DOI:10.1145/3060584

To view the accompanying paper, visit doi.acm.org/10.1145/3060586 rh

Technical Perspective
Building Knowledge Bases from Messy Data

By Alon Halevy

IMAGINE THE TASK of creating a database of all the high-quality specialty cafés around the world so you never have to settle for an imperfect brew. Relying on reviews from sites such as Yelp will not do the job because there is no restriction on who can post reviews there. You, on the other hand, are interested only in cafés that are reviewed by the coffee intelligentsia. There are several online sources with content relevant to your envisioned database. Cafés may be featured in well-respected coffee publications such as sprudge.com or baristamagazine.com, and data of more fleeting nature may pop up on your social media stream from coffee-savvy friends.

The task of creating such a database is surprisingly difficult. You would begin by deciding which attributes of cafés the database should model. Attributes such as address and opening hours would be obvious even to a novice, but you will need to consult a coffee expert who will suggest more refined attributes such as roast profile and brewing methods. The next step is to write programs that will extract structured data from these heterogeneous sources, distinguish the good extractions from the bad ones, and combine extractions from different sources to create tuples in your database. As part of the data cleaning process, you might want to employ crowd workers to confirm details, such as opening hours that were extracted from text or whether two mentions of cafés in text refer to the same café in the real world. In the extreme case, you might even want to send someone out to a café to check on some of the details in person. The process of creating the database is iterative because your extraction techniques will be refined and because the café scene changes frequently.

This Knowledge Base Construction task (KBC) has been an ongoing challenge and an inspiration for deep collaborations between researchers and practitioners in multiple fields, including data management and integration, information extraction, machine learning, natural language understanding, and probabilistic reasoning. Aside from the compelling application detailed here, the problem arises in many other settings where we need to construct databases from messy data. For example, imagine the task of creating a database (or ontology) of all job categories for a job-search site, or compiling a database of dishes served in Tokyo restaurants for the purpose of restaurant search or trend analysis.

The following paper is a prime example of groundbreaking work in the area of KBC. DeepDive, a project led by Chris Ré at Stanford, is an end-to-end system for creating knowledge bases. The input to DeepDive is a set of data sources such as text documents, PDF files, and structured databases. DeepDive extracts, cleans, and integrates data from the multiple sources and produces a database in which a probability is attached to every tuple. A user interacts with DeepDive in a high-level declarative language (DDLog) that uses predicates defined with functions in Python. The rules in DDLog specify how to extract entities, mentions of entities, and relationships from the data sources and the details of the extractions are implemented in Python. Based on this specification, DeepDive then uses an efficient statistical inference engine to compute probabilities of the facts in the database. Using a set of tools that facilitate examining erroneous extractions, the user can iteratively adjust the DDLog rules to obtain the desired precision and recall. DeepDive has already been used in several substantial applications, such as detecting human trafficking and creating a knowledge base for paleobiologists with quality higher than human volunteers.

> The following paper is a prime example of groundbreaking work in the area of Knowledge Base Construction.

One of the areas the DeepDive project focused on in particular is the incremental aspect of building a database. As noted, in several applications of the system, knowledge base construction is an iterative process. As the user goes through the process of building the knowledge base, the rules used to extract the data change and, of course, the underlying data may change as well. The DeepDive project developed algorithms to efficiently recompute the facts in the knowledge base and to efficiently recompute the probabilities of facts coming from the inference engine. The results show that efficient incremental computation can make a substantial difference in the usability of a KBC system.

Like with any deep scientific endeavor, there is much more research to be done (and for now, too many coffee lovers need to settle for over-roasted coffee because the database of cafés does not exist yet). We hope that reading this paper will inspire you to work on the KBC problem and hopefully to contribute ideas from far-flung fields. C

Alon Halevy is CEO of the Recruit Institute of Technology (R.I.T), Mountain View, CA.