Probabilistic Reasoning about Human Edits in Information Integration

Michael Wick MWICK@CS.UMASS.EDU
University of Massachusetts School of Computer Science, 140 Governors Drive, Amherst, MA, 01002

Ari Kobren AKOBREN@CS.UMASS.EDU
University of Massachusetts School of Computer Science, 140 Governors Drive, Amherst, MA, 01002

Andrew McCallum MCCALLUM@CS.UMASS.EDU
University of Massachusetts School of Computer Science, 140 Governors Drive, Amherst, MA, 01002

Abstract

We present a probabilistic framework for knowledge base (KB) management that allows users to correct data integration errors in the KB. Rather than allowing users to directly edit the KB, we instead treat user edits as assertions on the truth about the entities and relations in the KB. Probabilistic inference then considers the user assertions along with all other available evidence in order to predict the true data values in the KB (and jointly infer the reliabilities of the contributing users). We evaluate our approach on a bibliographic database of computer science authors and show that our system is more effective at correcting entity attribute and coreference errors than traditional approaches that grant users the ability to directly edit the database.

1. Introduction

Data integration is the task of combining data from different sources into a single representation. Usually this involves the automated integration of two databases (Doan & Halevy, 2005); however, the prevalence of computer systems that enable user contributions has made it important to determine how to best integrate data in preexisting knowledge bases (KBs) with user-supplied content (Doan et al., 2009).

One strategy for supporting this integration—of human and machine contributions—is to allow users to directly modify the state of the KB (Chai et al., 2009). However, this can be risky because sometimes humans will be wrong, sometimes humans disagree, and sometimes the human edits become out-of-date as the state of the world changes (and untiring automated extraction and integration should be allowed to displace the human edit).

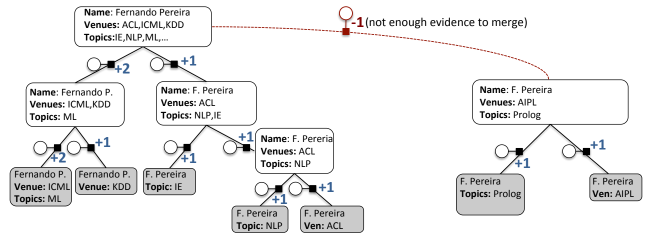
In this paper, we present a probabilistic framework for automatically integrating user-contributed corrections with a KB. Rather than allowing users to directly overwrite the KB's notion of entities and attributes, users instead provide additional records expressing proposed changes to the KB. These records are treated as *evidence* in support of the changes, but the changes are not inevitably adopted. Instead, our system reasons probabilistically about both the user contributed records and all raw pre-integration data. This reasoning continually strives to infer the set of entities and attributes most likely to be true given the evidence. We have termed such a KB an *epistemological database* because the truth about entities and attributes are never input directly into the database, but are instead inferred from raw evidence (Wick, 2012). Note that because entities are inferred from their evidence, coreference is foundational not just for assembling preexisting raw data into entities, but also for resolving user-contributed edits to inferred entities.

We implement an epistemological DB for managing user edits by using a factor graph to represent dependencies between the user-supplied evidence and the random variables that encode the values of the KB content. Further, we include random variables that represent the users' *reliability* (a user's reliability measures the quality of their contributed edits).

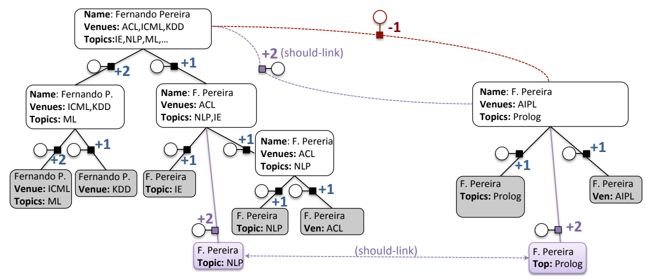
ICML Workshop: Machine Learning Meets Crowdsourcing, Atlanta, Georgia, USA, 2013.

We jointly estimate the values of the user reliability and KB-content variables using expectation maximization (EM). In the M-step we maximize the variables that encode KB content using temperature regulated Markov chain Monte Carlo (MCMC), and in the E-step we estimate the variables encoding user reliabilities.

We study our approach to managing user-edits on the problem of resolving author entities and their attributes in an automatically constructed KB of bibliographic data. In our setup, users may propose edits to both entity coreference and entity attributes. We find that treating user edits probabilistically is more robust to incorrect and malicious editing (50% error reduction) and is often better at integrating corrective edits than traditional deterministic data integration strategies (4.5% error reduction). Further, we find that modeling user reliabilities often increases the accuracy of the KB, and allows the system to distinguish between malicious and benevolent users.



(a) **A recursive coreference model** with two predicted *Fernando Pereira* entities. Black squares represent factors, and the numbers represent their log scores, which indicate the compatibilities of the various coreference decisions. Open circles are binary coreference decision variables. The recall error here is due to lack of contextual evidence.



(b) **Human edit corrects the coreference error** in the previous figure. A human asserts that "the Prolog F. Pereira is also the NLP F. Pereira." by submitting an edit-record containing two edit-mentions (shown in purple) and a SL operator (depicted as an arrow). During inference, the mentions are first moved into different entities. Then, when inference proposes to merge those two entities, the model will give a bonus to this possible world because the two edit-mentions are placed in the same entity.

Figure 1: A recall coreference error **(top)**, is corrected when a user edit arrives **(bottom)**.

2. Background

Our human editing framework employs epistemological databases (Wick, 2012) to integrate human edits with preexisting KB data by performing probabilistic inference on raw evidence. Our representation for data uncertainty is a factor graph with random variables for the truth, observed variables for evidence, and factors for capturing dependencies between these variables (the evidence and the truth).

2.1. Hierarchical coreference

We employ a hierarchical model of coreference that recursively summarizes the attributes of an entity with a tree (mention attributes at the leaves and the inferred entity attributes at the root) (Wick et al., 2012). We implement the model as a factor graph in which the coreference attachment decisions and attribute values are random variables ($\mathbf{Y}$), the mentions are observed evidence variables ($\mathbf{x}$), and the compatibility of the child-parent coreference decisions and inferred attributes values are evaluated using factors Ψ (Table 1 contains their factor specification details). For example, the child-parent factors might measure cosine similarity between the child's features and the parent's features; this factor would encourage the model to select possible worlds where children have similar features as their parents. We show an example of this recursive coreference factor graph instantiated on two predicted entities in Figure 1a.

We use MCMC in coreference to propose changes to entity trees. More specifically, MCMC proposes (1) the existence of a new entity by creating a new root node and attaching two entity trees as children (2) the deletion of an entity by attaching that entity as a subtree to another entity, and (3) changes to entities by moving a sub-tree from one entity to another. Recursive summarization of attributes and context is maintained during MCMC: detaching a subtree causes the removal of that subtree's features from its former ancestral tree, and attaching a subtree causes the addition of that subree's features to its new ancestral tree. For more details, see Wick et al (2012).

2.2. Application to bibliographic KBs

We have implemented a prototype epistemological KB of computer science authors, and use it to help test the viability of our human edit integration method on the problem of author coreference and attribute induction. Author coreference is important for bibliographic

KBs because it is necessary for compiling publication lists for authors. However, it is difficult to solve due to mispellings, alternative abbreviations and common first-initial last name combinations.

Context such as the list of co-authors, keywords in the paper title, and publication venue provide crucial evidence for resolving such ambiguity. Usually, this context is readily available because author mentions are records extracted from the headers and bibliography sections of research papers. In our model, context elements like paper title, co-author lists, publication venue, etc. are each represented as a bag-of-words. These context elements are used to score links between their corresponding mentions via factors.

3. Human edits

In our approach, human edits are additional evidence that influence the inferred truth values of the KB. Users communicate edits to the KB in the *language of mentions*. That is, rather than referring to inferred objects in the KB (e.g., entities, relations, attributes, rows, etc.) by their artificially generated keys, users instead refer to objects using an *edit-mention*: a user-created record containing distinguishing and corrected attributes of an object the editor wishes to change. The KB infers which objects the users intend to edit by resolving these edit-mentions to their target objects via coreference. This method of communication is essential for ensuring that user edits persist in the KB and remain valid throughout ongoing data integration. Entities are ephemeral; if they vanish during inference, direct references to them become invalid.

More formally, a user proposes an edit to the KB via an *edit-record* that contains the following components:

- A set of edit-mentions that identify the KB objects the user intends to edit.
- An edit-operation field that functions as a trigger for certain edit-specific factors in the model.
- An optional set of *edit-payload-mentions* that contain proposed values for the attributes of a target object (e.g., for editing attribute values).
- Meta-data fields (e.g., time-stamp, user id, etc.).

For our epistemological KB to interpret edit documents we include edit-specific factors for each supported edit operation. These factors augment the model which already includes coreference factors (for resolving edit-mentions to entities). The purpose of the edit-specific factors is to increase the score of possible worlds where the user's proposed edit is obeyed.

3.1. Edit operations on entities

We now describe edit operations on entities (attributes, and coreference decisions), but our framework can also be applied to relations and other types of KB objects.

3.1.1. Edits to entity coreference

There are two common types of entity coreference errors: recall errors (under-merging), and precision errors (over-merging). Recall errors occur when the system incorrectly predicts that two mentions refer to different entities; precision errors occur when the system incorrectly predicts that two mentions refer to the same entity.

Two types of edit-records are necessary for enabling users to communicate corrections to coreference: a should-link (SL) edit-record and a should-not-link (SNL) edit-record. An SL edit-record contains a set of mentions identifying (sub) entities that the user believes should be merged. An SNL edit-record consists of a set of mentions that reference (sub) entities the user believes should be in separate entity trees.

Edit-specific factors interpret these two types of edit-records. SL factors output a positive log-score whenever the mentions involved in an SL edit refer to the same entity (a log-score of zero otherwise); SNL factors output a negative log-score whenever the mentions involved in an SNL edit refer to the same entity (zero otherwise). In conjunction with the standard hierarchical coreference factors, these edit factors encourage inference to select possible worlds where the users' coreference assertions are satisfied.

For example, consider the problem of correcting the entity-coreference error in Figure 1a. In this example, the hierarchical model commits a recall-error by incorrectly separating Fernando Pereira into two separate entities. A user wishes to rectify this error and provides an SL edit-record containing the two Pereira mentions. The KB integrates this edit by running MCMC inference on the hierarchical coreference model. Since the edit-mentions contain similar attributes to the target entities, the parent-child factors yield high affinity scores for worlds in which the edit-mentions are resolved to their respective target entities. Furthermore, the SL factor yields a high score for worlds in which the two edit-mentions are in the same entity-tree. In this example , the edit successfully corrects the coreference error because the SL factor has more weight than the factor that opposes the merge.

factor type	input (variables)	parameters	output (log score)
BoW cosine similarity	parent bag (p), child bag (c)	w, t	$w \log(\|c\|_1 + 2) \left(\frac{(p-c)\cdot c}{\|p-c\|_2 \|c\|_2} + t\right)$
entity existence penalty	node (e)	w	$-w\mathbb{1}\{\text{isRoot}(e)\}$
subentity existence penalty	node (e)	w	$-w\mathbb{1}\{\neg\text{isRoot}(e) \wedge \neg\text{isLeaf}(e)\}$
BoW norm. entropy penalty	node's bag-of-words (b)	w	$-w\frac{H(b)}{\log \|b\|_0}$
BoW complexity penalty	node's bag-of-words (b)	w	$-w\frac{\|b\|_0}{\|b\|_1}$
names penalty	node's bag-of-names (b)	w	$-\min\left(w(\|b\|_0 - 1)^2, -16\right)$

Table 1: Precise definitions for factors in the hierarchical coreference model. We assume a sparse-vector representations for a bag of words (b), $|b|_n$ is the l_n norm of bag b, $H(b)$ is the Shannon-entropy of bag b, $\mathbb{1}\{\text{formula}\}$ is an indicator function.

3.1.2. Edits to attributes

Users edit entity attributes (e.g., an author's first name) by supplying attribute-edit-records. These consist of an edit-mention for identifying the target entity, and an edit-payload-mention that specifies the new attribute values. Once coreference identifies the target entity, the attributes in the edit-payload-mention accumulate in the target entity's bag-of-words. Attribute edit factors examine these bags and encourage the model to select the user supplied attributes.

For example, our author coreference model includes a set of factors for inferring canonical first, middle, and last names by using:

- weighted-mean Levenshtein-similarity to other strings in the bag-of-names (encourages the canonical string to be representative of the other name strings (Culotta et al., 2007b)).
- strong preference for proper nouns.
- weak preference for initials (`[A-Z]\.?`).
- an edit-factor that gives additional weight (×2) for strings that occur in the payload mention.

Thus, an attribute edit encourages the model to select the user-provided attribute as the canonical representation, but this is balanced with other factors that encourage the canonical representation to be "representative" of the mentions.

3.2. User reliabilities and reputations

Many real-world KBs (e.g. Wikipedia) have large, diverse populations of users. Many users may be benevolent, but some may be unreliable, and a few may be malicious . Additionally, all users likely have different levels of expertise; thus their contributions should not all be handled the same way.

To address this problem, we associate a reliability random variable, $r_u \in [0, 1]$, with each user u, and a single reliability variable for the population of users as a whole $\alpha \in [0, 1]$. In our setup, higher values of reliability variables correspond to more reliable users. We can learn the values of the reliability variables jointly using expectation-maximization (EM). Initially, each user's reliability variable is set to 0.5. In the *M-step*, MCMC performs MAP inference to estimate the most likely possible world given the evidence (including estimated user reliabilities). In the *E-step*, the reliability variable for each user is set to the percentage of their edit proposals that have been accepted by the model.

In order to incorporate reliabilities into the *M-step*, we modify our edit-specific factors to include the value of the user reliability variables as additional input. In particular, we incorporate the reliabilities into the SL, SNL, and attribute edit factors as follows. If s is the log-score of an edit-factor pertaining to an edit e_u submitted by user u, then the reliability-augmented version of the factor returns the log-score: $s\,(r_u/\alpha)^{\sigma}$, where r_u is the reliability of user u, α is the average reliability of the community, and $\sigma = 1$ is a parameter that controls how sensitive the model is to differences in reliability between users. This factor adjusts an individual user reliability by the average user reliability, ensuring that the model assigns extra weight ($r_u/\alpha > 1$) to evidence supplied by users with above average reliability, and reduced weight ($r_i/\alpha < 1$) to evidence of below average users.

4. Experiments

We evaluate our epistemological approach on the task of integrating edits to author attributes and coreference in automatically constructed KBs of computer science authors. Our study includes *should-link*, *should-not-link*, and *attribute-value* edits, and includes both *corrective* and *corruptive* edits. A corrective edit is one that if adopted would improve the quality of the KB, and a corruptive edit is one that if adopted would degrade the quality of the KB. In this section, we describe the data and experimental procedure.

4.1. Human edit integration strategies

We implement and test the following approaches for applying human edits to probabilistic KBs.

- **epistemological:** the approach advocated in this paper; human edits are treated as statistical evidence in support of a set of possible worlds.
- **epistemological w/ reliabilities:** as above but also reasons using user reliabilities.
- **overwrite** : in this strategy, the KB adopts *all* human edits (mimicking Wikipedia's model). An SL edit merges two entities; an SNL edit splits an entity into two disjoint subentities. When multiple edits express conflicting truth values to the KB, the more recent edit overwrites the older edit.
- **maximally-satisfy** : attempts to satisfy as many edits as possible through application of the transitivity rule. In the case where edits are all of the same type (SL/SNL) this strategy can optimally satisfy all edits together.

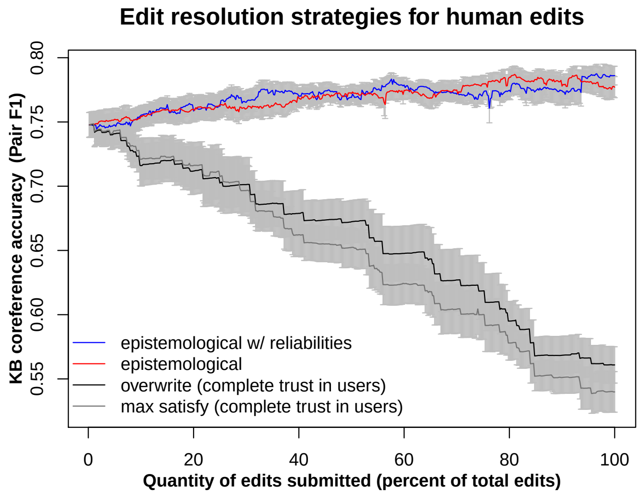


Figure 2: Epistemological integration of coreference edit (should-link, should-not-link, corruptive, corrective) with user reliabilities.

4.2. Experimental data and design

We use the REXA author disambiguation dataset for our experiments (Culotta et al., 2007a). This dataset contains 1,459 automatically extracted paper citations and 4,370 automatically extracted author mentions (1,459 of which are manually labeled with author coreference ground-truth). Each of the 1,459 labeled mentions belongs to one of eight common first-initial-last-name combinations: *D. Allen, A. Blum, S. Jones, L. Lee, J. McGuire, A. Moore, H. Robinson, S. Young.* Using this dataset we infer ten author coreference KBs. Each KB is initialized by running one million steps of MCMC inference in our hierarchical model (see Table **??**) on the entire dataset without any edits; the KBs differ due to randomness in the inference procedure. The average accuracy of these KBs, measured using pairwise F1 is 74.8%. The KBs contain an average of 227.4 predicted entities each (the ground truth contains 280 entities). For each KB, we then apply a set of human edits, run further inference, and report coreference accuracy averaged over the ten KBs.

4.3. Generating SL/SNL edits

Currently, we do not have access to a large epistemological KB with a community of contributing users. Therefore we exploit the manually annotated coreference data to experimentally generate human edits .

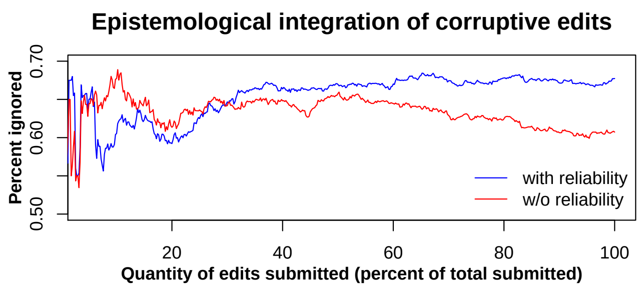
Determining whether an edit is corruptive or corrective can be ambiguous. We aim to avoid ambiguous cases by focusing on edits that are more obviously corrective or corruptive. Therefore, we have developed a set of heuristics for generating edits that we believe is both realistic and relatively unambiguous. First, when generating SL edits, we only consider merging pairs of predicted entities/sub-entities in the KB where (1) each entity/subentity tree has at least three mentions and (2) 80% of the mentions in each entity/subentity tree refer to the same real-world entity. Note that even if an entity does not satisfy this 80% rule, some of its subentities likely will. If merging a pair increases F1 accuracy we classify the edit as corrective (otherwise, corruptive). We generate SNL edits analogously: for each entity tree, we consider introducing an SNL edit that removes a subentity. If splitting the entity in this way increases F1 then we classify the SNL edit as corrective (otherwise, corruptive).

5. Experimental Results

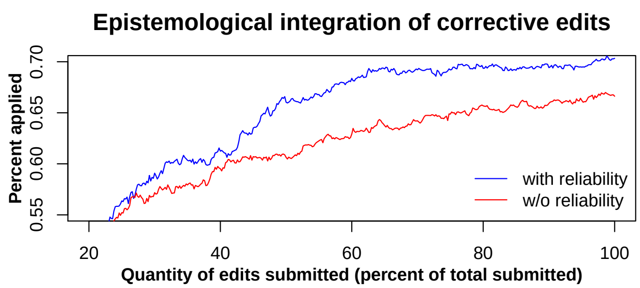
Our experiments evaluate the four edit integration strategies . We report the average accuracy of each strategy as a function of the number of user edits.

5.1. User reliability and edits integration

In Figure 2, we show the ability of the four systems to manage corrective and corruptive SL and SNL edits. The number of total edits ranges from 400-700, 30-50% of which are corruptive. This experiment consists of ten users, five of which are malicious and five of which are benevolent. Corruptive edits are randomly assigned to malicious users, and corrective edits are randomly assigned to benevolent users.



(a) Corruptive edits correctly ignored



(b) Corrective edits correctly applied

Figure 3: The user reliabilities improve epistemological integration of corrective edits (applies a higher percent) and ignore corruptive edits (ignores a higher percent).

Even though a large percent of the edits are corruptive, both epistemological integration systems (with and without reliability estimation) are able to increase the F1 accuracy of the original KBs from 74.8% to 78.6% and 77.7% respectively. In contrast, the overwrite and max-satisfy systems quickly succumb to the corruptive edits, resulting in a final KB accuracy of 56.0% and 54.0% respectively. Encouragingly, the reliability system is also able to correctly classify a user as malicious or benevolent with 97% accuracy.

In order to achieve a deeper understanding of the epistemological systems , we plot their ability to correctly integrate both corrective and corruptive edits into the KBs (Figure 3). These curves reveal interesting behavior of the systems. First, notice that corrective edits have an accumulative effect (Figure 3b): as more corrective edits arrive, the two systems are able to apply a larger percentage of them. This is expected because as edits continue to arrive, they begin to provide converging evidence for the user-asserted KB values (e.g., there are multiple edits in support of a particular change). Similarly, corruptive edits also produce an accumulative effect in the epistemological system (Figure 3a). Indeed, the accuracy of the sans-reliability

System	accuracy	% vandalized
Initial KB	64.5%	50%
Final KB (Overwrite)	78.2	23.1
Final KB (Epi)	81.1	9.10
Final KB (Epi w/ rel.)	**85.4**	**4.03**

Table 2: Integration of entity-attribute edits.

system steadily declines as corruptive edits continue to accumulate. Fortunately, the inclusion of user reliabilities successfully overcomes the negative accumulate effect of corruptive edits: as more corruptive edits arrive the KB can more confidently identify malicious users and reject more of their edits.

5.2. Should-link edit integration results

The epistemological system is substantially more robust to corruptive edits than the systems that place complete trust in users (overwrite and max satisfy). Indeed, the probabilistic approach considers multiple sources of evidence and is able to ignore 94.5% of these corruptive SL edits. Furthermore, the probabilistic system is better at applying the corrective SL edits than the two systems which place complete trust in the users. At first, it may seem surprising that the probabilistic system, would yield a better quality KB than the maximally-satisfy system (which applies *every* corrective SL edit). However, this improvement makes sense because the correction of even a small number of entities can trigger a cascading effect in probabilistic inference: as inference applies edits, the quality of the entities improve (bags of words have more context), and this in turn allows inference to infer further edits to coreference (beyond what is provided by the users).

5.3. Attribute edits

Finally, we study the ability of the systems to integrate user edits to entity attributes. In this experiment, we focus on edits to the first and middle name of authors. We define a strict notion of correctness for these attributes derived from the ground-truth coreference labels: only the most canonical form of the entity's name is acceptable as a first or middle name attribute (unless the complete form of the name does not occur in the data). For example, "Fernando" is considered a correct canonical first name, but "F." is considered incorrect (even though it is not *completely* incorrect); we only consider "F." to be a correct first name attribute, if the author *never* uses their full name in the entire dataset. In these experiments, we generate three types of edits: edits that are completely correct (e.g., "Fernando"), edits that are partially correct (e.g., "F."), and obvious vandalism edits (completely changes an author's name). We apply these edits to entities that have three or more mentions, 80% of which must have the same ground truth label. We generate a correct and partially correct edit for each of these entities and a malicious edit for 50% of these entities.

After applying the edits in a random order to the ten KBs, we record the average attribute accuracy, and the average amount of vandalism in Table 2. The three systems (resp. overwrite, epistemological with and without reliabilities) all improve the quality of the entity attributes both by reducing error in the initial KB (resp. 38.6%, 46.8%, and 58.9%), and by eradicating vandalism. The epistemological KB with reliabilities performs best, with highest error reduction (58.9%) and most resistance to vandalism (reducing the number of vandalized entities from 50% to 4.03%).

6. Related work

The problem of integrating user feedback with preexisting data for populating KBs has recently been recognized as a key challenge (Doan et al., 2009). However, research in this area is still in its infancy, and most current systems still adopt Wikipedia's integration model. For example, some approaches allow users to directly modify the content of information extraction (IE)(Chai et al., 2009), and others use IE to propose edits to manually curated KBs (Hoffmann et al.). The requirement that the KB curators accept or reject each proposal limits the scalability of the latter. Recent initiatives such as the TREC Knowledge Base Acceleration (KBA) task—keeping KBs such as Wikipedia up to date with large streams of real-world events—continue to foster research on this important problem (Frank et al., 2012).

There has also been interest in employing collective intelligence for reasoning about user-generated content and reliabilities (Whitehill et al., 2009; Bachrach et al., 2012). Techniques from collective intelligence have been applied to the problem of gathering user feedback to confirm or refute statements in RDF triple stores (Kasneci et al.; 2011). However, these models burden the users because they lack a data-integration component. In contrast, we combine the complementary strengths (accuracy and scalability respectively) of manual and automated approaches to KB construction by jointly reasoning about both the user-feedback and the data-integration processes (e.g., coreference).

Work concerning the task of identifying trustworthy users has become prevalent in the communities working with Amazon Mechanical Turk (AMT). Many AMT systems must also consider their users' reliabilities and often estimate them using learning algorithms like EM (Ipeirotis et al., 2010)—a method for reliability estimation that predates AMT (Dawid & Skene, 1979). In much of the AMT work, the user input is a multiple-choice selection and no attempt to integrate user inputs with preexisting KBs is made.

7. Conclusion

In this paper we presented an epistemological database framework for probabilistically managing user edits to preexisting KB content and automated data integration. We demonstrated the applicability of our approach on KBs of author entities by comparing it with multiple baselines. We found that our system is not only more robust to corruptive and malicious edits, but is also able to infer additional corrective changes to the KB from the user edits. In a real system, this would translate to more efficient KB curation. We further showed that our system was able to accurately estimate user reliabilities while inferring KB content.

References

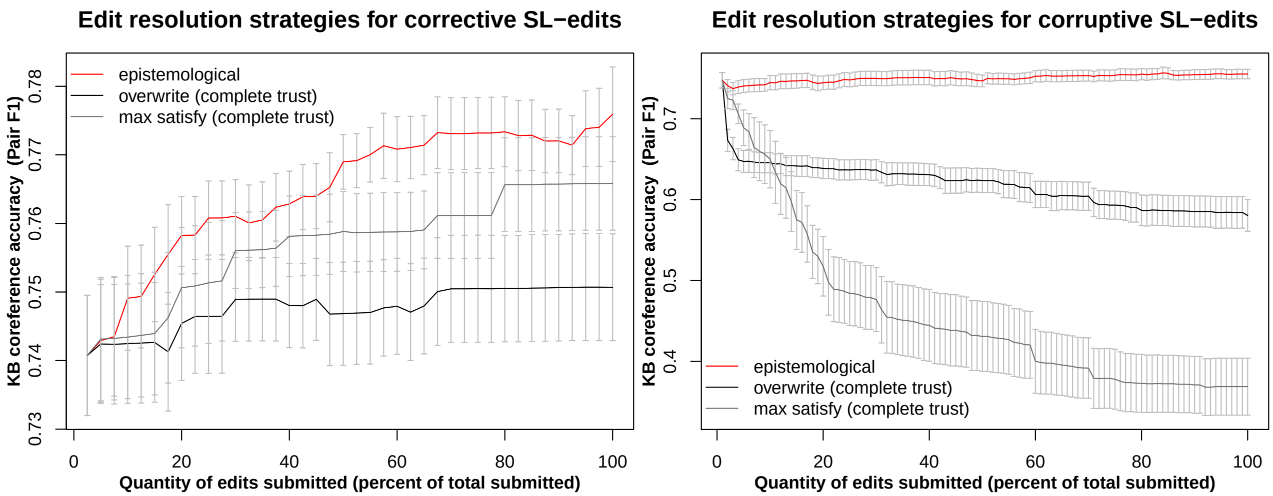
Bachrach, Yoram, Minka, Tom, Guiver, John, and Graepel, Thore. How to grade a test without knowing the answers - a Bayesian graphical model for adaptive crowdsourcing and aptitude testing. In *International Conference on Machine Learning*, Edinburgh, Scotland, UK, 2012.

Chai, Xiaoyong, Vuong, Ba-Quy, Doan, AnHai, and Naughton, Jeffrey F. Efficiently incorporating user feedback into information extraction and integration programs. In *SIGMOD Conference*, pp. 87–100, 2009.

Culotta, Aron, Kanani, Pallika, Hall, Robert, Wick, Michael, and McCallum, Andrew. Author disambiguation using error-driven machine learning with a ranking loss function. In *IIWeb WS*, Vancouver, Canada, 2007a.

Culotta, Aron, Wick, Michael, Hall, Robert, Marzilli, Matthew, and McCallum, Andrew. Canonicalization of database records using adaptive similarity measures. In *SIGKDD*, San Jose, CA, 2007b.

Dawid, A. P. and Skene, A. M. Maximum likelihood estimation of observer error-rates using the EM algorithm. In *Journal of the Royal Statistical Society.*, volume 28 of *Series C (Applied Statistics*, pp. 20–28. Wiley for the Royal Statistical Society, 1979. ISBN 978-1-4503-0222-7.



(a) Good quality human edits

(b) Poor quality human edits

Doan, AnHai and Halevy, Alon Y. Semantic integration research in the database community: a brief survey. *AI Magazine*, 26(1):83–94, 2005.

Doan, AnHai, Naughton, Jeffrey F., Ramakrishnan, Raghu, Baid, Akanksha, Chai, Xiaoyong, Chen, Fei, Chen, Ting, Chu, Eric, DeRose, Pedro, Gao, Byron, Gokhale, Chaitanya, Huang, Jiansheng, Shen, Warren, and Vuong, Ba-Quy. Information extraction challenges in managing unstructured data. *SIGMOD*, 37:14–20, March 2009. ISSN 0163-5808.

Frank, John R., Kleiman-Weiner, Max, Roberts, Danial A., Niu, Feng, Zhang, Ce, and Re, Christopher. Building an entity-centric stream filtering test collection for TREC 2012. In *TREK*, 2012.

Hoffmann, Raphael, Amershi, Saleema, Patel, Kayur, Wu, Fei, Fogarty, James, and Weld, Daniel S. Amplifying community content creation with mixed initiative information extraction. In *Proc. SIGCHI*.

Ipeirotis, Panagiotis G., Provost, Foster, and Wang, Jing. Quality management on Amazon Mechanical Turk. In *Proceedings of the ACM SIGKDD Workshop on Human Computation*, HCOMP '10, pp. 64–67, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0222-7.

Kasneci, Gjergji, Van Gael, Jurgen, Herbrich, Ralf, and Graepel, Thore. Bayesian knowledge corroboration with logical rules and user feedback. In *Proc ECML PKDD 2010*.

Kasneci, Gjergji, Gael, Jurgen Van, Stern, David H., and Graepel, Thore. CoBayes: bayesian knowledge corroboration with assessors of unknown areas of expertise. In *WSDM*, pp. 465–474, 2011.

McCallum, A. and Wellner, B. Toward conditional models of identity uncertainty with application to proper noun coreference. In *IJCAI IIWeb WS*, 2003.

Singh, Sameer, Subramanya, Amarnag, Pereira, Fernando C. N., and McCallum, Andrew. Large-scale cross-document coreference using distributed inference and hierarchical models. In *ACL*, pp. 793–803, 2011.

Whitehill, Jacob, Ruvolo, Paul, Wu, Tingfan, Bergsma, Jacob, and Movellan, Javier R. Whose vote should count more: optimal integration of labels from labelers of unknown expertise. In *NIPS*, 2009.

Wick, Michael. Epistemological databases for pobabilistic knowledge base construction. Ph.D. thesis proposal, University of Massachusetts, 2012.

Wick, Michael, Singh, Sameer, and McCallum, Andrew. A discriminative hierarchical model for fast coreference at large scale. In *Proc. ACL*, 2012.