Mining Scholarly Data for Fine-Grained Knowledge Graph Construction

Davide Buscaldi[1], Danilo Dessì[2], Enrico Motta[3], Francesco Osborne[3], and Diego Reforgiato Recupero[2]

[1] LIPN, CNRS (UMR 7030), University Paris 13, Villetaneuse, France
davide.buscaldi@lipn.univ-paris13.fr
[2] Computer Science Department, University of Cagliari, Cagliari (Italy)
{danilo_dessi, diego.reforgiato}@unica.it
[3] Knowledge Media Institute, The Open University, MK7 6AA, Milton Keynes, UK
{enrico.motta, francesco.osborne}@open.ac.uk

Abstract. Knowledge graphs (KG) are large networks of entities and relationships, typically expressed as RDF triples, relevant to a specific domain or an organization. Scientific Knowledge Graphs (SKGs) focus on the scholarly domain and typically contain metadata describing research publications such as authors, venues, organizations, research topics, and citations. The next big challenge in this field regards the generation of SKGs that also contain an explicit representation of the knowledge presented in research publications. In this paper, we present a preliminary approach that uses a set of NLP and Deep Learning methods for extracting entities and relationships from research publications, and then integrates them in a KG. More specifically, we i) tackle the challenge of knowledge extraction by employing several state-of-the-art Natural Language Processing and Text Mining tools, ii) describe an approach for integrating entities and relationships generated by these tools, iii) analyze an automatically generated Knowledge Graph including 10,425 entities and 25,655 relationships derived from 12,007 publications in the field of Semantic Web, and iv) discuss some open problems that have not been solved yet.

Keywords: Knowledge Graph · Semantic Web · Knowledge Extraction · Scholarly Data · Natural Language Processing

1 Introduction

Knowledge graphs (KG) are large networks of entities and relationships, usually expressed as RDF triples, relevant to a specific domain or an organization [6]. Many state-of-the-art projects such as DBPedia [9], Google Knowledge Graph, BabelNet, and YAGO build KGs by harvesting entities and relations from textual resources, such as Wikipedia pages. The creation of such KGs is a complex process that typically requires to extract and integrate various information from structured and unstructured sources.

Scientific Knowledge Graphs (SKGs) focus on the scholarly domain and typically contain metadata describing research publications such as authors, venues,

organizations, research topics, and citations. Good examples are Open Academic Graph[4], Scholarlydata.org [15], and OpenCitations [17]. These resources provide substantial benefits to researchers, companies, and policy makers by powering several data-driven services for navigating, analyzing, and making sense of research dynamics. One of their main limitations is that the content of scientific papers is represented by unstructured texts (e.g., title and abstract). Therefore, a significant open challenge in this field regards the generation of SKGs that contain also an explicit representation of the knowledge presented in scientific publications [2], and potentially describe entities such as approaches, claims, applications, data, and so on. The resulting KG would be able to support a new generation of content-aware services for exploring the research environment at a much more granular level.

Most of the relevant information for populating such a KG might be derived from the text of research publications. In the last year, we saw the emergence of several excellent Natural Language Processing (NLP) and Deep Learning methods for entity linking and relationship extraction [12, 2, 8, 11, 10]. However, integrating the outputs of these tools in a coherent KG is still an open challenge.

In this paper, we present a preliminary approach that uses a set of NLP and Deep Learning methods for extracting entities and relationships from research publications and then integrates them in a KG. Within our work, we refer to an entity as a linguistic expression that refers to an object (e.g., topics, tools names, a well-know algorithm, etc.). We define a relation between two entities when they are syntactically or semantically connected. As an example, if a tool T adopts an algorithm A, we may build the relation (T, *adopt*, A).

The main contributions of this paper are: i) a preliminary approach that combines different tools for extracting entities and relations from research publications ii) an approach for integrating these entities and relationships, iii) a qualitative analysis of a generated SKG in the field of Semantic Web, and iv) a discussion of some open problems that have not been solved yet.

2 Related Work

In textual resources there are both syntactical and semantic peculiarities that make hard the identification of entities and relations.

In previous works, entities in textual resources were detected by studying Part-Of-Speech (POS) tags. An example is constituted by [14], where authors provided a graph based approach for Word Sense Disambiguation (WSD) and Entity Linking (EL) named Babelfly. Later, some approaches started to exploit various resources (e.g., context information and existing KGs) for developing ensemble methodologies [11]. Following this idea, we exploited an ensemble of tools to mine scientific publications and get the input data. Subsequently, we have developed our methodology on top of the ensemble result.

Relations extraction is an important task in order to connect entities of a KG. For doing so, authors in [8] developed a machine reader called FRED which

[4] https://www.openacademic.ai/oag/

exploits Boxer [4] and links elements to various ontologies in order to represent the content of a text in a RDF representation. Among its features FRED extracts relations between frames, events, concepts and entities[5]. One more project that enables the extraction of RDF triples from text is [3], where a framework called PIKES has been designed to exploit the frame analysis to detect entities and their relations. These works consider a single text at a time and do not consider the type of text they parse. In contrast with them, our approach aims at parsing specific type of textual data and, moreover, at combining information from various textual resources. We decided to rely on open domain information extraction tool results refined by contextual information of our data, adapting open domain results on Scholarly Data. In addition, we combined entities and relations coming from different scientific papers instead of mining a single text at a time. With our approach the resulting KG represents the overall knowledge presented in the input scientific publications.

Recently, extraction of relations from scientific papers has also raised interest within the SemEval 2018 Task 7 *Semantic Relation Extraction and Classification in Scientific Papers* challenge [7], where participants had to face the problem of detecting and classifying domain-specific semantic relations. An attempt to build KGs from scholarly data was also performed by [10], as an evolution of their work at SemEval 2018 Task 7. Authors proposed both a Deep Learning approach to extract entities and relations, and then built a KG on a dataset of $110,000$ papers. Our work finds inspiration from it, but we used different strategies to address open issues for combining entities and relations. In fact, authors of [10] considered clusters of co-referenced entities to come up with a representative entity in the cluster and solving ambiguity issues. On the contrary, we adopted textual and statistics similarity to solve it.

3 The Proposed Approach

In this section, we describe the preliminary approach that we applied to produce a KG of research entities. We used an input dataset composed by $12,007$ abstracts of scientific publications about the Semantic Web domain. It was retrieved by selecting all publications from the Microsoft Academic Graph dataset which contains in the string "Semantic Web" in the "field of science" heading.

3.1 Extraction of Entities and Relations

For extracting entities and relations, we exploited the following resources:

- An extractor framework designed by [10], which is based on Deep Learning models and provides tools for detecting entities and relations from scientific literature. It detects six types of entities (*Task, Method, Metric, Material, Other-Scientific-Term, and Generic*) and seven types of relations among a list of predefined choices (*Compare, Part-of, Conjunction, Evaluate-for, Feature-of, Used-for, Hyponym-Of*).

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

– OpenIE [1] provided with Stanford Core NLP[6]. It detects general entities and relations among them, especially those which can be derived by verbs.
– The CSO Classifier [18], a tool for automatically classifying research papers according to the Computer Science Ontology (CSO)[7] [19] which is a comprehensive automatically generated ontology of research areas in the field of Computer Science.

We processed each sentence from the abstract and used the three tools to assign to each sentence s_i a list of entities E_i and a list of relations R_i. For each sentence s_i, we firstly extracted entities and relations with the extractor framework, and saved them in two lists (E_i and R_i, respectively). We discarded all relations with type *CONJUNCTION* because they were too generic. Then, we used CSO to extract all Computer Science topics from the sentence, further expanding E_i. Finally, we processed each sentence s_i with OpenIE, and retrieved all triples composed by subject, verb, and object in which both subject and object were in the set of entities E_i.

3.2 Entities Refinement

Different entities in E_i may actually refer to the same concept with alternative forms (e.g., machine learning, machine learning methods, machine-learning). In this section, we report the methodology used to merge these entities when they appeared together in the same abstract.

Cleaning up entities. First, we removed punctuation (e.g., dots and apostrophes) and stop-words (e.g., pronouns). Then we merged singular and plural forms by using the *WordNet Lemmatizer* available in the NLTK[8] library.

Splitting entities. Some entities actually contained multiple compound expressions, e.g., *machine learning and data mining*. Therefore, we split entities when they contained the conjunction *and*. Referring to our example, we obtained the two entities *machine learning* and *data mining*.

Handling Acronyms. Acronyms are usually defined, appearing the first time near their extended form (e.g., *Computer Science Ontology (CSO)*) and then by themselves in the rest of the abstract (e.g., *CSO*). In order to map acronyms with their extended form in a specific abstract we use a regular expression. We then substituted every acronym (e.g., *CSO*) in the abstract with their extended form (e.g., *Computer Science Ontology*).

3.3 Graph Generation

In order to generate the graph, we need to integrate all triples extracted from the abstracts. In this phase we have to address three main issues. First, multiple entities derived from different abstracts may refer to the same concept.

[6] https://stanfordnlp.github.io/CoreNLP/
[7] http://cso.kmi.open.ac.uk
[8] https://www.nltk.org/

Secondly, multiple relationships derived from the verbs in the abstract may be redundant (e.g., {*emphasize*, *highlight*, *underline*}), Finally, some entities may be too generic (e.g., paper, approach) and thus useless for a SKG.

Entity Merging For the entity merging task we exploit two data structures. The first one, labelled $W2LE$, maps each word to a list of entities that share the last token (e.g., *medical ontology*, *biomedical ontology*, *pervasive agent ontology*, and so on.). With $W2LE$ we avoided comparing those entities that syntactically could not refer to the same entity (e.g., the entities *ontology generation* and *ontology adoption* were not compared). The second one, labelled $E2E$, maps each original entity to the entity that will represent it in the KG.

Given an entity e and the list of its tokens $\{t_0, ..., t_n\}$, we took t_n. If t_n was not present in $W2LE$, a new entry key t_n was added to $W2LE$ and its value is a list with e as its unique element. If t_n was in $W2LE$, then we compute the Levenshtein string similarity[9] between the entity e and all other entities $e'_0, ..., e'_m \in W2LE[t_n]$. If the resulting score met a given threshold t_L, the entity e was mapped as e'_i in $E2E$. Otherwise e was mapped to itself in $E2E$. At the end, the entity e was added to $W2LE[t_n]$. Finally, the map $E2E$ was used to select the entities for the graph. For each entry key e_x, if its corresponding entity $e_y = E2E[e_x]$ was not in the graph, a new entity with label e_y was added.

Relationship Merging After selecting a unique set of entities, we need to take care the relationships among them. First we cluster all verbs labels in order to reduce their number. For such a reason, we exploited WordNet [13] and a set of Word2Vec word embeddings trained on a set of 9 milion research papers from Microsoft Academic Graph[10]. In details, given the set of all verbs $V = \{v_0, ..., v_n\}$, we built a distance matrix M considering as a distance between two verbs v_i and v_j the $1 - WuPalmer$[11] similarity between their synsets. Then, we apply a hierarchical clustering algorithm, cutting the dendrogram where the number of clusters had the highest value of overall silhouette-width [5]. Subsequently, clusters were refined as follows. Given a cluster c, we assigned each verb $v_{i_c} \in c$ with the word embedding w_i in the Word2Vec model, and computed the centroid ce of the cluster as the average of word embeddings of its elements. Then, we ordered verbs in ascending order by the distance from ce. All verbs with a distance over a threshold t were discarded. All the other verbs were mapped on the verb nearest to the centroid ce in a map $V2V$.

Finally, given each pair of entities $p = (e_1, e_2)$ and their relations $\{r_0, ..., r_n\}$, we took every relation label $l_{r_i} \forall r_i \in \{r_0, ..., r_n\}$. All relations label coming from the extractor framework were directly merged into a single label L. All verb labels were firstly mapped through $V2V$ and then merged.

[9] https://pypi.org/project/python-Levenshtein/
[10] Avaliable at http://tiny.cc/w0u43y
[11] http://www.nltk.org/howto/wordnet.html

3.4 Detection of Generic Entities

The resulting graph may contain several generic entities (e.g., content, time, study, article, input, and so on.) In order to discard them we used a frequency-based filter which detects generic terms by comparing the frequency of the entities in three set of publications:

1. the set of 12,007 publications about the *Semantic Web*;
2. a set of the same size covering *Computer Science* but not the *Semantic Web*;
3. a set of the same size containing generic papers not about the *Semantic Web* nor the *Computer Science*.

For each entity e, we computed the number of times it appeared in the above datasets, so that we had three different counts c', c'', c'''. Then we computed the ratios $r' = \frac{c'}{c''}$ and $r'' = \frac{c'}{c'''}$. If the ratio r' met a threshold t', and the ratio r'' met a threshold t'' the entity e was included in the graph. In addition, we automatically preserved all entities within a whitelist composed by CSO topics and all the paper keywords in the initial dataset.

4 The Knowledge Graph

In this section, we report our preliminary results about the KG produced from 12,007 papers about the Semantic Web. We used the following parameters $t_L = 0.9$, $t' = 2$, and $t'' = 3$, which have been determined empirically. The resulting KG has 10,425 entities and 25,655 relationships.

Table 1. Examples of relationships in the KG.

Subject Entity	Relation	Object Entity
content integration	help	linked data
context reasoning	support	web ontology language
machine readable information	PART-OF	semantic web
semantic wikis	USED-FOR	query interpretation
semantic relationship	establish	semantic link network
semantic relationship	determine	wordnet

Table 1 reports as example some relationships extracted by our framework. The KG contains both verb-based relations (from OpenEI, in lowercase) and default relations (from the Extractor Framework, in uppercase). Verbs are usually more informative, but also harder to extract. Conversely, the Extractor Framework is more flexible and it is able to extract a large number of relationships, but these are usually less specific. Using both systems allows us obtaining a good balance between coverage and specificity. Naturally, this set of relationships could also be expanded by reasoning methods. For instance, the last two relationships in Table 1 could be used to infer that *wordnet* is most likely a *semantic link network*.

Table 2. Contribution of Extractor Framework and CSO to the KG entities.

Tools Entities Contribution	Count	Percentage
CSO	1034	9.92%
Extractor Framework	8668	83.15%
Exclusive CSO	117	1.12%
Exclusive Extractor Framework	7751	74.35%
Entities where both tools contribute	917	8.8%
Derived Entities	1640	15.73%

4.1 Graph Statistics

In this section, we report some statistics about entities and relations of our KG. Table 2 reports statistics about entities. To weigh the actual contribution of each tool, we counted the number of entities that were detected by applying each tool. With the label *Exclusive* we indicate the number of entities detected only by that underlying tool. The row *Derived Entities* refers to the additional entities that were obtained by merging or splitting the original entities.

The majority of entities that are present in the resulting KG comes from the Extractor Framework tool which contributes to the $83,15\%$ of all entities, and exclusively contributes to $74,35\%$ of them. The CSO Classifier contributes with 9.92%, but only a minority are exclusive. This was expected, since CSO contains fairly established research topics that appeared in a minimum of 50 papers in the dataset from which it was generated [16]. Conversely, the Extractor Framework is able to identify many long tail entities [12] that may only appear in few research papers. It is worth nothing that in the final KG, $15,73\%$ of all entities are different from the original ones due to the transformations we applied. On average, each entity was extracted 3.69 times by one of the tools, with a maximum of 52 and a minimum of 1.

Table 3. Contribution of Extractor Framework and OpenIE to the KG relations.

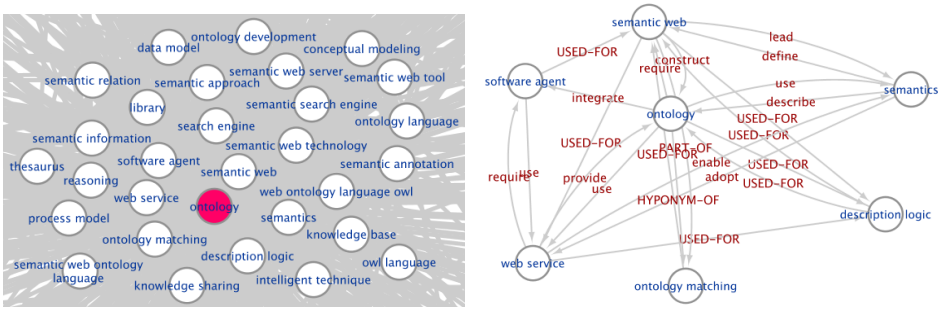
Tools Relations Contribution	Count	Percentage
Extractor Framework	23.624	92.09%
OpenIE	3.116	12.15%
Exclusive Extractor Framework	22.539	87.85%
Exclusive OpenIE	2.031	7.92%
Contribution of both tools	1.085	4.23%

Similarly to entities, the Extractor Framework produced also the majority of the relations with a coverage of 92.09%, 87.85% of which exclusive to this tool. However, the 12.15% of relations extracted by OpenIE are usually more informative since they are mapped to specific verbs.

On average, each relationship was extracted 1.32 times, with a maximum of 54 and a minimum of 1.

4.2 Limitations

In this section, we analyze some key entities of the Semantic Web and highlight some issues that still need to be solved to automatically produce high quality SKGs. In order to focus on specific subsections of the KG, we extracted three subgraphs containing all the entities directly linked to *ontology*, *natural language processing*, and *artificial intelligence*. For the sake of space, in the following figures we display only the most representative relationships between each pair of entities, considering the following priority order: any verb extracted from OpenEI, *Used-for, Part-of, Feature-of, Hyponym-Of, Evaluate-for, Compare*

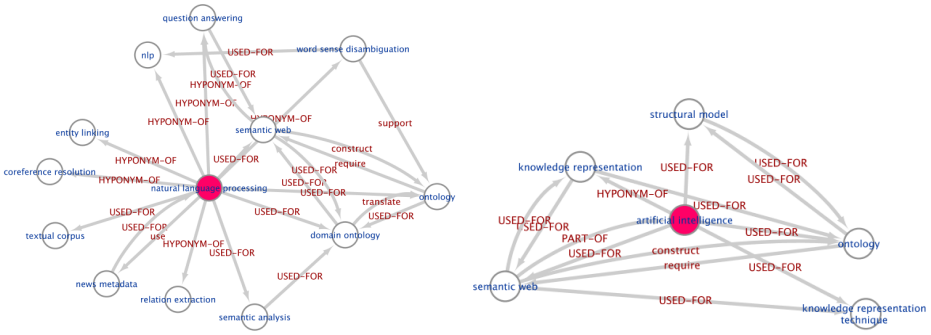


(a) (b)

Fig. 1. The subgraph of *ontology*. (a) A snippet where many entities related to *ontology* are showed. (b) A snippet where relations between its nearest entities are showed.

Figure 1 shows the subgraph of *ontology*, which is very dense since this concept is very well represented in the input dataset. The *ontology* entity was correctly connected to several relevant entities as *semantics*, *knowledge base*, *ontology language*, *description logic* and so on.

The subgraph of the *natural language processing* entity is showed in Figure 2a. It is less dense than that in Figure 1, since the *natural language processing* entity is less represented in the input dataset. The subgraph highlights an important issue that needs to be addressed. The entities *natural language processing* and *nlp* were not merged. This problem is due to the fact that acronyms are managed at abstract level, but not at graph level. Another issue regards the distinctive lack of verb-based relations, which are often useful to better specify a relationship between two entities. Similar considerations also apply to Figure 2b which shows the subgraph of the *artificial intelligence* entity. Some relationships between significant entities appear to be missing. For instance, *machine learning* and *artificial intelligence* are not connected here because they were originally linked by the *CONJUNCTION* relations, which was able to detect entities listed together, but we discarded since it is too generic. Another reason can be identified in textual forms that our approach may not be able to detect. We thus need to improve our pipeline to be able to handle similar instances and infer more specific relationships.



(a) (b)

Fig. 2. The subgraph of (a) *natural language processing* and (b) *artificial intelligence*

.

5 Conclusion and Future Work

In this paper we described a preliminary workflow for producing a Scientific Knowledge Graph from the text of research publication. We analysed a SKG derived from a set of 12,007 publication in the field of Semantic Web, with the aim of gaining a better understanding of the open problems that need to be solved when addressing this task. In summary, the analysis presented in this paper highlighted two main challenges. The first regards the disambiguation of entities that need to be further improved by also considering their semantic similarity. We also need to be able to resolve acronyms at a graph level by inferring to which extended form a certain acronym refers to in a specific publication. This may be addressed by representing entities according to word embedding learned from the input data or relevant textual resources. However, this representations would not consider long-tail entities that appear in few research papers. The second challenge regards the specificity of the relationships. While the Extractor Framework is quite good at extracting a large number of relationships, many of them are too generic. We thus intend to experiment with other techniques that combine Deep Learning and NLP for deriving specific predicates from research publications. Furthermore, we aim at validating the SKGs by human experts through a precision-recall analysis.

Acknowledgments

Danilo Dessì acknowledges Sardinia Regional Government for the financial support of his PhD scholarship (P.O.R. Sardegna F.S.E. 2014-2020).

References

1. Angeli, G., Premkumar, M.J.J., Manning, C.D.: Leveraging linguistic structure for open domain information extraction. In: Proceedings of the 53rd Annual Meeting of the ACL and the 7th IJCNLP. vol. 1, pp. 344–354 (2015)

2. Auer, S., Kovtun, V., Prinz, M., Kasprzik, A., Stocker, M., Vidal, M.E.: Towards a knowledge graph for science. In: Proceedings of the 8th International Conference on Web Intelligence, Mining and Semantics. p. 1. ACM (2018)
3. Corcoglioniti, F., Rospocher, M., Aprosio, A.P.: A 2-phase frame-based knowledge extraction framework. In: Proceedings of the 31st Annual ACM Symposium on Applied Computing. pp. 354–361. ACM (2016)
4. Curran, J.R., Clark, S., Bos, J.: Linguistically motivated large-scale nlp with c&c and boxer. In: Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions. pp. 33–36 (2007)
5. Dessì, D., Recupero, D.R., Fenu, G., Consoli, S.: A recommender system of medical reports leveraging cognitive computing and frame semantics. In: Machine Learning Paradigms, pp. 7–30. Springer (2019)
6. Ehrlinger, L., Wöß, W.: Towards a definition of knowledge graphs. SEMANTiCS (Posters, Demos, SuCCESS) **48** (2016)
7. Gábor, K., Buscaldi, D., Schumann, A.K., QasemiZadeh, B., Zargayouna, H., Charnois, T.: Semeval-2018 task 7: Semantic relation extraction and classification in scientific papers. In: Proceedings of The 12th International Workshop on Semantic Evaluation. pp. 679–688 (2018)
8. Gangemi, A., Presutti, V., Recupero, D.R., Nuzzolese, et al.: Semantic Web Machine Reading with FRED. Semantic Web **8**(6), 873–893 (2017)
9. Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P.N., et al.: Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. Semantic Web **6**(2), 167–195 (2015)
10. Luan, Y., He, L., Ostendorf, M., Hajishirzi, H.: Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. In: Proceedings of the EMNLP 2018 Conference. pp. 3219–3232 (2018)
11. Martinez-Rodriguez, J.L., Lopez-Arevalo, I., Rios-Alvarado, A.B.: Openie-based approach for knowledge graph construction from text. Expert Systems with Applications **113**, 339–355 (2018)
12. Mesbah, S., Lofi, C., Torre, M.V., Bozzon, A., Houben, G.J.: Tse-ner: An iterative approach for long-tail entity extraction in scientific publications. In: ISWC. pp. 127–143. Springer (2018)
13. Miller, G.A.: Wordnet: a lexical database for english. Communications of the ACM **38**(11), 39–41 (1995)
14. Moro, A., Raganato, A., Navigli, R.: Entity linking meets word sense disambiguation: a unified approach. Transactions of the Association for Computational Linguistics **2**, 231–244 (2014)
15. Nuzzolese, A.G., Gentile, A.L., Presutti, V., Gangemi, A.: Conference linked data: the scholarlydata project. In: ISWC. pp. 150–158. Springer (2016)
16. Osborne, F., Motta, E.: Klink-2: integrating multiple web sources to generate semantic topic networks. In: ISWC. pp. 408–424. Springer (2015)
17. Peroni, S., Shotton, D., Vitali, F.: One year of the opencitations corpus. In: ISWC. pp. 184–192. Springer (2017)
18. Salatino, A.A., Thanapalasingam, T., Mannocci, A., Osborne, F., Motta, E.: Classifying research papers with the computer science ontology. In: ISWC (P&D/Industry/BlueSky). CEUR Workshop Proceedings. vol. 2180
19. Salatino, A.A., Thanapalasingam, T., Mannocci, A., Osborne, F., Motta, E.: The computer science ontology: a large-scale taxonomy of research areas. In: ISWC. pp. 187–205 (2018)