Frame Embeddings for Event-Based Knowledge Reconciliation

Mehwish Alam[1], Diego Reforgiato Recupero[2,3], Misael Mongiovi[2], Aldo Gangemi[1,2], Petar Ristoski[4]

[1]LIPN, Université Paris 13, France, [2]ISTC-CNR, Rome, Catania, Italy, [3]University of Cagliari, Italy, [4]University of Mannheim, Germany.

Abstract. This paper focuses on reconciling knowledge graphs generated from two text documents about similar events described differently. The proposed approach employs and extends MERGILO, a tool for reconciling knowledge graphs extracted from text, using word similarity and graph alignment. Our approach effectively handles events using Frame Embeddings and Frame Based Similarities. It is evaluated over a coreference resolution task.

1 Introduction

This study addresses the problem of knowledge reconciliation (KR) [4] from the perspective of events. KR is useful in providing a combination of multiple graphs generated by multiple texts describing the same event. This merged graph provides a graph based summary of multiple texts which is more easily comprehensible by users and machines and usable by the algorithms providing interactive exploration of graphs/text analytics through visualization methods.

MERGILO [4] is a tool for reconciling knowledge graphs extracted from text, it first computes the word similarity between the node labels and then performs graph alignment over the whole graphs. When different verbs denote similar events and different agents play slightly different roles, the string matching techniques as introduced in MERGILO might be not appropriate in the KR process. For overcoming this limitation we use Frame Semantics which describes a situation in the text with the help of frames and roles. For identifying frames and semantic roles of entities in a text we use FRED [3], which generates event-centered knowledge graphs from two different texts. Then, the similarity between these events is computed by calculating the similarity between the corresponding FrameNet frames and semantic roles (frame elements). We adapt WordNet similarity measures to frames and roles and vector based similarities using the FrameNet graph and the subsumption hierarchy of roles as defined in Framester [2]. We follow the approach `RDF2Vec` [5] to generate graph based *frame embeddings*. It uses graph mining algorithms such as graph walks and graph kernels to traverse the graph for generating sequences, which are then fed to a neural model for generating its vector representations. Finally, we show experimentation over Cross-document Coreference Resolution (CCR) reporting significant improvements over a baseline.

2 Event-Based Knowledge Reconciliation

Consider the two sentences: *"The Spaniards conquered the Incas."* and *"The Incas were invaded by the Spaniards."* They are describing the same event in the past using different words i.e., event of an attack or an invasion from Spaniards to Incas. Figure 1 shows the FRED graph of the first sentence. Given two such knowledge graphs, MERGILO first performs graph compression by merging nodes in the same graph. The two compressed graphs are aligned by establishing a 1-1 correspondence between nodes of the two graphs by maximizing a score function, which combines the similarity between aligned nodes and the similarity between aligned edges. In such a case, the similarity between "conquered" and "invaded" is not effective since word similarity is low, although in this context such words describe the same event.

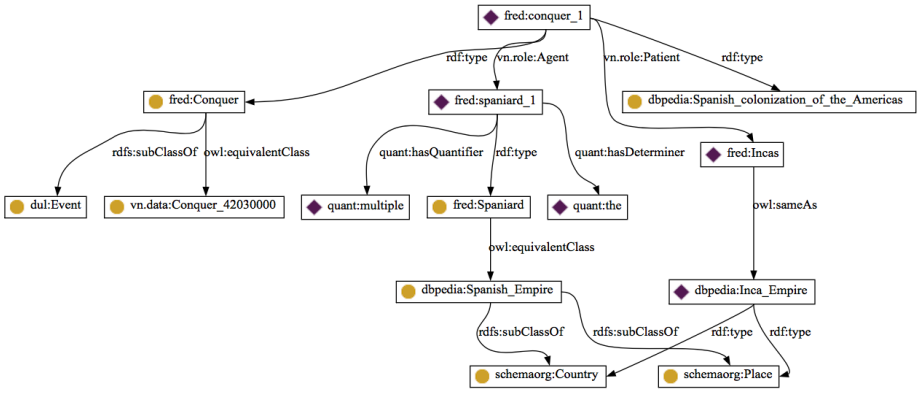

Fig. 1: FRED Knowledge Graph for *The Spaniards conquered the Incas.*

For computing similarity between two nodes containing verb senses, the verb senses are first mapped to frames using Framester mappings. For example, in Figure 1 s_1 =`vn.data`[1]`:Conquer_42030000` and for second sentence we have s_2 =`vn.data:Invade_10000000`. According to Framester mappings, we obtain $s_1 \rightarrow \{Conquering\}$ and $s_2 \rightarrow \{Attack\}$. These nodes are replaced by their corresponding frames. The edges containing the VN-roles are mapped to FN-roles. For example, in Figure 1, the verb sense `vn.data:Conquer_42030000` evokes the roles `vn.role:Agent` and `vn.role:Patient` which are mapped to `fe:Conqueror.conquering` and `fe:Theme.conquering` respectively.

Then the similarities are computed in two ways: *(i)* by considering the taxonomical structure imposed by the "inheritance" relation represented as `fnschema`[2]`:inheritsFrom` in Framester using Path Similarity, Wu-Palmers Similarity, Leacock-Chodorow Similarity; *(ii)* using Frame Embeddings.

[1] `prefix vn.data: http://www.ontologydesignpatterns.org/ont/vn/vn31/data/`
[2] `prefix fnschema: http://www.ontologydesignpatterns.org/ont/framenet/tbox/`

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.

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. *(i) Graph Walks:* 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. In the second iteration, for each of the previously explored edges, the algorithm visits the connected vertices. The final set of sequences for the given graph G is the union of the sequences of all the vertices $P_G = \bigcup_{v \in V} P_v$. *(ii) Graph Kernels:* it computes the number of sub-trees shared between two or more graphs by using the Weisfeiler-Lehman [1] test of graph isomorphism. This algorithm creates labels representing subtrees.

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. Once the training is finished, the cosine similarity is computed between two frames and roles.

3 Experimentation

The experimentations were conducted for the task of Cross-document Coreference Resolution on RDF graphs, which focuses on associating RDF nodes about a same entity (object, person, concept, etc.) across different RDF graphs generated from text. The dataset used for the experimentation was obtained by the EECB dataset which specifies coreferent mentions (text fragment). Our dataset was obtained by generating RDF graphs using FRED and associating text mensions to graph nodes by manual annotations. Following are the metrics used for evaluations[3]: (1) MUC is a link-based metric that quantifies the number of merges necessary to cover predicted and gold clusters (ground truth). (2) B^3 is a mention-based metric that quantifies the overlap between predicted and gold clusters for a given mention. (3) CEAFM (Constrained Entity Aligned F-measure Mention-based) measures the number of corresponding mentions in the optimal

[3] The formulas of Precision, Recall and F1 are suppressed because of space constraints.

one-to-one alignment between gold and predicted clusters. (4) CEAFE (Constrained Entity Aligned F-measure Entity-Based) measures the overlap between gold and predicted clusters in their optimal one-to-one alignment.

MERGILO was considered as the baseline. 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, *(ii)* graph kernels. Here `Frame2Vec` and `Role2Vec` refers to the vector representations generated for FrameNet frames and frame elements i.e., semantic roles respectively. We further built 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, 500 and 800 dimensions. These results are compared with MERGILO. Each model used for graph walks and graph kernels perform better for all the considered metrics, showing a clear advantage of using the proposed approach. The generated models are freely available on-line[4].

		muc	**bcub**	**ceafm**	**ceafe**
MERGILO Baseline		24.05	17.36	28.61	26.20
Similarity Measures					
Wu-Palmer		27.14	19.91	31.91	29.41
Path		27.16	19.93	31.85	29.38
Leacock Chodorow		27.04	19.80	31.74	29.21
Graph Walks					
Frame2Vec	**Role2Vec**	**muc**	**bcub**	**ceafm**	**ceafe**
CBOW_200	CBOW_200	27.34	**19.99**	32.15	29.82
CBOW_200	SG_800	**27.38**	19.97	**32.29**	**29.98**
CBOW_200	SG_500	27.28	19.95	31.99	29.54
Graph Kernels					
Frame2Vec	**Role2Vec**	**muc**	**bcub**	**ceafm**	**ceafe**
CBOW_200	CBOW_200	26.76	19.57	31.50	29.06
CBOW_200	SG_200	26.70	19.52	31.45	28.99
CBOW_200	SG_500	26.70	19.52	31.45	28.99
SG_500	CBOW_200	26.90	19.68	31.58	29.08

Table 1: Results and comparisons between MERGILO and the proposed approach.

4 Perspectives

Ongoing work includes application of frame embeddings in real systems, such as news series integration, knowledge graph evolution with robust event reconciliation (e.g. text streaming) etc.

References

1. G. de Vries and S. de Rooij. Substructure counting graph kernels for machine learning from rdf data. *J. Web Sem.*, 35, 2015.
2. A. Gangemi, M. Alam, L. Asprino, V. Presutti, and D. Reforgiato. Framester: A wide coverage linguistic linked data hub. In *EKAW 2016.*, 2016.
3. A. Gangemi, V. Presutti, D. R. Recupero, A. G. Nuzzolese, F. Draicchio, and M. Mongiovì. Semantic web machine reading with FRED. *Semantic Web*, 8(6), 2017.
4. M. Mongiovì, D. Reforgiato, A. Gangemi, V. Presutti, and S. Consoli. Merging open knowledge extracted from text with MERGILO. *Knowl.-Based Syst.*, 108, 2016.
5. P. Ristoski and H. Paulheim. RDF2Vec: RDF Graph Embeddings for Data Mining. In *ISWC 2016*, 2016.

[4] `http://lipn.univ-paris13.fr/~alam/Frame2Vec/`