Co-LOD: Continuous Space Linked Open Data

Mayank Kejriwal and Pedro Szekely

Information Sciences Institute, University of Southern California
{kejriwal,pszekely}@isi.edu

Abstract. The Linked Open Data (LOD) initiative has been one of the successful manifestations of Semantic Web efforts over the last two decades, with near-exponential growth of LOD datasets in the initial years. Entities and datasets on LOD are naturally discrete, making them amenable to both well-defined reasoning and retrieval procedures that ultimately return lists or sets of resource identifiers fulfilling some criteria (whether stating user intent or using pattern-matching query languages like SPARQL). In recent years, representation learning algorithms have witnessed a powerful ascent in mainstream Artificial Intelligence, fueled in part by the adoption and refinement of neural network architectures like Recurrent Neural Nets and skip-grams, and by empirical successes such as achieved in the natural language processing and knowledge discovery communities by word and graph embeddings. Large datasets, which are almost always required by such algorithms, make it possible to train and release models openly. In some cases, open models can even be released based on proprietary datasets like Twitter corpora. We propose that the Semantic Web community position itself as a pre-eminent research leader in this space by leveraging the vast and diverse collection of structured datasets that are currently available on Linked Open Data, to build out a corresponding continuous-space equivalent.

Keywords: Linked Open Data, Knowledge Graphs, Embeddings, Continuous Space, Representation Learning

Linked Open Data (LOD)[1] has been one of the success stories of the Semantic Web community, involving unifying investments in our research agenda over many years. In the early years, LOD started with only a handful of datasets, but with the modeling and publication of datasets like DBpedia, GeoNames, Freebase and the NYT ontology, LOD became a large-scale resource that was initially adopted broadly by our community but has since gained greater acceptance in other communities of AI like Natural Language Processing (NLP) and Knowledge Discovery.

Yet, the LOD has not been immune from criticism; both anecdotally, and in a few formal studies, it has been noted that the collection of datasets is noisy, containing missing, redundant and even contradictory information, that the datasets are relatively schema-free and not designed for facilitating the kinds of reasoning that semantic agents may need to do for powerful processing of queries posed by Web users, and that the data can get stale very quickly since many datasets do not update in real time. A bigger, fruitful (in our opinion) and even provocative debate that has arisen in our community in recent years[2] is whether the growth in schema.org threatens the premises of LOD. Despite utilizing similar technologies, many of which have been doubtlessly inspired by work in our own community, schema.org is based on a completely different rationale, with the focus being on facilitating a better search experience for users, rather than on connectivity (thereby doing away with the notion of 'linked' altogether). That the search engine providers have pushed hard for embedded schema.org markups, particularly for websites describing restaurants, movies and other consumer-facing products and services, in an effort to ingest more standard datasets into their knowledge graphs and ranking algorithms has also led to the popularity of the schema.org movement. Website publishers and service providers have an incentive to provide clean, up-to-date schema.org data for some of these high-priority categories since it plays a non-trivial role in whether (and how) they will be found and listed by the search engine when users search for terms that relate to the business they are in. In short, publishing good schema.org for a subset of ontological classes influences modern-day search optimization, a must for any online provider.

There is no doubt that various ambitious research agendas are already in place all over the world to address some of the problems we noted above with LOD, and some have been trying to bridge the gap between LOD and schema.org, usually by showing how we could possibly extract and LODify schema.org markup on webpages with high accuracy. But we believe that there is a bigger opportunity with LOD that will allow us to significantly expand its scope, and make it a vital resource for the AI and Deep Learning community as a whole.

To lay the groundwork for this vision, we briefly present the preliminaries on continuous-space representation learning aka 'embeddings'. Simply put, an embedding is a continuous, real-valued vector, usually of relatively low dimensionality (a common range is from 20-100 depending on the application and dataset) that serves as a *distributed* representation of a data unit. The definition of a data unit depends on the algorithm e.g., a word embedding algorithm treats words as data units and 'embeds' words into continuous, real-valued and low-dimensional vectors. Graph embeddings generally embed the nodes in a graph into such spaces, though more advanced knowledge graph embeddings are also capable of embedding relations. Data units can even be heterogeneous e.g., the paragraph2vec algorithm was an example of a 'document embedding' model that jointly embedded words and documents into a single vector space. Even more recently, the StarSpace package released by Facebook Research, is a general-purpose representation learning package that models data units very abstractly as a graph-like data structure before embedding them. Because of this abstraction, it is able to jointly embed all kinds of units, including nodes, text, documents, users etc. as long as the data is correctly modeled and formatted.

[2] Some of these debates have been encouraged by ISWC workshops like HSSUES (2017).

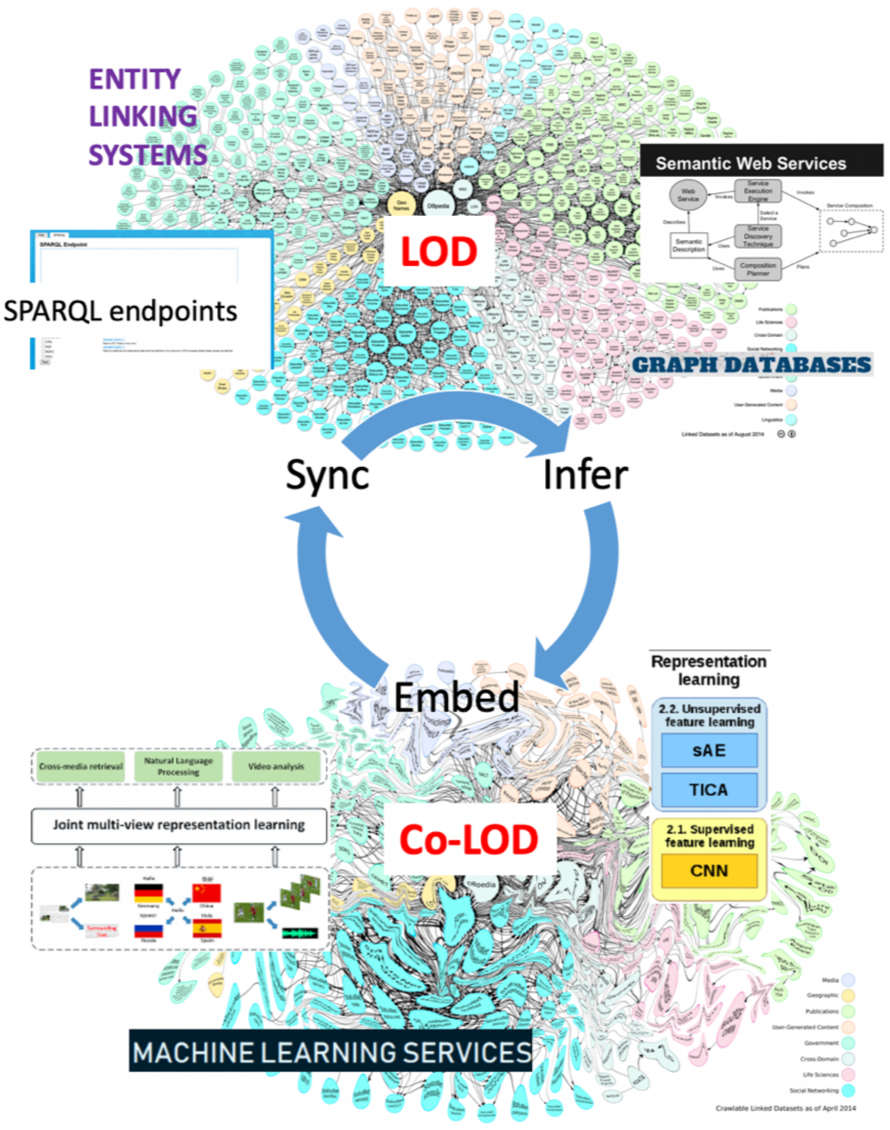

Fig. 1. An illustration of our materialized vision, where LOD and co-LOD co-exist synchronously, mutually benefit one another and support a much wider class of services and communities than the current LOD collection of datasets.

We argue that some of the critical features that make these continuous-space representation learning algorithms so successful are, by a fortuitous coincidence, exactly in alignment with the features of LOD. First, embedding algorithms rely on the presence of large datasets that do not necessarily have to be of high quality[3]. Second, embedding algorithms perform well when there is enough 'context'. In a graph-theoretic setting, this usually implies connectivity i.e. the denser the graph, the more likely that 'good' embeddings will be learned. Similarly, in a text-theoretic setting, context means that words are not 'sparsely' used i.e. if a word is only used once or twice in the corpus, it is unlikely that good embeddings will be generated for it by a model. Given corpora like Wikipedia or the Google News Corpus, this problem rarely arises since most words are used several times throughout the corpus. We argue that LOD provides context both because of connectivity of resources within each dataset, but also because the linked data principles ensure that resources across datasets are connected using agreed-upon OWL, SKOS or RDFS properties like owl:sameAs.

With the groundwork and arguments in place, we present the crux of our vision in Figure 1. The top portion of the figure shows the LOD ecosystem as it stands today. In essence, it is a 'discrete' system in that it can be visualized as a giant graph of domain-specific (and in the center, open-world domain) datasets that have connections between them due to the fourth Linked Data principle. These datasets are typically accessed as dumps, or via SPARQL endpoints, and were designed with Semantic Web agents in mind.

The bottom portion of the figure shows our vision of co-LOD, which is a continuous space version of Linked Open Data. Our vision can be stated very simply: embed the *entire* LOD collection of datasets into a continuous space, and make the space accessible to machine learning, data mining and recommendation services that rely so heavily on general-purpose embeddings (such as of the Wikipedia corpus) for good performance. Our vision is currently just theoretical and aspirational, due to several wildcard challenges: how can we make all of LOD accessible to a representation learning algorithm? Which algorithm should we use (e.g., PyTorch-BigGraph [9])? How do we take meta-data, data, literals (including text, dates and numbers) and ontologies all into account when embedding? How do we evaluate the quality of the embedding? Where should such an embedding be hosted? Should there be a single continuous space for all of Linked Data? How do we access the embeddings for machine learning services? How do we ensure co-LOD and LOD stay in sync?

We suspect that each of these questions has the potential to spawn a host of papers in the short, medium, and long-term future, and we hope to have the opportunity to read, critique and write some of them.

[3] It is known that lower quality degrades some embedding algorithms, but recent algorithms, like fastText (also from Facebook Research) have been designed to deal with many different kinds of noise, including misspellings and out-of-vocabulary words. A complete study analyzing dependence of embedding quality on noise is lacking, to the best of our knowledge.

References

1. C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann. Dbpedia-a crystallization point for the web of data. *Web Semantics: science, services and agents on the world wide web*, 7(3):154–165, 2009.
2. M. Färber, B. Ell, C. Menne, and A. Rettinger. A comparative survey of dbpedia, freebase, opencyc, wikidata, and yago.
3. A. Grover and J. Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864. ACM, 2016.
4. W. L. Hamilton, R. Ying, and J. Leskovec. Representation learning on graphs: Methods and applications. *arXiv preprint arXiv:1709.05584*, 2017.
5. M. Kejriwal. *Populating a linked data entity name system: A big data solution to unsupervised instance matching*, volume 27. IOS Press, 2016.
6. M. Kejriwal and P. Szekely. Neural embeddings for populated geonames locations. In *International Semantic Web Conference*, pages 139–146. Springer, 2017.
7. M. Kejriwal and P. Szekely. Scalable generation of type embeddings using the abox. *Open Journal of Semantic Web (OJSW)*, 4(1):20–34, 2017.
8. M. Kejriwal and P. Szekely. Supervised typing of big graphs using semantic embeddings. In *Proceedings of The International Workshop on Semantic Big Data*, page 3. ACM, 2017.
9. A. Lerer, L. Wu, J. Shen, T. Lacroix, L. Wehrstedt, A. Bose, and A. Peysakhovich. Pytorch-biggraph: A large-scale graph embedding system. *arXiv preprint arXiv:1903.12287*, 2019.
10. B. Perozzi, R. Al-Rfou, and S. Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710. ACM, 2014.
11. M. Schmachtenberg, C. Bizer, and H. Paulheim. Adoption of the linked data best practices in different topical domains. In *The Semantic Web–ISWC 2014*, pages 245–260. Springer, 2014.
12. J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei. Line: Large-scale information network embedding. In *Proceedings of the 24th international conference on world wide web*, pages 1067–1077. International World Wide Web Conferences Steering Committee, 2015.
13. Q. Wang, Z. Mao, B. Wang, and L. Guo. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*, 29(12):2724–2743, 2017.