SWEEP: A Database for Automated Segmentation With Economical Editing by People

Sean Goldberg
University of Florida
sean@cise.ufl.edu

Jeff Depree
University of Florida
jdepree@cise.ufl.edu

Daisy Zhe Wang
University of Florida
daisyw@cise.ufl.edu

Abhiram Jagarlapudi
University of Florida
abhiram@cise.ufl.edu

Sahil Puri
University of Florida
sahil@cise.ufl.edu

ABSTRACT

Temporary, will be rewritten.–SLG The need to automatically process and store large amounts of uncertain and imprecise machine learned data has necessitated the use of Probabilistic Databases (PDBs) which maintain and allow queries that carry a degree of uncertainty. An integral part of the data cleaning process is finding efficient ways to reduce this uncertainty. In this paper we propose the Crowd Assisted Machine Learning (CAMeL) paradigm which seeks to utilize crowdsourcing techniques such as Amazon Mechanical Turk to optimally improve the accuracy of learned data. This paradigm is implemented on top of a probabilistic database which we call CAMeL-DB. A subset of the automatically populated fields are converted into questions to be answered by the crowd. We demonstrate the efficacy of our approach on an information extraction problem consisting of automated segmentation of bibliographic citations, showing that a relatively small subset of questions can lead to a large boost in accuracy.

1 Introduction

Motivate need for automatic information extraction

Discuss reasons for wanting to automatically segment citations

Introduce system

Briefly describe dataflow from extraction to selection and integration

2 Related Work

Discuss previous attempts at citation IE (esp. by McCallum)

Identify various methods of data cleaning/integration

Compare selection process to that used in active learning

Introduce crowdsourcing and related quality control efforts

3 System Overview

Graphic showing all system components

Extraction (CRF)

Storage

Selection

Crowd

Integration

4 Extraction

More in depth description of CRF compared to related work

Briefly describe Viterbi for best path and Forward-Backward for marginals

CRF graphic of example citation mapped onto CRF

5 Storage

Presentation of schema

Discussion of how underlying probabilities are stored and manipulated

Screenshot of system interface

6 Selection

Process for selecting a token from each citation

Clustering of tokens by various metrics (same token neighborhood, same label neighborhood, etc)

Ranking of clusters by various metrics (high entropy, size of cluster)

7 Crowdsourcing

Mapping of top clusters to questions via XML

Describe specific task Turkers required to do

Screenshot of AMT example question

8 Integration

Assessing Turker quality via EM

Mapping Turkers to probability distributions

Combining distributions using Dawid-Skene

Integration into DB (Clamped Viterbi for new best path)

9 Experiments

Compare four different clustering algorithms.
t 0,l 0,t-1,t+1 Retained: 5221 Skipped: 1778 Errors: 1
t 0,l 0,l-1,l+1 Retained: 3224 Skipped: 3775 Errors: 67
E 0,L 0,L-1,L+1 Retained: 6197 Skipped: 802 Errors: 0
t 0,l 0,t-1,l-1,t+1,l+1 Retained: 5236 Skipped: 1763 Errors: 0

Selection: Compare accuracy before and after replacing tokens with ground truth.

Selection: Compare accuracy before and after re-running inference.

Selection: Measure accuracy vs. clusters asked to find where diminishing returns occur.

Integration: Compare DS vs. MV for 3, 5, 7, and 9 Turkers per question.

Integration: Find appropriate entropy threshold to use as decision boundary.

Integration: Compare accuracy before, after w/o entropy threshold, and after applying entropy. threshold.

10 Conclusion

11 Future Work

12 References