Semantic reconciliation of knowledge extracted from text through a novel machine reader

Misael Mongiovì[*]
misael.mongiovi@istc.cnr.it

Diego Reforgiato Recupero[*]
diego.reforgiato@istc.cnr.it

Aldo Gangemi[*†]
aldo.gangemi@istc.cnr.it

Valentina Presutti[*]
valentina.presutti@istc.cnr.it

Andrea Giovanni Nuzzolese[*]
andrea.nuzzolese@istc.cnr.it

Sergio Consoli[*]
sergio.consoli@istc.cnr.it

[*]Semantic Technology Lab, ISTC-CNR
Rome and Catania, Italy

[†]Paris Nord University, Sorbonne Citè
CNRS UMR7030, France

ABSTRACT

This paper describes a novel method for generating and integrating knowledge graphs extracted from multiple natural language sources by FRED, a machine reading tool for generating abstract representations of text documents. This is a key problem in human-robot spoken dialogue interaction, issue which arises from a current research project related to active and healthy ageing using caring service robots where we are involved. The problem is also relevant in many application scenarios requiring the creation and dynamic evolution of a knowledge base, such as automatic news summarisation. Solving this problem requires solving sub-tasks that have only been studied individually, so far. We propose a holistic approach to handle FRED's graphs related to different input texts and output a knowledge graph representing the reconciled knowledge.

Keywords

Semantic reconciliation, knowledge integration, machine reading

1. INTRODUCTION

[1]This paper presents a novel method for integrating knowledge extracted from multiple *natural language* (NL) sources into an *integrated formal representation.* Our tool relies on FRED[2], a machine reading tool for generating abstract representations of text documents, and integrates the generated knowledge by means of an optimization technique for graph alignment. FRED is a tool that extracts knowledge from text and represents it by well-connected RDF graphs with a formal semantic interpretation. FRED links the extracted knowledge to existing linked data and ontologies, so providing the entity-centric grounding to unstructured data and their relations as well. Knowledge related to different input text might refer to the same concepts, events, named entities, etc. To reconcile FRED's knowledge graphs we need to identify when entities (events, persons, organisations, concepts, etc.) present in different texts are referring to the same one.

This problem, referred to as *semantic reconciliation*, is relevant in most application scenarios that require to create and update or evolve a knowledge base from multiple and/or dynamic NL sources, for example: (1) supporting human-machine dialogue in the context of assistive robotics by collecting a user's personal memories, which are provided by NL inputs over time; (2) building an integrated knowledge view, e.g. a summary, about a specific event, by analysing news.

Within active and healthy ageing with use of caring service robots[3], one of the challenges is represented by the lexical/semantic communication/interaction with a person. The robot needs to be able to communicate with humans on a natural language basis as well as to detect, interpret, and express emotional expressions, and to react to such interactions with a behaviour, which adapts and evolve dependently on the environment they live in. The robot learns and evolves thanks to its specific with-humans relationships, but it also exploits a general encyclopaedic background knowledge from the Semantic Web. Moreover, it shares what it learns with the other robots, hence creating a knowledge-sharing virtuous cycle that during time makes them more and more "cultivated" at their starting phase as well as beyond it. In doing so, one of the problem that needs to be addressed is related to the identification and reconciliation of entities when they lie in different parts of the dialogue. Notice that our machine reader FRED is general-purpose and the semantic reconciliation we have built on top works not just with the robot-human dialoguing but in any domain

[1]The research leading to these results has received funding from the European Union Horizons 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.

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

[3]H2020 research project MARIO, `http://www.mario-project.eu/portal/`

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org. *K-CAP* 2015, October 07 - 10, 2015, Palisades, NY, USA. Copyright is held by the owner/author(s). Publication rights licensed to ACM. ACM 978-1-4503-3849-3/15/10 ...$15.00. DOI: http://dx.doi.org/10.1145/2815833.2816945

(news, social networks chatting, comments, etc.).

Let us consider the following news from two different sources: *"Tony awards: "Fun Home" and "Curious incident" big winners."*
and
"On Broadway's biggest night "Fun Home" wins Tony award for Best Musical"

In an ideal scenario, the goal is to automatically produce an integrated knowledge graph[4] such as the one depicted in Figure 1. Solving the problem of semantic reconciliation requires semantic parsing of multiple natural language texts, transforming them to a formal representation, and identifying common vs. different parts in order to reason over an integrated knowledge graph associated with its textual provenance. On the one hand, the problem is very challenging considering that natural language can use very heterogeneous forms for expressing similar knowledge; on the other hand solving it is crucial in order to fully exploit the knowledge produced by FRED. The remainder of the paper is organised as follows: Section 2 discusses relevant related research. Section 3 presents FRED, our machine reader that transforms input text into knowledge representation. Section 4 introduces the method to solve the semantic reconciliation problem. Finally Section 5 is dedicated to the evaluation of the proposed approach and Section 6 ends the paper.

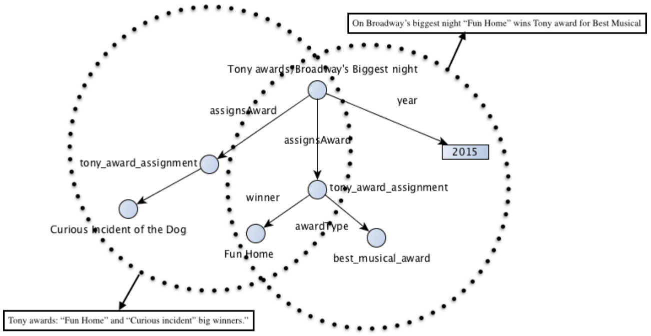
On Broadway's biggest night "Fun Home" wins Tony award for Best Musical
Tony awards/Broadway's Biggest night
assignsAward
year
assignsAward
2015
tony_award_assignment
tony_award_assignment
winner
awardType
Curious Incident of the Dog
Fun Home
best_musical_award
Tony awards: "Fun Home" and "Curious incident" big winners."

Figure 1: An integrated formal representation resulting from knowledge acquired from two different news about 2015 Tony Awards. The link between text segments and extracted knowledge needs to be preserved.

2. RELATED WORK

Ontology learning.

A nice survey presented in [18] identifies seven systems representing the state of the art in the area, and describes the typical tasks addressed by ontology learning systems, as well as their functionalities and implemented techniques. Most ontology learning and population systems focus on deriving a schema-level formal representation of the knowledge expressed by a text source (e.g. concepts and taxonomical relations, axioms, etc.), while fact-level knowledge extraction is mainly addressed by ontology population tools, which require an existing target ontology and large-size text corpora. Many of them also need some manual intervention. Recent work includes [2], which defines a semantic language to represent the meaning of thousands of English sentences. However an implementation is not provided yet.

Knowledge base integration and ontology matching.

A rich overview of ontology matching methods is provided by [6]. As for knowledge base integration, relevant work include [16] that leverages the interplay between schema and instance matching. Similarly, [9] shows a simple 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 rarely (probably never) derive from text sources. We aim at handling knowledge graphs derived from text sources, which are modelled by using a frame-semantics-based representation. We align such graphs accordingly to similarity measures that exploit frame semantics features, combined with an integer linear programming (ILP) graph matcher.

Coreference resolution.

This task deals with identifying elements in a text that refer to the same entities. Relevant work addressing cross-document coreference resolution include [13, 15, 1]. [5] uses spectral clustering and graph partitioning, and [12] is based on bag of words, latent similarity and clustering techniques. The main difference between our method and those ones is that we identify coreference relations by analysing a formal representation of the input texts, and by exploiting their formal semantic features. When extracted entities are events, the problem changes to resolution of event coreference within documents [8, 4] and across documents [3, 10]. Authors in [10] jointly model named entities and events. Clusters of entity and event mentions are constructed and merged accordingly to a linear regression model. 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. The authors of this method show that it achieves 61.2% recall, 75.9% precision and 67.8% F1 score, under the MUC metric and on the EECB corpus, an extension of the ECB corpus [3] annotated with both event and entity coreferences.

3. KNOWLEDGE EXTRACTION

Knowledge extraction is performed by our tool FRED, which converts a text into a knowledge graph. A knowledge graph is a fully labeled multi-digraph, such as the RDF abstract data structure, characterised by multiple semantic layers, i.e. nodes and edges may represent schema entities, data entities, meta-data entities, linguistic entities, (named) sub-graphs, etc. NL constructions can be recognised from parsing text fragments, but their formal semantics needs to be represented as a knowledge graph in a formalisation step. In our approach, we start by parsing and formalising texts into RDF-OWL. We mainly expect to target relatively short texts, and we need to represent concepts, relations, and factual knowledge, with less emphasis to schema-level axioms such as disjointness, cardinality restrictions, etc.

Entities of FRED graphs can be classified in two macro-groups: *individuals* and *classes*. Individuals, in turn, can be categorized into *named entities*, *skolemized entities* (entities that have no proper name), *events/situations* (occur-

[4] This paper refers to RDF/OWL as knowledge representation languages for knowledge graphs.

rences of n-ary predicates), and *qualities.* Events have a type `dul:Event`. Since FRED performs word sense disambiguation and entity linking, some entities of the resulting graph are linked to external sources (DBpedia, VerbNet). Properties of FRED graphs are divided in two macro-categories: *roles* and *non-roles.* Roles are outgoing edges from event nodes. All other edges are non-role edges. Role edges are broadly classified into *agentive*, *passive*, and *oblique* roles. For further information on the FRED semantics, please refer to `http://wit.istc.cnr.it/stlab-tools/fred/`.

4. KNOWLEDGE RECONCILIATION

We present a method for reconciling knowledge extracted from text using FRED. The main issue in reconciling two FRED graphs consists in detecting nodes of the two graphs that correspond to the same entity.

In the next sections, we define a graph-alignment-based method for solving the problem defined above. The graphs are first compressed by merging nodes and removing unnecessary labels. 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 maximises a *score function.* The score function combines the similarity between aligned nodes and the similarity between aligned properties. Maximising 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 node/edge similarity and structural information are considered. At the end, the aligned nodes are mapped to individuals of the original graph and `sameAs` relations are added between aligned nodes. If a node of the compressed graph corresponds to more than one individual in the original graph, one of them is picked at random.

Graph compression. Graph compression aggregates clusters of nodes in order to obtain abstract graphs with less, more informative, nodes. This step is necessary for two reasons. First the same entity may be represented by different equivalent nodes. Collapsing all equivalent nodes reduces the number of cross-graph associations to be found and increases their quality. Second, it enables aggregating type information to nodes, therefore increasing the amount of information that helps associating nodes across graphs.

Node and edge similarity. Some similarity measures for nodes and edges are used by the optimizer to define the alignment scoring function. The similarity can be positive or negative. Elements that have negative similarity tend not to be associated, while element with positive similarity tend to be associated. Note that the alignment algorithm performs a global optimisation, and hence local parts of the alignment may be penalised in favour 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.

Node similarity. We distinguish among three kinds of node pairs: *relevant*, *compatible* and *incompatible.* We first check if both nodes refers 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 pair of nodes is considered *relevant.* Otherwise, they are considered *incompatible.* If one of the two nodes does not refer to a named entity, we check the similarity of all cross-node pairs of labels (we remind the reader that a node may have more than one label) to see if the nodes share equivalent or similar concepts. Label similarity is computed by word-to-word similarity using SEMILAR [14] if the corresponding entities are of the same kind, otherwise it is zero. If the two nodes share the same label or refer to words with similarity higher than a predefined threshold (0.5 in our experimental evaluation), the two nodes are considered compatible. In all other cases, the pair of nodes is considered *incompatible.* Node similarity is assigned as follow: 1 on relevant pairs, -1 on compatible pairs and $-\infty$ on incompatible pairs.

Edge similarity. 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. Edge similarity is set to 1.34 for compatible edges and $-\infty$ for incompatible edges.

Alignment. Once the similarity among nodes and edges have been defined, our problem can be defined in terms of a graph alignment problem. Graph alignment is a widely studied problem that has many applications in several fields [17, 9]. It can be formulated as a quadratic assignment problem [17] and reduced to Integer Linear Programming. The problem formulation we adopt here is designed specifically for directed multi-graphs (a pair of nodes can be connected by more than one edge) and is similar to other previously proposed formulations [7][11]. Computing the optimal alignment is a NP-hard problem and hence no polynomial-time algorithm for it is known. However, since the size of knowledge graphs generated from text is not very high, and this kind of graphs are usually sparse, standard optimization techniques are affordable. We reduce our problem to ILP (Integer Linear Programming) and use a standard solver for the optimization. ILP optimizers often converge to proved optimal solutions on small or medium problem instances and provide good approximations with proved error bounds on larger instances (in our experiments an optimal solution was found in fractions of seconds in most cases). For large problem instances it is possible to apply known efficient heuristics in change of a slight lost in quality [17].

5. EXPERIMENTAL ANALYSIS

We implemented the reconciliation method as a Python tool[5]. We used IBM ILOG CPLEX 12.6.1 for solving the Integer Linear Program.

Although there are no competing approaches and benchmarks for evaluation available, we evaluated our method "by analogy" against an existing benchmark for a related problem: Cross-document Coreference Resolution (CCR) and to do that we used a part of the EECB 1.0 [10] gold standard (cluster 1).

EECB is an extension of ECB [3], a corpus annotated with event coreferences, that also contains entity coreference an-

[5] We omit the tool URL in the submitted version of the paper for keeping anonymity.

Table 1: Performances of our reconciliation tool in resolving cross-document coreferences

METRIC	Recall		Precision		F1	
	Avg	St.dev.	Avg	St.dev.	Avg	St.dev.
MUC	50,03%	0,18	72,75%	0,22	58,03%	0,19
B^3	37,88%	0,17	66,22%	0,23	46,31%	0,18
CEAFM	49,58%	0,18	69,75%	0,22	56,24%	0,17
CEAFE	36,23%	0,18	46,14%	0,23	36,89%	0,16
BLANC	46,79%	0,23	77,07%	0,25	56,73%	0,24

notations. Since we are interested in reconciling pairs of documents, we aligned pairs of documents from the corpus in all possible ways, and evaluated the results for each pair (171 pairs in total). The evaluation compares clusters of mentions generated by our tool with clusters of mentions from the ground truth, restricted to the documents under consideration. We removed singleton clusters (clusters containing only one mention). To correctly match corresponding mentions, we considered a mention to match a ground truth mention if the first one was entirely included in the second one. We used this procedure instead of exact matching because a mention produced by our tool often refers to the head word or to a few important words that express the corresponding entity in the FRED graph, while mentions in the gold standard may include a long sentence.

We computed precision, recall and the F1 score of several metrics. We report in Tab. 1 the average value and standard deviation on all pairs of reconciled documents.

Our tool is able to perform 72.75% of precision on average with 50.03% of recall (under the MUC measure). Considering the difficulty of the cross-document coreference task, these results are promising. Note that the state-of-the-art tool for CCR achieves 75.9% of precision with 61.2% of recall on the EECB dataset [10]. However, that tool is specifically designed for CCR, while our tool is more general in that it performs reconciliation over an abstract representation of the knowledge contained in multiple text documents, with CCR being just a possible application. Furthermore, our method provides a knowledge graph as output that can be directly used as machine readable data.

6. CONCLUSIONS

We have presented a novel method for generating and integrating knowledge graphs extracted from multiple text documents. Our tool relies on FRED, a machine reading tool for generating abstract representations of text documents, and integrates the generated knowledge by means of an optimization technique for graph alignment. We have assessed the performance of our tool by analogy in resolving cross-document co-references. The results show that our method is effective. Ongoing work is focused on building a specific gold standard for the semantic reconciliation problem and performing large-scale evaluation.

7. REFERENCES

[1] N. Andrews, J. Eisner, and M. Dredze. Robust entity clustering via phylogenetic inference. In *ACL*, 2014.

[2] L. Banarescu, C. Bonial, S. Cai, M. Georgescu, K. Griffitt, U. Hermjakob, K. Knight, P. Koehn, M. Palmer, and N. Schneider. Abstract meaning representation for sembanking, 2013.

[3] C. Bejan and S. Harabagiu. Unsupervised event coreference resolution with rich linguistic features. In *48th ACL Conference*, pages 1412–1422, Uppsala, Sweden, July 2010. ACL.

[4] B. Chen, J. Su, and L. C. Tan. Resolving event noun phrases to their verbal mentions. In *EMNLP2010*, pages 872–881. ACL, 2010.

[5] S. Dutta and G. Weikum. Cross-document co-reference resolution using sample-based clustering with knowledge enrichment. *Transactions of ACL, 3, 1*, pages 15–28, 2015.

[6] J. Euzenat and P. Shvaiko. *Ontology matching.* Springer-Verlag, Heidelberg (DE), 2nd edition, 2013.

[7] G. W. Klau. A new graph-based method for pairwise global network alignment. *BMC bioinformatics*, 10(Suppl 1):S59, 2009.

[8] F. Kong and G. Zhou. Improve tree kernel-based event pronoun resolution with competitive information. In T. Walsh, editor, *IJCAI*, pages 1814–1819. IJCAI/AAAI, 2011.

[9] S. Lacoste-Julien, K. Palla, A. Davies, G. Kasneci, T. Graepel, and Z. Ghahramani. Sigma: Simple greedy matching for aligning large knowledge bases. In *KDD2013*, KDD '13, pages 572–580, New York, NY, USA, 2013. ACM.

[10] H. Lee, M. Recasens, A. Chang, M. Surdeanu, and D. Jurafsky. Joint entity and event coreference resolution across documents. In *EMNLP*, EMNLP-CoNLL '12, pages 489–500, Stroudsburg, PA, USA, 2012. ACL.

[11] M. Mongiovi and R. Sharan. Global Alignment of Protein-Protein Interaction Networks. *Data Mining for Systems Biology, Methods in Molecular Biology Series*, 2011.

[12] A.-C. Ngonga Ngomo, M. Röder, and R. Usbeck. Cross-document coreference resolution using latent features. In *LD4IE—Linked Data for Information Extraction at ISWC 2014*, 2014.

[13] D. Rao, P. McNamee, and M. Dredze. Streaming cross document entity coreference resolution. In *23rd Int. Conf. on Computational Linguistics*, COLING '10, pages 1050–1058, Stroudsburg, PA, USA, 2010. ACL.

[14] V. Rus, M. C. Lintean, R. Banjade, N. B. Niraula, and D. Stefanescu. Semilar: The semantic similarity toolkit. In *ACL (Conference System Demonstrations)*, pages 163–168. Citeseer, 2013.

[15] S. Singh, A. Subramanya, F. Pereira, and A. McCallum. Large-scale cross-document coreference using distributed inference and hierarchical models. In *49th ACL*, HLT '11, pages 793–803, Stroudsburg, PA, USA, 2011. ACL.

[16] F. M. Suchanek, S. Abiteboul, and P. Senellart. Paris: Probabilistic alignment of relations, instances, and schema. *Proc. VLDB Endow.*, 5(3):157–168, nov 2011.

[17] J. T. Vogelstein, J. M. Conroy, V. Lyzinski, L. J. Podrazik, S. G. Kratzer, E. T. Harley, D. E. Fishkind, R. J. Vogelstein, and C. E. Priebe. Fast approximate quadratic programming for graph matching. *PLOS one*, 2015.

[18] W. Wong, W. Liu, and M. Bennamoun. Ontology learning from text: A look back and into the future. *ACM Comput. Surv.*, 44(4):20:1–20:36, Sept. 2012.