Knowledge-Based Systems 135 (2017) 192–203

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Event-based knowledge reconciliation using frame embeddings and frame similarity

Mehwish Alam [a], Diego Reforgiato Recupero [b,d,*], Misael Mongiovi [c], Aldo Gangemi [a,d], Petar Ristoski [e]

[a] Université Paris 13, 99 avenue JB Clément, Villetaneuse, Paris 93430, France
[b] Università degli Studi di Cagliari, Department of Mathematics and Computer Science, Via Ospedale 72, Cagliari 09124, Italy
[c] CNR, ISTC, Catania, Italy
[d] CNR, ISTC, Via S. Martino della Battaglia 44, Rome, Italy
[e] University of Mannheim, Mannheim, Germany

ARTICLE INFO

Article history:
Received 6 April 2017
Revised 9 August 2017
Accepted 14 August 2017
Available online 16 August 2017

Keywords:
Knowledge reconciliation
Frame semantics
Frame embeddings
Frame similarity
Role similarity
Role embeddings
FrameNet
Framester

ABSTRACT

This paper proposes an evolution over MERGILO, a tool for reconciling knowledge graphs extracted from text, using graph alignment and word similarity. The reconciled knowledge graphs are typically used for multi-document summarization, or to detect knowledge evolution across document series. The main point of improvement focuses on event reconciliation i.e., reconciling knowledge graphs generated by text about two similar events described differently. In order to gather a complete semantic representation of events, we use FRED semantic web machine reader, jointly with Framester, a linguistic linked data hub represented using a novel formal semantics for frames. Framester is used to enhance the extracted event knowledge with semantic frames. We extend MERGILO with similarities based on the graph structure of semantic frames and the subsumption hierarchy of semantic roles as defined in Framester. With an effective evaluation strategy similarly as used for MERGILO, we show the improvement of the new approach (MERGILO plus semantic frame/role similarities) over the baseline.

1. Introduction

Several approaches have been proposed for extracting knowledge graphs from text. These knowledge graphs are generated with the aim of making unstructured text machine-readable [1]. In case of multiple texts explaining similar events, it is more efficient and usable to provide the machine with a combination of multiple graphs generated by multiple texts. Using this merged graph, a machine reader can obtain knowledge contained in multiple texts from a single consolidated graph instead of reading several graphs. This problem, termed as "Knowledge Reconciliation" (KR), has recently been addressed by *MERGILO* [2], a tool for reconciling knowledge graphs using graph alignment and word similarity. These reconciled knowledge graphs can further be utilized by specific NLP applications, in particular by graph-based text summarization (which aims at *summarizing* knowledge represented in multiple closely related pieces of text), for assessing sentence or document similarity, etc.

The current study mainly targets the problem of knowledge reconciliation from the perspective of events. In a text, a complete description of an event is syntactically denoted by a verb, since it defines a relation between event participants. The first step in the event-based knowledge reconciliation is to extract event-oriented knowledge graphs. For doing so, we use FRED, a machine reader presented in [1], which generates an RDF/OWL graph of any open domain input text.

For dealing with different lexical units describing the same or similar events, we enhance the existing pipeline by enriching the knowledge graphs generated by FRED with semantic frames as defined in FrameNet[1]. For this purpose, this study further makes use of mappings between VerbNet[2] (i.e., VerbNet verb classes and VerbNet roles) and FrameNet, as contained in Framester [3]. Framester is a linguistic linked data hub formulated using a novel formal semantics for frames for improving semantic interoperability between linguistic resources. Framester uses the RDF version of

* Corresponding author at: via simeto 5, san gregorio, Italy.
E-mail addresses: alam@lipn.univ-paris13.fr (M. Alam), diego.reforgiato@unica.it (D. Reforgiato Recupero), misael.mongiovi@istc.cnr.it (M. Mongiovi), aldo.gangemi@lipn.univ-paris13.fr (A. Gangemi), petar.ristoski@informatik.uni-mannheim.de (P. Ristoski).

[1] https://framenet.icsi.berkeley.edu/fndrupal/.
[2] https://verbs.colorado.edu/~mpalmer/projects/verbnet.html.

http://dx.doi.org/10.1016/j.knosys.2017.08.014
0950-7051/

FrameNet [4][3], formalizes the FrameNet graph in OWL, and introduces a very rich subsumption hierarchy related to FrameNet frame elements *(semantic roles)*.

We use Framester graph representations as a way to improve similarity between the nodes and the edges, where nodes represent the frames and edges represent the roles. When different verbs denote similar events, i.e. different verbs evoke different frames which are somehow connected in the FrameNet graph using the semantic relations already defined in FrameNet (such as *Inheritance, SubFrame, ...*), we can greatly improve simple string matching techniques introduced in MERGILO with frame as well as semantic role similarity measures. For doing so we considered the similarities based on the graph structure of the FrameNet frames as well as the subsumption hierarchy associated to the semantic roles defined in Framester. FrameNet graph organizes frames using semantic relations; to benefit from this graphical structure we adapt WordNet similarity measures [5] to FrameNet graph. We further exploit the vector representations of frames using the FrameNet graph and the subsumption hierarchy of roles as represented in Framester. We follow the approach `RDF2Vec` [6] to generate graph based *frame embeddings* referred to as `Frame2Vec`. These graph-based embeddings make use of graph mining algorithms such as graph walks and graph kernels to traverse over the graph, which is further used for generating its vector representations. In order to find the similarity between two frames and between two roles, this study uses WordNet similarities and cosine similarity for obtaining better consolidation between multiple graphs, which lead to an improvement over the results of a baseline algorithm for knowledge reconciliation, MERGILO [2]. MERGILO already computes the similarity between the roles represented as edges in the FRED graphs but it merely performs string matching for finding if the roles are similar. These embeddings can further be used for any NLP application, however in the current scenario we use it for knowledge reconciliation purposes.

More in detail, the paper is organized as follows. Section 2 introduces state of art and related work. Section 3 lists the data sources, resources and tools we have adopted in our methodology. Then, Section 4 gives some details of MERGILO and its functionalities for use as basis for the Section 5, which explains how frame semantics have been employed for improving MERGILO. Section 6 shows a precision-recall analysis for the presented approach on the dataset introduced in [2]. Finally, Section 7 concludes the paper with discussions, remarks and highlights some future directions.

2. State of the art

2.1. From text to knowledge graphs

Given the large amount of unstructured text, it has become a key challenge to extract structured information and knowledge from that and integrate it into a coherent knowledge graph. There are several applications which aim at extracting these structures such as digital assistants (Siri, Alexa, Cortana, and Google Now), question answering, summarization. Projects such as Never Ending Language Learning (NELL) [7], OpenIE [8], YAGO [9], and Google Knowledge Vault [10] proposed various technologies and methodologies to extract new structured information from the web and represented a significant progress in the field of information retrieval and relation extraction. Three categories of methodologies for relation extraction have been defined i.e., supervised, semi-supervised, and distant supervision approaches. Supervised approaches formulate the problem of text extraction as a classification problem. They generally extract a set of features (context words, part of speech tags, dependency path between entity, edit distance, etc.) from the sentence and the corresponding labels are obtained from a large annotated training corpus. Usually these approaches are neither general nor scalable and computationally very expensive due to the requirement of large amount of training data. Semi-supervised approaches start with seed triples and iterate through the text to extract patterns that match them. Patterns become new seed triples and the process is recursively repeated until no other pattern is found. Some of the most popular approaches in this category are Dual Iterative Pattern Relation Extractor [11], Snowball [12], Text Runner [13]. For the last category, distant supervision approaches, existing knowledge bases are used with large text corpus to generate a large number of relation triples. These relations are located within the text and from them new hypothesis are learned to obtain a generalized model for relation extraction. Projects such as NELL use predefined ontology and bootstrap relations from the web and text using seed examples of ontology constraints. Then they use multi-view learning paradigm to extract entities and relations from unstructured text.

2.2. Knowledge integration

Approaches for integrating knowledge include cross-document coreference resolution (when knowledge is represented as text documents) and ontology matching (when knowledge is in a machine-readable form). Cross-document coreference resolution aims at associating mentions about a same entity (object, person, concept, etc.) across different texts [14–17]. When extracted entities are events, the problem changes to resolution of event coreference across documents [18,19]. Authors in [19] jointly model named entities and events. Clusters of entities and event mentions are constructed and merged accordingly to a similarity threshold based on linear regression. Then, information flows between entity and event clusters through features that model semantic role dependencies. The system handles nominal and verbal events as well as entities, and the joint formulation allows information from event coreference to help entity coreference, and vice-versa.

A rich overview of ontology matching methods is provided by [20]. Relevant work includes [21] that leverages the interplay between schema and instance matching. Similarly, [22] shows a greedy iterative algorithm for aligning knowledge bases with millions of entities and facts. These approaches are characterised by the preferred large size of the ontologies/datasets treated (for best performance), which is rarely (probably never) derived from text sources. MERGILO, as other knowledge integration tools [22], employs graph alignment, a more general and widely studied problem [23–25]. To note that all these approaches are connected and related to the classical graph matching problem [26].

2.3. Word embeddings and its applications

Word Embeddings are the Vector Space Representation (VSM) of words in a low-dimensional semantic space. A conventional way of computing these representations is to create a term-document frequency matrix and then perform dimensionality reduction on that matrix using Singular Value Decomposition [27,28]. Recent techniques convert the two step approach to single step using neural networks [29] which also proves to have significant gain in efficiency. It computes continuous vector representations of the words in very large data sets. Another variation of this approach that learns fixed-length feature representations from text of different lengths such as sentences, paragraphs and documents has been proposed in [30] and is called as ParagraphVector. GloVe [31] is another similar technique which uses statistical methods for im-

[3] http://www.ontologydesignpatterns.org/ont/framenet/abox/cfn.ttl.

proving the efficiency over state-of-the-art methods for vector representations.

These vectors obtained by the above defined methods can be used in a variety of applications such as information retrieval, document classification, question answering, named entity recognition and parsing etc. One recent application is used for generating vector representations of word senses [32] and then these vector representations are used for improving the results of word similarity and word analogy tasks based on BabelNet word senses formally known as SensEmbed. [33,34] apply Frame Semantics and Distributional Semantics for slot filling in Spoken Dialogue System. In [35], the authors use Word and Frame Embeddings for generating categories of annoying behaviors where each category contains a set of words specific to that category. The frame embeddings are generated using 3.8 million tweets tagged by FrameNet frames using SEMAFOR. By contrast, in this study we are using graph-based Frame Embeddings. However, as a perspective, the frame embeddings generated using SEMAFOR on tweets will be compared to embeddings generated using Word Frame Disambiguation API as discussed in [3] over Wikipedia Data Dump. Finally, [36] reviews several methods for analysing relational data in the form of graphs. It focuses on how models based on latent features and pattern mining can be trained on large knowledge graphs and used for prediction.

3. Data sources and tools

3.1. VerbNet

VerbNet [37] is a broad coverage verb lexicon in English with links to other data sources such as WordNet [38] and FrameNet [39]. VerbNet contains semantic roles and patterns which allows to form a verb class called as Levin's classes. It generalizes the verbs based on their shared syntactic behavior. These verb classes are structured into a hierarchy of parents and their subclasses. For example, the verb *conquer* is a member of the class *subjugate-42.3* which means to bring under domination.

VerbNet further contains thematic roles which correspond to the relation between the predicate and its arguments. These thematic roles are further organized into a hierarchy. For each class contained in VerbNet, there exists a list of roles which identifies the behavior of a verb in the class. For example, the roles defined for the class *subjugate-42.3* are *Agent, Patient* and *Instrument* meaning that an agent subjugates the patient with some instruments. Here, *Agent* and *Patient* are the necessary roles and *Instrument* is an optional role. Verb senses help in determining if a particular verb instance conforms to the underlying semantics of the class, in case of the verb *conquer* its only sense is included in the class *subjugate-42.3*. VerbNet further maps the verb to a FrameNet frame e.g., the verb *conquer* is mapped to the frame *Conquering*.

3.2. FrameNet

FrameNet [39] contains descriptions and annotations of English words using Frame Semantics. FrameNet contains *frames*, which describe a situation, state or action. Each frame has *frame elements* usually consisting of agent, patient, time and location and are also known as *semantic roles*. FrameNet also defines a subsumption relation between the frame elements. Each frame can be evoked by *Lexical Units (LUs)* belonging to different parts of speech. In version 1.5, FrameNet covers about 10,000 lexical units and 1024 frames. These LUs can be nouns, verbs, adjectives and adverbs representing closely related sets of meanings.

For example, in the frame *Conquering* the argument for the role *Conqueror* overtakes the argument of the role *Theme* where the theme loses its autonomy. Such constructs describing the situation of conquering or invasion are referred to as *frame elements* and the LUs such as *conquer, overtake* etc. are example words, typically used to denote conquering situations in text. Let us consider the following sentence:

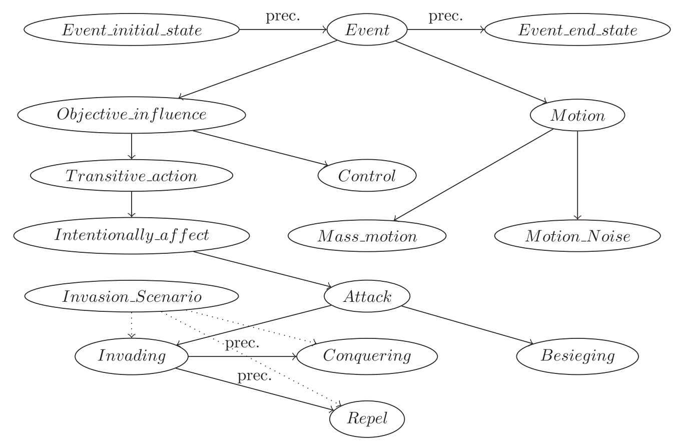

Fig. 1. A part of FrameNet graph. "prec." represents the relation "precedes", dotted lines represent "SubFrame" relation and solid lines represent the "Inheritance" relation as defined in FrameNet.

$$[The\ Spaniards]_{Conqueror}[conquered]_{Lexical\ Unit}[the\ Incas]_{Theme}. \quad (1)$$

In the above example, *The Spaniards* is the argument of the role *Conqueror* and Incas is the argument of the role *Theme* and *conquered* is the LU evoking the frame.

3.3. Framester

Framester [3] is a large RDF[4] knowledge graph (currently including about 30 million RDF triples) acting as a hub between FrameNet, WordNet, VerbNet [37], BabelNet [40], Predicate Matrix [41], etc. It leverages this wealth of links to create an interoperable and homogeneous *predicate space* represented in a formal rendering of frame semantics [42] and semiotics [43]. Framester uses a mapping between WordNet, BabelNet, VerbNet and FrameNet at its core using detour based approach, expands it to other linguistic resources transitively, and represents all of this formally. It further links these resources to important ontological and linked data resources such as DBpedia [44], YAGO [9], DOLCE-Zero [45], schema.org, [46], NELL [7], etc. Further links to DeepKnowNet [47] topic signatures, as well as SentiWordNet [48] and DepecheMood [49] mood mappings, are also available.

Framester keeps the original node-arc-labeled graph as introduced in FrameNet where the nodes represent the FrameNet frames and the edges represent different semantic relations between the frames i.e., *Inheritance, SubFrame, CausativeOf* etc. Fig. 1 shows a part of FrameNet graph. It re-uses the RDF graph introduced in [4][5]. Framester has also cleaned up the subsumption hierarchy of semantic roles (i.e., frame elements) and added generic roles on top of the frame specific roles. Fig. 2 shows a part of the Framester role hierarchy associated with the framester role *agent*.

3.4. FRED

FRED [1][6] is machine reader which generates ontological structure from natural language text using Discourse Representation

[4] https://www.w3.org/TR/rdf11-primer/.
[5] http://www.ontologydesignpatterns.org/ont/framenet/abox/cfn.ttl.
[6] http://wit.istc.cnr.it/stlab-tools/fred.

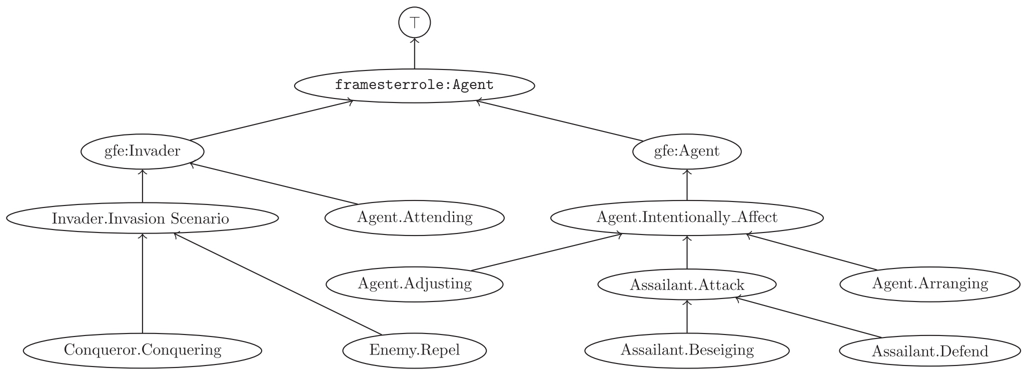

Fig. 2. A part of Subsumption Hierarchy with FrameNet and Framester Roles. The prefixes for http://www.ontologydesignpatterns.org/ont/framenet/abox/gfe/ and http://www.ontologydesignpatterns.org/ont/framester/data/framesterrole.ttl# are `gfe:` and `framesterrole:` respectively.

Theory (DRT) [50], frame semantics [42] and Ontology Design Patterns [51]. FRED uses Boxer [52] which is an open source tool for deep parsing of natural language using Combinatory Categorial Grammar (CCG) and produces event-based, semantic representations of natural language. The Discourse Representation Structures (DRS) produced by Boxer use VerbNet thematic roles. These functionalities implemented in FRED help in the event detection task for our method. FRED further uses logical rules on top of the First Order Logic (FOL) representations generated by Boxer to generate ontological models. For further details, refer to [1]. Fig. 3 shows the output of FRED for the sentence in the running example. FRED also implements linguistic frame detection task and performs semantic role labeling which is comparable to Semafor[7]. However, in the current study we use VerbNet roles as a base because the coverage of VerbNet roles is targeted better in FRED as compared to FrameNet roles (i.e., frame elements). FRED is also available online as REST service. We further exploit Framester for the mappings between VerbNet and FrameNet as described in Section 3.3. Section 5 further details how we used these mappings for providing event-based knowledge reconciliation.

4. MERGILO

MERGILO [2] is a method for generating and integrating knowledge graphs extracted from multiple text documents by using FRED, a machine reader. Given two input sentences, it extracts the associated knowledge graph using FRED.

Knowledge Extraction: The graphs generated by FRED can be viewed as a fully labeled multi-digraph which consists of nodes and edges representing schema entities, data entities, meta-data entities, linguistic entities, etc. As a first step, MERGILO parses the text into an RDF-OWL. MERGILO basically focuses on four objects from frame semantics perspective, i.e., (i) named and skolemized entities (machine generated entities) e.g., persons, places, etc. (ii) event occurrences i.e., an event is represented by a verb in a sentence with attached a semantic role R having the arguments of an event A, (iii) classes (public names, machine generated names) such as city, country, etc, and finally (iv) qualities, which represent characteristics of an entity such as nice, strong etc.

The semantic roles in FREDs graphs are represented as an edge and are divided into two macro-categories: roles and non-roles. Roles are outgoing edges from event nodes. Role edges are broadly classified into agentive, passive, and oblique roles. All other edges are non-role edges. Some of the non-role edges include `owl:sameAs`, `owl:equivalentClass`, `rdf:type` and `rdfs:subClassOf`, with standard meaning from RDFS and OWL ontology specification languages.

Knowledge Reconciliation: Given two sentences and their FRED graphs $G_1 = (V_1, E_1, P_1)$ and $G_2 = (V_2, E_2, P_2)$, V_1 and V_2 represent nodes (entities), E_1 and E_2 represent edges (relations) and P_1 and P_2 represent edge labels (properties). G_1 and G_2 are first compressed by merging nodes and removing unnecessary URIs. The two compressed graphs are aligned by establishing a 1-1 correspondence between nodes of the first graph and nodes of the second graph that maximizes a score function, which combines the similarity between aligned nodes and the similarity between aligned edges. Maximizing the score function has the effect of aligning nodes that have high similarity and that are in turn connected to edges with high similarity. Therefore both element similarities and structural information are considered. At the end, the aligned nodes are mapped to individuals in the original graphs and sameAs relations are added between aligned nodes. Fig. 4 (taken from [2]) shows two input sentences and their corresponding FRED graphs. Fig. 5 (taken from [2]) reports the final output of MERGILO for the two input sentences after the compression of some of the nodes in the original graphs. Red dashed lines represent cross-graph co-references.

Similarity measures for nodes and edges are used by the optimizer to define the alignment score function. The similarity can be positive or negative. Elements that have negative similarity tend not to be associated, while elements with positive similarity tend to be associated. Note that the alignment algorithm performs a global optimization, and hence local parts of the alignment may be penalized in favor of a global reward. For instance, two edges with positive similarity may not be aligned because this would imply aligning their endpoint nodes with negative similarity. Similarly, two nodes with negative similarity may be aligned to enable aligning incident edges with positive similarity.

We distinguish among three kinds of node pairs whose definitions are given below: **relevant, compatible** and **incompatible**. We first check if both nodes refer to named entities. If so, we check whether they refer to the same named entity or to different ones. Labels of named entities are compared both by string matching and by their alignment to public resources (DBpedia). If the labels are equal or are associated with the same DBpedia entity, the node pair is considered **relevant**. If the two nodes share the same URI or refer to words with similarity higher than a predefined threshold that we call similarity threshold, they are considered **compatible**. In all other cases, the nodes are considered **incompatible**. Therefore, the similarity between two nodes v_1 and v_2 is assigned as

[7] http://www.cs.cmu.edu/~ark/SEMAFOR/.

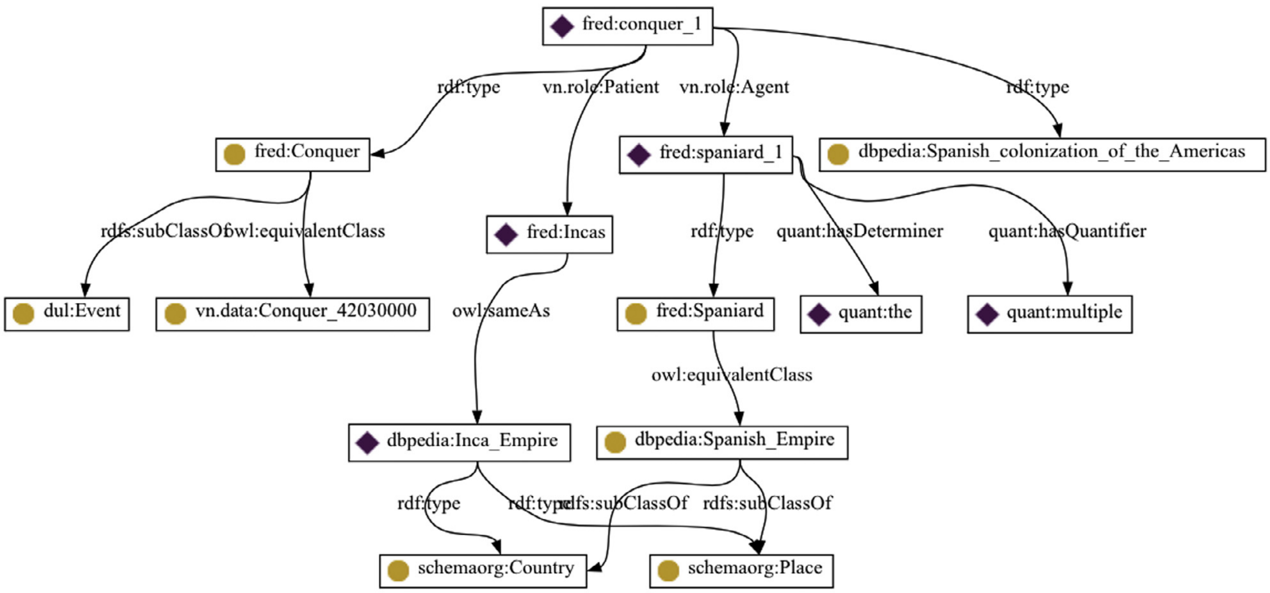

Fig. 3. FRED Knowledge Graph for the Sentence 1.

follows:

$$sim(v_1, v_2) = \begin{cases} 1 \text{ if } v_1 \text{ and } v_2 \text{ are relevant} \\ -1 \text{ if } v_1 \text{ and } v_2 \text{ are compatible} \\ -inf \text{ if } v_1 \text{ and } v_2 \text{ are incompatible} \end{cases}$$

The similarity between two edges is defined in terms of their type. Specifically, we distinguish between compatible and incompatible edges based on their property type and possibly their thematic role. If both edges are non-role edges, they are considered compatible. If both edges are role edges, they are considered compatible only if their roles are both agentive (AGNT) or passive (PTNT). In all other cases the edges are considered incompatible. The similarity between two edges is defined as:

$$sim(e_1, e_2) = \begin{cases} 1 + w + eps \text{ if } e_1 \text{ and } e_2 \text{ are compatible} \\ -inf \text{ if } e_1 \text{ and } r_2 \text{ are incompatible} \end{cases}$$

where *eps* is a very small number (0.001) introduced to break ties and w is a parameter that enables associating sets of compatible nodes if they are connected by a sufficiently high numbers of edges. For more details about MERGILO the reader is invited to see [2]. Here it can clearly be noticed that the similarity is performed using only string matching between the edges representing some roles. However, the next section discusses the novelty of our approach by introducing several ways of computing similarities between the edges representing a role as well as the nodes representing an event.

5. Event-based knowledge reconciliation

Let us consider the two sentences: *"The Spaniards conquered the Incas."* and *"The Incas were attacked by the Spaniards."* The two sentences are addressing two actions related to the same happening in the past i.e., event of an attack or an invasion from Spaniards to Incas. In such a case, the similarity measures introduced by MERGILO will not be able to effectively consider the similarity between the two events because the two verbs are different. Figs. 3 and 6 show the FRED graphs of the first and the second sentence, respectively.

For finding the similarity between these two sentences, the following extensions were mainly performed based on node and edge similarities. Several similarity measures were applied on the FrameNet frame graph and the subsumption hierarchy of the roles. This section focuses on:

–improved subsumption hierarchy of roles in Framester,
–improved node similarities (based on frame similarity) and
–improved edge similarities (based on role similarity).

Two kinds of similarities were used *(i)* by traversing only inheritance relation in the FrameNet graph using depth first search algorithm and *(ii)* using graph walks and graph kernels for generating vector representations of frames and roles (`Frame2Vec`) and then computing the cosine similarity between the corresponding vectors.

Node Similarity: In the current study we improved the alignment score function as described in Section 4 using FrameNet. We introduce the similarity between two nodes. For doing so, the first step is to verify that the nodes represent the verb senses. Let s_1 and s_2 be two verb senses from two different graphs G_1 and G_2 generated from two different texts. To compute the similarity between two such nodes, the verb senses are further mapped to frames using Framester mappings. Each verb sense s_i can have one or more mappings represented as follows: $s_1 \rightarrow \{f_{11}, f_{12}, f_{13}\}$ and $s_2 \rightarrow \{f_{21}, f_{22}, f_{23}\}$. Then we compute pairwise similarity between two sets of frames i.e., $sim_t(f_{11}, f_{21}), sim_t(f_{11}, f_{22}), \ldots,$ where t represents the type of similarity measure. Finally, we obtain a set of similarity scores (*sim_score*) where each value varies in [0–1] i.e, where 0 indicates that the two frames are completely dissimilar and 1 represents the same frame and the values between 0 to 1 represent the degree to which the frames are similar. After obtaining the set of similarity scores we choose the maximum score $max(sim\_score)$.

For example, in Figs. 3 and 6 $s_1 = vn.data : Conquer\_42030000$ and $s_2 = vn.data : Attack\_33000000$. According to Framester mappings, we obtain $s_1 \rightarrow \{Conquering\}$ and $s_2 \rightarrow \{Attack\}$. These nodes are replaced by their corresponding frames. Further, the similarity is computed between these two frames, as discussed in Section 5.1.

These similarities are computed in two ways: *(i)* by considering the taxonomical structure imposed by the "inheritance" relation represented as `fnschema`[8]`:inheritsFrom` in Framester; *(ii)* by

[8] prefix fnschema: http://www.ontologydesignpatterns.org/ont/framenet/tbox/.

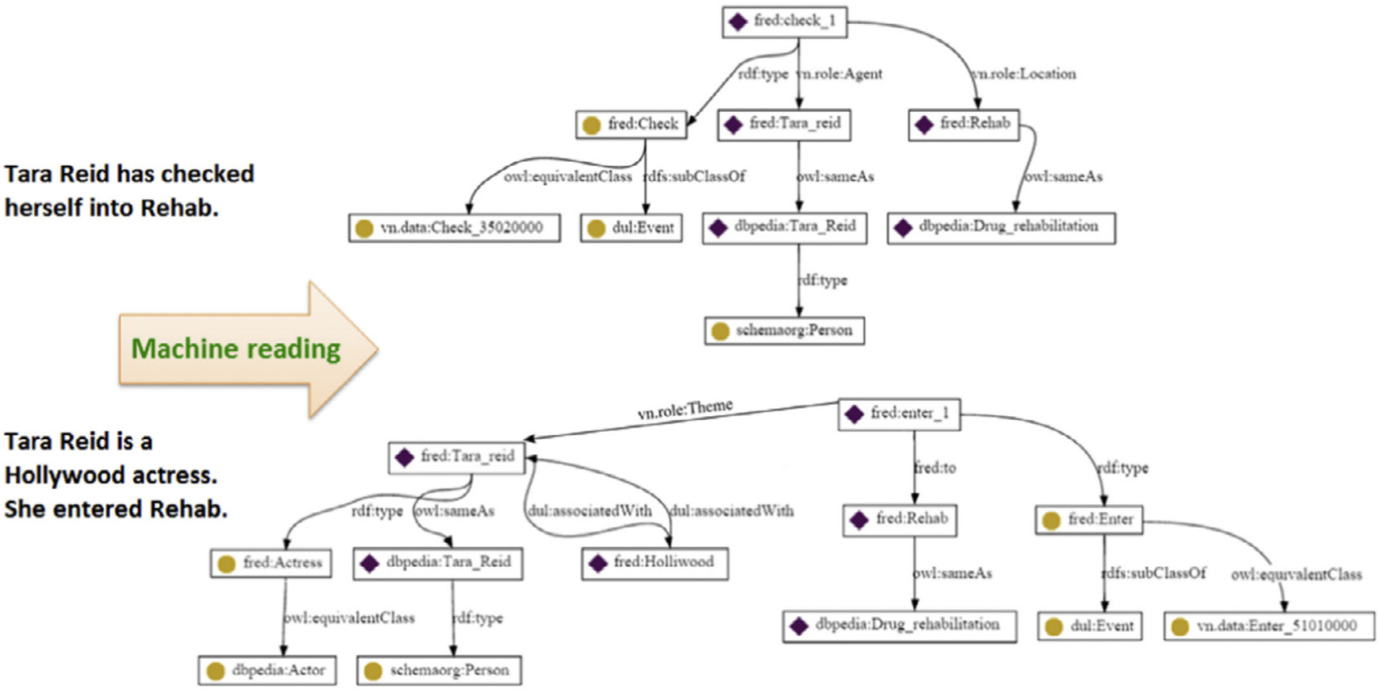

Fig. 4. FRED graphs for two input sentences shown on the left side of the image. The example has been taken from [2].

considering the graphical structure of the FrameNet graph without putting any constraints over the kinds of relations by performing graph walks and using graph kernels (see Section 5.2). For example, in Fig. 1, it can be clearly seen that the first kind of similarity is not a fair measure because it does not consider the "precedes" or "SubFrame" relation. Accordingly, the similarity between "Invading" and "Conquering" will be 0 in case of the first kind of similarity which is semantically not true. However, the second similarity score for the second kind (i.e., graph walks and graph kernels) will be higher. The types of similarity measures implemented using "inheritsFrom" are *Path Similarity, Wu-Palmers Similarity* and *Leacock-Chodorow Similarity*. These WordNet similarities are adapted to FrameNet graphs. The similarity used for vector representations of FrameNet graph is the cosine similarity.

Edge Similarity: MERGILO computes the similarity between the edges based on the types of the edges i.e., they are compatible if both the roles are agentive or passive. Moreover, it only checks if two roles are compatible or not, hence generating a number which can be either 0 or 1. In our extension the similarities are assigned the values belonging to the interval [0–1] which enables the system to judge the degree to which the two roles are similar. The similarity measures used for this purpose are computed on the subsumption hierarchy of the roles provided in Framester. As a first step, the edges containing the VerbNet roles are identified, these VerbNet roles are then mapped to the FrameNet semantic roles using the extended version of the mappings from VerbNet roles to FrameNet roles. In case of multiple mappings, pairwise similarity is computed.

For example, in Fig. 3, the verb sense `vndata`[9]`:Conquer_42030000` evokes the roles `vndata:Agent` and `vndata:Patient`. In the sentence in Fig. 6, the roles evoked by the verb sense `vndata:Attack_33000000` are `vndata:Agent` and `vndata:Theme`. The Framester mappings contains the following records for these roles:

```
vndata:Agent.conquer_42030000 skos:closeMatch
  fe:Conqueror.conquering.
vndata:Patient.conquer_42030000
  skos:closeMatch fe:Theme.conquering.
vndata:Agent.attack_33000000 skos:closeMatch
  fe:Assailant.attack.
vndata:Theme.attack_33000000 skos:closeMatch
  fe:Victim.attack
```

Then the similarity between `fe`[10]`:Assailant.attack` and `fe:Conqueror.conquering` is computed in three ways:

1. by considering the subsumption hierarchy represented by the subsumption relation represented as "subsumedUnder" in Framester;
2. using the refined subsumption hierarchy of the roles in Framester and,
3. without putting any constraints over the kinds of relations by performing graph walks and using graph kernels.

5.1. Semantic similarity between frames and roles

This section details automated ways to compute the similarity measures between two frames based on the relations already present in FrameNet. This notion has been partly discussed in [53]. In the following we mainly use the inheritance relation i.e., the hierarchical structure of the FrameNet graph. The WordNet similarity measures were adjusted to deal with the frames. Fig. 7 shows the part of taxonomical structure of the FrameNet graph for the running example.

Path Similarity is based on shortest distance between two nodes in the taxonomy. Let us consider two nodes c_1 and c_2; then the shortest path similarity between these two nodes is given as follows:

$$sim_{path}(c_1, c_2) = \frac{1}{len(c_1, c_2) + 1} \quad (2)$$

[9] `prefix vndata:` http://www.ontologydesignpatterns.org/ont/vn/vn31/data/.

[10] `prefix fe:` http://www.ontologydesignpatterns.org/ont/framenet/abox/fe/.

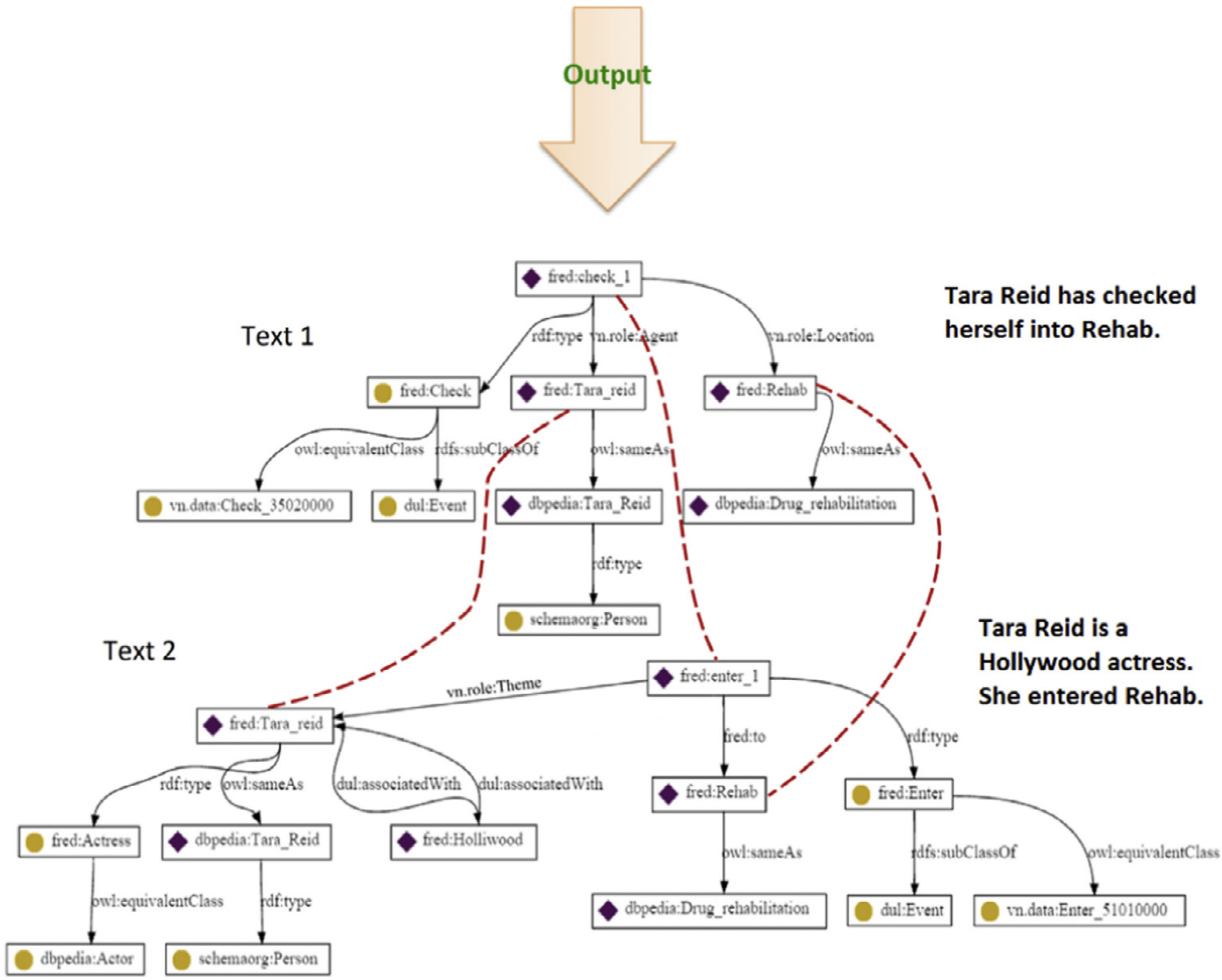

Fig. 5. Reconciled graph for sentences in Fig. 4 after the execution of MERGILO. The example has been taken from [2].

where $len(c_1, c_2)$ is the shortest path between the two nodes c_1 and c_2. For example, according to Fig. 7, the similarity between the frames *Invading* and *Besieging* would be 0.33 because $len(c_1, c_2) = 2$. In Fig. 2, the similarity between the roles *Assailant.Beseiging* and *Assailant.Defend* is 0.33. It is important to mention here that the similarity between *Conqueror.Conquering* and *Assailant.Attack* will be 0.14. This similarity is obtained because of the generic roles defined by Framester i.e., `framesterrole:Agent` otherwise the similarity between the two roles will be 0.

Wu-Palmers Similarity [54] calculates the similarity by considering the depths of the two nodes in the taxonomy and their least common subsumer. Let c_1 and c_2 be two nodes in the taxonomy then the least common subsumer of the two nodes is represented as $lcs(c_1, c_2)$.

Finally, the Wu-Palmer's similarity between two nodes c_1 and c_2 is given as follows:

$$sim_{wup}(c_1, c_2) = \frac{2 * depth(lcs(c_1, c_2))}{depth(c_1) + depth(c_2)} \quad (3)$$

Here the $lcs(Invading, Besieging) = Attack$. The Wu-Palmer similarity between the frames *Invading* and *Besieging* would be 0.8.

Leacock-Chodorow Similarity [55] takes into account the shortest path between two nodes and the depth of the taxonomy.

$$sim_{lc}(c_1, c_2) = -log\left(\frac{len(c_1, c_2) + 1}{2 * D}\right) \quad (4)$$

Where $len(c_1, c_2)$ is the shortest path between the two nodes c_1 and c_2 and D is the maximum depth of the taxonomy. The Leacock-Chodorow (LC) similarity between the frames *Invading* and *Besieging* would be 0.522.

5.2. Frame embeddings using RDF2vec

To learn latent numerical representation of the frames and roles in the FrameNet graph, we follow the RDF2Vec approach. First we transform the graph into a set of sequences of entities, which is then fed into a neural language models, resulting into vector representation of all the nodes in the graph in a latent feature space. The algorithm follows both the unique name assumption (UNA) and the open world assumption (OWA).

Definition 1. An RDF graph is a labeled graph G = (V, E), where V is a set of vertices, and E is a set of directed edges, where each vertex $v \in V$ is identified by a unique identifier, and each edge $e \in E$ is labeled with a label from a finite set of edge labels.

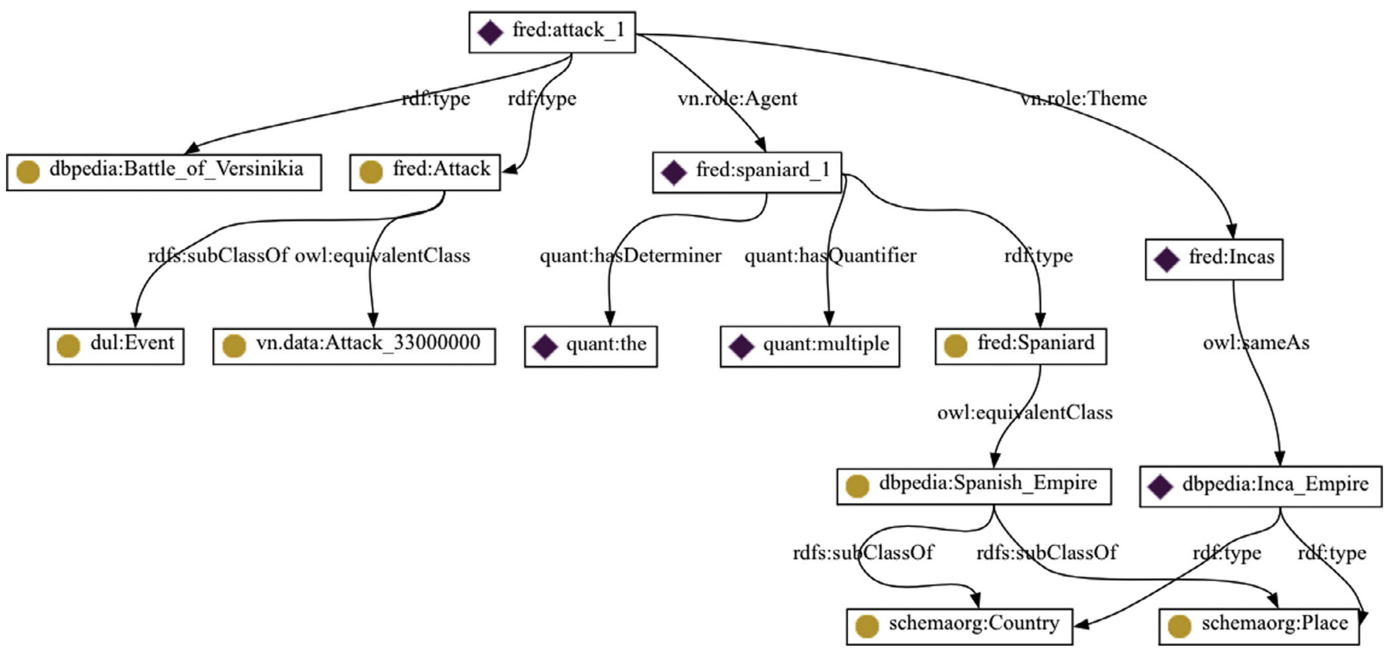

Fig. 6. FRED Knowledge Graph for the Sentence *The Incas were attacked by the Spaniards.*

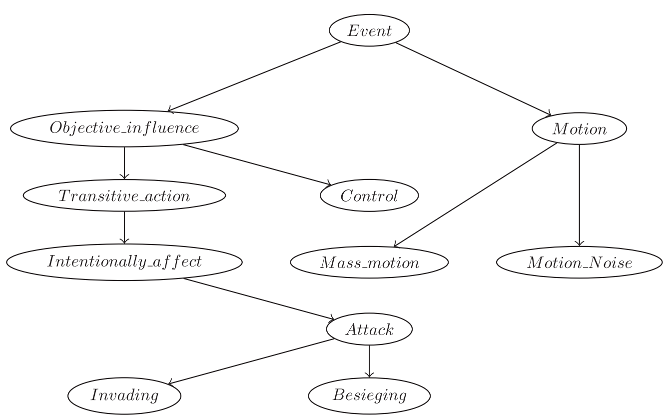

Fig. 7. A part of FrameNet graph using only inheritance relation.

To convert the graph into a set of sequences of entities we use two approaches, i.e., graph walks and Weisfeiler-Lehman Subtree RDF Graph Kernels. The objective of the conversion functions is for each vertex $v \in V$ to generate a set of sequences S_v, where the first token of each sequence $s \in S_v$ is the vertex v followed by a sequence of tokens, which might be edge labels, vertex identifiers, or any substructure extracted from the RDF graph, in an order that reflects the relations between the vertex v and the rest of the tokens, as well as among those tokens.

In the first approach, given a graph $G = (V, E)$, for each vertex $v \in V$, we generate all graph walks P_v of depth d rooted in vertex v. To generate the walks, we use the breadth-first algorithm. In the first iteration, the algorithm generates paths by exploring the direct outgoing edges of the root node v_r. The paths generated after the first iteration will have the following pattern $v_r \rightarrow e_i$, where $e_i \in E_{v_r}$, and E_{v_r} is the set of all outgoing edges from the root node v_r. In the second iteration, for each of the previously explored edges, the algorithm visits the connected vertices. The paths generated after the second iteration will follow the following pattern $v_r \rightarrow e_i \rightarrow v_i$. The algorithm continues until d iterations are reached. The final set of sequences for the given graph G is the union of the sequences of all the vertices $P_G = \bigcup_{v \in V} P_v$.

In the second approach, we use the subtree RDF adaptation of the Weisfeiler-Lehman algorithm presented in [56,57]. The Weisfeiler-Lehman Subtree graph kernel is a state-of-the-art, efficient kernel for graph comparison [58]. The kernel computes the number of sub-trees shared between two (or more) graphs by using the Weisfeiler-Lehman test of graph isomorphism. This algorithm creates labels representing subtrees in h iterations. The rewriting procedure of Weisfeiler-Lehman works as follows: (i) the algorithm creates a multiset label for each vertex based on the labels of the neighbors of that vertex; (ii) this multiset is sorted and together with the original label concatenated into a string, which is the new label; (iii) for each unique string a new (shorter) label replaces the original vertex label; (iv) at the end of each iteration, each label represents a unique full subtree.

There are two main modifications of the original Weisfeiler-Lehman graph kernel algorithm in order to be applicable on RDF graphs, as explained in [56,57]. The algorithm takes as input the RDF graph $G = (V, E)$, a labeling function l, which returns a label of a vertex or edge in the graph based on an index, the subraph depth d and the number of iterations h. The algorithm returns the labeling functions for each iteration l_0 to l_h, and a label dictionary f. Furthermore, the neighborhood $N(v) = (v', v) \in E$ of a vertex is the set of edges going to the vertex v and the neighborhood $N((v, v')) = v$ of an edge is the vertex that the edge comes from. The procedure of converting the RDF graph to a set of sequences of tokens works as follows: (i) for a given graph $G = (V, E)$, we define the Weisfeiler-Lehman algorithm parameters, i.e., the number of iterations h and the vertex subgraph depth d, which defines the subgraph in which the subtrees will be counted for the given vertex; (ii) after each iteration, for each vertex $v \in V$ of the original graph G, we extract all the paths of depth d within the subgraph of the vertex v on the relabeled graph. We set the original label of the vertex v as the starting token of each path, which is then considered as a sequence of tokens. The sequences after each iteration will have the following pattern $v_r \rightarrow l_n(e_i, j) \rightarrow l_n(v_i, j)$, where l_n returns the label of the edges and the vertices in the n^{th} iteration. The sequences could also be seen as $v_r \rightarrow T_1 \rightarrow T_1 \ldots T_d$, where T_d is a subtree that appears on depth d in the vertex's subgraph; (iii) we repeat step (ii) until the maximum iterations h are reached. (iv)

The final set of sequences is the union of the sequences of all the vertices in each iteration $P_G = \bigcup_{i=1}^{h} \bigcup_{v \in V} P_v$. In the RDF2vec with random paths the cycles are not addressed, i.e., a walk can contain a cycle. However, the experiments show that cycles are not causing a problem. For the kernels, again cycles can exist, but the label of the node contains also the level where the node appears, therefore the cycle is not a problem. In case of any cycles, explicitly breaking the cycles is trivial, i.e., it is just a matter of removing predicates that lead to an already visited node. In case of subsumption hierarchy of the roles, no cycles exist, however, in case of FrameNet graph the experimentation does not seem to cause any issues.

Once the set of sequences of entities is extracted, we build a word2vec model. Word2vec is a particularly computationally-efficient two-layer neural net model for learning word embeddings from raw text. There are two different algorithms, the Continuous Bag-of-Words model (CBOW) and the Skip-Gram model. The CBOW model predicts target words from context words within a given window. The input layer is comprised from all the surrounding words for which the input vectors are retrieved from the input weight matrix, averaged, and projected in the projection layer. Then, using the weights from the output weight matrix, a score for each word in the vocabulary is computed, which is the probability of the word being a target word.

The skip-gram model does the inverse of the CBOW model and tries to predict the context words from the target words.

Once the training is finished, semantically similar nodes appear close to each other in the feature space. Therefore, the problem of calculating the similarity between two nodes is a matter of calculating the distance between two instances in the given feature space. To do so, we use the standard cosine similarity measure, which is applied on the vectors of the entities. Formally, the similarity between two nodes c_1 and c_2, with vectors V_1 and V_2, is calculated as the cosine similarity between the vectors V_1 and V_2:

$$sim(c_1, c_2) = \frac{V_1 \cdot V_2}{||V_1|| \cdot ||V_2||} \tag{5}$$

6. Experimentation

We conducted several experiments to evaluate the feasibility of our approach. We built on top of the EECB 1.0 [19] gold standard for CCR (cluster 1) and transferred the coreferences between mentions into coreferences between entities with a semi-automatic process. The EECB gold standard is an extension of ECB [18], a corpus annotated with event coreferences, that also contains entity coreference annotations. ECB contains text found through Google Search that was annotated with mentions, events and their times, locations, human and non-human participants as well as within and cross-document event and entity coreference information. We chose this corpus because our tool aligns both entities and events. As mentioned in [2] we performed the following operations to build the corpus:

– build the RDF graph of each document using FRED;
– map RDF entities with mentions in the EECB gold standard;
– build clusters of entities from clusters of mentions.

The hardest task was to establish the correspondence between entities and mentions. To do that, we took advantage of entity-associated text spans generated by FRED during the construction of the RDF graph. Each text span maintains the character offset of the part of original text associated to an entity. Often this text span differs from the corresponding mention in the gold standard. For example, in the following sentence: *Tara Reid, 33, who starred in 'American Pie' and appeared on U.S. TV show 'Scrubs', has entered the Promises Treatment Center*, FRED creates an entity **fred:Tara_reid** and connects it to the text span corresponding to *Tara Reid*. In contrast, in the EECB gold standard the whole text *Tara Reid, 33, who starred in 'American Pie' and appeared on 17 U.S. TV show 'Scrubs'* is associated to a mention that refers to *Tara Reid*. In this example the text span given by FRED is wholly contained in the EECB mention, but this is not always true in general. Indeed containment is neither a necessary nor sufficient condition for a FRED's text span and an EECB mention to correspond. To solve the mapping we used the same process mentioned in [2] where CrowdFlower[11] has been leveraged to recruit a number of workers and assign them tasks to establish the correspondence between mentions.

Therefore, we aligned pairs of documents from the corpus in all possible ways, and evaluated the results of each pair (171 pairs in total). We employed standard metrics to evaluate the results of our method. In particular, we employ the following metrics:

– MUC [59]: Link-based metric that quantifies the number of merges necessary to cover predicted and gold clusters. Precision, recall and F1-measures are given by: $P = \frac{\sum(|S_i| - |p(S_i)|)}{\sum(|S_i - 1|)}$; $R = \frac{\sum(|G_i| - |p(G_i)|)}{\sum(|G_i - 1|)}$; $F1 = \frac{2 \times PR}{P+R}$, where G_i is a gold mention cluster, $p(G_i)$ is a partition of G_i, S_i is a system mention cluster and $p(S_i)$ is a partition of S_i.
– B^3 [60]: Mention-based metric that quantifies the overlap between predicted and gold clusters for a given mention. Precision, recall and F1-measures are computed as following: $P = \sum_i \frac{|G_{m_i} \cap S_{m_i}|}{S_{m_i}}$, $R = \sum_i \frac{|G_{m_i} \cap S_{m_i}|}{G_{m_i}}$ and $F1 = \frac{2 \times PR}{P+R}$, where G_{m_i} is the gold cluster of mention m_i and S_{m_i} is the system cluster of mention m_i.
– CEAFM (Constrained Entity Aligned F-measure Mention-based) [61]: Mention-based metric based on a one-to-one alignment between gold and predicted clusters. For best alignment $g^* = argmax_{g \in G_m} \phi(g)$ where S is the system mention clusters, G is the gold mention clusters to S and $\phi(g)$ is the total similarity of g, a one-to-one mapping from G. Precision, recall and F1-measures are given as following: $P = \frac{\phi(g^*)}{\sum_i \phi(S_i, S_i)}$, $R = \frac{\phi(g^*)}{\sum_i \phi(G_i, G_i)}$, and $F1 = \frac{2 \times PR}{P+R}$. For CEAFM, we use $\phi(G, S) = |G \cap S|$.
– CEAFE (Constrained Entity Aligned F-measure Entity-Based) [61]: Entity-based metric based on a one-to-one alignment between gold and predicted clusters. For best alignment $g^* = argmax_{g \in G_m} \phi(g)$ where S is the system mention clusters, G is the gold mention clusters to S and $\phi(g)$ is the total similarity of g, a one-to-one mapping from G. Precision, recall and F1-measures are given as following: $P = \frac{\phi(g^*)}{\sum_i \phi(S_i, S_i)}$, $R = \frac{\phi(g^*)}{\sum_i \phi(G_i, G_i)}$, and $F1 = \frac{2 \times PR}{P+R}$. For CEAFE, we use $\phi(G, S) = \frac{2 \times |R \cap S|}{|R| + |S|}$.
– BLANC (Bilateral Assessment of NounPhrase Coreference) [62]: Rand-index-based metric that considers both coreference and non-coreference links. Precision, recall and F1-measures are given as following: $P_c = \frac{rc}{rc+wc}$, $P_n = \frac{rn}{rn+wn}$, $R_c = \frac{rc}{rc+wn}$, $R_n = \frac{rn}{rn+wc}$, $F_c = \frac{2 \times P_c \times R_c}{P_c + R_c}$, $F_n = \frac{2 \times P_n \times R_n}{P_n + R_n}$, $BLANC = \frac{F_c + F_n}{2}$, where rc is the number of correct coreference links, wc is the number of incorrect coreference links, rn is the number of correct non-coreference links, wn is the number of incorrect non-coreference links.

In our experiments we compared the results of MERGILO (which we considered as the baseline) against the method we are proposing in this paper which extends MERGILO by leveraging semantic frame theory (which we consider as MERGILO plus frame similarities). Table 1 shows the results for the baseline method and the results of the extended MERGILO using different models. Due to space constraints, we report only the results with the best thresholds and models found among all the combinations (clearly,

[11] http://www.crowdflower.com.

Table 1
F1 score for Graph Walks, Graph Kernels with Framester Roles and FrameNet Roles.

		muc	bcub	ceafm	blanc	ceafe
MERGILO Baseline		24.05	17.36	28.61	10.70	26.20
Similarity Measures						
Wu-Palmer		27.14	19.91	31.91	**12.81**	29.41
Path		27.16	19.93	31.85	12.73	29.38
Leacock Chodorow		27.04	19.80	31.74	12.77	29.21
Graph walks (with Framester roles)						
Frame2Vec	**Role2Vec**	**muc**	**bcub**	**ceafm**	**blanc**	**ceafe**
CBOW_200	CBOW_200	27.34	**19.99**	32.15	12.66	29.82
CBOW_200	SG_800	27.23	19.89	32.01	12.63	29.66
CBOW_200	SG_500	23.13	16.57	26.99	10.46	24.82
CBOW_200	CBOW_500	27.23	19.89	32.01	12.63	29.66
CBOW_500	SG_800	27.28	19.90	31.96	12.65	29.54
SG_200	SG_800	26.76	19.79	31.73	12.58	29.32
SG_500	SG_800	27.08	19.97	31.99	12.69	29.54
Graph walks (with Framenet roles)						
Frame2Vec	**Role2Vec**	**muc**	**bcub**	**ceafm**	**blanc**	**ceafe**
CBOW_200	CBOW_200	27.34	**19.99**	32.15	12.66	29.82
CBOW_200	SG_800	**27.38**	19.97	**32.29**	12.69	**29.98**
CBOW_200	SG_500	27.28	19.95	31.99	12.69	29.54
CBOW_200	CBOW_500	27.09	19.03	29.95	11.91	28.97
CBOW_500	SG_500	26.90	19.68	31.58	12.60	29.08
SG_200	SG_500	26.87	19.57	31.33	12.10	29.01
SG_500	SG_500	26.85	19.45	31.12	12.08	28.98
Graph kernels (with Framenet roles)						
Frame2Vec	**Role2Vec**	**muc**	**bcub**	**ceafm**	**blanc**	**ceafe**
CBOW_200	CBOW_200	26.76	19.57	31.50	12.45	29.06
CBOW_200	CBOW_500	26.76	19.57	31.50	12.45	29.06
CBOW_200	SG_200	26.70	19.52	31.45	12.40	28.99
CBOW_200	SG_500	26.70	19.52	31.45	12.40	28.99
CBOW_500	CBOW_200	26.76	19.51	31.45	12.45	28.96
SG_200	CBOW_200	26.86	19.62	31.67	12.48	29.18
SG_500	CBOW_200	26.90	19.68	31.58	12.60	29.08

also the baseline results are reported with the best thresholds found among all the combinations). More in detail, Table 1 shows the results for Wu-Palmer's similarity, Path similarity and Leacock-Chodorow similarity and the results for cosine similarity using *(i)* graph walks with Framester roles, *(ii)* graph walks with FrameNet roles and *(iii)* graph kernels with FrameNet roles respectively. Here `Frame2Vec` refers to the vector representations generated for FrameNet frames and `Role2Vec` refers to the vector representations generated for frame elements i.e., semantic roles.

For the first approach with graph walks, for each entity in the FrameNet graph we generate 200 and 500 random walks, each of depth 4 and 8. For each entity in the subsumption hierarchy of roles we generate 400 random walks with depth 4. For the Weisfeiler-Lehman algorithm, we use $h = 2$ iterations and subgraph depth $d = 2$, and after each iteration of the algorithm we extract all walks for each entity with the same depth. We use these sequences to build both CBOW and Skip-Gram models with the following parameters: window size = 5; number of iterations = 10; negative sampling for optimization; negative samples = 25; with average input vector for CBOW. We experiment with 200 and 500 dimensions for the entities' vectors. We have built on top of the original MERGILO code, which was released as a Python tool[12] and, on top of FredLib[13]. We used IBM ILOG CPLEX 12.6.1 for solving the Integer Linear Program and run the experiments on a MacOs server with 6-Core Intel Xeon E5 3.50GHz and 64GB of RAM. Without taking into account the linear problem fed to CPLEX, which might take time in the order of minutes to be solved with basic settings (there are several optimization techniques that can be applied to improve CPLEX performance, but this is out of the scope of the present paper), our tool takes few seconds to be run for a given pair of texts.

As shown in the table, it can be noticed that each model used for graph walks and graph kernels performs better than the MERGILO baseline for all the considered metrics, showing a clear advantage of using the proposed frame similarities to reconcile knowledge graphs.

Although the rationale of our approach was to show that using similarities based on the graph structure of semantic frames and the subsumption hierarchy of semantic roles as defined in Framester outperformed the baseline (Mergilo), we provide more insights into the results. The Wu-Palmer, Path and Leacock Chodorow measures use the inheritance relations only whereas Frame2Vec employs either graph walks or graph kernels over the FrameNet frame graph as well as subsumption hierarchy of FrameNet roles using either only FrameNet roles or improved subsumption hierarchy of FrameNet roles as introduced in Framester. Based on these settings, vector representations are generated which are further used for computing the cosine similarity. In general, Frame2Vec, for its intrinsic construction, exploits more semantics than the other similarity measures (Wu-Palmer, Path and Leacock Chodorow); for such a reason, Frame2Vec provides the highest results for almost each evaluation measure except for BLANC. BLANC is more sensitive to wrong assignments when clusters of mentions are larger, since a wrong assignment lead to a higher number of wrong non-coreference links. Therefore, although BLANC is case-by-case coherent with the other measures (when BLANC is low, the other measures are low and vice-versa), in the few cases when Frame2Vec is outperformed by other measures (Wu-Palmer, Path and Leacock Chodorow), the BLANC measure, and in particular the contribution given by non-coreference link, gives a much smaller score. These cases influence the overall average and for this reason in Table 1 BLANC seems to have a different behaviour than the other measures.

The generated models i.e., vector representations of FrameNet frames generated using FrameNet graph and subsumption hierarchy of FrameNet roles using RDF2Vec are freely available on-line[14].

7. Conclusions

This paper presents an extension of MERGILO, a tool for reconciling knowledge graphs using graph alignment and word similarity. This study exploits Framester, a linguistic data hub formulated using a novel formal semantics for frames, in order to enhance semantic interoperability between linguistic resources. This paper introduces several ways for improving the basic MERGILO pipeline to deal with event-based knowledge reconciliation. In particular, several path-based similarity measures for frames and semantic roles were used. Following the approach *RDF2Vec*, graph-based frame embeddings were generated. Our experimentation shows that the introduced approach improves over the MERGILO baseline.

Ongoing work concentrates on practical applications of frame embeddings in real systems, such as news series integration, knowledge graph evolution with robust event reconciliation (e.g. in streaming of texts where we expect relatedness or updates), or conflict detection across texts describing similar facts with different narratives or perspectives. In order to deal with these challenging real world use cases, we will test optimization procedures for CPLEX in order to achieve scalability. We will also explore the scenario of existing corpora, benchmarks, gold standards, and challenges related to the aforementioned tasks.

[12] http://wit.istc.cnr.it/stlab-tools/mergilo.
[13] http://wit.istc.cnr.it/stlab-tools/fred/fredlib.
[14] http://lipn.univ-paris13.fr/~alam/Frame2Vec/.

As a future perspective, we also want to further apply the presented approach to NLP tasks such as text summarization or dialogue, e.g. taking advantage of frame similarities. We also want to introduce information-content-based similarity measures along with corpus-based frame embeddings.

Acknowledgment

The research leading to these results has received funding from the European Union Horizon 2020 the Framework Programme for Research and Innovation (2014-2020) under grant agreement 643808 Project MARIO Managing active and healthy aging with use of caring service robots as well as by a public grant overseen by the French National Research Agency (ANR) as part of the program "Investissements d'Avenir" (reference: ANR-10-LABX-0083). Moreover, the authors gratefully acknowledge Sardinia Regional Government for the financial support (Convenzione triennale tra la Fondazione di Sardegna e gli Atenei Sardi Regione Sardegna L.R. 7/2007 annualit 2016 DGR 28/21 del 17.05.201, CUP: F72F16003030002).

References

[1] A. Gangemi, V. Presutti, D.R. Recupero, A.G. Nuzzolese, F. Draicchio, M. Mongiovì, Semantic web machine reading with FRED, Semantic Web J. (2016).
[2] M. Mongiovì, D.R. Recupero, A. Gangemi, V. Presutti, S. Consoli, Merging open knowledge extracted from text with MERGILO, Knowl.-Based Syst. 108 (2016) 155–167, doi:10.1016/j.knosys.2016.05.014.
[3] A. Gangemi, M. Alam, L. Asprino, V. Presutti, D.R. Recupero, Framester: a wide coverage linguistic linked data hub, in: E. Blomqvist, P. Ciancarini, F. Poggi, F. Vitali (Eds.), Knowledge Engineering and Knowledge Management - 20th International Conference, EKAW 2016, Bologna, Italy, November 19–23, 2016, Proceedings, Lecture Notes in Computer Science, 10024, 2016, pp. 239–254, doi:10.1007/978-3-319-49004-5_16.
[4] A.G. Nuzzolese, A. Gangemi, V. Presutti, Gathering lexical linked data and knowledge patterns from FrameNet, in: M.A. Musen, Ó. Corcho (Eds.), Proceedings of the 6th International Conference on Knowledge Capture (K-CAP 2011), June 26–29, 2011, Banff, Alberta, Canada, ACM, 2011, pp. 41–48, doi:10.1145/1999676.1999685.
[5] A. Budanitsky, G. Hirst, Evaluating wordnet-based measures of lexical semantic relatedness, Comput. Linguist. 32 (1) (2006) 13–47, doi:10.1162/coli.2006.32.1.13.
[6] P. Ristoski, H. Paulheim, Rdf2vec: RDF graph embeddings for data mining, in: P.T. Groth, E. Simperl, A.J.G. Gray, M. Sabou, M. Krötzsch, F. Lécué, F. Flöck, Y. Gil (Eds.), The Semantic Web - ISWC 2016 - 15th International Semantic Web Conference, Kobe, Japan, October 17–21, 2016, Proceedings, Part I, Lecture Notes in Computer Science, 9981, 2016, pp. 498–514, doi:10.1007/978-3-319-46523-4_30.
[7] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E.R. Hruschka Jr, T.M. Mitchell, Toward an architecture for never-ending language learning., in: AAAI, 5, 2010, p. 3.
[8] A. Fader, S. Soderland, O. Etzioni, Identifying relations for open information extraction, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, in: EMNLP '11, Association for Computational Linguistics, Stroudsburg, PA, USA, 2011, pp. 1535–1545. http://dl.acm.org/citation.cfm?id=2145432.2145596.
[9] J. Hoffart, F.M. Suchanek, K. Berberich, G. Weikum, Yago2: a spatially and temporally enhanced knowledge base from wikipedia, Artif. Intell. 194 (2013) 28–61.
[10] X. Dong, E. Gabrilovich, G. Heitz, W. Horn, N. Lao, K. Murphy, T. Strohmann, S. Sun, W. Zhang, Knowledge vault: a web-scale approach to probabilistic knowledge fusion, in: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, in: KDD '14, ACM, New York, NY, USA, 2014, pp. 601–610, doi:10.1145/2623330.2623623.
[11] S. Brin, Extracting patterns and relations from the world-wide web, in: Proceedings of the 1998 International Workshop on the Web and Databases (WebDB98).
[12] E. Agichtein, L. Gravano, Snowball: extracting relations from large plain-text collections, in: Proceedings of the Fifth ACM Conference on Digital Libraries, in: DL '00, ACM, New York, NY, USA, 2000, pp. 85–94, doi:10.1145/336597.336644.
[13] M. Banko, M.J. Cafarella, S. Soderland, M. Broadhead, O. Etzioni, Open information extraction from the web, in: Proceedings of the 20th International Joint Conference on Artifical Intelligence, in: IJCAI'07, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2007, pp. 2670–2676. http://dl.acm.org/citation.cfm?id=1625275.1625705.
[14] S. Dutta, G. Weikum, Cross-document co-reference resolution using sample-based clustering with knowledge enrichment, Trans. Assoc. Comput.Linguist. 3 (1) (2015) 15–28.
[15] D. Rao, P. McNamee, M. Dredze, Streaming cross document entity coreference resolution, in: 23rd International Conference on Computational Linguistics, in: COLING 2010, ACL, Stroudsburg, PA, USA, 2010, pp. 1050–1058.
[16] S. Singh, A. Subramanya, F. Pereira, A. McCallum, Large-scale cross-document coreference using distributed inference and hierarchical models, in: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, in: HLT '11, 1, ACL, Stroudsburg, PA, USA, 2011, pp. 793–803.
[17] N. Andrews, J. Eisner, M. Dredze, Robust entity clustering via phylogenetic inference, in: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, 1, ACL, Stroudsburg, PA, USA, 2014.
[18] C. Bejan, S. Harabagiu, Unsupervised event coreference resolution, Comput. Linguist. 40 (2) (2014) 311–347, doi:10.1162/COLI_a_00174.
[19] H. Lee, M. Recasens, A. Chang, M. Surdeanu, D. Jurafsky, Joint entity and event coreference resolution across documents, in: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, in: EMNLP-CoNLL '12, Association for Computational Linguistics, Stroudsburg, PA, USA, 2012, pp. 489–500. http://dl.acm.org/citation.cfm?id=2390948.2391006.
[20] J. Euzenat, P. Shvaiko, Ontology matching, Second, Springer-Verlag, Heidelberg, 2013.
[21] F.M. Suchanek, S. Abiteboul, P. Senellart, PARIS: Probabilistic alignment of relations, instances, and schema, in: Proceedings of the VLDB Endowment, 5, VLDB Endowment, 2011, pp. 157–168.
[22] S. Lacoste-Julien, K. Palla, A. Davies, G. Kasneci, T. Graepel, Z. Ghahramani, Sigma: simple greedy matching for aligning large knowledge bases, in: KDD2013, ACM, New York, USA, 2013, pp. 572–580.
[23] J.T. Vogelstein, J.M. Conroy, V. Lyzinski, L.J. Podrazik, S.G. Kratzer, E.T. Harley, D.E. Fishkind, R.J. Vogelstein, C.E. Priebe, Fast approximate quadratic programming for graph matching, PLoS ONE 10 (4) (2015). e0121002. PMID: 25886624.
[24] G.W. Klau, A new graph-based method for pairwise global network alignment, BMC Bioinform. 10 (1) (2009) 1–9.
[25] M. Mongiovì, R. Sharan, Global alignment of protein–protein interaction networks, in: Data Mining for Systems Biology: Methods and Protocols, 2013, pp. 21–34.
[26] D.R. Recupero, Efficient graph matching, in: Encyclopedia of Data Warehousing and Mining, 2009, pp. 736–743.
[27] J.A. Bullinaria, J.P. Levy, Extracting semantic representations from word co-occurrence statistics: stop-lists, stemming, and svd, Behav. Res. Methods 44 (3) (2012) 890–907, doi:10.3758/s13428-011-0183-8.
[28] S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer, R. Harshman, Indexing by latent semantic analysis, J. Am. Soc. Inf.Sci. 41 (6) (1990) 391–407, doi:10.1002/(SICI)1097-4571(199009)41:6⟨391::AID-ASI1⟩3.0.CO;2-9.
[29] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, CoRR, abs/1301.3781, 2013.
[30] Q.V. Le, T. Mikolov, Distributed representations of sentences and documents, in: Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21–26 June 2014, in: JMLR Workshop and Conference Proceedings, vol. 32, JMLR.org, 2014, pp. 1188–1196. http://jmlr.org/proceedings/papers/v32/le14.html.
[31] J. Pennington, R. Socher, C.D. Manning, Glove: global vectors for word representation, in: A. Moschitti, B. Pang, W. Daelemans (Eds.), Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25–29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL, ACL, 2014, pp. 1532–1543. http://aclweb.org/anthology/D/D14/D14-1162.pdf.
[32] I. Iacobacci, M.T. Pilehvar, R. Navigli, Sensembed: learning sense embeddings for word and relational similarity, 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 1: Long Papers, The Association for Computer Linguistics, 2015, pp. 95–105. http://aclweb.org/anthology/P/P15/P15-1010.pdf.
[33] Y.-N. Chen, W.Y. Wang, A.I. Rudnicky, Leveraging frame semantics and distributional semantics for unsupervised semantic slot induction in spoken dialogue systems., in: SLT, IEEE, 2014, pp. 584–589.
[34] Y. Chen, W.Y. Wang, A.I. Rudnicky, Jointly modeling inter-slot relations by random walk on knowledge graphs for unsupervised spoken language understanding, in: R. Mihalcea, J.Y. Chai, A. Sarkar (Eds.), NAACL HLT 2015, The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, Colorado, USA, May 31, - June 5, 2015, The Association for Computational Linguistics, 2015, pp. 619–629. http://aclweb.org/anthology/N/N15/N15-1064.pdf.
[35] W.Y. Wang, D. Yang, That's so annoying!!!: a lexical and frame-semantic embedding based data augmentation approach to automatic categorization of annoying behaviors using #petpeeve tweets, in: L. Màrquez, C. Callison-Burch, J. Su, D. Pighin, Y. Marton (Eds.), Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17–21, 2015, The Association for Computational Linguistics, 2015, pp. 2557–2563. http://aclweb.org/anthology/D/D15/D15-1306.pdf.
[36] M. Nickel, K. Murphy, V. Tresp, E. Gabrilovich, A review of relational machine learning for knowledge graphs, Proc. IEEE 104 (1) (2016) 11–33, doi:10.1109/JPROC.2015.2483592.
[37] K. Kipper Schuler, Verbnet: A Broad-coverage, Comprehensive Verb Lexicon, Ph.D. thesis, Philadelphia, PA, USA, 2005. AAI3179808.
[38] C. Fellbaum (Ed.), Wordnet: An Electronic Lexical Database, MIT Press, 1998.

[39] C.F. Baker, C.J. Fillmore, J.B. Lowe, The Berkeley frameNet project, in: C. Boitet, P. Whitelock (Eds.), 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, COLING-ACL '98, August 10–14, 1998, Université de Montréal, Montréal, Quebec, Canada. Proceedings of the Conference., Morgan Kaufmann Publishers / ACL, 1998, pp. 86–90. http://aclweb.org/anthology/P/P98/P98-1013.pdf.

[40] R. Navigli, S.P. Ponzetto, Babelnet: the automatic construction, evaluation and application of a wide-Coverage multilingual semantic network, Artif. Intell. 193 (2012) 217–250.

[41] M.L. de Lacalle, E. Laparra, G. Rigau, Predicate Matrix: extending SemLink through WordNet mappings, in: N. Calzolari, K. Choukri, T. Declerck, H. Loftsson, B. Maegaard, J. Mariani, A. Moreno, J. Odijk, S. Piperidis (Eds.), Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014), Reykjavik, Iceland, May 26–31, 2014., European Language Resources Association (ELRA), 2014, pp. 903–909. http://www.lrec-conf.org/proceedings/lrec2014/summaries/589.html.

[42] C.J. Fillmore, Frame semantics and the nature of language, Ann. N. Y. Acad. Sci. 280 (1) (1976) 20–32.

[43] A. Gangemi, What's in a Schema?, Cambridge University Press, Cambridge, UK, pp. 144–182.

[44] J. Lehmann, C. Bizer, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, S. Hellmann, DBpedia - a Crystallization point for the web of data, J. Web Semantics 7 (3) (2009) 154–165, doi:10.1016/j.websem.2009.07.002.

[45] A.G. Nuzzolese, A. Gangemi, V. Presutti, P. Ciancarini, A. Musetti, Automatic typing of DBpedia entities, in: Proc. of the International Semantic Web Conference (ISWC), Boston, MA, US, 2012.

[46] R. Speer, C. Havasi, Representing general relational knowledge in conceptnet 5., in: LREC, 2012, pp. 3679–3686.

[47] M. Cuadros, L. Padró, G. Rigau, Highlighting relevant concepts from topic signatures, in: N. Calzolari, K. Choukri, T. Declerck, M.U. Dogan, B. Maegaard, J. Mariani, J. Odijk, S. Piperidis (Eds.), Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012), Istanbul, Turkey, May 23–25, 2012, European Language Resources Association (ELRA), 2012, pp. 3841–3848. http://www.lrec-conf.org/proceedings/lrec2012/summaries/374.html.

[48] S. Baccianella, A. Esuli, F. Sebastiani, SentiWordNet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining, in: N. Calzolari, K. Choukri, B. Maegaard, J. Mariani, J. Odijk, S. Piperidis, M. Rosner, D. Tapias (Eds.), Proceedings of the International Conference on Language Resources and Evaluation, LREC 2010, 17–23 May 2010, Valletta, Malta, European Language Resources Association, 2010. http://www.lrec-conf.org/proceedings/lrec2010/summaries/769.html.

[49] J. Staiano, M. Guerini, Depeche mood: a Lexicon for emotion analysis from crowd annotated news, in: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22–27, 2014, Baltimore, MD, USA, Volume 2: Short Papers, The Association for Computer Linguistics, 2014, pp. 427–433. http://aclweb.org/anthology/P/P14/P14-2070.pdf.

[50] J. Bos, M. Nissim, Combining discourse representation theory with framenet, in: Frames, Corpora, and Knowledge Representation, R. Rossini Favretti and Bononia University Press, 2008, pp. 169–183.

[51] A. Gangemi, V. Presutti, Ontology design patterns, in: S. Staab, R. Studer (Eds.), Handbook on Ontologies, International Handbooks on Information Systems, Springer, 2009, pp. 221–243, doi:10.1007/978-3-540-92673-3_10.

[52] J. Bos, Wide-coverage semantic analysis with boxer, in: J. Bos, R. Delmonte (Eds.), Semantics in Text Processing, College Publications, 2008, pp. 277–286.

[53] M. Pennacchiotti, M. Wirth, Measuring frame relatedness, in: A. Lascarides, C. Gardent, J. Nivre (Eds.), EACL 2009, 12th Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference, Athens, Greece, March 30, - April 3, 2009, The Association for Computer Linguistics, 2009, pp. 657–665. http://www.aclweb.org/anthology/E09-1075.

[54] Z. Wu, M. Palmer, Verbs semantics and lexical selection, in: Proceedings of the 32Nd Annual Meeting on Association for Computational Linguistics, in: ACL '94, Association for Computational Linguistics, Stroudsburg, PA, USA, 1994, pp. 133–138, doi:10.3115/981732.981751.

[55] C. Leacock, M. Chodorow, Combining local context and wordNet similarity for word sense identification, in: C. Fellbaum (Ed.), WordNet: An Electronic Lexical Database., MIT Press, 1998, pp. 265–283.

[56] G.K.D. de Vries, A fast approximation of the Weisfeiler-Lehman graph kernel for RDF data., ECML/PKDD (1), 2013.

[57] G.K.D. de Vries, S. de Rooij, Substructure counting graph kernels for machine learning from rdf data, Web Semantics 35 (2015) 71–84.

[58] N. Shervashidze, P. Schweitzer, E.J. Van Leeuwen, K. Mehlhorn, K.M. Borgwardt, Weisfeiler-lehman graph kernels, J. Mach. Learn. Res. 12 (2011) 2539–2561.

[59] M. Vilain, J. Burger, J. Aberdeen, D. Connolly, L. Hirschman, A model-theoretic coreference scoring scheme, in: Proceedings of the 6th Conference on Message Understanding, Association for Computational Linguistics, 1995, pp. 45–52.

[60] A. Bagga, B. Baldwin, Algorithms for scoring coreference chains, in: The first International Conference on Language Resources and Evaluation Workshop on Linguistics Coreference, 1, Citeseer, 1998, pp. 563–566.

[61] X. Luo, On coreference resolution performance metrics, in: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2005, pp. 25–32.

[62] M. Recasens, E. Hovy, Blanc: implementing the rand index for coreference evaluation, Nat. Lang. Eng. 17 (04) (2011) 485–510.