Technical Perspective: Incremental Knowledge Base Construction Using DeepDive

Alon Halevy
Recruit Institute of Technology

Imagine the task of creating a database of all the high-quality specialty cafes around the world so you never have to settle for an imperfect brew. There are plenty of online sources with content relevant to your envisioned database. Cafes may be featured in well-respected coffee publications such as sprudge.com or baristamagazine.com. Data of more fleeting nature may pop up when your coffee-savvy friends note their location by checking in on Facebook or tweeting. Naturally, there is a plethora of books that studied cafes around the world in even more detail.

The task of creating such a database is surprisingly hard. You would begin by deciding which attributes of cafes 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 cafes in text refer to the same cafe in the real world. In the extreme case, you might even want to send someone out to a cafe 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 cafe 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 extaction, machine learning, natural language understanding, and probabibilistic reasoning. Aside from the compelling application detailed above, the problem arises in many other settings. 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 cafes for the purpose of restaurant search or trend analysis.

The paper you are about to read 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 that are 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. 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.

This particular paper focuses on the incremental aspects of DeepDive. 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 are modified and of course, the underlying data may change as well. The paper describes the algorithms used in DeepDive 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 endeavour, 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 cafes 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.