on DBpedia, finding that it yields encouraging results across different domains. Finally, we have outlined future work to create even more integration rules involving complex patterns.

References

[1] H. Halpin and P. J. Hayes, "When owl: sameAs isn't the Same: An Analysis of Identity Links on the Semantic Web," in *LDOW'10*, 2010.

[2] G. de Melo, "Not Quite the Same: Identity Constraints for the Web of Linked Data," in *AAAI'13*, 2013.

[3] J. David, J. Euzenat, F. Scharffe, and C. Trojahn dos Santos, "The Alignment API 4.0," *Semantic Web Journal*, vol. 2, no. 1, pp. 3–10, 2011.

[4] F. Scharffe and D. Fensel, "Correspondence patterns for ontology alignment," in *Proceedings of the 16th International Conference on Knowledge Engineering: Practice and Patterns*, ser. EKAW '08. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 83–92.

[5] R. Dhamankar, Y. Lee, A. Doan, A. Halevy, and P. Domingos, "iMAP: Discovering Complex Semantic Matches Between Database Schemas," in *SIGMOD 2004*, 2004, pp. 383–394.

[6] D. Ritze, C. Meilicke, O. Sváb-Zamazal, and H. Stuckenschmidt, "A Pattern-based Ontology Matching Approach for Detecting Complex Correspondences," in *OM'10*, 2008.

[7] J. Rouces, G. de Melo, and K. Hose, "FrameBase: Representing N-ary Relations Using Semantic Frames," in *ESWC'15*, 2015.

[8] ——, "Representing Specialized Events with FrameBase," in *DeRiVE'15*, 2015.

[9] J. Rouces, G. de Melo, and K. Hose, "A Frame-Based Approach for Connecting Heterogeneous Knowledge," 2016, Submitted.

[10] C. F. Baker, C. J. Fillmore, and J. B. Lowe, "The Berkeley FrameNet Project," in *Proceedings of the 17th international conference on Computational linguistics – Volume 1*, ser. ICCL '98, 1998, pp. 86–90.

[11] C. Fellbaum, *WordNet: An Electronic Lexical Database*. MIT Press, 1998.

[12] L. Galárraga, G. Heitz, K. Murphy, and F. M. Suchanek, "Canonicalizing Open Knowledge Bases," in *CIKM'14*, 2014, pp. 1679–1688.

[13] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann, "DBpedia-A crystallization point for the Web of Data," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 7, no. 3, pp. 154–165, 2009.

Paper G

Klint: Assisting Integration of Heterogeneous Knowledge

Jacobo Rouces, Gerard De Melo, Katja Hose

The demo paper has been accepted for publication in the
Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI), 2016

The layout has been revised.

Abstract

An increasing number of structured knowledge bases have become available on the Web, enabling many new forms of analytics and applications. However, the fact these are being published by different parties with heterogeneous vocabularies and ontologies also leads to formidable data integration challenges. This paper presents Klint, a web-based system that automatically creates mappings to transform knowledge from original data sources into a large unified schema, and allows them to be reviewed and edited by users with a streamlined interface. In this way, it allows human-level accuracy with minimum human effort.

1 Introduction

The Web of Data now includes a rich and increasing amount of structured knowledge bases and has enabled many new applications and forms of analytics. These are usually available in a format based on subject-predicate-object triples, such as RDF, yet they are modelled in different ways and querying them jointly becomes a daunting task, even if they are available under a single endpoint. The reason is that, in order to capture all relevant 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.

Automatic data integration would solve this, but is often an AI-hard problem, especially since many applications require knowledge with a precision of 90% or higher. Moreover, existing work in this area has mostly focused on connecting entries via binary properties such as `owl:sameAs`. However, these only connect individual identifiers but cannot easily capture mappings between more complex patterns of triples that represent the same information but are structurally different.

In this paper, we present Klint (Knowledge integrator), a Web-based system enabling semi-automatic schema integration. Given one or more existing RDF ontologies, Klint generates tentative integration rules from these ontologies into a unified schema. For this unified schema, Klint relies on FrameBase [1], a wide-coverage, highly expressive and extensible schema that can be used to represent and integrate [2] a wide range of knowledge from many sources in a homogeneous and seamless way. 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.

2 Assisted Schema Integration

Klint allows a user to integrate one or more entire knowledge bases into FrameBase with minimum effort. An input knowledge base can be loaded

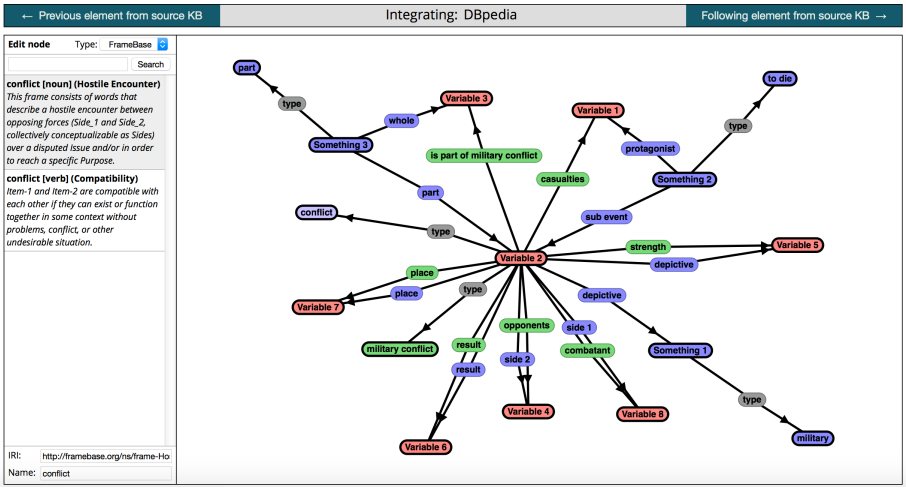


Fig. G.1: The Klint interface integrating elements from DBpedia – Klint used the contextual and lexical information from the source elements to suggest two candidate values for the integrated type (selected node, "conflict"), for which the correct assigned value, `Hostile_encounter-conflict.n` was the 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.

from an RDF file or a SPARQL endpoint. Other structured data formats can also easily be used after being pre-processed with a suitable RDF converter[1].

Integration Heuristics. Klint automatically creates complex integration rules for each element in the source schema, using intelligent integration algorithms based on linguistic annotations in FrameBase [2], extended with a support vector machine learning from a labeled training set.

Interface. Each integration rule is then represented as a graph in the right pane (see Figure G.1). Users can navigate across different integration rules with the buttons at the top bar, making modifications in a given graph if this is found necessary. **Variable nodes** are shown in red and represent universally quantified variables over entities. They bind the pattern from the source KB with the integrated FrameBase pattern. The remaining nodes are classified according to the type of entity they represent.

- **Source nodes** (shown in green) represent resources from the source KB and connect two different variable nodes, forming a knowledge pattern in the source KB.
- **FrameBase nodes** (colored in different shades of blue) represent FrameBase resources and also connect variable nodes. They provide the *translation* of the source pattern to FrameBase.

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

- **Auxiliary nodes** (colored gray) represent resources from third-party knowledge bases, usually representing common idioms or very specific entities.

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

Both edges and nodes can be added, deleted, or edited. When a node is selected, the left panel is activated, where the user can make changes.

Automatic Suggestions. When selecting a FrameBase node from an automatically created integration rule, the integration engine behind Klint provides an ordered list of alternative suggestions for its value in the left pane, in case the default choice was not the correct one. If users still do not find an appropriate choice in the list, they can use the search box to conduct a custom search. This search re-uses the algorithm of the integration engine, but allows free input.

Klint also assists when a FrameBase node is created from scratch, but not given a value. Once the new node is connected to others (and therefore is given a context), Klint will be able to use the integration engine and the constraints given by the context to suggest possible values.

When an element from the candidate list on the left is chosen, and this element is a class, associated FrameBase predicates are added as well, connected via subject-predicate edges. Users can select the ones they find relevant by completing the property-object edges, and delete or just ignore the rest, which will not produce complete triples. In some cases, our system will also automatically produce entire triples for certain predicates.

3 Conclusion

We have presented Klint, a web-based framework that allows the user to supervise the automatic integration of heterogeneous knowledge bases, by providing a user-friendly graph-based interface that allows to review and curate complex integration rules produced by state-of-the-art integration algorithms.

References

[1] J. Rouces, G. de Melo, and K. Hose, "FrameBase: Representing N-ary Relations Using Semantic Frames," in *ESWC'15*, 2015.

[2] ——, "Complex Schema Mapping and Linking Data: Beyond Binary Predicates," in *LDOW'16*, 2016.

ISSN (online): 2246-1248
ISBN (online): 978-87-7112-557-3

AALBORG UNIVERSITY PRESS