Knowledge Base Construction in the Machine-learning Era

ALEX RATNER AND CHRIS RÉ

EXPERT-CURATED GUIDES TO THE BEST OF CS RESEARCH

Research for Practice combines the resources of the ACM Digital Library, the largest collection of computer science research in the world, with the expertise of the ACM membership. In every RfP column experts share a short curated selection of papers on a concentrated, practically oriented topic.

This installment of Research for Practice features a curated selection from Alex Ratner and Chris Ré, who provide an overview of recent developments in Knowledge Base Construction (KBC). While knowledge bases have a long history dating to the expert systems of the 1970s, recent advances in machine learning have led to a knowledge base renaissance, with knowledge bases now powering major product functionality including Google Assistant, Amazon Alexa, Apple Siri, and Wolfram Alpha. Ratner and Ré's selections highlight key considerations in the modern KBC process, from interfaces that extract knowledge from domain experts to algorithms and representations that transfer knowledge across tasks. Please enjoy! —*Peter Bailis*

More information is accessible today than at any other time in human history. From a software perspective, however, the vast majority of this data is unusable, as it is locked away in *unstructured* formats such as text, PDFs, web pages, images, and other hard-to-parse formats. The goal of KBC (knowledge base construction) is to extract structured information automatically from this "dark data," so that it can be used in downstream applications for search, question-answering, link prediction, visualization, modeling, and much more. Today, KBs (knowledge bases) are the central components of systems that help fight human trafficking,[18] accelerate biomedical discovery,[9] and, increasingly, power web-search and question-answering technologies.[4]

KBC is extremely challenging, however, as it involves dealing with highly complex input data and multiple connected subtasks such as parsing, extracting, cleaning, linking, and integration. Traditionally, even with machine learning, each of these subtasks would require arduous *feature engineering* (i.e., manually crafting attributes of the input data to feed into the system). For this reason, KBC has traditionally been a months- or years-long process that was approached only by academic groups (e.g., YAGO,[8] DBPedia,[7] KnowItNow,[2] DeepDive,[19] etc.) or large, well-funded teams in industry and government (e.g., Google's Knowledge Vault, IBM Watson, Amazon's Product Graphs, etc.).

Today, however, there is a renewed sense of democratized progress in the area of KBC, thanks to powerful but easy-to-use deep-learning models that largely obviate the burdensome task of feature engineering. Instead, modern deep-learning models operate directly over raw input data such as text or images and get state-of-the-art performance on KBC sub-tasks such as parsing, tagging, classifying, and linking. Moreover, standard commodity architectures are often suitable for a wide range of domains and tasks such as the "hegemony"[11] of the bi-LSTM (bidirectional long short-term memory) for text, or the CNN (convolutional neural network) for images. Open-source implementations can often be downloaded and run in several lines of code.

For these emerging deep-learning-based approaches to make KBC faster and easier, though, certain critical design decisions need to be addressed—such as how to piece them together, how to collect training data for them efficiently, and how to represent their input and output data. This article highlights three papers that focus on these critical design points: (1) *joint-learning* approaches for pooling information and coordinating among subcomponents; (2) more efficient methods of *weakly supervising* the machine-learning components of the system; and (3) new ways of representing both inputs and outputs of the KB.

JOINT LEARNING: SHARING INFORMATION AND AVOIDING CASCADED ERRORS

Mitchell, T. M., Cohen, W. W., Hruschka Jr., E. R., Talukdar, P. P., Betteridge, J., Carlson, A., Mishra, B. D., Gardner, M., Kisiel, B., Krishnamurthy, J., et al. 2015. Never-ending learning. In *Proceedings of the Conference on Artificial Intelligence (AAAI)*, 2302-2310.

KBC is particularly challenging because of the large number of related subtasks involved, each of which may use one or more ML (machine-learning) models. Performing these tasks in disconnected pipelines is suboptimal in at least two ways: it can lead to cascading errors (for example, an initial parsing error may throw off a downstream tagging or linking task); and it misses the opportunity to pool information and training signals among related tasks (for example, subcomponents that extract similar types of relations can probably use similar representations of the input data). The high-level idea of what are often termed *joint inference* and *multitask learning*—which we collectively refer to as *joint learning*—is to learn multiple related models jointly, connecting them by logical relations of their output values and/or shared representations of their input values.

NELL (Never-Ending Language Learner) is a classic example of the impact of joint learning on KBC at an impressive scale. NELL is a system that has been extracting various facts about the world (e.g., `ServedWith(Tea, Biscuits)`) from the Internet since 2010, amounting to a KB containing (in 2015) more than 80 million entries. The problem setting approached by NELL consists of more than 2,500 distinct learning tasks, including categorizing noun phrases into specific categories, linking similar entities, and extracting relations between entities. Rather than learning all these tasks separately, NELL's formulation includes known (or learned) *coupling constraints* between the different tasks, which Mitchell et al. cite as critical to training NELL. These include logical relations such as subset/superset (e.g., `IsSandwhich(Hamburger)` ⇒ `IsFood(Hamburger)`) and mutual-exclusion constraints, which connect the many disparate tasks during inference and learning.

NELL is a classic example of the impact of joint learning on KBC at an impressive scale.

In other systems, the importance of connecting or coupling multiple tasks is echoed in slightly different contexts or formulations: for example, as a way to avoid cascading errors between different pipeline steps such as extraction and integration (e.g., DeepDive[19]), or implemented by sharing weights or learned representations of the input data between tasks as in multitask learning.[3,17] Either way, the decision about how to couple different subtasks is a critical one in any KBC system design.

WEAK SUPERVISION: PROGRAMMING ML WITH TRAINING DATA

Ratner, A. J., Bach, S. H., Ehrenberg, H., Fries, J., Wu, S., and Ré. C., 2017. Snorkel: rapid training data creation with weak supervision. In *Proceedings of the Very Large Database (VLDB) Endowment* 11(3), 269-282.

In almost all KBC systems today, many or all of the critical tasks are performed by increasingly complex machine-learning models, such as deep-learning ones. While these models indeed obviate much of the feature-engineering burden that was a traditional bottleneck in the KBC development process, they also require large volumes of labeled *training data* from which to learn. Having humans label this training data by hand is an expensive task that can take months or years, and the resulting labeled data set is frustratingly static: if the schema of a KB changes, as it frequently does in real production settings, the training set must be thrown out and relabeled. For these reasons, many KBC systems today use some form of *weak supervision*:[15] noisier, higher-level supervision provided more efficiently by a domain expert.[6,10] For example, a popular heuristic technique is distant supervision, where the entries of an existing knowledge base are heuristically aligned with new input data to label it as training data.[1,13,16]

Snorkel provides an end-to-end framework for weakly supervising machine-learning models by having domain experts write LFs (labeling functions), which are simply black-box functions that programmatically label training data, rather than labeling any training data by hand. These LFs subsume a wide range of weak supervision techniques and effectively give non-machine-learning experts a simple way to "program" ML models. Moreover, Snorkel automatically learns the accuracies of the LFs and reweights their outputs using statistical modeling techniques, effectively denoising the training data, which can then be used to supervise the KBC system. In this paper, the authors demonstrate that Snorkel improves over prior weak supervision approaches by enabling the easy use of many noisy sources. Snorkel and comes within several percentage points of performance using massive hand-labeled training sets, showing the efficacy of weak supervision for making high-performance KBC systems faster and easier to develop.

EMBEDDINGS: REPRESENTATION AND INCORPORATION OF DISTRIBUTED KNOWLEDGE

Riedel, S., Y ao, L., McCallum, A., and Marlin, B. M. 2013. Relation extraction with matrix factorization and universal schemas. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics–Human Language Technologies*, 74–84.

Finally, a critical decision in KBC is how to represent data: both the unstructured input data and the resulting output constituting the knowledge base. In both KBC and more general ML settings, the use of dense vector embeddings to represent input data, especially text, has become an omnipresent tool.[12] For example, word embeddings, learned by applying PCA (principal component analysis) or some approximate variant to large unlabeled corpora, can inherently represent meaningful semantics of text data, such as synonymy, and serve as a powerful but simple way to incorporate statistical knowledge from large corpora. Increasingly sophisticated types of embeddings, such as hyperbolic,[14] multimodal, and graph[5] embeddings, can provide powerful boosts to end-system performance in an expanded range of settings.

A critical decision in KBC is how to represent data

In their paper, Riedel et al. provide an interesting perspective by showing how embeddings can also be used to represent the knowledge base itself. In traditional KBC, an output schema (i.e., which types of relations are to be extracted) is selected first and fixed, which is necessarily a manual process. Instead, Riedel et al. propose using dense embeddings to represent the KB itself and learning these from the union of all available or potential target schemas.

Moreover, they argue that such an approach unifies the traditionally separate tasks of extraction and integration. Generally, *extraction* is the process of going from input data to an entry in the KB—for example, mapping a text string `X likes Y` to a KB relation `Likes(X,Y)`—while *integration* is the task of merging or linking related entities and relations. In their approach, however, both input text and KB entries are represented in the same vector space, so these operations become essentially equivalent. These embeddings can then be learned jointly and queried for a variety of prediction tasks.

KBC BECOMING MORE ACCESSIBLE

This article has reviewed approaches to three critical design points of building a modern KBC system and how they have the potential to accelerate the KBC process: (1) coupling multiple component models to learn them jointly; (2) using weak supervision to supervise these models more efficiently and flexibly; and (3) choosing a dense vector representation for the data. While ML-based KBC systems are still large and complex, one practical benefit of today's interest and investment in ML is the plethora of state-of-the-art models for various KBC subtasks available in the open source, and well-engineered frameworks such as PyTorch and TensorFlow with which to run them. Together with techniques and systems for putting all the pieces together like those reviewed, high-performance KBC is becoming more accessible than ever.

References

1. Bunescu, R. C., Mooney, R. J. 2007. Learning to extract relations from the web using minimal supervision. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, 576–583.
2. Cafarella, M. J., Downey, D., Soderland, S., Etzioni, O. 2005. KnowItNow: fast, scalable information extraction from the web. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in*

Natural Language Processing, 563–570.

3. Caruana, R. 1993. Multitask learning: A knowledge-based source of inductive bias. In *Proceedings of the 10th International Conference on Machine Learning*, 41-48.
4. Dong, X., Gabrilovich, E., Heitz, G., Horn, W., Lao, N., Murphy, K., Strohmann, T., Sun, S., Zhang, W. 2014. 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*, 601–610.
5. Grover, A., Leskovec, J. 2016. node2vec: scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 855–864.
6. Hoffmann, R., Zhang, C., Ling, X., Zettlemoyer, L., Weld, D. S. 2011. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL)–Human Language Technologies,* Volume 1, 541-550.
7. Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P., Hellmann, S., Morsey, M., van Kleef, P., Auer, S., Bizer, C. 2014. DBpedia—a large-scale, multilingual knowledge base extracted from Wikipedia. *Semantic Web* 6(2), 167–195.
8. Mahdisoltani, F., Biega, J. Suchanek, F. M. 2013. YAGO3: a knowledge base from multilingual wikipedias. *In Proceedings of the 7th Biennial Conference on Innovative Data Systems Research (CIDR).*
9. Mallory, E. K., Zhang, C., Ré, C., Altman, R. B. 2015. Large-

scale extraction of gene interactions from full-text literature using DeepDive. *Bioinformatics* 32(1),106–113.

10. Mann, G. S., McCallum, A. 2010. Generalized expectation criteria for semi-supervised learning with weakly labeled data. *Journal of Machine Learning Research* 11, 955–984.
11. Manning, C. 2017. Representations for language: from word embeddings to sentence meanings. Presented at Simons Institute for the Theory of Computing, UC Berkeley; https://nlp.stanford.edu/manning/talks/Simons-Institute-Manning-2017.pdf.
12. Mikolov, T., Chen, K., Corrado, G., Dean, J. 2013. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.
13. Mintz, M., Bills, S., Snow, R., Jurafsky, D. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the Association for Computational Linguistics (ACL) and the 4th Conference of the Asian Federation of Natural Language Processing (AFNLP),* 1003–1011.
14. Nickel, M., Kiela, D. 2017. Poincaré embeddings for learning hierarchical representations. In *Advances in Neural Information Processing Systems* 30, 6341–6350.
15. Ratner, A., Bach, S., Varma, P., Ré, C. Weak supervision: the new programming paradigm for machine learning. Hazy Research; https://hazyresearch.github.io/snorkel/blog/ws_blog_post.html.
16. Ren, X., He, W., Qu, M., Voss, C. R., Ji, H., Han, J. 2016. Label noise reduction in entity typing by heterogeneous partial-label embedding. In *Proceedings of the 22nd*

ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 1825–1834.

17. Ruder, S. 2017. An overview of multi-task learning in deep neural networks. arXiv preprint arXiv: 1706.05098.
18. Zhang, C., Shin, J., Ré, C., Cafarella, M., Niu, F. 2016. Extracting databases from dark data with DeepDive. In *Proceedings of the International Conference on Management of Data*, 847–859.
19. Zhang, C., Ré, C., Cafarella, M., De Sa, C., Ratner, A., Shin, J., Wang, F., Wu, S. 2017. DeepDive: declarative knowledge base construction. *Communications of the ACM* 60(5),93–102.

Alex Ratner *is a Ph.D. candidate in computer science at Stanford University, advised by Chris Ré, where his research focuses on weak supervision—the idea of using higher-level, noisier input from domain experts to train complex state-of-the-art models where limited hand-labeled training data is available. He leads the development of the Snorkel framework for weakly supervised ML, which has been applied to KBC problems in domains such as genomics, clinical diagnostics, and political science. He is supported by a Stanford Bio-X SIGF fellowship.*

Christopher Ré *is an associate professor of computer science at Stanford University. His work's goal is to enable users and developers to build applications that more deeply understand and exploit data. His contributions span database theory, database systems, and machine learning, and have won best paper awards from PODS (Principles of Database Systems),* *SIGMOD (Special Interest Group on Management of Data), and ICML (International Conference on Machine Learning). Work from his group has been incorporated into major scientific and humanitarian efforts, including the IceCube neutrino detector, PaleoDeepDive, and MEMEX in the fight against human trafficking, and into commercial products from major web and enterprise companies. He has been the recipient of a SIGMOD Dissertation Award, an NSF CAREER Award, an Alfred P. Sloan Fellowship, a Moore Data Driven Investigator Award, the VLDB Early Career Award, the MacArthur Foundation Fellowship, and an Okawa Research Grant.*

CONTENTS