Epistemological Databases and Human-Machine Cooperation for KB Construction

Michael Wick
University of Massachusetts
140 Governors Drive
Amherst, MA
mwick@cs.umass.edu

Karl Schultz
University of Massachusetts
140 Governors Drive
Amherst, MA
kschultz@cs.umass.edu

Andrew McCallum
University of Massachusetts
140 Governors Drive
Amherst, MA
mccallum@cs.umass.edu

ABSTRACT

Large knowledge bases (KBs) support real-world decision making by providing access to information in a structured format that facilitates pattern analysis and semantic queries. However, this structured information must first be gathered from disparate sources, typically requiring both extraction and integration. Traditionally, an external system performs these tasks and injects its resulting entities and relations directly into the database; but this is undesirable because when new augmenting and correcting data arrives later, past integration decisions need to be reconsidered. In this paper we propose a new framework for knowledge base construction and maintenance that we term an epistemological database, so named because the database isn’t given the truth; it must infer the truth from available evidence. More specifically, the raw pieces of evidence (text and structured data to be integrated) are presented to the DB and are stored along with both the intermediate variables and final results of extraction and integration. Truth-discovering inference can then run inside the DB—allowing the system to efficiently revisit previous conclusions as new evidence becomes available. Full provenance is thus a natural by-product of this approach. Moreover this framework readily supports an advantageous representation of human-proposed “corrections” to the KB as simply additional pieces of evidence (e.g., mini-documents expressing that user X claimed that Y was true on date Z)—allowing our system to reason jointly and robustly about old and new textual evidence, old and new human edits, their provenance and reliability. We empirically validate the advantages of our approach including (1) the ability to jointly revisit previous inference decisions 33% error reduction, (2) better incorporation of human edits 43% error reduction, (3) robustness to incorrect human edits, and (4) performance, by disambiguating a database with nearly six million mentions in a few hours on a single CPU.

1. INTRODUCTION

Consider a large database (e.g., of business facts, biomedical papers, or gene ontologies) containing a combination of categorical, relational, real-valued and textual data. There is much structured and unstructured information available for populating this database, but it needs integration: schema alignment, deduplication, and value alignment. To be successful we must manage and combine multiple sources of evidence and uncertainty—uncertainty about the reliability of different sources, uncertainty about the accuracy of extraction, uncertainty about correct integration, and uncertainty about changes over time.

Populating the DB from information extraction (IE) is especially appealing because so much information is provided in unstructured text. Performing joint inference across the typical steps of IE (e.g., part-of-speech tagging, parsing, named entity recognition, relation extraction, coreference) can significantly reduce cascading errors [18, 28, 16]. Joint inference involving cross-document considerations quickly involves more random variables than can fit in memory.

Furthermore, database population is not just “one-shot”, it’s an ongoing process. The database will be presented with new information throughout its life time: natural language text and HTML documents streamed from web-crawlers, rows gathered from external databases and web tables, and RDF triples culled from web ontologies. This new evidence has the potential to augment and correct previous extractions in the database, but this requires having efficient access to the necessary provenance, and intermediate results of inference. We would like to use the support of database technology not just for storing the output of an IE system, but also for IE joint inference itself.

This paper presents epistemological databases, a new paradigm for knowledge base construction and data integration in which the canonical “true” entities and relations in the database are always inferred from extracted/integrated or human-entered data, never injected directly. Throughout the lifetime of the database, new evidence will arrive: new data rows, new field values in old rows, new relational linkages. Meanwhile, truth-discovering inference continues to run inside the database responding to new evidence, revisiting past inference decisions, and retroactively correcting errors.

Furthermore, human editors will notice DB errors and want to make corrections. How should these edits be managed? It is risky to allow users to just directly modify the DB’s notion of “the truth” because sometimes humans will be wrong; other times the human was right, but a new event makes
the old human edit wrong, and new automated integration should overwrite the human’s value; and sometimes humans disagree. We propose an alternative where human edits are modeled as additional “documents” or “data”, to be treated as evidence and reasoned about. Thus we can perform probabilistic reasoning not only about the integration process, changes to the world state over time, and which human edits to incorporate, but also about the reliability/reputation of the human editors themselves (addressed in future work).

We show that an epistemological database can be implemented at large scale using factor graphs to represent the distribution over the truth and Markov chain Monte Carlo (MCMC) inference for truth discovery. Note that our approach deviates substantially from most current probabilistic database systems, which use inference primarily for answering probabilistic queries with respect to a source of uncertainty injected into the database by an external process.

Experimentally, we demonstrate that our epistemological database results in a higher quality KB. First, we show that it is indeed able to revisit and correct previous inference decisions resulting in an 33% error reduction. Second, we demonstrate that treating user edits as evidence allows corrections to propagate throughout the database resulting in an additional 43% improvement over an approach that deterministically treats edits as the truth. We also demonstrate more robustness to incorrect human edits. Finally, we show that our prototype system can scale to a large joint coreference task, disambiguating almost six million mentions in a matter of hours.

2. EPISTEMOLOGICAL DATABASES

Imagine a large knowledge base where information continues to arrive in a stream: web spiders continue to provide raw text and HTML documents, rows of external databases are gradually gathered as input, RDF tuples stream in from a large web ontology, human edits are intermittently proposed by groups of collaborative users. Much of this new evidence is relevant to previously made extraction and integration decisions, but the traditional approach to knowledge base construction cannot take advantage of this streaming data without redoing inference from scratch. Instead, we would like to keep around the intermediate results of extraction/integration so that truth discovering inference can respond to the new evidence, revisit previous inference decisions, and correct errors: all using the infrastructure support of large databases. This is especially important for joint inference across multiple modalities where future data is likely to have a large effect on previous conclusions.

Epistemological Database

Definition: A database in which the existence and properties of entities and relations are not directly input into the DB, they are instead determined by inference on raw evidence input into the DB. Typically the DB stores (a) the original raw evidence, (b) some intermediate random variables capturing the (partial) state of inference, and (c) the entities and relations resulting from inference (and possibly probability distributions thereon).

In a sense, the difference between traditional and epistemological databases is merely a matter of where the boundary lines between systems are drawn (See Figure 1). But this shift in thinking has dramatic and practical implications.

An epistemological database typically comprises three components: (1) a database, (2) a model of uncertainty, and (3) an inference procedure for reasoning under uncertainty. The database component contains a schema, a software infrastructure for managing large data, and optionally a scheme for storing uncertainty. The model (e.g., a graphical model) captures the statistical relationships among the prediction variables and the evidence, and is capable of providing a richer representation of uncertainty than what might be stored in the database. Finally, the truth discovering component supports never-ending inference by not only providing procedures for visiting inference on any part of the data so that any errors can be corrected, but by also providing procedures for prioritizing inference so that it can quickly respond and incorporate new evidence: if inference is to be run forever, what should it run on? We now describe known frameworks for implementing these components.

2.1 Database and Model Components

In order to make epistemological databases feasible, we must be able to represent uncertainty for large interconnected data. We require a probabilistic database (PDB) which is simply a set of possible database instances (worlds) \( W \) endowed with a probability distribution \( p : W \rightarrow [0,1] \) s.t. \( \sum_{w \in W} p(w) = 1 \). However, most current PDB implementations focus on probabilistic query-answering and make design decisions that sacrifice modeling power. This limits their ability to handle the large number of statistical dependencies required for jointly reasoning across the variables of data integration/extraction.

Instead of depending on a PDB technology that is still in its infancy, we will take full advantage of modern machine learning algorithms such as Markov chain Monte Carlo (MCMC) that only require storing a single world-state at a time.
Thus, allowing us to leverage decades of research from the systems community on efficiently storing single possible worlds (using classic deterministic databases). Note however, that because the model captures all the uncertainty in the database, we do not lose any information by choosing to store only a single possible world because other worlds can be lazily materialized with inference [25].

Therefore, it is important that we choose a framework for expressing the models-of-uncertainty that can represent highly complex dependencies. To this end, we propose using graphical models, in particular, factor graphs.

Factor graphs are a formalism for compactly specifying random variables and their dependencies making them capable of representing highly complex probability distributions with succinct relational expressions. Inference and learning algorithms can take advantage of these compact representations enabling a striking combination of efficiency, accuracy and generality, placing them at the forefront of a wide array of application areas, including information extraction [15], coreference resolution [5, 18], information integration [26, 28], and machine vision [24]. Let \( \mathbf{y} \) be the set of unobserved random variables. For example, a random variable might be a distribution over the uncertain value of a person’s job title extracted from their homepage. We call an assignment to all the random variables \( \mathbf{y} \rightarrow \mathbf{w} \). For example, the job title variable might be assigned the value “Professor” in world \( \mathbf{w} \). We call \( \mathbf{W} \) the set of all possible worlds. Further, let \( \mathbf{x} \) be the set of observed statistical evidence, and let \( \Psi \) be the set of factor functions. Intuitively, a factor function \( \psi(y^k, x) \rightarrow \mathbb{R}_+ \) takes as input the assignment to \( k \) random variables \( y^k \) and the evidence \( x \), and outputs a positive real-valued compatibility score indicating how “compatible” the setting of those random variables are with each other in context of the evidence. The factor graph encodes a probability distribution that decomposes into a product of factors:

\[
p(y|x) = \frac{1}{Z} \prod_{\psi \in \Psi} \psi(y^k, x), \quad Z = \sum_{\mathbf{w} \in \mathbf{W}} p(\mathbf{w}|\mathbf{x})
\]

The factor function in the epistemological database might include the same types of compatibility functions borrowed from the information extraction models. For example, named entity recognition would have transition and emission factors, and coreference resolution might have entity-wise compatibility factors. Of course, additional factors that model dependencies across the various IE tasks and examine other evidence such as human edits might also be included.

### 2.2 Truth Discovery Component

Inference is the task of recovering the truth from the uncertainty model (which is a compact representation of the probability distribution over the truth). The goal of inference is to find the possible world \( \mathbf{w}^* \) that is most probable under the model given the evidence (i.e., maximum a posteriori (MAP) estimate):

\[
\mathbf{w}^* = \arg \max_{\mathbf{w} \in \mathbf{W}} p(\mathbf{w}|\mathbf{x})
\]

In many real-world applications of extraction/integration, the factor graph required to encode the probability distribution has such a highly connected dependency structure that computing the most probable world is intractable. Thus, epistemological databases require a never-ending any-time inference routine that constantly improves the quality of the best known solution. This inference routine should also have procedures for visiting any variable, as well as procedures for prioritizing inference efforts (e.g., in response to recent evidence).

Fortunately, Markov chain Monte Carlo (MCMC) is a viable approach to inference in complex graphical models [18, 16] that is sufficiently flexible to satisfy these requirements [25]. Metropolis-Hastings (MH) is an MCMC algorithm that is particularly well suited for inference in epistemological databases because (1) it operates on a single possible world at a time, (2) sampling possible worlds requires evaluating only a small handful of compatibility functions, even for large interconnected factor graphs, allowing inference to operate on portions of the graph that fit in memory, and (3) MH is a highly flexible framework with customizable jump-functions that can be adapted to rapidly respond to new evidence.

MH discovers the truth (e.g., seeks the solution to Equation 2) by initializing to a possible world \( \mathbf{w} \) and then iteratively making incremental local modifications to the current world \( \mathbf{w} \) using a user-defined proposal distribution \( q(\mathbf{w}'|\mathbf{w}) \). That is, conditioned on the current world \( \mathbf{w} \), sample a new world \( \mathbf{w}' \) by modifying the assignment to a small handful of the variables in \( \mathbf{y} \). Then, this proposed world \( \mathbf{w}' \) is accepted with probability

\[
\min \left( \frac{p(\mathbf{w}) q(\mathbf{w}'|\mathbf{w})}{p(\mathbf{w}') q(\mathbf{w} |\mathbf{w}')} \right)
\]

Intuitively, the higher the probability of the proposed world, the higher the probability that the proposed world will be accepted. The second ratio of \( q \)'s compensates for any bias in the proposal distribution. Note, however, that this second term is often omitted for MAP inference [16]. Observe that even if the proposed world has less probability than the current world, it may still be accepted with positive probability. This helps the algorithm avoid local optima.

Much of the flexibility in MH comes from the ability to specify a customized proposal distribution \( q \). We can inject domain-specific knowledge about how to explore the space of possible worlds, and even bias \( q \) so that it is more likely to modify portions of the graph affected by new evidence, as done in [27]. Further \( q \) can be parameterized and these parameters can be learned during inference to provide more fruitful proposals. An example of a proposal function might be to first select a some variable at random, for example, the variable encoding a person’s job title, and then assign that variable a new value from its domain (for example, we change the value from “Professor” to “Student”). Note that in practice there can be multiple MCMC inference workers working in parallel [25, 20].

Much of the efficiency of MH comes from the form of the acceptance function (Equation 3): when taking the ratios of the two worlds \( Z \) cancels as well as all the factors with variables that were not affected by the proposal. Thus to determine whether a new world should be accepted as a sample, we only need to evaluate factors that neighbor variables...
whose values have changed. For a wide variety of information extraction problems, including large-scale cross document coreference [20], the number of variables that need to be in memory is proportional to the number of variables modified by the proposal, and not proportional to the size of the database. This is essential if we hope to do inference across an entire database.

3. HUMAN EDITS

We can imagine a large database serving the web and tapping a large user-base of human collaborators. For example, the impact of Wikipedia has been revolutionary in this respect. However, integrating user edits with information extraction in the context of structured relational data poses many challenges for systems that allow users to directly edit the “truth” by changing values in the database. First, the constraints and logic of the domain must remain intact when a user makes an edit. One user may make an edit that violates the logical constraints imposed by another, then how should this violation be resolved? Users may simply disagree about what the value of the truth is, which edit should we trust more? Additionally, how could such a system accommodate users that may only have partial information about what the true value is? For example, the user is only 80% sure that the color of the car is blue. What if a user submits an edit that is partially correct? Finally, the truth itself changes over time in response to new real-world events: relationships between people change, companies fail, new political parties come to power, etc: this is problematic because user edits quickly become obsolete in response to such changes and should be overwritten by incoming IE.

Epistemological databases address all of these concerns because user edits are merely additional pieces of evidence that can be reasoned about probabilistically in the context of other evidence. Edits are simply just mini-documents containing statements like “user X said on July 21st that person Y worked at organization Z”. Note that the edit can provide information about the user, mentions of entities (person Y and organization Z) and expresses a relationship between these mentions. These documents can then be parsed, the entities identified through coreference resolution, and the relationships reasoned about probabilistically by the system.

4. REXA 2.0: A PROTOTYPE EPistemOLOGICAL DATABASE

Reasoning about academic research, the people who create it, and the venues/institutions/grants that foster it is a current area of high interest because it has the potential to revolutionize the way scientific research is conducted. For example, if we could predict the next hot research area, or identify researchers in different fields who should collaborate, or facilitate the hiring process by pairing potential faculty candidates with academic departments, then we could rapidly accelerate and strengthen scientific research. A first step towards making this possible is gathering a large amount of bibliographic data, extract mentions of papers, authors, venues, and institutions, and perform massive-scale cross document entity resolution (coreference) and relation extraction to identify the real-world entities.

To this end, we implement a prototype epistemological data-base for bibliographic data that will eventually replace the original REXA. Currently, we have spidered the web for BibTeX files, and have combined this with DBLP to create a database with over ten million mentions (6 million authors, 2.3 million papers, 2.2 million venues, and 500k institutions). We perform joint coreference between authors, venues, papers, and institutions at this scale. Recall that an epistemological database has three components: the database, the model, and the truth discovering inference procedure. For the database component we use the key-value store MongoDB, which allows indexing and querying over JSON style documents. For the uncertainty model we implement a hierarchical model of coreference that recursively structures the predicted entities into mention trees, and also contains additional factors to jointly reason about user corrections. Finally, for truth discovering inference we employ the Metropolis-Hastings algorithm. We now describe further details of the model and inference and how they interact to combine user edits.
4.1 Hierarchical Coreference inside the DB

Entity resolution is difficult at any scale, but is particularly challenging on large bibliographic data sets or other domains where there are large numbers of mentions. Traditional pairwise models (e.g., [21, 17]) of coreference—that measure compatibility between pairs of mentions—lack both scalability and modeling power to process these datasets. Instead, inspired by a recently proposed three-tiered hierarchical coreference model [20], we employ an alternative model that recursively structures entities into trees. Rather than measuring compatibilities between all mention pairs, instead, internal tree nodes might summarize thousands of leaf-level mentions, and compatibilities are instead measured between child and parent nodes. For example, a single intermediate node might compactly summarize one thousand “F. Pereira” mentions. Compatibility functions (factors) measure how likely a mention is to be summarized by this intermediate node. Further, this intermediate node may be recursively summarized by a higher level node in the tree. We show an example of this recursive coreference factor graph instantiated on two entities in Figure 2a.

For inference, we use a modified version of the Metropolis-Hastings algorithm that proposes multiple worlds for each sample. In particular, each proposal selects two tree nodes uniformly at random. If the nodes happen to be in the same entity tree, then one of the nodes is made the root of a new entity. Otherwise, the two nodes are in different entity trees, then we propose to merge the two sub-tree’s together by either merging the second subtree into the first subtree, or merging the second subtree into the root of the first subtree. If two leaf-level nodes (mentions) are chosen, then a new entity is created and the two mentions are merged into this newly created entity. We describe these proposals and the hierarchical coreference model in more detail in a forthcoming paper [1].

4.2 Human edits for entity resolution

Broadly speaking, there are two common types of errors for entity coreference resolution: recall errors, and precision errors. A recall error occurs when the coreference system predicts that two mentions do not refer to the same entity when they actually do. Conversely, a precision error occurs when the coreference error incorrectly predicts that two mentions refer to the same entity when in fact they do not. In order to correct these two common error types, we introduce two class of user edits: should-link and should-not-link. These edits are analogous to the must-link and must-not-link constraints used in constrained clustering problems; however, they are not deterministic constraints, but extra suggestions via factors.

Each coreference edit in fact introduces two new mentions which are each annotated with the information pertinent to the edit. For example, consider the recall error depicted in Figure 2a. This is a real error that occurred in our system: there is simply not enough evidence for the model to know that these two Fernando Pereira entities are the same person because the co-authors do not overlap, the venues hardly overlap, and the topics they write about do not overlap. A user might notice this error and wish to correct it with an edit: “user X declared on this day that the Fernando Pereira who works with conditional random fields (CRFS) is the same Fernando Pereira who works on natural language processing (NLP) tasks”. Presenting this edit to the Rixa 2.0 database involves creating two mentions, one with keywords about CRFs and the other with keywords about NLP; and both are annotated with a note indicating user X’s belief: “user x: should-link”. Then, special factors in the model are able to examine these edits in the context of other coreference decisions. As MCMC inference explores possible worlds by moving mentions between entities, the factor graph is able to reward possible worlds where the two mentions belong to the same entity. For example, see Figure 2b. In our experiments, this coreference error was actually corrected by a human edit of this nature.

5. EXPERIMENTS

In this section we empirically evaluate the ability of the epistemological database to incorporate evidence from a variety of structured and unstructured sources including user input. We also demonstrate scalability of our prototype epistemological database by evaluating it on a problem of large scale author disambiguation.

5.1 Data

We create an epistemological version of a bibliographic research database. For the purpose of these experiments we focus on the problem of author coreference (disambiguation). Author coreference is a notoriously difficult problem due to common first and last names, spelling errors, extraction errors, and lack of “within document boundaries.”

In order to evaluate our approach, we label a highly ambiguous “F. Pereira” dataset from BibTeX files. We constructed this dataset using two strategies. First, we spidered the web for BibTeX files and retained only entries that have an author with last name “Pereira” and first name beginning with “F.” The purpose of this initial phase is to retrieve a realistic variety in spelling variations (e.g., “F.”, “Fernando”, “Francisco”). Each of the Pereira mentions gathered in this manner are manually disambiguated by identifying the real-world author to which they refer. Second, we identified five prominent Pereira entities from the initial labeling and for each of these we were able to find their publication page and enter each publication into our dataset manually. The number of mentions in the five entities is as follows: (181 mentions, 92 mentions, 43 mentions, 7 mentions, 2 mentions). We will make this dataset publicly available for other researchers.

5.2 Author coreference and model description

We create mention records from the BibTeX entries in the fpereria database by extracting the first/middle/last names, and then create bags-of-words from co-authors, venues, titles, and also topics (trained using latent Dirichlet allocation [2] on 1.3 million titles). Then, in order to perform author coreference resolution, we employ the recursive coreference model discussed in Section 4.1. In particular, we implement two coreference factor templates. The first is a parent-child template that measures the compatibility between a child entity-node and its parent. Due to the paucity of labeled data we cannot afford to train the model, but instead set the weights of our factors by hand to punish a mismatch of first, middle, and last name, (−8); reward a match (+2); and
reward for initials matching (+1). Additionally, to exploit context, we use the cosine similarity between various bags of words: venue tokens (shifted and scaled between −1 and 1), topics (shifted and scaled between −2.5 and 2.5), and co-author last names (shifted and scaled between −2 and 2). Second, we implement priors over the latent structure of the entity-trees. We encourage each node to have eight children using 1/(number of children−8)+1), manage tree depth by placing a cost on the creation of intermediate tree nodes of −2 and encourage clustering by placing a cost on the creation of root-level entities of −3. For human edits, we add a template that rewards a should-link constraint with a score of 1.5 whenever the two mentions are in the same entity, and punishes a should-not-link constraint with a score of -16 when they are not. We use the MCMC inference algorithm discussed in Sections 2.2 and 4.

5.3 Ability to incorporate streaming evidence
An important advantage of epistemological databases is their ability to revisit inference decisions and harness new arriving evidence (from both structured and unstructured sources) in order to retroactively correct errors in the DB. In this section we experimentally evaluate the ability of our epistemological DB to retroactively correct previous errors. Our methodology is as follows. We first construct an initial author coreference database by randomly selecting 25% of the mentions from the fperereia dataset. We call this the “original” database, and for these experiments we evaluate pairwise F1 accuracy exclusively on these original databases in order to isolate the effects of retroactive error correction on database quality. As explained next, we experimentally incorporate different types of evidence.

Streaming structured data In order to simulate the arrival of structured data, we use the remaining 75% of the mention from the fperereia dataset and incrementally present them to the original database in a random order, presenting 2.5% (27 mentions) of them at a time. After the arrival of each new batch, inference is run for a fixed number of proposals (20,000), and the F1 accuracy is recorded. The experiment proceeds in this way until the remainder of the mentions have been incorporated. We repeat this process 30 times, and present the average in Figure 3a.

Streaming unstructured data In order to simulate the arrival of unstructured data, we employ a similar procedure. Again, we create the initial database using a randomly selected 25% of the mentions. Then, instead of streaming in the remaining 75%, we build an extractor to segment author mentions from the homepages of two of the most prominent F. Pereira entities, and a handful of their co-authors. This results in an additional 550 mentions, which are divided into batches of 27 mentions. Again, this procedure is run 30 times and averaged; we present the results in Figure 3b.

Discussion As we expect, the epistemological database is able to retroactively correct errors as new evidence arrives. It is interesting that—despite the use of a crude extractor—the mentions extracted from the unstructured sources provide an equally useful signal for error correction. One reason is that the topic models we run on the title segments may be mitigating the negative impact of segmentation errors.

5.4 Human edits
We also evaluate the ability of the epistemological database to incorporate human edits. We argued earlier that users should not be allowed to directly edit the value of the truth because of the complications that may arise: domain-specific constraint/logical violations, disagreement about the truth, incorrect edits, etc. In this section, we test the hypothesis that the epistemological database is better able to incorporate human edits than a more direct approach where users can edit the database content at will. To this end, we design two experiments to evaluate database quality as the number of human edits increase. In the first experiment, we stream “good quality” human edits to the database, and in the second experiment we stream “poor quality” human edits (we will define what we mean by this in more detail later). As in the previous experiment, we first create an initial database using the mentions in the fperereia dataset, and run MCMC until convergence reaching a precision of 80, and F1 of 54.

Next, given this initial database of predicted author entities,
we measure the influence of both “good quality” (correct) and “poor quality” (incorrect) human edits. Although assessing the quality of a user edit is a subjective endeavor, we are still able to implement a relatively objective measure. In particular, we take the set of Pereira author entities initially discovered in the “original” DB and consider all possible pairs of these entities. If merging a pair into the same entity would increase the overall F1 score we consider this a correct human edit; if the merge would decrease the score we consider this an incorrect edit. Note that this reflects the types of edits that might be considered in a real-world bibliographical database where a user would browse two author pages and decide (correctly or incorrectly) that they should be the same entity. For example, one of the good quality pairs we discover in this way encodes the simulated “user’s” belief that the “the Fernando Pereira who works on NLP is the same Fernando Pereira who works on machine learning”. An example of a poor quality edit is “the Fernando Pereira that researches NLP is the same Fernando Pereira that works on MPEG compression”.

Once we have determined which author pairs result in higher or lower F1 accuracy, we can then construct simulated edits of various quality. We consider three ways of incorporating these edits into the database. The first approach, epistemological, which we advocate in this paper, is to treat the edits as evidence and incorporate them statistically with MCMC (using the edit factors from Section 5.2). We convert each entity pair into edit-evidence as follows: two mentions are created (one for each entity), the attributes of the entities are copied into the features of these corresponding mentions, and a should-link constraint is placed between the mentions. The second two approaches simulate users who directly modify the database content. The first baseline, overwrite, resolves conflicts by simply undo-ing previous edits and overwriting them, and the second baseline, maximally satisfy, applies all edits by taking their transitive closure.

**Good quality edits**

In Figure 4a we compare our epistemological approach to the two baselines overwrite and maximally satisfy on the set of good user edits (averaged over 10 random runs). What is interesting about this result is that the epistemological approach, which is not obligated to merge the edited entities, is actually substantially better than the two baselines (which are deterministically required to merge the edited entities). After some error analysis, we determine that a major reason for this improvement is that the user edits propagate beyond the entity pair they were initially intended to merge. In particular, as the user edits become applied, the quality of the entities increase. As the quality of the entities increase, the model is able to make more accurate decisions about other mentions that were errorfully merged. For example, we observed that after MCMC inference merged the natural language processing Fernando with the machine learning Fernando, that an additional 18 mentions were correctly incorporated into the new cluster by inference. In a traditional approach, these corrections could not propagate thus placing the burden the users to provide additional edits.

**Poor quality user edits**

In Figure 4b we evaluate the robustness of our epistemological database to poor quality (incorrect) human edits. In this figure, we evaluate quality in terms of precision instead of F1 so that we can more directly measure resistance to the overzealous recall-oriented errorful must-link edits. The baseline approach that deterministically incorporates the errorful edits suffers rapid loss of precision as entities become merged that should not be. In contrast, the epistemological approach is able to veto many errorful edits when there is enough evidence to warrant such an action (the system is completely robust for twenty straight errorful edits). Surprisingly, the F1 (not shown) of the epistemological database actually increases with some errorful edits because some of the edits are partially correct, indicating that this approach is well suited for incorporating partially correct information.

### 5.5 Scalability experiments

Finally, we demonstrate the scalability of our prototype epistemological database, Reza 2.0, on three large datasets. The
5: Coreference scalability to large databases.

first dataset consists of 1.3 million author mentions extracted from BibTeX files gathered by a web-crawl. The second dataset is DBLP and has 4.4 million author mentions. The third dataset is a combination of the two, and contains over 5.7 million author mentions. We supplemented each of these datasets with an additional 2000 labeled author mentions from the original REXA dataset. For each experiment we initialize REXA 2.0 to the singleton configuration (every mention is in its own cluster), and then run MCMC inference in the hierarchical model until convergence. We compare accuracy versus time for the three datasets in Figure 5.5. Impressively, we are able to achieve a high accuracy in a relatively small amount of time on all three datasets (only a few hours). This is a promising result as it indicates that epistemological databases have the potential to be real-world deployable in the near-future. We hope to make REXA 2.0 available soon, with full user-editing capabilities.

6. RELATED WORK

There has been a tremendous amount of interest in knowledge base construction, including both automatic and manual approaches. Examples of manually constructed knowledge bases include Wikipedia, it’s structured counterparts Freebase and Yago-NAGA [12]; Cyc; and ACM. While these manually constructed knowledge bases arguably contain highly accurate data, constructing structured knowledge bases can be achieved at much larger scale when combined with automatic methods such as information extraction.

Examples of automatically populated knowledge bases include bibliographic and academic web portals such as CiteSeer, Google Scholar, REXA, and ArnetMiner [23]; machine reading systems such as TextRunner [9], ReVerb/R2A2 [11, 10], NELL [3], and SOFIE [22]; and many more. However, the accuracy of these automated systems does not always meet the level of reliability required by real-world decision makers. Combining both manual and automatic methods has been recognized as an important challenge in knowledge base construction [7]. We argue that such a combination would lead to knowledge bases with sufficient accuracy to be used for downstream applications.

A common approach for knowledge base construction is to harness the collective wisdom of large groups of collaborative users (e.g., “crowd-sourcing”, “wisdom of crowds”, “collective intelligence”), for a survey see [8]. Wikipedia is one example of this, but the information is natural text, not rich semantically structured data. We are interested in constructing structured data at large scale, which becomes significantly more difficult to manage when user annotations are introduced. Another prominent example is Amazon’s Mechanical Turk, which has been tremendously useful in gathering labeled data for training and evaluation, for example, in machine vision [19]. Mechanical Turk aids us in generating more structured data, and is a natural setting for gathering human edits to improve an automatically generated KB.

An example of a structured database where there is active research in harnessing user feedback is the DBLife project [6]. Chai et al. [4] propose a solution that exposes the intermediate results of extraction for users to edit directly. However, their approach deterministically integrates the user edits into the database and may potentially suffer from many of the issues discussed earlier; for example, conflicting user edits are resolved arbitrarily, and incorrect edits can potentially overwrite correct extractions or correct user edits.

There has also been recent interest in using probabilistic models for correcting the content of a knowledge base. For example, Kasneci et al. [13] use Bayesian networks to incorporate user feedback into an RDF semantic web ontology. Here users are able to assert their belief about facts in the ontology being true or false. The use of probabilistic modeling enables them to simultaneously reason about user reliability and the correctness of the database. However, there is no observed knowledge base content taken into consideration when making these inferences. In contrast, we jointly reason over the entire database as well as user beliefs, allowing us to take all available evidence into consideration. Koch et al. [14] develop a data-cleaning “conditioning” operator for probabilistic databases that reduces uncertainty by ruling out possible worlds. However, the evidence is incorporated as constraints that eliminate possible worlds. In contrast, we incorporate the evidence probabilistically which allows us to reduce the probability of possible worlds without eliminating them entirely; this gives our system the freedom to revisit the same inference decisions not just once, but multiple times if new evidence arrives that is more reliable.

7. CONCLUSION

In this paper we proposed a new paradigm for knowledge base construction called an epistemological database where the truth is always inferred from raw evidence, providing a new way of handling human edits. We implemented a scalable instantiation of this paradigm using factor graphs to represent uncertainty, never-ending MCMC inference to infer the truth, and a deterministic database to store evidence and inference results. Experimentally, we demonstrated that such a system is able to exploit new evidence (from structured, unstructured, and human resources) to correctly revisit old inference decisions. We observed that incorporating
user feedback in this manner allowed corrections to propagate and further reduce errors. Additionally, we observed that the database was more robust to poor quality user edits, a desirable characteristic for real-world deployment. Finally, we demonstrated the scalability of our coreference system with a bibliographic database of DBLP size. In future work we would like use epistemological databases to create large knowledge bases in various domains, and also incorporate inference about user reliabilities.

8. ACKNOWLEDGMENTS
This work was supported in part by the Center for Intelligent Information Retrieval and in part by IARP A via DoI/NBC contract #D11PC20152. The U.S. Government is authorized to reproduce and distribute reprint for Governmental purposes notwithstanding any copyright annotation thereon. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor.

9. REFERENCES
[26] M. Wick, K. Rohanimanesh, A. McCallum, and A. Doan. A discriminative approach to ontology alignment. In In proceedings of the 14th NTII WS at the conference for Very Large Databases (VLDB),