Stop the Truthiness and Just Be Wrong

Oliver Kennedy
Based on work with Ying Yang, Niccolò Meneghetti, Poonam Kumari, Will Spoth, Aaron Huber, Arindam Nandi, Vinayak Karuppasamy, Dieter Gawlick, Zhen Hua-Liu, Boris Glavic, Ronny Fehling, and Lisa Lu
University at Buffalo
okennedy@buffalo.edu

Since their earliest days, databases have held themselves to a strict invariant: Never give the user a wrong answer. So ingrained into psyche of database theoreticians, researchers, and DBAs is this invariant, that those few attempts to violate it have included cumbersome data models [3, 7, 8], huge warning signs [4], or obnoxious language constructs that break classical SQL [1, 5].

Sadly, by focusing on the quality of the database itself, database systems fail to acknowledge that the it is the data that is rarely precise, correct, valid, or unambiguous. The emphasis on certain, deterministic data forces the use of complex, hard-to-manage, slow-to-create extract-transform-load (ETL) pipelines that emit deceptively certain, "truthy" data rather than acknowledging ambiguity or error. Database systems have always had a key role in automating complex data-driven decision processes. However, as more decisions are automated, even small errors in source data can have drastic impacts on peoples lives, from denying a person credit[1], to deciding that an 8-year old is a terrorist[2].

User-facing software companies have surpassed databases in this respect. For example, Apple applications[3] use facts data-mined from email to automatically populate databases in their contacts and calendar applications. Figure 1 illustrates how two applications presents extracted information to the user. Three features are prominent: First, these facts are explicitly kept distinct, clearly marked as being guesses and thus uncertain. Second, the interface includes intuitive provenance mechanisms that help the user understand the extraction system's intent and puts the extracted information in context. Finally, the interface includes overt solicitations for feedback to help the user quickly correct unwanted extractions or mark the extracted data as valid.

The illusion of accuracy in database query results can no longer be maintained. Database systems must learn how to acknowledge errors in source data, and how to use this

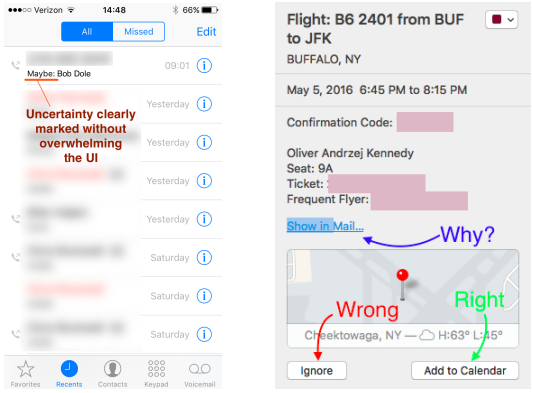

Figure 1: **Extraction in Calendar.app and iOS Phone**

information to *effectively* communicate ambiguity to users. Moreover, this needs to happen without overwhelming users, breaking the decades-old abstractions that people understand and use on a day-to-day basis in their workflows, or requiring a statistics background from all users. In short, we need tooling [2, 9] and interfaces [6] that fit into people's existing workflows and allow databases to be less truthy and more wrong.

[1] http://money.cnn.com/2016/04/11/pf/john-oliver-credit-reports/index.html
[2] http://www.nytimes.com/2010/01/14/nyregion/14watchlist.html?_r=0
[3] Apple is used here purely as an example; Google and Microsoft software has similar functionality.

1. REFERENCES

[1] L. Antova, C. Koch, and D. Olteanu. $10^{(10^6)}$ worlds and beyond: Efficient representation and processing of incomplete information. *VLDBJ*, 18(5):1021–1040, 2009.
[2] X. Chu, J. Morcos, I. F. Ilyas, M. Ouzzani, P. Papotti, N. Tang, and Y. Ye. Katara: Reliable data cleaning with knowledge bases and crowdsourcing. *pVLDB*, 8(12):1952–1955, Aug. 2015.
[3] A. Deshpande and S. Madden. Mauvedb: Supporting model-based user views in database systems. In *SIGMOD*, 2006.
[4] J. M. Hellerstein, P. J. Haas, and H. J. Wang. Online aggregation. In *SIGMOD*, 1997.
[5] R. Jampani, F. Xu, M. Wu, L. L. Perez, C. Jermaine, and P. J. Haas. Mcdb: A monte carlo approach to managing uncertain data. In *SIGMOD*, 2008.
[6] P. Kumari, S. Achmiz, and O. Kennedy. Communicating data quality in on-demand curation. In *QDB*, 2016.
[7] A. Thiagarajan and S. Madden. Querying continuous functions in a database system. In *SIGMOD*, 2008.
[8] D. Z. Wang, E. Michelakis, M. Garofalakis, and J. M. Hellerstein. Bayesstore: Managing large, uncertain data repositories with probabilistic graphical models. *PVLDB*, 1(1):340–351, 2008.
[9] Y. Yang, N. Meneghetti, R. Fehling, Z. H. Liu, and O. Kennedy. Lenses: An on-demand approach to ETL. *VLDB*, 8(12):1578–1589, 2015.