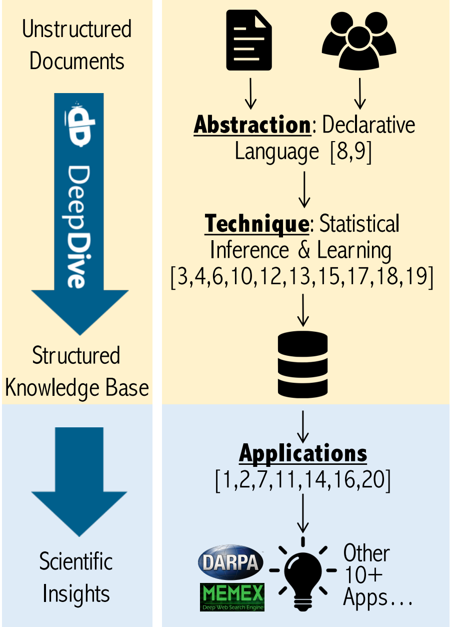
Research Statement

Ce Zhang


Unstructured Documents
DeepDive
Structured Knowledge Base
Scientific Insights
Abstraction: Declarative Language [8,9]
Technique: Statistical Inference & Learning [3,4,6,10,12,13,15,17,18,19]
Applications [1,2,7,11,14,16,20]
DARPA
MEMEX
Other 10+ Apps...

Modern science, analytics, and business intelligence often require macroscopic analysis—it is often necessary to consult a diverse array of information in order to achieve insights, make discoveries, or reach decisions. Unfortunately, much of the critical information that exists in digital form, while increasingly *available*, is often not *accessible* to scientists and enterprise users in a directly usable form. In order to be accessible, data often needs to reside in structured formats such as relational tables in databases; thus, unstructured information, which is in formats such as text, tables, or figures, often remains out of reach. Recently, the term "dark data" has been coined to describe these available but inaccessible data. Should dark data become accessible, it could facilitate a range of scientific and enterprise efforts. With this goal in mind, my research focuses on bringing dark data into the light. Specifically, I focus on *Knowledge Base Construction (KBC)*, the process that takes unstructured documents as input and constructs a knowledge base, i.e., a relational database that stores factual information.

Executive Summary. My work on KBC has primarily involved creating DeepDive, a *data management system* designed to facilitate the end-to-end workflow of building a KBC system. DeepDive's central goal is to *enable experts in domains outside of the computer sciences to build high-quality KBC systems that provide data they need for their applications*. To achieve this goal, my research consists of three areas of investigation:

(1) Application and Quality. I have collaborated with experts in more than ten scientific domains, including geology, paleontology, pathology, and medical genetics, to understand *how KBC could help their applications*. One barrier to applying KBC to scientific applications is the quality of its extractions; thus, I studied the quality aspect of KBC systems [2, 5, 16]. These studies, somewhat surprisingly, show that it is possible to build a KBC system that achieves comparable, and sometimes better, quality results than professional human volunteers in many domains. The key is to process a diverse set of resources, such as text, tables, external KBs, and domain knowledge, all in a single joint probabilistic framework. This work not only supports the development of KBC systems that facilitate scientific discovery within each specific domain [1, 7, 11, 14], but also sheds light on core system challenges on which other investigations need to focus.

(2) Performance and Scalability. One lesson I learned from the investigations around quality was that, to get high-quality results, DeepDive needs to be able to jointly process multiple noisy sources in a single framework. This requirement is also consistent with the recent arms race of scalable statistical inference and learning. For example, in DeepDive, the diversity of the input sources and volume of input often produces statistical inference tasks that are #P-hard in general and involve terabytes of data. Scaling statistical inference and learning up to these statistical workloads is a difficult computational task. In my research, I have completed studies on this problem that have led to an increase in speed of up to two orders of magnitude for workloads such as Gibbs sampling for Bayesian inference [17], generalized linear models for statistical analytics [18], and convolutional neural networks for deep learning [6]. This set of techniques serves not only as the cornerstone of DeepDive, but can also be applied more broadly to statistical analytics in general.

(3) Abstraction and Usability. One goal of DeepDive is to allow experts in domains outside of computer science to specify their applications in a *declarative* way. That is, they are not required to understand the inference techniques underlying their applications. Deriving scalable and efficient execution from declarative descriptions is a core database challenge. Based on my experience interacting with domain experts, I designed a *relational*, *probabilistic*, and *iterative* language model inside DeepDive that allows users to specify a large range of applications and to iteratively debug the quality of their applications. In creating this model, I faced numerous technical challenges—for instance, how to maintain the statistical program's iterative execution incrementally and efficiently. To help resolve such challenges, I conducted studies in this area on popular workloads, such as feature selection [15] and feature engineering [12]. As a result of this language model and these techniques, most DeepDive applications are currently built mainly by its end users.

Impact. DeepDive has already had an impact in a range of scientific and social domains. In the social arena, DeepDive is the main data provider for the DARPA MEMEX program, whose goal is to fight human trafficking. As such, law enforcement agencies are evaluating DeepDive for real-world deployment. In the sciences, DeepDive is being used by more than a dozen scientists from Stanford Hospital, the earth sciences, bioinformatics, and genetics. One application I built for paleontology [11] has been featured in the *Nature* magazine. DeepDive also won the popular TAC-KBC competition in 2014 [2]. The DeepDive system is now being commercialized as a company to expand its availability to more users. Last but not least, the inference engine is also wildly used by industrial tools such as Impala and MADlib.

Thesis Research

My thesis and postdoctoral research have proceeded as a series of iterations of *application*, *technique*, and *abstraction*. Applications from domain scientists provide the motivation and prioritize a series of core computer science challenges; these challenges become the focus of a series of studies of relevant techniques, and these techniques are then made available to the user via a declarative abstraction that in turn facilitates more applications.

I believe solid engineering is the prerequisite of solid systems research. My research into KBC started with DARPA's machine-reading program. The goal was to extract relationships between people, organizations, and locations from Web pages. During the first year of my PhD, I led the engineering efforts in constructing a system that participated in DARPA's evaluation. Although this experience mainly consists of intensive engineering, it established the basis for my whole thesis and postdoctoral research. This prototype system sparked the possibility of building high-quality KBC systems, but the principled solutions for challenges such as application, performance, scalability, and abstraction remain unclear.

Application and Quality: PaleoDeepDive Paleontology is based on the description of fossils. Unfortunately, most fossil data are buried in journal articles published over the past four centuries. To make these data usable by paleontologists, nearly two decades ago, a team of more than 300 scientists spent nearly 10 continuous person years manually reading 40K publications to compile one of the largest paleontological knowledge bases, PaleoBioDB, which has been the basis of more than 200 scientific publications. This stunning amount of human effort and the huge scientific impact of PaleoBioDB raised the question: *Can we apply our prototype system built for DARPA to automate this process but still achieve human levels of quality?* This motivated my collaboration with geoscientist Shanan Peters to build PaleoDeepDive [11]. The goal was to take as input journal articles and construct knowledge bases in the same schema as PaleoBioDB. I conducted a series of studies on the quality of this process.

(1) Rule-based vs. Inference-based Systems. State-of-the-art KBC researches are usually conducted along one of two different lines—those based on hard rules and those based on statistical inference. I conducted studies [8, 9] to investigate the impact of statistical inference on the quality of KBC. We found that, with *similar engineering efforts* and *enough training examples*, across a range of five different applications, the inference-based approach often achieves higher quality. This is not surprising, because for the inference-based approach, the user only needs to specify what features are *potentially relevant* and let the inference algorithm decide their *quality (weights)*; however, for a rule-based system, the developer must deal with both. This decoupling of *relevancy* and *quality* allows users to add many relevant but possibly noisy sources to boost the quality of a KBC system quickly.

(2) Directly Supervised vs. Distantly Supervised Systems. As my study about inference-based KBC systems reveals, for an inference system to be able to make decisions about the quality of features, it is important to have a large set of training examples. Thus, I investigated different ways of creating such examples. As for textbook supervised machine-learning systems, many such training examples can be manually labeled by experts, which results in high-quality training examples, but this process is usually expensive and time-consuming. Instead, I studied [16] two less expensive but potentially lower quality, noisy ways of generating training examples, namely *distant supervision* and *crowd-sourcing*. Distant supervision allows the user to write rules to generate training examples from an unstructured corpus by linking it to an existing KB, while crowd-sourcing allows the user to delegate the labelling procedure to non-professional workers. This study found that distant supervision can achieve better quality given a large enough unstructured corpus, and the key insight is that the noise produced by the supervision procedure can be mitigated by a large number of training examples with statistical learning.

(3) Single-modality vs. Multi-modality Systems. The above studies formed the basis for building the first version of PaleoDeepDive, which extracts information from sources such as text and tables separately. However, this first prototype had a less than satisfying quality—To understand a table inside a document, we often need to consult information in other tables or text in the rest of the document. This motivated a follow-up study [5] in which I designed a joint inference scheme to conduct inference concurrently across text and tables. This study shows the significant impact of such a joint inference on the quality of PaleoDeepDive; however, it introduced statistical inference tasks that often require terabytes of data, which pose challenges for the scalability and performance of the system. This motivated the main technical focus of my dissertation.

Discussion. With this engine and the series of study on quality, for all fossil-related relations in PaleoBioDB, PaleoDeepDive achieves comparable, and sometimes better, quality than human volunteers [11] and forms the foundation of another dozen of applications [1, 2, 7, 11, 14, 20] that follow similar approaches.

Performance and Scalability: Scalable Statistical Inference and Learning. The array of studies shown above demonstrate the necessity for a scalable and efficient engine that can conduct joint statistical inference and learning. Compared with a traditional relational workload, one key difference of a statistical workload, as I found in many applications [18], is the decoupling of *hardware efficiency* and *statistical efficiency*. That is, an implementation that takes full advantage of the hardware might converge slowly from a statistical perspective, and thus lead to suboptimal end-to-end efficiency compared with a more balanced implementation. Worse, there is only a limited body of theory to guide the choice between these two angles. Thus, my research takes a *systems approach* for both scalability and efficiency.

(1) Scalable Statistical Inference with Gibbs Sampling. As statistical inference is often #P-hard in general, one workhorse approximate algorithm is Gibbs sampling. I conducted a study [17] on running Gibbs sampling over data that does not fit in the main memory. My approach is to revisit three classic database techniques that are used in storage managers: *materialization*, *page-oriented layout*, and *buffer-replacement policy*. After studying the tradeoffs associated with the different choices of these techniques, I developed a prototype system that achieves an increase in speed of up to two orders of magnitude over traditional baseline approaches. This study also revealed the potential of adapting classic database techniques to this new workload after a systematic revisit.

(2) Performant Main-memory Statistical Analytics. I then conducted a study on how to speed up the execution of inference inside the main-memory buffer. The need for this type of research is also relevant to an industrial trend; today, even small organizations have access to machines with large main memories. Not surprisingly, there has been a flurry of activity to support main-memory analytics in both industry and research. However, each of these systems often picks one design point in a larger tradeoff space. Thus, my research [18] seeks to define and study this tradeoff space, focusing on commodity multi-socket, multi-CPU, non-uniform memory access (NUMA) machines. This tradeoff space contains axes such as *data access methods*, *data replication*, and *model replication*. From this study, I found that today's research and industrial systems often underutilize commodity modern hardware for analytics, sometimes by two orders of magnitude. This study results in a system called DimmWitted, the workhorse inference engine inside DeepDive that supports all its execution of statistical inference algorithms.

Abstraction and Usability: Iterative Feature Engineering. The above studies, along with my other work in systems [6, 10] and *collaborative* work in theory [3,4,13,19], forms the cornerstone of an efficient DeepDive engine. One remaining challenge is determining what abstraction and interface DeepDive should provide to the user. One observation I made by observing scientists' use patterns is that building high-quality KBC systems is not a one-shot process; instead, it is an iterative process in which the user keeps trying different combinations of features and executes statistical inference on different but similar tasks. Therefore, I focused on studying an iterative abstraction for two popular workloads: feature selection and feature engineering.

(1) Iterative Feature Selection. Feature selection is the process of selecting a set of features that will be used to build a statistical model—a process that is widely regarded as the most critical step of statistical analytics. We found that [15] feature selection is an interactive human-in-the-loop process. For this reason, we designed a declarative language to specify a feature selection workload. This declarative language and the iterative model mean that feature selection workloads are rife with reuse opportunities. Thus, I studied how to materialize portions of this computation, using not only classical database optimizations but also methods that have not previously been used in databases, including *structural decomposition methods* and *warmstart*. This study found that traditional database-style approaches that ignore these new opportunities are more than two orders of magnitude slower than an optimal plan in this new tradeoff space across multiple execution backends, and that a simple cost-based optimizer can often automatically select a near-optimal execution plan for feature selection.

(2) Iterative Feature Engineering. The workload of developing a KBC system is often more complicated than just feature selection. In most applications built with DeepDive, we have seen a spectrum of changes that can be made during the development—quality requirements change, new data sources arrive, and new concepts are needed in the application. This finding motivates my desire to develop techniques for making the entire pipeline incremental in the face of changes, both to the data and to the DeepDive program. The first component of this study is a language similar to Datalog that allows the user to specify feature extraction, distant supervision, and statistical inference and learning using a unified, relational, and declarative language. The main technical challenge is to incrementally maintain statistical inference and learning given a changed factor graph. I propose two methods for incremental inference based respectively on sampling and variational techniques, and I further study the tradeoff space of these methods in order to develop a simple rule-based optimizer [12]. These techniques speed up KBC inference tasks by up to two orders of magnitude with negligible impact on quality. This work forms DeepDive's current language and interaction model, which most of our applications build on.

Future Research

I believe that in the coming decade scientists, analysts, and business users will fully embrace the advancement of statistical inference and machine learning. If this comes to pass, I envision an entity, most likely a research center, that provides *statistical inference as a service* to the whole university or private enterprise. The central goal of this center would be to enable scientists in domains outside of the computer sciences to integrate sophisticated inference and learning into their research without worrying about, or making *ad hoc* compromises on, issues of *quality*, *performance*, *scalability*, and *computational resources*. My dream is to devote the first part of my future career to creating such a center, serving my university and the wider community as well.

I believe that for most grand dreams to come true, they need to be broken into a series of components and milestones that are *systematic*, *concrete*, and *testable*. In this case, I plan to continue my research on *data management systems*; continue exercising the iterations of *application*, *technique*, and *abstraction*; and continue to collaborate with a large, diverse set of users from hospitals, natural sciences, social sciences, and business. Given this plan, I believe the future research activities described below will become feasible soon.

Application and Quality: Statistical Multimodality Fusion. My dissertation speeds up and scales up a range of statistical models, sometimes by orders of magnitude. However, I have observed that, after scientists are equipped with these *individual* tools, they often require a *combination* of these tools for their research. For example, when one of my collaborators at Stanford Hospital tried to develop models to predict the survival rate of lung cancer patients according to their histopathological images, he used both a convolutional neural network for image processing and factor graphs to encode state-of-the-art survival analysis models. Without a proper framework, his system was built by an *ad hoc* pipeline between these two systems. I see similar examples in other domains, such as paleontology and the DARPA MEMEX program, in which users need to concurrently take advantage of images, text, and their domain knowledge. These observations raise the question, *Is there a principled way that different statistical tools interact with each other, and can this interaction facilitate users' applications?* I believe the answer is positive. A principled "fusion" framework has the potential to contribute a major boost to the quality of a range of scientific applications by taking advantage of a much more diverse set of information.

Performance and Scalability: "Black Box" Acceleration. My dissertation regards each statistical model as a "white box." That is, to speed up a workload, such as Gibbs sampling or deep learning, I assumed a full understanding of the underlying execution. This approach has been shown to be effective in speeding up each model after systematic studies of the system tradeoffs. However, there is a range of existing statistical tools that cannot be regarded as a white box, many of which involve decades of engineering efforts, such OCR software and linguistic parsers. Treating these tools as white boxes is challenging, because that requires a full understanding of their own engineering complexity. Thus, one potential question to ask is, *To what extent can we scale up and speed up these tools by treating them as black boxes?* Promising candidate techniques include boosting to decrease the sample space and error-correcting tournaments to decrease the prediction space; however, a systematic study needs to be conducted to understand, adapt, or improve these techniques. I believe the exploration of this question will open a new line of research on par with the white box approach, as in my dissertation. If this black box approach succeeds, users will have a much larger set of efficient and scalable statistical tools in their arsenal.

Abstraction and Usability: Declarative Statistical Inference. Nowadays, given a machine learning model, users have an abundance of choices of systems to use. Each tool has a different abstraction and syntax. This phenomenon appears for statistical inference, in which systems like Church, Alchemy, PrDB, DeepDive, or Infer.Net can all be used to specify an inference task. Despite their similarity, these tools often provide a *complementary* set of functionality. Take Bayesian inference and learning, for example—out of all 49 related papers published in NIPS 2014, there is no single system that can implement all these workloads. This is a barrier to users wishing to take advantage of these systems. One potential question to ask is, *Is it possible to build a single (meta) language that provides a unified interface for statistical inference and learning that supports a superset of semantics of all these tools?* My experience in benchmarking the performance of DeepDive with all these tools suggests that the answer to this question might be positive, and I conjecture that a model that combines factor graphs and stochastic process could form the basis of this unified interface. If building this unified interface is feasible, I expect users will enjoy a greater degree of freedom in modelling their applications instead of worrying about a specific tool.

References

[1] ———. *Under Review (Nature Communications)*, 2015.

[2] G. Angeli, S. Gupta, M. J. Premkumar, C. D. Manning, C. Ré, J. Tibshirani, J. Y. Wu, S. Wu, and C. Zhang. Stanford's distantly supervised slot filling systems for KBP 2014. In *Text Analysis Conference Proceedings*, 2015.

[3] C. De Sa, C. Zhang, K. Olukotun, and C. Ré. Taming the wild: A unified analysis of Hogwild!-style algorithms. *NIPS*, 2015.

[4] C. De Sa, C. Zhang, K. Olukotun, and C. Ré. Rapidly mixing Gibbs sampling for a class of factor graphs using hierarchy width. *NIPS*, 2015. **Spotlight**.

[5] V. Govindaraju, C. Zhang, and C. Ré. Understanding tables in context using standard NLP toolkits. In *ACL*, 2013.

[6] S. Hadjis, F. Abuzaid, C. Zhang, and C. Ré. Caffe con Troll: Shallow ideas to speed up deep learning. In *DanaC*, 2015.

[7] E. Mallory, C. Zhang, C. Ré, and R. Altman. Large-scale extraction of gene interactions from full text literature using DeepDive. *Bioinformatics*, 2015.

[8] F. Niu, C. Zhang, C. Ré, and J. W. Shavlik. DeepDive: Web-scale knowledge-base construction using statistical learning and inference. In *VLDS*, 2012.

[9] F. Niu, C. Zhang, C. Ré, and J. W. Shavlik. Elementary: Large-scale knowledge-base construction via machine learning and statistical inference. *Int. J. Semantic Web Inf. Syst.*, 2012.

[10] F. Niu, C. Zhang, C. Ré, and J. W. Shavlik. Scaling inference for Markov logic via dual decomposition. In *ICDM*, 2012.

[11] S. Peters, C. Zhang, M. Livny, and C. Ré. A machine-compiled macroevolutionary history of Phanerozoic life. *PLoS One*, 2014.

[12] J. Shin, S. Wu, F. Wang, C. D. Sa, C. Zhang, and C. Ré. Incremental knowledge base construction using DeepDive. *PVLDB*, 2015. **Invited to VLDB Journal "Best of VLDB 2015"**.

[13] S. Sridhar, S. J. Wright, C. Ré, J. Liu, V. Bittorf, and C. Zhang. An approximate, efficient LP solver for LP rounding. In *NIPS*, 2013.

[14] C. Zhang, V. Govindaraju, J. Borchardt, T. Foltz, C. Ré, and S. Peters. GeoDeepDive: statistical inference using familiar data-processing languages. In *SIGMOD*, 2013.

[15] C. Zhang, A. Kumar, and C. Ré. Materialization optimizations for feature selection workloads. In *SIGMOD*, 2014. **SIGMOD 2014 Best Paper Award**.

[16] C. Zhang, F. Niu, C. Ré, and J. W. Shavlik. Big data versus the crowd: Looking for relationships in all the right places. In *ACL*, 2012.

[17] C. Zhang and C. Ré. Towards high-throughput Gibbs sampling at scale: a study across storage managers. In *SIGMOD*, 2013.

[18] C. Zhang and C. Ré. Dimmwitted: A study of main-memory statistical analytics. *PVLDB*, 2014.

[19] Y. Zhou, U. Porwal, C. Zhang, H. Q. Ngo, L. Nguyen, C. Ré, and V. Govindaraju. Parallel feature selection inspired by group testing. In *NIPS*, 2014.

[20] Y. Zhu, C. Zhang, C. Ré, and L. Fei-Fei. Building a large-scale multimodal knowledge base for visual question answering. *ArXiv e-prints*, 2015.