


Winter 2015

Epistemological Databases for Probabilistic Knowledge Base Construction

Michael Louis Wick
Computer Sciences

Follow this and additional works at: https://scholarworks.umass.edu/dissertations_2

 Part of the [Artificial Intelligence and Robotics Commons](#), [Databases and Information Systems Commons](#), [Numerical Analysis and Scientific Computing Commons](#), and the [Other Computer Sciences Commons](#)

Recommended Citation

Wick, Michael Louis, "Epistemological Databases for Probabilistic Knowledge Base Construction" (2015). *Doctoral Dissertations*. 334.
https://scholarworks.umass.edu/dissertations_2/334

This Open Access Dissertation is brought to you for free and open access by the Dissertations and Theses at ScholarWorks@UMass Amherst. It has been accepted for inclusion in Doctoral Dissertations by an authorized administrator of ScholarWorks@UMass Amherst. For more information, please contact scholarworks@library.umass.edu.

**EPISTEMOLOGICAL DATABASES FOR
PROBABILISTIC KNOWLEDGE BASE CONSTRUCTION**

A Dissertation Presented

by

MICHAEL LOUIS WICK

Submitted to the Graduate School of the
University of Massachusetts Amherst in partial fulfillment
of the requirements for the degree of

DOCTOR OF PHILOSOPHY

February, 2015

Computer Science

© Copyright by Michael Louis Wick 2015

All Rights Reserved

EPISTEMOLOGICAL DATABASES FOR PROBABILISTIC KNOWLEDGE BASE CONSTRUCTION

A Dissertation Presented

by

MICHAEL LOUIS WICK

Approved as to style and content by:

Andrew McCallum, Chair

Ben Marlin, Member

Gerome Miklau, Member

Jon Machta, Member

Tom Mitchell, Member

Lori A. Clarke, Department Chair
Computer Science

DEDICATION

To family.

ACKNOWLEDGMENTS

I am extremely fortunate to have received such a superb education here at UMass. My experiences during this time have truly transformed me into an accomplished researcher from my humble beginnings as a fledgling, but enthusiastic undergraduate. This transformation could not have occurred without my mentors, friends, family, and colleagues.

I am deeply grateful to my advisor, Andrew McCallum, who has been a shepherd for this work. Through his ideas and guidance, Andrew has had a tremendous impact on my research; through his seemingly endless supply of energy, passion and drive, he has always kept me motivated—even through the toughest failures and rejections. I am also grateful to my committee members Ben Marlin, Gerome Miklau, Jon Machta, and Tom Mitchell, whose feedback and ideas have helped improve, develop and mature this work.

I owe much of my graduate school success to the nurturing research environment of our lab where I was surrounded by wonderful people, many of whom will remain life long friends, mentors, and collaborators. While the composition of the lab has changed over the years, the delightful brand of spontaneous research discussion, serendipitous interactions, “let’s take it to the white-board” arguments, and culture of collaboration has always been a constant. Aron Culotta, Khashayar Rohamanesh and Gideon Mann were my early mentors, and I’d like to thank them along with other senior members of the lab including Charles Sutton, Kedare Bellare, Pallika Kanani, Chris Pal, David Mimno, Sebastian Reidel, Laura Dietz, Hanna Wallach, and Andres Corrada-Emmanuel. I will continue to share their wisdom with those whom I have the privilege of mentoring. I am also thankful for interactions and fruitful collaborations with my peers and lab-mates, including Sameer Singh, Karl Schultz, Rob Hall, Anton Bakalov, Limin Yao, Greg Druck, Ari Kobren, Harshal Pandya, Adam Saunders, Alexandre Passos, Jack Sullivan, David Bellanger, Sam Anzaroot, Luke

Vilnis, Tim Vieira, Brian Martin, David Soergel, Jinho Choi. We have shared many experiences including the agony of a Somoa-time deadline, the sting of a rejected paper, the uncertainty of a new research direction, the anticipation of an ongoing experiment, the thrill of acquiring a new result, and the reward of sharing the result with other enthusiastic colleagues at conferences. I'm also grateful for my external collaborators including Erik Learned Miller, Michael Ross, Daisy Zhe Wang, Christan Grant. Thank you to Dan Parker, Kate Maruzzi, Leeanne Leclerc, and Glen Stowell who have assisted me technically and administratively, and have bailed me out more times than I care to remember.

Finally, I would like to thank my friends and family. I am fortunate to have a family that values education, but emphasizes creativity. My parents have always nurtured my intellectual curiosity and have encouraged me academically. Their enthusiasm for life-long learning has been infectious. My wife, Farahnaz, has been a constant source of inspiration and support throughout graduate school and beyond. Her focus, intellect and entrepreneurialism have set a lofty standard for which I strive.

ABSTRACT

EPISTEMOLOGICAL DATABASES FOR PROBABILISTIC KNOWLEDGE BASE CONSTRUCTION

FEBRUARY, 2015

MICHAEL LOUIS WICK

B.Sc., UNIVERSITY OF MASSACHUSETTS

M.Sc., UNIVERSITY OF MASSACHUSETTS

Ph.D., UNIVERSITY OF MASSACHUSETTS AMHERST

Directed by: Professor Andrew McCallum

Knowledge bases (KB) facilitate real world decision making by providing access to structured relational information that enables pattern discovery and semantic queries. Although there is a large amount of data available for populating a KB; the data must first be gathered and assembled. Traditionally, this integration is performed automatically by storing the output of an information extraction pipeline directly into a database as if this prediction were the “truth.” However, the resulting KB is often not reliable because (a) errors accumulate in the integration pipeline, and (b) they persist in the KB even after new information arrives that could rectify these errors.

We envision a paradigm-shift in KB construction for addressing these concerns that we term an “epistemological” database. In epistemological databases the existence and properties of entities are not directly input into the DB; they are instead determined by

inference on raw evidence input into the DB. This shift in thinking is important because it allows inference to revisit previous conclusions and retroactively correct errors as new evidence arrives. Evidence is abundant and in steady supply from web spiders, semantic web ontologies, external databases, and even groups of enthusiastic human editors. As this evidence continues to accumulate and inference continues to run in the background, the quality of the knowledge base continues to improve. In this dissertation we develop the machine learning components necessary to achieve epistemological knowledge base construction at scale with key contributions in modeling, inference and learning.

TABLE OF CONTENTS

	Page
ACKNOWLEDGMENTS	v
ABSTRACT	vii
LIST OF TABLES	xiii
LIST OF FIGURES	xv
 CHAPTER	
1. INTRODUCTION	1
1.1 Summary of contributions	5
1.2 Declaration of previous work and collaboration	8
2. BACKGROUND	9
2.1 Graphical models and factor graphs	9
2.1.1 Inference	11
2.1.1.1 MAP inference	11
2.1.1.2 Marginal inference and statistical queries	11
2.1.1.3 Markov chain Monte Carlo	12
2.1.1.4 Metropolis Hastings	13
2.1.2 Remarks on MCMC for MAP inference	14
2.1.3 Parameter estimation	15
2.2 Coreference	16
2.2.1 Pairwise models for coreference	17
2.2.2 MCMC inference for coreference	18
2.2.3 Entity-wise models	19
2.2.4 Entity linking	20
2.2.5 Evaluation	20

3. HIERARCHICAL COREFERENCE	22
3.1 Overview	22
3.2 Related work	25
3.3 Hierarchical models of coreference	27
3.4 Model details	29
3.4.1 Representing entities, sub-entities, and mentions	29
3.4.2 Model factors	31
3.4.3 Capturing context with bags of words	32
3.5 MCMC for hierarchical coreference	33
3.5.1 Proposal distribution	34
3.5.2 Efficient proposal evaluations	36
3.5.3 Learning	37
3.6 Applications	39
3.6.1 Author coreference	39
3.6.2 Entity-linking and wikification	40
3.7 Experiments	45
3.7.1 Data	45
3.7.2 Scalability	46
3.7.3 Studies on the role of the hierarchy	49
3.8 Conclusion and future work	53
4. INFERENCE WITH PRIORITY-DRIVEN MCMC	58
4.1 Overview	58
4.2 Related work	59
4.3 Query-aware MCMC	60
4.3.1 Prioritization via influence	62
4.3.2 Convergence rates	63
4.4 Experiments	68
4.4.1 Synthetic experiments	68
4.4.2 Real-world data	71
4.5 Conclusion	72

5. PARAMETER ESTIMATION WITH SAMPLERANK	76
5.1 Overview and related work	76
5.2 SampleRank	79
5.2.1 Pairwise Objectives	81
5.2.2 Optimization	82
5.2.3 SampleRank Algorithm	83
5.2.4 Sample Complexity	84
5.2.5 Implementation Details for Efficient Learning	85
5.2.5.1 Exploit Locality of MCMC	86
5.2.5.2 Efficient Parameter Averaging	87
5.2.5.3 Efficient Implementations of L2 Regularization	88
5.2.6 Extensions to SampleRank	89
5.2.6.1 SampleRank for SVM	89
5.2.6.2 SampleRank with delayed reward	90
5.2.7 Experiments	91
5.2.7.1 Noun-phrase Coreference	92
5.2.7.2 Multi-Label Classification	93
5.2.7.3 Named Entity Recognition	94
5.2.7.4 Hierarchical coreference	95
5.3 Training Restricted Boltzmann Machines with SampleRank	98
5.3.1 SampleRank for RBMs	101
5.3.2 RBM Experiments	105
5.3.3 SampleRank for RBM Conclusion	109
5.4 Conclusion	111
6. EPISTEMOLOGICAL DBS AND APPLICATIONS	116
6.1 Epistemological databases	116
6.1.1 Database and Model Components	118
6.1.2 Truth Discovery Component	119
6.1.3 Prioritized inference for incremental KB construction	121
6.1.4 Prioritized MCMC for Coreference	123
6.2 Bibliometrics: constructing a database of all scientists in the world	125
6.2.1 Author mention-finding	126

6.2.2	Author coreference	127
6.2.3	Joint Mention-Finding and Coreference	128
6.3	EpiDB Experiments (bibliographic)	128
6.3.1	Datasets	129
6.3.2	KB management systems and models	129
6.3.3	Experimental design	130
6.3.4	Incremental KBC with Epistemological DBs	131
6.3.5	Prioritized inference	136
6.3.6	Joint inference for multi-modal tasks	137
6.4	Text and Entity Linking Experiments	139
6.4.1	Data	140
6.4.2	Systems and baselines	142
6.4.3	Results	143
6.5	Human machine cooperation	145
6.5.1	Overview	145
6.5.2	Related work	146
6.5.3	Human edits to coreference predictions	147
6.5.3.1	Experiments on author coreference	148
6.5.3.2	Attribute edits	151
6.6	Conclusion	152
7.	CONCLUSION	156
7.1	Summary of contributions	156
7.2	Potential Impact	157
7.3	Limitations, Discussion, and Future Work	158
	BIBLIOGRAPHY	164

LIST OF TABLES

Table	Page	
3.1	Factor definitions for 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.	33
3.2	The comprehensive set of factor templates for our hierarchical author coreference model. See Table 3.1 for generic factor functions.	41
3.3	REXA dataset.	46
3.4	Large-scale hierarchical coreference runs.	46
3.5	The effect of structuring subentities on coreference accuracy.	53
3.6	Greedy vs. non-greedy inference on Rexa author coref (one-million samples). Aggressiveness (how aggressively the model wishes to merge mentions into entities) is varied across rows (0 is the cross-validated value; rows highlighted in gray are aggression values for which the model performs well). For all but one value in this range, greedy inference is more unstable than non-greedy inference (measured by relative standard error).	54
5.1	A comparison of SampleRank with other MCMC learning algorithms on an entity-wise model of coreference.	95
5.2	Labeling error rates of various algorithms on multi-label classification data sets averaged over 10 random runs. Results in bold indicate significant error reduction ($p = 0.05$).	96
5.3	SampleRank and exact parameter estimation algorithms on CoNLL NER.	96
5.4	SampleRank training of the hierarchical coreference model on the BoNY subset of Wikilinks.	98
5.5	Weights on basic features.	98

5.6	Run-time statistics for the SampleRank-data-only algorithm. Δ -hidden is the average number of hidden units that differ between the data and the sample during the course of the run, Δ -visible is analogously defined for the visible units, and Δ -total is their sum. %-updated is the percentage of Gibbs samples that lead to an update (some samples might not lead to an update if the ranking constraint is already satisfied). Error is the reconstruction error on the held out test data.	112
6.1	Evaluation of Linking and Discovery using pairwise F1 (PW F1) and linking accuracy.	143
6.2	Evaluating the ability of our system to discover entities, when the various pre-known (Wikipedia) entities are withheld (metric is pairwise F1).....	143
6.3	Evaluating the effect of additional mentions on the performance of coreference resolution (NY dataset).	144
6.4	Integration of entity-attribute edits.	151

LIST OF FIGURES

Figure	Page
2.1 Pairwise model on six mentions: Open circles are the binary coreference decision variables, shaded circles are the observed mentions, and the black boxes are the factors of the graphical model that encode the pairwise compatibility functions.	18
3.1 Discriminative hierarchical factor graph for coreference: Latent entity nodes (white boxes) summarize subtrees. Pairwise factors (black squares) measure compatibilities between child and parent nodes, avoiding quadratic blow-up. Corresponding decision variables (open circles) indicate whether one node is the child of another. Mentions (gray boxes) are leaves. Deciding whether to merge these two entities requires evaluating just a single factor (red square), corresponding to the new child-parent relationship.	28
3.2 A mergeUp(x_1, x_2) operation on two orthogonal vectors.	32
3.3 A mergeLeft(x_0, p) operation.	36
3.4 Joint linking and discovery example. As we cluster mentions into entities, it becomes more clear where they should link based on the proximity of their outer cluster (entity) to the Wikipedia mentions.	41
3.5 Coreference of 67 million author mentions from PubMed enables us to attribute papers to real-world authors. Shown here is the distribution over publication counts for PubMed authors (i.e., the frequency of an author with x publications) resulting from our inference.	47
3.6 Traditional KB (top) vs. epistemological KB (bottom).	48

3.7	A comparison of the infinitely deep hierarchical coreference model with a flat entity-based model (with and without block moves). These results confirm our analysis in which we conclude that a hierarchical structure is a modeling necessity in the absence pairwise factors. We vary the magnitude of an additional entity penalty factor and record the impact on precision, recall, and F1. The original setting of the entity penalty factor is 1 (determined on held-out data), and the point $x = 0$ (no additional penalty) corresponds to this parameter setting. Larger penalties encourages the model to predict fewer entities and obtain higher recall, hence the term “aggressiveness.” The F1 of these model are compared directly in Figure 3.8	56
3.8	A comparison of different coref models. The curves for the hierarchical and two-tiered are the average of three runs; error bars for these two models are the standard error.	57
4.1	Intuition for why query aware MCMC is likely to perform better than traditional MCMC: fast convergence to the incorrect distribution is often more desirable than slow convergence to the correct distribution. These hypothetical scenarios illustrate the importance of error and convergence rates of the two chains. The figure on the right demonstrates a scenario in which query-aware MCMC would not work well (strong statistical dependencies might give rise to such a case).	65
4.2	Graphical models for evaluating QAM	72
4.3	Convergence to the query marginals of the stationary distribution from an initial uniform distribution.	73
4.4	Convergence to the full joint stationary distribution from an initial uniform distribution.	73
4.5	Improvement over uniform p as the number of variables increases. Above the line $x = 0$ is an improvement in marginal convergence, and below is worse than the baseline. As number of variables increase, the improvements of the query specific techniques increase.	74
4.6	Query-aware sampling on two NER models. Linear-chain model (left) and skip-chain model (right).	75
5.1	Local optima in coreference examples (left: with ten mentions, right: with twenty mentions). In the left figure, small circles represent mentions and their color is their ground-truth entity label; big ovals represent hypothesized entities.	90

5.2	A factor graph representation of an RBM as a Markov random field with weights between the visible and hidden units, and bias weights on the visible and hidden units.	99
5.3	A comparison of different SampleRank variations for training RBMs.	108
5.4	RBM filters learned by SampleRank on MNIST digit recognition dataset. The filters are typical for this dataset: some filters represent full digits or composition of multiple digits, others represent edges. The R code for generating these images was adapted from Andrew Landgraf’s post: http://www.r-bloggers.com/restricted-boltzmann-machines-in-r/	113
5.5	Ranking constraints produced by SampleRank during the first four rounds of learning. Each row is a round of learning in which the sampler is initialized to the truth; the first five constraints are shown for each round. The “better” configuration is shown on the left (odd numbers) and the “worse configuration is shown on the right (even numbers). Each row is a new round of learning, initialized at the truth.	114
5.6	Rate at which the different RBM trainers reduce test-time reconstruction error.	115
6.1	Traditional KB (top) vs. epistemological KB (bottom).	117
6.2	The benefits of revisitation upon acquisition of new data. New data provides evidence for inference on old data.	131
6.3	Epistemological DB management of an incremental KB construction task. The author coreference accuracy of the original mentions in the KB is plotted as a function of time. Accuracy is recorded every 1000 samples, blue dots indicate a new batch of author mentions. Standard errors reported over three random runs (with different splits of the data).	133

6.4	Epistemological DB management of an incremental KB construction task vs a baseline approach that periodically re-runs inference from scratch (and runs greedy inference in between these restarts). The author coreference accuracy of all existing mentions in the KB is plotted as a function of time. Accuracy is recorded every 1000 samples, blue/red dots indicate a new batch of author mentions: blue dots show the accuracy for the EpiDB system and red dots show the accuracy for the baseline system. Red X's show accuracy of baseline DB after inference is redone from scratch (1M samples). Dashed relines show the accuracy of the KB if no more inference is done. Standard errors reported over three random runs (with different splits of the data).	134
6.5	The benefits of revisitation upon acquisition of new data. New data provides evidence for inference on old data.	137
6.6	New evidence allows multi-modal joint inference to correct citation extraction errors on old data via predictions on new data. Error reduction is 20%.	140
6.7	A recall coreference error (top), is corrected when a user edit arrives (bottom).	153
6.8	Epistemological integration of coreference edit (should-link, should-not-link, corruptive, corrective) with user reliabilities.	154
6.9	The user reliabilities improve epistemological integration of corrective edits (applies a higher percent) and ignore corruptive edits (ignores a higher percent).	154
6.10	A comparison of various approaches to applying human edits to KBs.	155

CHAPTER 1

INTRODUCTION

Informed decision making hinges on having prompt access to complete, reliable, and up-to-date information. Tantalizingly, for many decisions, there exists a vast quantity of relevant information—in structured databases, on the web, and in text documents—but this data is not directly useful because it first needs to be gathered, deduplicated and assembled. Indeed, much of the world’s knowledge is not available in a form that is immediately accessible to decision makers. This very observation is echoed in Kenneth Kosik’s essay *The Wikification of Knowledge* [46], in which he laments:

“It is perhaps an irony of our time that with all of these avenues to discover knowledge at our command, we can find ourselves starved for information in a sea churning with nothing but information.”

A key challenge in satiating our demand for information is *knowledge base (KB) construction*, the task of organizing data from different sources into a single cohesive representation where it can be more easily queried and utilized. KBs store and organize knowledge (e.g., entities and relations) using formalisms that readily supports semantic queries, data mining, and decision-making. Wikipedia is an example of a KB that has already revolutionized the way we gather and share information. Although Wikipedia contains some useful structure (hyper-links between pages of related entities, and info boxes), most of its information is still expressed in natural language text. In order to unlock the true potential of this knowledge, we require richer structure in the form of entities, their attributes, and relations between them.

Thus, there has been a tremendous interest in creating KBs like Wikipedia, but with richer ontologies and relational structures. Freebase [8] and Yago [94, 43] are two re-

cent KB initiatives that aim to provide such structure to Wikipedia. Much like Wikipedia, Freebase depends on the contributions of collaborative users; however, instead of writing articles, Freebase users identify entities, define relations between them, and map them into a target ontology. Yago adopts a more automated approach, employing high accuracy heuristics to automatically map between Wikipedia and the ontologies of WordNet [60] and GeoNames.¹ Freebase also exploits structured sources of information such as the movie database IMDB.² For example, both Wikipedia and IMDB contain information on many of the same movies and people, such as *James Cameron, the movie director*. In order to combine knowledge between James Cameron’s Wikipedia page and his IMDB page, a contributor must first make the inference that these two pages are both indeed discussing the same entity: James Cameron, the director (and not, for example, James Cameron, *the American Historian*). This fundamental problem, known as coreference, is foundational for KB construction because it allows us to combine information about a single entity that has been *mentioned* in multiple sources. For example, once we have coreference of James Cameron’s mentions, perhaps only an insignificant amount of additional effort is required to extract and combine the attributes in the Wikipedia info box (associated with his Wikipedia mention) with the relations accompanying his IMDB mention (e.g., directedBy(“The Terminator”, “James Cameron”). Additional attributes and relations can be derived from a combination of information provided by the different sources. Of course, coreference is a difficult problem to solve manually because there are many entities and mentions (e.g., the entity James Cameron might be mentioned millions of times on the web: in blogs, newswire articles, social media, and IMDB).

Although both Freebase and Yago have already achieved impressive scales (40 million entities and 10 million entities respectively), because of their dependence on crowdsourcing and crowd-sourced KBs (specifically, Wikipedia), they are limited in size to what

¹www.geonames.org

²www.imdb.com

can be achieved manually by altruistic contributors (in fact, keeping all but the most prominent entities in Wikipedia up to date is a challenge—the median lag time for updating an entity in Wikipedia is two years [29]). In contrast, consider the possibilities afforded by automated systems such as the never-ending language learner (NELL) [12]. Given a desired set of relations and examples of how these relations are expressed in natural language, NELL is able to read the web, find more examples of these target relations, learn from these additional examples, and improve its ability to extract examples of such relations in the future. NELL has the potential to read the entire web and map the knowledge into a target ontology of relations, types and attributes, with almost no manual effort by humans.³ In general, automated approaches such NELL, Yago, and others [23], have tremendous potential—the potential to initiate an information revolution similar to Wikipedia, but with richer structure, more data, and on a much grander scale.

However, automatically constructing a KB is a difficult problem, requiring automated solutions to a variety of extraction and integration tasks: we must identify mentions of entities in different sources (mention finding), determine which mentions actually refer to the same real-world entities (coreference), aggregate the information across the entity’s mentions to infer attributes and relations (attributes/relation extraction), and finally extract additional relations between entities that might not be immediately available from the mentions themselves. 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. Such uncertainty makes automation difficult, but probabilistic machine learning models have proven to be a promising solution for various individual integration tasks [48, 93, 42, 24]. Traditionally, these models are combined into an external information integration system that outputs inferred entities/relations and stores them in a database where

³Though, automated systems still benefit from human input, and NELL specifically is capable of incorporating human feedback.

they can be queried and browsed [30, 23, 99, 94, 12]. However, this is undesirable because when new augmenting and correcting data arrives later, past integration decisions need to be reconsidered.

Throughout the lifetime of a KB, new evidence will become available: new data rows, new field values in old rows, new relational linkages. Unfortunately, many current automated KB construction architectures—from Yago to NELL—are unable to efficiently use the new evidence to improve integration because they do not store the intermediate results of inference (and often must regenerate the KB from scratch to fully incorporate the new evidence). Yago’s data integration strategy is based on heuristics instead of probabilistic models making it difficult to reason about integration uncertainty and revisit past conclusions. NELL is capable of life-long learning in which it improves its ability to extract future relations from current extractions, but NELL does not store much of the intermediate variables and inference provenance necessary to efficiently reconsider and revise past extractions when new evidence arrives (in general this problem is non-trivial because incorrect relations might participate in “multi-hop” compound relations, have profound impact on coreference decisions, or influence canonicalization of entity attributes).

We envision a paradigm-shift for KB construction termed *epistemological databases*, in which the canonical “true” entities and relations in the database are always inferred from extracted/integrated or human-entered data, never injected directly from external systems. The KB stores and manages uncertainty about what it believes is the truth. As new evidence arrives, truth-discovering inference continues to run inside the database inferring values for missing attributes, responding to new evidence, revisiting past inference conclusions, and retroactively correcting errors. In comparison to NELL, epistemological DBs focus on improving the quality of the KB through never-ending inference on new evidence, rather than improving the future quality of the KB through never-ending learning on existing data.

Implementing such a KB construction system is a formidable challenge because its success depends heavily on having accurate probabilistic models for data integration and

efficient statistical algorithms that operate on these models at scale. The key aspects of these problems are model structure, inference, and learning. We have contributions in each of these areas. First, we propose a novel discriminative model for coreference resolution that achieves tremendous scalability by reasoning about entities as trees. Coreference is foundational because it is the key step in populating the KB with entities and even makes the extraction of many attributes and relations trivial (though certain types of relation extraction are more involved [74], and other tasks such as mention finding are also important [3], but not a focus of this work). Second, we employ MCMC for inference and propose a novel extension that is able to prioritize which variables to sample in order to efficiently answer statistical queries and integrate new evidence. Third, since we are using MCMC for inference, we propose a novel learning algorithm that efficiently estimates model parameters with MCMC. Finally, we empirically study the ability of the epistemological DB to reconsider past inference conclusions when new evidence arrives. We find that traditional greedy pipelines commit correctable errors and that epistemological DBs are able to rectify these errors without having to re-run inference from scratch. We further demonstrate that epistemological DBs readily support an advantageous representation of human-contributed “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 provide a summary of our contributions below.

1.1 Summary of contributions

Thesis Statement A key assumption of this dissertation is that the accuracy of probabilistic data integration models improves as the amount of data available for inference increases. In this dissertation we make progress towards building a system that is able to take advantage of this assumption by making key contributions in model structure, inference, and learning. We also provide empirical evidence in support of this assumption. Our thesis statement

is that probabilistic KB construction systems are more accurate at maintaining KBs than traditional greedy approaches and that we can build such probabilistic KB construction systems by using hierarchical coreference for reasoning about entities, MCMC for statistical inference, and MCMC-based learning algorithms for parameter estimation.

The main contributions are:

- **Epistemological databases.** We present a new paradigm-shift for knowledge base construction and maintenance termed an epistemological DB. In contrast to traditional deterministic and probabilistic databases in which the content is input into the database by an external process, epistemological databases infer their own content (and distributions over this content) from raw evidence input into the DB. We provide experimental evidence that epistemological DBs are better at constructing and maintaining KBs than traditional approaches. The chapters of this thesis will address important sub-problems for epistemological DBs.
- **Hierarchical coreference** (Chapter 3). Coreference resolution is foundational for epistemological databases because most pieces of evidence are mentions that must be resolved to known entities and relations already present in the DB. We propose a new model of coreference that recursively structures entities into trees of latent attributes. These trees compactly summarize mentions allowing coreference to scale to large datasets including DBLP, Rexa, PubMed, and Web of Science (5 million, 20 million, 67 million, and 150 million mentions respectively). We study the role and importance of the hierarchical structure from the perspective of the statistical inference procedure and the probabilistic model itself. We find that the hierarchy is not only advantageous for MCMC inference, but also for modeling the problem of coreference in the absence of pairwise factors.

- **Priority-driven MCMC** (Chapter 4). Epistemological DBs will need to respond quickly to arriving evidence and incorporate it without re-running inference from scratch. Furthermore, the DB should support probabilistic query answering over the entities and relations in the DB; user queries are often highly focused and only a subset of the random variables are relevant. Prioritizing which variables to select during MCMC sampling is a fundamental problem for epistemological DBs. We propose a query-aware MCMC inference algorithm for answering marginal queries, reveal conditions under which it is more accurate than traditional MCMC samplers, and demonstrate it is indeed faster on both real and synthetic data. We also propose a prioritized MCMC algorithm for MAP inference in hierarchical coreference, and demonstrate that it is orders of magnitude faster at resolving new mentions to an existing KB.
- **SampleRank parameter estimation** (Chapter 5). Parameter estimation often requires expensive inference as subroutines. We present a new parameter estimation algorithm called SampleRank that learns from each MCMC samples. We derive the family of objective function that SampleRank optimizes and find that this family is a generalization of structured support vector machines (SVMs). Given this insight, we also derive a stochastic approximation algorithm for structured SVMs. We compare SampleRank to a variety of exact and approximate learning algorithms on a diverse range of problems and models (including both discriminative and generative models).
- **Human edits to KBs** We show that by treating human input as evidence, epistemological DBs naturally support human edits to automatically constructed KBs. Rather than deterministically applying each edit directly to the KB, edits instead influence the value of the predicted truth by participating in inference as evidence to the model. We find that an epistemological treatment of human edits is more accurate (makes better use of the user edit) and robust (to mistakes) than a deterministic treatment.

1.2 Declaration of previous work and collaboration

Much of the work presented in this document has been previously published or will appear as published work in the near future.

- Work on probabilistic databases and preliminary ideas for epistemological DBs appear in Wick, Miklau, McCallum [109]
- The work on human machine cooperation (Chapter 6.5) appears as Wick, Schultz, and McCallum in a workshop [114] and Wick, Kobren, and McCallum in a follow-up workshop paper[106].
- Query-aware MCMC (Chapter 4) appears as Wick and McCallum [117].
- Hierarchical coreference (Chapter 3) appears as Wick, Singh, and McCallum [115], and entity/attribute-based coreference appears as Wick and McCallum[105].
- The work on SampleRank (Chapter 5) has appeared as Wick, Rohanimanesh, Bellare, Culotta, McCallum [111, 110], a handful of technical reports including Wick and McCallum [107] and Rohanimanesh, Wick and McCallum [76]. Earlier work on SampleRank appears as Culotta, Wick, Hall, and McCallum [21], and as Culotta [19].

CHAPTER 2

BACKGROUND

In this section we provide preliminaries on graphical models, inference, parameter estimation (learning), and coreference resolution. We also introduce notation.

2.1 Graphical models and factor graphs

Graphical models represent probability distributions over a set of random variables as a product of factors. Each factor captures statistical dependencies between some subset of the random variables by outputting a positive-real valued score to a particular variable assignment. Examples of graphical models including Bayesian networks (all factors are required to be normalized conditional probability distributions), Markov networks (factors are arbitrary and the distribution must be globally normalized), conditional random fields (CRFs), and factor graphs (useful for representing CRFs and Markov networks). Graphical models are also capable of representing artificial neural networks and deep learning components such as restricted Boltzmann machines [1].

Factor graphs are a particularly useful representation because they decompose probability distributions into a product of *factors* that capture statistical dependencies between the variables. The topology of the graph is useful for understanding the underlying probability distribution (e.g., the conditional independence assumptions), and the abstractions allow for efficient implementations of general purpose learning and inference algorithms [53].

Factor graphs are a bipartite graph between random variables \mathbf{V} and factors Ψ . Each random variable $V \in \mathbf{V}$ realizes a value v from its associated domain $\text{DOM}(V)$, and

the entire domain space is denoted $\mathcal{V} = \bigotimes_{V \in \mathbf{V}} \text{DOM}(V)$. A factor $\psi : \text{DOM}(\mathbf{V}^k) \rightarrow \mathbb{R}_+$ is a function that takes as input an assignment to k random variables, and outputs a non-negative real-valued score that quantifies the compatibility of the variable assignment. The probability of an assignment to all variables is given as a normalized product of the factors. For notational consistency, upper case letters denote random variables, lowercase letters denote their values, bold face denotes sets of random variables (sometimes with a superscript representing the size of the subset), and domains of variables are represented with uppercase scripted letters.

It is often useful to distinguish between two different variable types: observed and hidden (denoted resp. \mathbf{X} , \mathbf{Y}) each with their own domain space (denoted resp. \mathcal{X} , \mathcal{Y}). Observed variables are fixed to a value in their domain, hidden variables are the variables whose value we predict. The probability of a particular assignment to the hidden variables $\mathbf{Y} = \mathbf{y}$ is

$$\pi(\mathbf{Y} = \mathbf{y} | \mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{\psi \in \Psi} \psi(\mathbf{y}^r, \mathbf{x}), \quad Z(\mathbf{x}) = \sum_{\mathbf{y} \in \mathcal{Y}} \prod_{\psi \in \Psi} \psi(\mathbf{y}^r, \mathbf{x}) \quad (2.1)$$

In this dissertation we will further assume a log-linear parameterization of the model, in which the value of each factor is determined by real-valued weights $\boldsymbol{\theta}$ on real-valued features $\boldsymbol{\phi}$

$$\psi(\mathbf{y}^r, \mathbf{x}) = \exp \boldsymbol{\theta}^T \boldsymbol{\phi}(\mathbf{y}^r, \mathbf{x}) \quad (2.2)$$

Typically, for information extraction and data integration problems, the observed variables are pieces of text, rows in a database, or RDF triples to be integrated. Hidden variables represent the alignments, clusterings, and labelings to this data. Statistical inference can then be used to reason about these various extraction and integration decisions.

2.1.1 Inference

In this dissertation we consider two important inference problems: MAP inference and marginal inference. In the context of epistemological DBs, the former is useful for prediction while the latter is useful for giving probabilistic answers to user queries.

2.1.1.1 MAP inference

In data integration problems we are often most interested in finding the assignment to the hidden variables that results in the most likely alignment, clustering, or labeling of the data. This is known as *maximum a posteriori* (MAP) inference. More generally, the MAP assignment $\mathbf{y}_{MAP} \in \mathcal{Y}$ is the setting to the random variables that maximizes the conditional probability:

$$\mathbf{y}_{MAP} = \operatorname{argmax}_{\mathbf{y} \in \mathcal{Y}} \pi(\mathbf{Y} = \mathbf{y} | \mathbf{x}) = \operatorname{argmax}_{\mathbf{y} \in \mathcal{Y}} \boldsymbol{\theta}^T \boldsymbol{\phi}(\mathbf{y}, \mathbf{x}) \quad (2.3)$$

Finding the exact solution for many graphical models of interest is known to be NP-hard; however, approximate algorithms are able to find reasonable estimates.

2.1.1.2 Marginal inference and statistical queries

Another type of inference problem relevant to knowledge bases and probabilistic databases is marginal inference for answering probabilistic queries. Informally, a query on a graphical model is a request for some quantity of interest that the graphical model is capable of providing. That is, a query is a function over the random variables of the model. Answering the query is equivalent to finding the marginal distribution over the function's range. A query on a graphical model specifies a set of latent variables \mathbf{Z} , a set of query variables \mathbf{Y} , and a set of observed variables \mathbf{X} . The answer to the query is a distribution over the query variables \mathbf{Y}

$$\pi(\mathbf{Y} | \mathbf{x}) = \sum_{\mathbf{z} \in \mathcal{Z}} \pi(\mathbf{Y}, \mathbf{Z} = \mathbf{z} | \mathbf{x}) \quad (2.4)$$

Users can query the knowledge base as they would a normal deterministic database, but the answer to the query is a distribution rather than a single element from the domain.

2.1.1.3 Markov chain Monte Carlo

Markov chain Monte Carlo (MCMC) algorithms are critical for inference in complex models of information extraction [68], data integration [21], and machine vision [80]. Depending on the application, MCMC either draws samples from the factor graph’s distribution (useful for marginal inference), or locally searches the space of assignments to variables (useful for MAP inference).

Markov chain Monte Carlo produces a sequence of states $\{s_i\}_1^\infty$ in a state space S according to a transition kernel $K : S \times S \rightarrow \mathbb{R}_+$, which in the discrete case is a stochastic matrix: for all $s \in S$ $K(s, \cdot)$ is a valid probability measure and for all $s \in S$ $K(\cdot, s)$ is a measurable function. Since we are concerned with MCMC for inference in graphical models, a state is defined as an assignment to all hidden variables, and we will from now on let $S := \mathcal{Y}$. Under certain conditions the Markov chain is said to be ergodic, then the chain exhibits two types of convergence. The first is of practical interest: a law of large numbers convergence

$$\mathbb{E}_\pi[f(\cdot)] = \int_{\mathbf{y} \in \mathcal{Y}} f(\mathbf{y}) \mu(\pi) = \lim_{t \rightarrow \infty} \frac{1}{t} \sum f(\mathbf{y}_t) \quad (2.5)$$

where the \mathbf{y}_t are sampled variable assignments from the chain.

The second type of convergence is to a distribution π . At each time step, the Markov chain is in a time-specific distribution over the state space (encoding the probability of being in a particular state at time t). For example, given an initial distribution π_0 over the state space, the probability of being in a next state \mathbf{y}' is the probability of all paths beginning in starting states \mathbf{y} with probabilities $\pi_0(\mathbf{y})$ and transitioning to \mathbf{y}' with probabilities $K(\mathbf{y}, \mathbf{y}')$. Thus the time-specific ($t = 1$) distribution over all states is given by $\pi^{(1)} = \pi_0 K$; more generally, the distribution at time t is given by $\pi^{(t)} = \pi_0 K^t$. Under certain conditions

and regardless of the initial distribution, the Markov chain will converge to the stationary (invariant) distribution π . A sufficient (but not necessary) condition for this is to require that a transition kernel K_1 obey the reversibility condition [9]:

$$\pi(\mathbf{y})K_1(\mathbf{y}, \mathbf{y}') = \pi(\mathbf{y}')K_2(\mathbf{y}', \mathbf{y}) \quad \forall \mathbf{y}, \mathbf{y}' \in \mathcal{Y} \quad (2.6)$$

Where K_1 and K_2 are two stochastic matrices over the state space of π ¹. Convergence of the chain is established when repeated applications of the transition kernel maintain the invariant distribution $\pi = \pi K_1^2$, and convergence is traditionally quantified using the total variation norm:

$$\|\pi^{(t)} - \pi\|_{\text{tv}} := \sup_{\mathcal{A} \in \Omega} |\pi^{(t)}(\mathcal{A}) - \pi(\mathcal{A})| = \frac{1}{2} \sum_{s \in \mathcal{Y}} |\pi^{(t)}(\mathbf{y}) - \pi(\mathbf{y})| \quad (2.7)$$

The rate at which a Markov chain converges to the stationary distribution is proportional to the spectral gap of the transition kernel, and so there exists a large body of literature proving bounds on the second eigenvalues.

2.1.1.4 Metropolis Hastings

Metropolis-Hastings (MH) is an MCMC algorithm traditionally used for marginal inference, but which can also be tuned for MAP inference. MH is a flexible framework for specifying customized search transition functions and provides a principled way of deciding which search moves to accept as samples. A proposal function $T(\mathbf{y}, \cdot)$, which we also represent as a stochastic matrix, conditions on a current assignment to the variables and

¹detailed balance equations occur when $K_1 = K_2$

²this invariance follows directly from Equation 2.6 by summing both sides over the state space

proposes a new assignment by reassigning values to a small subset of the variables. The proposed change is accepted with probability $A(\mathbf{y}, \mathbf{y}')$:

$$A(\mathbf{y}, \mathbf{y}') = \min(1, R(\mathbf{y}, \mathbf{y}')), \quad R(\mathbf{y}, \mathbf{y}') = \frac{Pr(\mathbf{y}') T(\mathbf{y}', \mathbf{y})}{Pr(\mathbf{y}) T(\mathbf{y}, \mathbf{y}')} \quad (2.8)$$

MH for MAP inference requires only the condition of irreducibility so the term $T(\mathbf{y}', \mathbf{y})/T(\mathbf{y}, \mathbf{y}')$ is optional, and omitted in practice. Moves that reduce model score may be accepted and an optional temperature can be used for annealing. From Equation 2.8 the resulting MH transition kernel is

$$K_{MH}(\mathbf{y}, \mathbf{y}') = \begin{cases} T(\mathbf{y}, \mathbf{y}') & \text{if } R(\mathbf{y}, \mathbf{y}') \geq 1, \mathbf{y} \neq \mathbf{y}' \\ T(\mathbf{y}, \mathbf{y}')R(\mathbf{y}, \mathbf{y}') & \text{if } R(\mathbf{y}, \mathbf{y}') < 1 \\ T(\mathbf{y}, \mathbf{y}') + \sum_{\mathbf{y}'' : R(\mathbf{y}, \mathbf{y}'') < 1} K(\mathbf{y}, \mathbf{y}'')(1 - R(\mathbf{y}, \mathbf{y}'')) & \text{if } \mathbf{y} = \mathbf{y}' \end{cases} \quad (2.9)$$

We can interpret the kernel as follows. In the first two cases, the chain transitions to a new assignment to the variables. In Case 1 the acceptance ratio $R(\mathbf{y}, \mathbf{y}')$ is greater than or equal to one so the transition probability of the kernel is given by the proposal distribution $T(\mathbf{y}, \mathbf{y}')$. In Case 2, the acceptance ratio is less than one so we accept the move proportional to the acceptance ratio. Finally, in Case 3 the chain stays in the same state: either the proposal function proposes to stay in the same state, or it proposes to move to some other state which is subsequently rejected by R . A simple case analysis reveals that the MH kernel obeys the detailed balance condition (Equation 2.6 with $K_1 = K_2 = K_{MH}$).

2.1.2 Remarks on MCMC for MAP inference

Since most of the models discussed in this dissertation do not contain latent variables over which we must marginalize, we do not need conditions as restrictive as detailed balance when using MCMC for MAP inference. In particular, the only property we desire is

irreducibility, which guarantees full exploration of the state space. A consequence of relaxing these restrictions on the Markov chain is that we have more flexibility to use MCMC as a stochastic local search algorithm. For example, we can ignore the forward/backward ratio term in the MH acceptance ratio (Equation 2.8), and introduce a temperature parameter that controls the frequency of moves that increase probability. Often, it is useful to employ a high temperature because for many problems, we can use domain specific knowledge to find a good initialization point (and it would not make sense to use a low temperature that might wander away from this point). For example, in clustering problems such as coreference resolution, a good initialization point is to place every data point into its own cluster (termed the singleton configuration).

2.1.3 Parameter estimation

The goal of weight learning is to find a setting to factor graph parameters θ that yields high quality predictions $\hat{y} \in \mathcal{Y}$ for a set of ground truth labels $y^* \in \mathcal{Y}$. This is often achieved by minimizing a risk function over the training data. Given a training set \mathcal{D} consisting of n instances $\{\mathbf{x}_i \in \mathcal{X}\}_1^n$ with corresponding labels $\{y_i^* \in \mathcal{Y}(x_i)\}_1^n$; a regularizer $\mathcal{R}(\theta)$ that penalizes the complexity of the solution; and a loss function $\mathcal{L}(\mathcal{D}; \theta)$; the process of learning can be described as minimizing an equation of the form:

$$\hat{\theta} = \underset{\theta \in \mathbb{R}^k}{\operatorname{argmin}} \mathcal{R}(\theta) + \mathcal{L}(\mathcal{D}; \theta) \tag{2.10}$$

For example, in part-of-speech tagging, the training data would be a set of English sentences where each word is labeled with its part of speech. The goal is to set the parameters of the model so that MAP assignment of part of speech tags on an input sentence is likely to agree with the ground-truth part of speech labels.

In structured support vector machines (SVM) [100, 102] the goal is to learn a set of parameters that minimizes structured prediction risk on the training set \mathcal{D} . In particular, we

are interested in defining risk in terms of some domain-specific evaluation metric ω such as F1 prediction accuracy.

Let $\omega : \mathcal{Y} \rightarrow \mathbb{R}$ be a training signal that determines the traditional cost function Δ for a structured SVM: $\Delta : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}_+$ s.t. $\Delta(\mathbf{y}_i, \mathbf{y}_j) \mapsto \omega(\mathbf{y}^+) - \omega(\mathbf{y}^-)$, where $\mathbf{y}^+ := \operatorname{argmax}_{\mathbf{y} \in \{\mathbf{y}_i, \mathbf{y}_j\}} \omega(\mathbf{y})$ and $\mathbf{y}^- := \operatorname{argmin}_{\mathbf{y} \in \{\mathbf{y}_i, \mathbf{y}_j\}} \omega(\mathbf{y})$. Let the ground-truth label for an input \mathbf{x} be $\mathbf{y}_x^* = \operatorname{argmax}_{\mathbf{y} \in \mathcal{Y}(\mathbf{x})} \omega(\mathbf{y})$.

The empirical risk on the dataset is the expected cost of making a prediction $\hat{\mathbf{y}}_x$ when the truth is \mathbf{y}_x^* :

$$r(D) = \frac{1}{n} \sum_{x \in D} \Delta(\mathbf{y}_x^*, \hat{\mathbf{y}}_x) \quad (2.11)$$

Structured SVM minimizes a penalized upper bound on the empirical risk [102]:

$$\hat{\boldsymbol{\theta}} = \operatorname{argmin}_{\boldsymbol{\theta} \in \mathbb{R}^k} \frac{\boldsymbol{\theta}^T \boldsymbol{\theta}}{C} + \frac{1}{n} \sum_{\mathbf{x} \in D} \xi_{\text{svm}}(\mathbf{y}_x^*, \hat{\mathbf{y}}_x) \quad (2.12)$$

with one slack variable $\xi_{\text{svm}}(\mathbf{y}_x^*, \hat{\mathbf{y}}_x)$ per instance:

$$\xi_{\text{svm}}(\mathbf{y}_x^*, \hat{\mathbf{y}}_x) = \max_{\mathbf{y} \in \mathcal{Y}(\mathbf{x})} [\Delta(\mathbf{y}_x^*, \mathbf{y}) - \boldsymbol{\theta}' \phi(\mathbf{y}_x^*) + \boldsymbol{\theta}' \phi(\mathbf{y})]_+$$

Where $[r]_+ = \max(0, r)$ is the hinge loss for $r \in \mathbb{R}$.

2.2 Coreference

Coreference is the problem of clustering mentions into sets such that all the mentions in a particular set refer to the same entity. For example, in author coreference, each mention is a record extracted from the author field of a textual citation or BibTeX record. The mention record may contain attributes for the author's first, middle, and last name, as well as contextual information occurring in the citation string, including co-authors, titles, topics, and institutions. The goal is to cluster these mention records into sets, each containing all the mentions of the author to which they refer; we use this task as a running pedagogical example.

Let \mathcal{X} be the space of observed mention records; then the traditional pairwise coreference approach scores candidate coreference solutions with a compatibility function $\psi : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ that measures how likely it is that the two mentions refer to the same entity.³ In discriminative log-linear models, the function ψ takes the form of weights θ on features $\phi(x_i, x_j)$, i.e., $\psi(x_i, x_j) = \exp(\theta^T \phi(x_i, x_j))$. For example, in author coreference, the feature functions ϕ might test whether the name fields for two author mentions are string identical, or compute cosine similarity between the two mentions’ bags-of-words, each representing a mention’s context. The corresponding real-valued weights θ determine the impact of these features on the overall pairwise score.

Coreference can be solved by introducing a set of binary coreference decision variables for each mention pair and predicting a setting to their values that maximizes the sum of pairwise compatibility functions. While it is possible to independently make pairwise decisions and enforce transitivity *post hoc*, this can lead to poor accuracy because the decisions are tightly coupled. For higher accuracy, a graphical model such as a conditional random field (CRF) is constructed from the compatibility functions to jointly reason about the pairwise decisions [54]. We now describe the pairwise CRF for coreference as a factor graph.

2.2.1 Pairwise models for coreference

Each mention $x_i \in \mathcal{X}$ is an observed variable, and for each mention pair (x_i, x_j) we have a binary coreference decision variable Y_{ij} whose value determines whether x_i and x_j refer to the same entity (i.e., 1 means they are coreferent and 0 means they are not coreferent). The pairwise compatibility functions become the factors in the graphical model. Each factor examines the properties of its mention pair as well as the setting to the coreference decision variable and outputs a score indicating how likely the setting of that

³We can also include an *incompatibility* function for when the mentions are not coreferent, e.g., $\psi : \mathcal{X} \times \mathcal{X} \times \{0, 1\} \rightarrow \mathbb{R}$

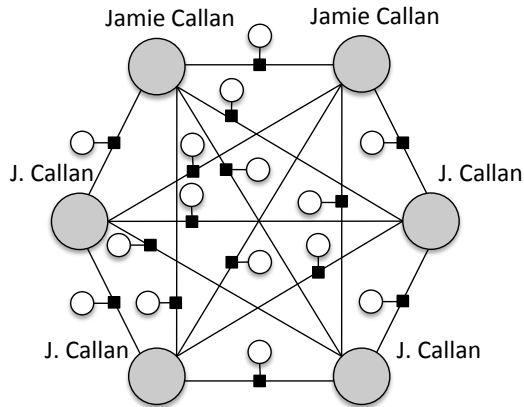


Figure 2.1: **Pairwise model on six mentions:** Open circles are the binary coreference decision variables, shaded circles are the observed mentions, and the black boxes are the factors of the graphical model that encode the pairwise compatibility functions.

coreference variable is. The joint probability distribution over all possible settings to the coreference decision variables (\mathbf{y}) is given as a product of all the pairwise compatibility factors:

$$Pr(\mathbf{Y} = \mathbf{y} | \mathbf{x}) \propto \prod_{i=1}^n \prod_{j=i+1}^n \psi(x_i, x_j, y_{ij}) \quad (2.13)$$

We illustrate the pairwise model instantiated on six mentions in Figure 2.1. Given the pairwise CRF, the problem of coreference is then solved by searching for the setting of the coreference decision variables that has the highest probability according to Equation 2.13 subject to the constraint that the setting to the coreference variables obey transitivity;⁴ this is the maximum probability estimate (MPE) setting. However, the solution to this problem is intractable, and even approximate inference methods such as loopy belief propagation can be difficult due to the cubic number of deterministic transitivity constraints.

2.2.2 MCMC inference for coreference

The primary advantages of MH for coreference are (1) only the compatibility functions of the changed decision variables need to be evaluated to accept a move, and (2) the pro-

⁴We say that a full assignment to the coreference variables \mathbf{y} obeys transitivity if $\forall ijk y_{ij} = 1 \wedge y_{jk} = 1 \implies y_{ik} = 1$

positional function can enforce the transitivity constraint by exploring only variable settings that result in valid coreference partitionings.

A commonly used proposal distribution for coreference is the following: (1) randomly select two mentions (x_i, x_j) , (2) if the mentions (x_i, x_j) are in the same entity cluster according to \mathbf{y} then move one mention into a singleton cluster (by setting the necessary decision variables to 0), otherwise, move mention x_i so it is in the same cluster as x_j (by setting the necessary decision variables). Typically, MH is employed by first initializing to a singleton configuration (all entities have one mention), and then executing the MH for a certain number of steps (or until the predicted coreference hypothesis stops changing).

This proposal distribution always moves a single mention x from some entity e_i to another entity e_j and thus the configuration \mathbf{y} and \mathbf{y}' only differ by the setting of decision variables governing to which entity x refers. In order to guarantee transitivity and a valid coreference equivalence relation, we must properly remove x from e_i by untethering x from each mention in e_i (this requires computing $|e_i| - 1$ pairwise factors). Similarly—again, for the sake of transitivity—in order to complete the move into e_j we must coref m to each mention in e_j (this requires computing $|e_j|$ pairwise factors). Clearly, all the other coreference decision variables are independent and so their corresponding factors cancel because they yield the same scores under \mathbf{y} and \mathbf{y}' . Thus, evaluating each proposal for the pairwise model scales linearly with the number of mentions assigned to the entities, requiring the evaluation of $2(|e_i| + |e_j| - 1)$ compatibility functions (factors).

2.2.3 Entity-wise models

The pairwise model can be extended to include features over entire entities (sets of mentions) by introducing factors over each entity. In previous work we show that these factors can lead to higher coreference accuracy [21] and have further extended the model to include latent entity attributes [105].

2.2.4 Entity linking

Entity linking is a simplification of the full coreference problem in which all (or most) of the entities are known in advance. The goal of entity linking is to match the set of mentions against the set of known entities. For example, the task of “wikification” is to extract mentions in newswire and link them to their corresponding Wikipedia page. In the passage

Obama returned to Washington.

we would link the string “Obama” to the Wikipedia page http://en.wikipedia.org/wiki/Barack_Obama and “Washington” to http://en.wikipedia.org/wiki/Washington,_D.C. (and be careful not to link it to the page corresponding to George Washington).

2.2.5 Evaluation

Coreference is most commonly evaluated with F1 because it balances the false positive and false negative error rates. A set of general definitions for precision (prec), recall (rec), and F1 in terms of true positives (tp), false positives (fp), and false negatives (fn) is

$$\text{prec} = \frac{\text{tp}}{\text{tp} + \text{fp}}, \quad \text{rec} = \frac{\text{tp}}{\text{tp} + \text{fn}}, \quad f1 = \frac{2(\text{prec} \times \text{rec})}{\text{prec} + \text{rec}}$$

Let the predicate $\hat{Y}(m_i, m_j)$ be true if and only if m_i and m_j are coreferent according to the configuration \hat{y} , and similarly let the predicate $Y^*(m_i, m_j)$ be true if and only if m_i and m_j refer to the same entity according to the ground-truth labels. In *pairwise fl*, we define the false positives, false negatives, and true positives as

$$\text{tp} = \sum_{i < j} \mathbb{1} \left(\hat{\mathbf{Y}}(m_i, m_j) \wedge \mathbf{Y}^*(m_i, m_j) \right) \quad (2.14)$$

$$\text{fp} = \sum_{i < j} \mathbb{1} \left(\hat{\mathbf{Y}}(m_i, m_j) \wedge \neg \mathbf{Y}^*(m_i, m_j) \right) \quad (2.15)$$

$$\text{fn} = \sum_{i < j} \mathbb{1} \left(\neg \hat{\mathbf{Y}}(m_i, m_j) \wedge \mathbf{Y}^*(m_i, m_j) \right) \quad (2.16)$$

An alternative to pairwise F1 is B-Cubed F1 [2], which is popular for evaluating within document coreference. However, because B-Cubed gives equal weight to all entities regardless of their size, it is poorly suited for cross-document coreference problems that exhibit long-tailed distributions over entity sizes. Thus, in this dissertation, we primarily employ pairwise F1 when reporting numbers for cross document coreference.

CHAPTER 3

HIERARCHICAL COREFERENCE

3.1 Overview

Coreference resolution is the problem of determining which mentions of entities (e.g., in text) actually refer to the same real-world entities. The task is foundational for other high-level information extraction and data integration tasks, including semantic search, question answering, and knowledge base construction. For example, coreference is vital for compiling author publication lists in bibliographic knowledge bases such as CiteSeer and Google Scholar, in which the DB must know if the “R. Hamming” who authored “Error detecting and error correcting codes” is the same “R. Hamming” who authored “The unreasonable effectiveness of mathematics.” Features of the mentions (e.g., bags-of-words in titles, contextual snippets and co-author lists) provide evidence for resolving such entities.

Over the years, various machine learning techniques have been applied to different instantiations of the coreference problem. A commonality in many of these approaches is that they model the problem of entity coreference as a collection of decisions between mention pairs [4, 93, 54, 91, 5]. That is, coreference is solved by answering a quadratic number of questions of the form “does *mention A* refer to the same entity as *mention B*?” by employing a compatibility function that outputs a scalar indicating how likely A and B refer to the same entity. While these models have been successful in some domains, they also exhibit several undesirable characteristics. The first is that pairwise models lack the expressive power required to represent aggregate properties of the entities. Recent work has shown that these entity-level properties allow systems to correct coreference errors made

from myopic pairwise decisions [66, 21, 120, 69, 105], and can even provide a strong signal for unsupervised coreference [6, 33, 34].

A second problem, that has received significantly less attention in the literature, is that the pairwise coreference models scale poorly to large collections of mentions especially when the expected number of mentions in each entity cluster is also large. Current systems cope with this by either dividing the data into blocks to reduce the search space [35, 55, 7], using fixed heuristics to greedily compress the mentions [72, 70], employing specialized Markov chain Monte Carlo procedures [59, 73, 90], or introducing shallow hierarchies of sub-entities for MCMC block moves and super-entities for adaptive distributed inference [88]. However, while these methods help manage the search space for medium-scale data, evaluating each coreference decision in many of these systems still scales linearly with the number of mentions in an entity, resulting in prohibitive computational costs associated with large datasets. This scaling with the number of mentions per entity seems particularly wasteful because although it is common for an entity to be referenced by a large number of mentions, many of these coreferent mentions are highly similar to each other. For example, in author coreference the two most common strings that refer to Richard Hamming might have the form “R. Hamming” and “Richard Hamming.” In newswire coreference, a prominent entity like Barack Obama may have millions of “Obama” mentions (many occurring in similar semantic contexts). Deciding whether a mention belongs to this entity need not involve comparisons to all contextually similar “Obama” mentions; rather we prefer a more compact representation in order to efficiently reason about them.

In this chapter we explore a novel hierarchical discriminative factor graph for coreference resolution that recursively structures each entity as a tree of latent sub-entities with mentions at the leaves. Our hierarchical model avoids the aforementioned problems of the pairwise approach: not only can it jointly reason about attributes of entire entities (using the power of discriminative conditional random fields), but it is also able to scale to datasets with enormous numbers of mentions because scoring entities does not require comput-

ing a quadratic number of compatibility functions. The key insight is that each node in the tree functions as a highly compact information-rich summary of its children. Thus, a small handful of upper-level nodes may summarize thousands of mentions (for example, a single node may summarize all contextually similar “R. Hamming” mentions). Although inferring the structure of the entities requires reasoning over a larger state-space, the latent trees are actually beneficial to inference (as shown for shallow trees in [88]), resulting in rapid progress toward high probability regions, and mirroring known benefits of auxiliary variable methods in statistical physics (such as [97]). Moreover, each step of inference is computationally efficient because evaluating the cost of attaching (or detaching) sub-trees requires computing just a single compatibility function (as seen in Figure 3.1). Further, our hierarchical approach provides a number of additional advantages. First, the recursive nature of the tree (arbitrary depth and width) allows the model to adapt to different types of data and effectively compress entities of different scales (e.g., entities with more mentions may require a deeper hierarchy to compress). Second, the model contains compatibility functions at all levels of the tree enabling it to simultaneously reason at multiple granularities of entity compression. Third, the trees can provide split points for finer-grained entities by placing contextually similar mentions under the same subtree. Finally, if memory is limited, redundant mentions can be pruned by replacing subtrees with their roots.

The rest of this chapter is organized as follows. In Section 3.2 we review related work on coreference, in Section 3.3 we introduce our hierarchical model for coreference. We first provide a general definition of our hierarchical model, describe how to perform MCMC inference in the general model, and then discuss specific concrete modeling choices and how they improve MCMC efficiency. In these sections, we recommend specific model and inference considerations for hierarchical coreference including (1) a way to represent entities from their children, (2) a specific set of factor functions that are broadly applicable to multiple coreference domains, and (3) a set of MCMC proposal functions that are useful

for manipulating the entity hierarchy. Finally, we present our experimental results and conclude.

3.2 Related work

Singh et al. [88] introduce a hierarchical model for coreference that treats entities as a two-tiered structure, by introducing the concept of sub-entities and super-entities. Super-entities reduce the search space in order to propose fruitful jumps. Sub-entities provide a tighter granularity of coreference and can be used to perform larger block moves during MCMC. However, the hierarchy is fixed and shallow. In contrast, our model can be arbitrarily deep and wide. Even more importantly, their model has pairwise factors and suffers from the quadratic curse, which they address by distributing inference. Our arbitrarily deep structure allows us to replace pairwise factors with factors between a child and its parent; we avoid expensive unroll operations during inference by adopting factor functions with a special form that (1) if desired can succinctly capture certain types of pairwise interactions between mentions and (2) allows each MCMC transition to be evaluated in constant time (independent of the number of child nodes in a cluster). Furthermore, as we show later, flat and fixed-depth models lack the power necessary to eliminate the pairwise factors. Indeed, a model in which the pairwise factors are replaced with child-parent factors inherently requires arbitrarily deep entities in order to properly model coreference.

One of the key advantages of the hierarchical coreference model is that it exploits data redundancy by using higher nodes in the tree to compactly summarize many contextually similar mentions. A tempting alternative to the hierarchical coreference algorithm is to modify the inference procedure in the pairwise model to similarly exploit redundancy in the data. For example, if many of the mentions in an entity cluster were redundant, most of the pairwise factors would be nearly identical to each other, and could potentially be ignored. In a separate paper, we explore this very idea. We attempt to scale pairwise coreference by approximating the MH acceptance ratio using an algorithm we term Monte Carlo Markov-

chain Monte Carlo (MCMCMC) [89]. Normally, in pairwise coreference, computing the MH acceptance ratio requires computing a linear-number of compatibility functions. The main idea in that work is to approximate the computation of the MH acceptance ratio by sampling a subset of the factors to construct a Monte Carlo estimate of the acceptance ratio. Unfortunately, this approximation introduces auxiliary variables that must be marginalized over in order to produce a correct solution. Since counting over the state-space of all possible coreference configurations is difficult, we instead resort to maximizing over the auxiliary variables. Although this approximation works well on other tasks, it fails for pairwise coreference. In particular, we observe that the MCMCMC procedure produces wild fluctuations in accuracy and fails to converge [89].

Another approach to scaling coreference is to solve the problem using a streaming clustering algorithm [70]. Streaming algorithms are particularly appealing due to their remarkable speed: the running time of streaming coreference is essentially the time it takes to read in the data (mentions) from disk. However, streaming algorithms are greedy in the sense that they are unable to revisit previous coreference decisions. For example, the streaming algorithm might merge a mention into an entity cluster only because that cluster happened to be the most compatible with the mention at the time the decision was made. Later, new entity clusters will form, and it might become apparent that the mention should belong to one of these new clusters, but there is no way of correcting this error in the streaming setting. Indeed, through empirical studies, we find that the ability to revisit initial coreference decisions improves accuracy, and reduces noise (variance) in the predicted entities [116]. Fortunately, the strengths of both approaches are complementary and can be combined; for example, streaming algorithms could provide a good initialization for non-greedy MCMC-based hierarchical coreference.

Our hierarchical model provides the advantages of recently proposed entity-based coreference systems that are known to provide higher accuracy [33, 21, 120, 105, 34]. However,

these systems are not hierarchical, and are therefore difficult to scale to large amounts of data without distributing inference.

Techniques such as lifted inference [92] for graphical models exploit redundancy in the data, but typically do not achieve any significant compression on coreference data because the observations usually violate any symmetry assumptions. On the other hand, our model is able to compress similar (but potentially different) observations together in order to make inference fast even in the presence of asymmetric observed data.

Finally, we remark that coreference is an application of the more general problem of clustering, and that hierarchical clustering algorithms exist in many other application domains. For example, greedy agglomerative clustering and hierarchical affinity propagation generate dendrograms for visualizing data [32], KD-trees provide efficient nearest neighbor queries for fast K-means implementations [67], and BIRCH generates trees of sufficient statistics on which a fast post-processing clustering algorithm can efficiently operate [122]. However, none of these approaches are well-suited for the specific problem of coreference, which involves clustering a large number of high-dimensional sparse data points into an unknown number of clusters of heterogeneous sizes (e.g., power-law distribution). Greedy agglomerative clustering is prohibitively expensive for large data because it requires a quadratic number of comparisons. KD-trees are expensive for points that lie in high dimensional spaces. BIRCH and other algorithms related to K-means are unable to predict heterogeneous cluster sizes, and often assume a fixed number (k) of clusters.

3.3 Hierarchical models of coreference

In this section we describe our discriminative hierarchical model for coreference. In contrast to the pairwise model (Figure 2.1), in which each entity is a flat cluster of mentions, our proposed model structures each entity recursively as a tree (Figure 3.1). The leaves of the tree are the observed mentions with a set of attribute values. Each internal node of the tree is latent and contains a set of unobserved attributes; recursively, these *node*

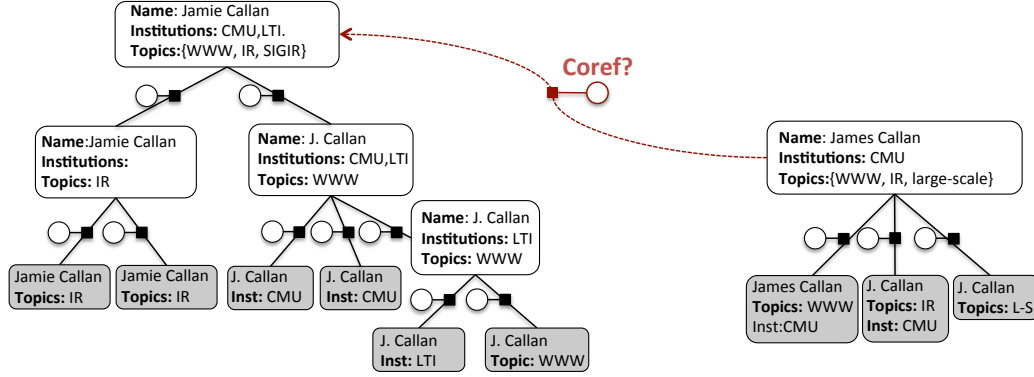


Figure 3.1: **Discriminative hierarchical factor graph for coreference:** Latent entity nodes (white boxes) summarize subtrees. Pairwise factors (black squares) measure compatibilities between child and parent nodes, avoiding quadratic blow-up. Corresponding decision variables (open circles) indicate whether one node is the child of another. Mentions (gray boxes) are leaves. Deciding whether to merge these two entities requires evaluating just a single factor (red square), corresponding to the new child-parent relationship.

records summarize the attributes of their child nodes, for example, they may aggregate the bags of context words of the children. The root of each tree represents the entire entity, with the leaves containing its mentions. Formally, the coreference decision variables in the hierarchical model no longer represent pairwise decisions directly. Instead, a decision variable $y_{ij} = 1$ indicates that node-record r_j is the parent of node-record r_i . We say a node-record *exists* if either it is a mention, has a parent, or has at least one child. Let $R = \mathbf{z} \cup \mathbf{x}$ be the set of all existing node records with observed node-record mentions \mathbf{x} and latent node-records \mathbf{z} . Let r^p denote the parent for node r , that is $y_{r,r^p} = 1$, and $\forall r' \neq r^p, y_{r,r'} = 0$. As we describe in more detail later, the structure of the tree and the values of the unobserved attributes are determined during inference.

In order to represent our recursive model of coreference, we include two types of factors: child-parent factors ψ_{cp} that measure compatibility between a child node-record and its parent, and unit-wise factors ψ_{rw} that measure compatibilities of the node-records themselves. For efficiency we enforce that parent-child factors only produce a non-zero score when the corresponding decision variable is 1. The unit-wise factors can examine compatibility of settings to the attribute variables for a particular node (for example, the set of

topics may be too diverse to represent just a single entity), as well as enforce priors over the tree’s breadth and depth. Our recursive hierarchical model defines the probability of a configuration as:

$$Pr(\mathbf{Y} = \mathbf{y}, \mathbf{z} | \mathbf{x}) \propto \prod_{r \in R} \psi_{rw}(r) \psi_{cp}(r, r^p, \mathbf{y}), \quad R = \mathbf{x} \cup \mathbf{z} \quad (3.1)$$

We require that settings to the child-parent variables induce a valid forest over nodes R (by assigning zero probability to configurations $\langle \mathbf{y}, R \rangle$ that violate this condition). Thus, the model defines a probability distribution over the space of possible ways of organizing the observed mentions \mathbf{x} into a forest with nodes R and child-parent relations \mathbf{y} .

3.4 Model details

There are two important modeling decisions pertinent to the hierarchical coreference model: (1) how to represent the nodes (entities, subentities, mentions), and (2) given this representation, what factor/compatibility functions best capture coreference similarity.

3.4.1 Representing entities, sub-entities, and mentions

We represent each node $v \in \mathcal{R}^k$ as a vector. Mention vectors are observed (given) \mathbf{X} , but the vector representation of the other node-records \mathbf{Z} (entities and subentities) must be inferred from their respective children. In particular, we want to infer these vectors such that node-records similar in vector space are more likely to be coreferent than node-records that are further apart; but we also want the representation to admit efficient MCMC calculations. Specifically, we are interested in the following desiderata:

- **Representative:** the inferred vector should be a comprehensive representation of the node’s children.
- **Compact:** the inferred vector should capture the information concisely (compressed).

- Computationally efficient: the vector representation should be easy to manipulate and efficiently support inference operations.
- Comparable: the vectors for each node should reside in the same space in order to admit meaningful comparisons (between mentions and their inferred parents, and also between inferred subentities and their inferred parents). An alternative would be to allow parents to reside in different space as the children, and learn a matrix to project them onto the same space.

With these desiderata in mind, we choose to represent each inferred node-record as a sum of its children node-records. The sum is both representative and compact (the various dimensions are aggregated through the sum, and redundant elements are compressed into single frequency values), while also yielding vectors that are comparable (in the same vector space). Although we might lose information via the sum (e.g., because multi-modal word/topic distributions might better represent the entities), we hope that the hierarchical structure will compensate for this deficiency because sub-entities act as “mixture components.”

Further, this representation is computationally efficient because the sums can easily be maintained during MCMC inference by maintaining the invariance that a node is the sum of its children: each time a node is moved from one tree to another, we subtract the node’s vector from each node in its former ancestral tree, and add the node’s vector to each node in its new ancestral tree. Assuming balanced trees, such operations are logarithmic in the number of mentions in the tree (and linear in the *sparse* vector representing the moved node). Furthermore, in the presence of certain compatibility functions, the sums admit efficient inference approximations. Note that if we were to use a more complex summary function instead of the sum, we would likely have to resort to sampling in order to infer the values of the latent vectors.

3.4.2 Model factors

In the hierarchical model there are two primary types of factors. The first is a factor on the individual nodes. These factors are able to examine the attributes of an entity node and output a number representing the cohesiveness of the entity’s attributes. The second type of factor measures the compatibility between a child node and its parent through a *child-parent compatibility function*. In this section we motivate the appropriate form for the child-parent compatibility functions.

Consider two representational vectors $x_1 = \langle 0, 1 \rangle$ and $x_2 = \langle 1, 0 \rangle$ that correspond to two nodes (that have nothing in common). Imagine a proposal which hypothesizes merging these nodes under a common parent p (as shown in Figure 3.2). Since p maintains the invariance that it is the sum of its children we have $p = x_1 + x_2 = \langle 1, 1 \rangle$. Since x_1 and x_2 are orthogonal, it is desirable that our compatibility function assign zero compatibility to a world in which they shared a common parent; however, in the hierarchical model, compatibilities are not measured directly between nodes, but between nodes and their parent. Thus, a straightforward application of traditional pairwise compatibility functions (from the coreference literature) would assign them a non-zero compatibility. For example, suppose we employ the common cosine similarity to evaluate the child-parent factors, then the compatibility of the merge is $\text{cossim}(\langle 0, 1 \rangle, \langle 1, 1 \rangle) + \text{cossim}(\langle 1, 0 \rangle, \langle 1, 1 \rangle) = 2/\sqrt{(2)}$. This directly motivates the following class of *child-parent compatibility functions*.

Definition $g : \mathbb{R}^k \times \mathbb{R}^k \rightarrow \mathbb{R}$ is a pairwise compatibility function.

Definition Let g be a pairwise compatibility function as defined above. Then a child-parent compatibility function is a function $f : \mathbb{R}^k \times \mathbb{R}^k \rightarrow \mathbb{R}$ such that $f(\text{child}, \text{parent}) \mapsto g(\text{child}, \text{parent} - \text{child})$.

Note that if we take g to be the cosine similarity function, then our definition of the corrected compatibility function f assigns 0 compatibility to our children in the example

above: $f(x_1, p) + f(x_2, p) = g(x_1, p - x_2) + g(x_2, p - x_1) = g(x_1, x_2) + g(x_2, x_1) = \text{cossim}(x_1, x_2) + \text{cossim}(x_2, x_1) = 0$.

3.4.3 Capturing context with bags of words

We conclude the section on modeling by providing concrete definitions for the generic hierarchical compatibility functions described earlier in the chapter that are able to capture the typical coreference compatibility functions employed in the literature. For example, it is common to use a bag-of-words (BoW) representation for mentions, and then measure cosine similarity between the BoWs. Of course, not all context is equally important for disambiguating the mentions. For example, in author coreference, co-author similarity provides a stronger signal for coreference than title-token similarity. Note that just because we chose to describe our model using only a single vector per-node, this does not preclude the use of multiple sources of context: trivially, individual context vectors can be concatenated into a single vector, and specialized compatibility functions would only operate over the appropriate dimensions of these vectors. Thus, we can implement weighted cosine similarities between different context BoWs. Further, we can exploit the fact that we explicitly represent the entities, and include factors that penalize BoWs with higher entropy. Finally, we can include factors that penalize the existence of entities and sub-entities. The former controls the aggressiveness of the model (the degree to which the model will merge mentions into entities, or keep them split apart), and the latter controls the depth and bushiness of the entity trees. We define our complete set of factor templates in Table 3.1.

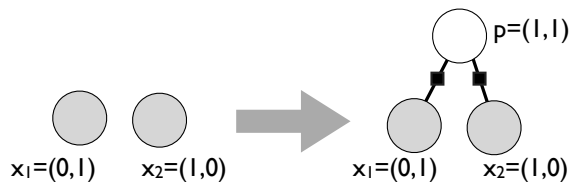


Figure 3.2: A $\text{mergeUp}(x_1, x_2)$ operation on two orthogonal vectors

Factor type	Input	params.	output (log score)
BoW cosine similarity	parent (p), child (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 BoW (b)	w	$-w \frac{H(b)}{\log \ b\ _0}$
BoW complexity penalty	node BoW (b)	w	$-w \frac{\ b\ _0}{\ b\ _1}$
names penalty	node BoW (b)	w	$-\min(w(\ b\ _0 - 1)^2, -16)$

Table 3.1: Factor definitions for 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.

3.5 MCMC for hierarchical coreference

The state space of our hierarchical model is substantially larger (theoretically infinite) than the pairwise model due to the arbitrarily deep (and wide) latent structure of the cluster trees. Inference must simultaneously determine the structure of the tree, the latent node-record values, as well as the coreference decisions themselves.

While this may seem daunting, these auxiliary structures are actually beneficial to coreference inference. Indeed, despite the enlarged state space, inference in the hierarchical model is substantially faster than a pairwise model with a smaller state space. One explanatory intuition comes from the statistical physics community: we can view the latent tree as auxiliary variables in a data-augmentation sampling scheme that guides MCMC through the state space more efficiently.¹ For example, each sub-tree identifies a potential split-point for the entity, and moving the root of the sub-tree from one entity to another is an MCMC block move that changes the coreference decisions for many mentions in a single sample.

Further, evaluating each proposal during inference in the hierarchical model is substantially faster than in the pairwise model. Indeed, under certain modeling assumptions,

¹there is a large body of literature in the statistics community describing how these auxiliary variables can lead to faster convergence despite the enlarged state space (classic examples include [97] and slice samplers [64]).

we can replace the linear number of factor evaluations (as in the pairwise model) with a constant number of factor evaluations for most proposals (for example, adding a subtree requires re-evaluating only a single parent-child factor between the subtree and the attachment point, and a single node-wise factor). We describe the sampling procedure in more details next.

3.5.1 Proposal distribution

Recall the MH proposal step for performing coreference in the pairwise model (Section 2.2.2) first randomly samples two mentions; second decides how to move them: if the two mentions are in the same cluster then the algorithm proposes to move one of the mentions out of the cluster, otherwise they are in different clusters in which case the algorithm proposes to merge the mentions so that they are in the same cluster; and finally decides whether or not to accept or reject based on the change in model score.

We modify this basic MCMC procedure to perform inference in the hierarchical model as follows. First, instead of randomly selecting two mentions, we randomly select two nodes. If the two nodes are in the same tree, we propose “splits” that detach one of the subtrees. Otherwise, the nodes are in different trees, and we propose “merges” that move one of the nodes (and the subtree rooted thereon) into the other node’s tree. Since inference must determine the structure of the entity trees in addition to coreference, it is advantageous to consider multiple MH proposals per sample. Therefore, we employ a modified variant of MH that is similar to multi-try Metropolis [50]. During each time-step, our modified MH algorithm makes k proposals and samples one according to its model ratio score (the first term in Equation 2.8) normalized across all k (See Algorithm 1). As shown in Algorithm 2, the proposals procedure comprises two steps. In the first step, the procedure calls the *nextNodePair* function which selects two nodes from the set of all known nodes in the current coreference state. In the second step, the procedure calls the *proposeMoves* function which generates a set of proposals (e.g., merges and splits) from the nodes se-

lected in the previous step. We provide our specific implementation of these two functions in Algorithms 3&4.

In designing a set of moves for our proposeMoves function, we favored a simple minimal set of moves that covers the entire search space of the hierarchical model. More formally, we sought a set of moves for which the model is irreducible and can thus retroactively undo previous inference decisions. We use the following moves

- *Merge Left (merges two subtrees together)*: inputs two nodes, a left node and a right node, and assigns the left node as the right node's parent. In order to maintain the property that a node is the sum of its children, we must subtract the vector encoding the bags-of-words from the right node's former ancestral tree, and add this vector to the new ancestral tree (i.e., add it to left, left's parent, \dots , all the way to the root of the tree).
- *Merge Up: (merges two subtrees together:)* similar to merge-up but first creates a new parent node p , then assigns the left and right node's parents to be p . Again, to maintain the sum-of-the-children property, vectors are subtracted from former ancestral tree and added to the new ancestral tree.
- *Split Right: (removes a subtree)*: split right is the opposite of merge left in that it detaches a node from its parent. If detaching causes the former parent to have only a single child, then the former parent is removed to eliminate "linked-list structures" from the tree. Split right subtracts the removed node's vector from its former ancestral tree.
- *Collapse Node: (removes intermediate node)*: inputs a subentity (intermediate node), removes that subentity and attaches its children to the deleted subentity's parent.
- *Sample Attributes*: samples a canonical attribute value for a field (sampled from the distribution implied by the appropriate bag of words).

Thus, merge-up hypothesizes new nodes, collapse-node and split-right are capable of removing nodes, and sample attributes is used for inferring the canonical attributes of the entity. It is easy to check that these moves allow exploration of the entire space of possible entity trees. We hope that these moves will allow efficient exploration of the coreference state space. Since the entity trees reflect the recent coreference “moves,” we hypothesize that our sampling strategy is particularly robust to local optima in the coreference state space.

3.5.2 Efficient proposal evaluations

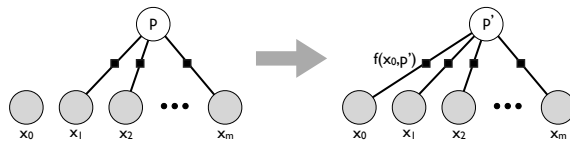


Figure 3.3: A $\text{mergeLeft}(x_0, p)$ operation.

Consider the evaluation of proposals such as MergeLeft, MergeUp, and SplitRight: these proposals change two types of variables (1) the binary variables variables that indicate the parent of each node (2) the node-record variable of the former/new parent (e.g., the variables that capture the weights in the bags of words). For example, consider the MergeLeft proposal in Figure 3.3 in which node x_0 is attached to parent p . This move changes two variables: (1) the variable $Y_{0,p}$ (not depicted) from false to true, and (2) the vector representation of p to $p' = p + x_0$. Since p changes, we normally would evaluate all the factors in Figure 3.3 to correctly compute the change in model score for MCMC (first term in Equation 2.8).

However, it is tempting to instead approximate the change in model score by only computing $f(x, p)$ (and ignoring the other factors). In general, we find that for our choice of compatibility function, we can approximately compute the model difference by only evaluating factors that neighbor the variables involved in changing a node’s parent. For child parent compatibility functions that are linear and symmetric, we can show that this approx-

imation is a constant factor from the exact computation (specifically 2 times larger). In practice, we find that this approximation is sufficient (even for compatibility functions that do not obey this property). Furthermore, we can show that asymptotically, the approximation error goes to zero for our cosine similarity based compatibility function (under certain assumptions about the representation of the entities). The intuition is that as the entities get larger, the norms of their bags also increase. Thus, adding or removing a small number of mentions from an enormously large entity changes that entities representation by only a negligible amount (relative to the norm of the entity’s representation vector).

Algorithm 1 Multi-try MH for Coreference.

Input: initial coreference state $\mathbf{y}^{(0)}$, number of samples n , temperature τ
Output: final coreference state $\mathbf{y}^{(t)}$
 $\mathbf{y}^{(t)} \leftarrow \mathbf{y}^{(0)}$
for i in 1 to n **do**
 $S \leftarrow \text{proposals}(\mathbf{y}^{(t)})$ (e.g., Algorithm 2)
 $S \leftarrow S \cup \{\mathbf{y}^{(t)}\}$ //Includes the no-op “stay here” proposal.
 $\mathbf{y}^{(t+1)} \sim Pr(\cdot|S)$ where $Pr(\mathbf{y}^{(t+1)}|S) \propto (\pi(\mathbf{y}^{(t+1)})/\pi(\mathbf{y}^{(t)}))^\tau$
 $\mathbf{y}^{(t)} \leftarrow \mathbf{y}^{(t+1)}$
end for

Algorithm 2 Generic *proposals* algorithm for hierarchical coreference.

Input: current state $\mathbf{y}^{(t)}$
Output: set of proposed next states $S = \{\mathbf{y}_i^{(t+1)}\}$
 $\langle n_i, n_j \rangle \leftarrow \text{nextNodePair}(\mathbf{y}^{(t)})$ (e.g., Algorithm 4)
 $S \leftarrow \text{proposeMoves}(n_i, n_j)$ (e.g., Algorithm 3)

3.5.3 Learning

The compatibility factors in our model take the usual exponential family form in which each factor is a log-linear combination of weights on features. For example, the child-parent factor (ψ_{cp}) takes the form

$$\psi(r_i, r_j, y_{ij})_{\text{cp}} = \exp(\boldsymbol{\theta}^T \phi(r_i, r_j, y_{ij})) \quad (3.2)$$

Algorithm 3 Implementation of *proposeMoves* for hierarchical coreference.

Input: node n_i , node n_j
Output: set of proposed next states $S = \{\mathbf{y}_i^{(t+1)}\}$
 $S \leftarrow \{\}$
 $r_i \leftarrow \text{root}(n_i)$
 $r_j \leftarrow \text{root}(n_j)$
if $r_i \neq r_j$ /*(sampled nodes n_i, n_j are in the different entities)*/ **then**
 //Merge subtrees into same tree; try various attachment points.
 $a_i \leftarrow n_i$
 while $a_i \neq \text{NULL}$ **do**
 $S \leftarrow S \cup \text{mergeLeft}(a_i, n_j)$
 $a_i \leftarrow \text{parentOf}(a_i)$
 end while
 if $\text{isMention}(n_i) \ \&\& \ \text{isMention}(n_j)$ **then**
 $S \leftarrow S \cup \text{mergeUp}(n_i, n_j)$
 end if
else
 //Nodes in same entity, try splitting or rearranging the tree.
 $S \leftarrow S \cup \text{splitRight}(n_i)$
 $S \leftarrow S \cup \text{sampleAttributes}(n_i)$
 $S \leftarrow S \cup \text{collapse}(n_i)$
end if

Algorithm 4 Implementation of the *nextNodePair* method for hierarchical coref.

Input: current state $\mathbf{y}^{(t)}$
Output: a pair of nodes $\langle n_i, n_j \rangle_1^m$
 $n_i \sim \text{Pr}(\cdot | \mathbf{y}^{(t)})$ where $\text{Pr}(n_i | \mathbf{y}^{(t)}) = \frac{1}{|\mathbf{y}^{(t)}|_{\#nodes}}$
 $c_k \sim \text{Pr}(\cdot | n_i)$ where $\text{Pr}(c_k | n_i) = \frac{1}{|\text{canopiesOf}(n_i)|}$
 $n_j \sim \text{Pr}(\cdot | c_k)$ where $\text{Pr}(n_j | c_k) = \frac{1}{|c_k|}$

where θ are the weights and ϕ are the features. The weights for the features can either be tuned manually or learned automatically from training data. In Chapter 5 we present an MCMC-based learning algorithm, SampleRank, and show how it can be employed to estimate the parameters for the hierarchical model. In the meantime we set the weights manually.

3.6 Applications

Up to now we have mostly defined the generic components of hierarchical coreference. In this section, we give details on employing hierarchical coreference to model domain-specific coreference problems.

3.6.1 Author coreference

The first application we consider is the problem of author coreference. In author coreference, the mentions are usually extracted from paper citations. For example, consider the following citations:

Approximate lineage for probabilistic DBs. C. Re, D. Suciu. VLDB, 2008.

Nilesh N. Dalvi, C. Re, and Dan Suciu Probabilistic databases: Diamonds in the dirt Commun. ACM Volume 52, 2009.

The first citation yields two author mentions (C. Re, D. Suciu) while the second citation yields three author mentions (Nilesh N. Dalvi, C. Re, and Dan Suciu). In author coreference, we must determine whether the C. Re mentioned in the first citation refers to the same author as the C. Re mentioned in the second citation. This example highlights several clues (or features) that we might use to make such a determination: first, the two C. Re mentions have a co-author in common (D. Suciu); second, the two papers are on the same topic

(probabilistic databases). We would like to exploit such information in our hierarchical coreference model. Therefore we infer topical information from the mentions using LDA, and extract bags-of-words from tokens in the titles, venues, names, and co-author fields of the citation. For example, from the first citation (above), we would extract a “C. Re” author mention and populate a hierarchical coreference mention with the following bags-of-words (BoW):

- First name BoW: $\{C. \mapsto 1\}$
- Last name BoW: $\{Re \mapsto 1\}$
- Venue BoW: $\{vldb \mapsto 1\}$
- Co-author BoW: $\{dsuciu \mapsto 1\}$
- Title: $\{approximate \mapsto 1, lineage \mapsto 1, for \mapsto 1, probabilistic \mapsto 1, dbs \mapsto 1\}$
- Topics (LDA on title+venue): $\{14 \mapsto 0.7, 54 \mapsto 0.2, 99 \mapsto 1.0\}$

Then, we endow our hierarchical coreference model with factors (e.g., child-parent compatibility functions) that examine each of these fields. See Table 3.2 for a complete list of author coreference factors.

3.6.2 Entity-linking and wikification

The problem of entity-linking is similar to coreference resolution, except that in entity-linking, the entities are already known in advance (these entities usually contain a plethora of rich context available for features). Perhaps the best known example of entity-linking is the task of wikification: given a text document (e.g., newswire article), identify and link mentions in the text to their corresponding Wikipedia page [61]. For example, consider the newswire snippet:

The Red Sox defeated the Yankees yesterday at Fenway

factor type	inputs	bag-of-words	weights (w,t)
BoW cosine similarity	parent bag, child bag	topics	$w = 8, t = -0.25$
BoW cosine similarity	parent bag, child bag	co-authors	4, -0.125
BoW cosine similarity	parent bag, child bag	venues	4, -0.25
entity existence penalty	root node (entity)	—	$w = -1.0$
subentity existence penalty	interm. node (subentity)	—	-0.5
BoW norm. entropy penalty	node	topics	0.5
BoW complexity penalty	node	co-authors	2
BoW complexity penalty	node	venues	1
names penalty	root node (entity) bag	first names	1
names penalty	root node (entity) bag	first initials	1
names penalty	root node (entity) bag	middle names	1
names penalty	root node (entity) bag	middle initials	1
names penalty	root node (entity) bag	last name	∞

Table 3.2: The comprehensive set of factor templates for our hierarchical author coreference model. See Table 3.1 for generic factor functions.

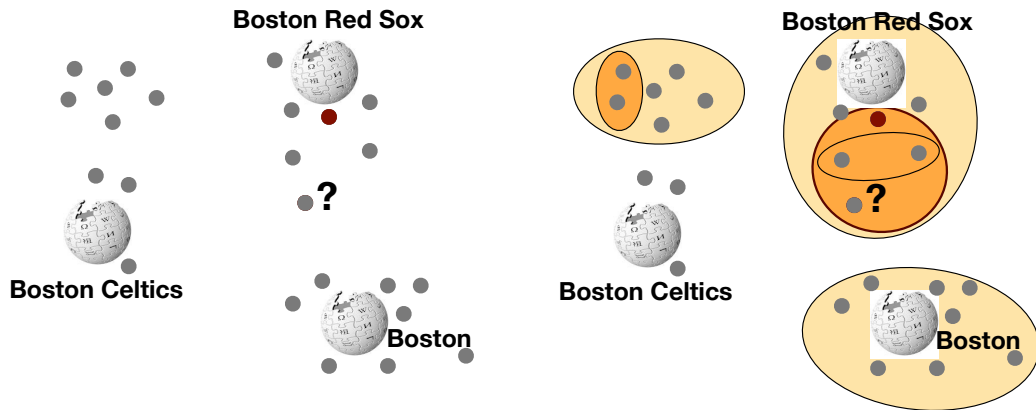


Figure 3.4: Joint linking and discovery example. As we cluster mentions into entities, it becomes more clear where they should link based on the proximity of their outer cluster (entity) to the Wikipedia mentions.

Entity-linking links the phrases “Red Sox,” “Yankees,” and “Fenway” to their appropriate Wikipedia pages (e.g., Red Sox \mapsto http://en.wikipedia.org/wiki/Boston_Red_Sox). However, the problem is not always easy; often, there are multiple ways of referring to the same entity. For example, another newswire article might use the following language instead:

Boston defeated NY yesterday at Fenway

Here, we must correctly link “Boston” to the Red Sox Wikipedia page (and not the city, the basketball team, or any other Boston organization), and similarly link “NY” to the Yankees Wikipedia page (and not the city, or another NY sports team). The difficulty lies in the ambiguity of the names; both Boston and NY might refer to a large number of potential entities that happen to have Boston or NY in their titles. Although the problem is difficult, it is not impossible; in this example, “Fenway,” a baseball stadium, provides evidence that Boston and NY should be linked to the respective baseball teams rather than the respective basketball teams. Thus, as long as the hierarchical model can capture such context, we can hope to do well on this task.

Traditionally, the problem of entity-linking is solved in a cascade of several steps [57, 61, 71]. Once mentions are identified,² we link each mention by performing the following operations. For each mention m perform:

1. **Candidate generation:** generate a set of candidate entities C
2. **Ranking:** Rank the entities in C according to similarity to m
3. **Linking:** Make a binary decision about whether to link m to the top ranked candidate in C or to NIL

Finally, after linking is complete, the remaining “NIL” entities are clustered into *unknown* entities, in a process termed entity discovery.

²in some approaches, most notably in TACKBP, mentions are identified jointly with one or more of the following tasks

A problem with this pipelined approach is that the entities discovered in the entity-discovery step are not able to inform linking decisions in the entity-linking steps. For example, consider the entity-linking problem depicted in Figure 3.4. The labeled Wikipedia logo's represent Wikipedia entities, and the small circles represent the mentions that we must link to the appropriate entities. In this example, we position the entities and mention in a two-dimensional feature space: mentions and entities closer in this space are more likely to be linked. Although it is clear to which entity the “red” mention links, the mention with the question mark is much more ambiguous (halfway between Boston and Boston Red Sox). However, if we perform some clustering of the entities right portion of (Figure 3.4), it becomes more clear that the question-marked mention should link to the Red Sox. Thus, we hypothesize that jointly solving entity-linking and entity-discovery should improve entity-linking results. However, solving these problems jointly in practice entails coreference over a large number of mentions; thus, until now, such an approach would be prohibitively expensive. We now describe how to model this problem with the hierarchical coreference model.

We propose to solve entity-linking and discovery jointly by running large-scale hierarchical coreference. Specifically, we demote all “known” entities to mentions and simply perform hierarchical coreference using the constraint that only one Wikipedia mention can be coreferent to an entity. Although demoting entities to mentions might seem like a disadvantage, it potentially gives our model the chance to discover even richer representations of the entities. For example, in Wikification, most approaches can only use features derived from the entities Wikipedia page for making linking decisions. However, in our approach, the hierarchical model builds up representations of the entities that contain features derived from many mentions (not just Wikipedia). Thus, the entities will contain all the rich feature information from not just the Wikipedia mention, but all the other mentions to which it is coreferent.

We extract Wikipedia mentions as follows. First, for each Wikipedia page, we create a mention. Next, for each anchor text in Wikipedia (that links to a Wikipedia page) we create a mention. For example, the anchor text “husband” might appear on Michelle Obama’s Wikipedia and link to Barack Obama’s Wikipedia page. We would then consider “husband” a mention of Barack, but we also consider the Barack Obama Wikipedia page a mention. In addition, we exploit this linking structure when initializing the hierarchical model. Specifically, we require that the anchor text mentions be children of the page mentions to which they link. For example, “husband” would be a child of “Barack Obama” and all the features we extract for husband would automatically propagate to the “Barack Obama” mention. All mentions (whether from Wikipedia pages, Wikipedia anchor texts, or raw text documents) comprise the following fields:

- *Name BoW*: the surface forms for the mention.
- *Context BoW*: context extracted in a ± 15 token window around the mention
- *Mentions BoW*: other mentions that appear in the document.
- *Combined BoW*: top four words from mention and context.
- *Topics*: LDA topics on the document from which the mention is extracted.

As usually, we include factors in the hierarchical model that examine these fields. However, we must also respect the domain specific constraints inherent to the problem of entity-linking. Thus, in addition to the contextual BoW, we also include BoW that summarize other information about the entities

- *Source BoW*: the source of the document from which the mention is extracted (Wikipedia, newswire, N/A)
- *Document BoW*: the id of the document from which the mention is extracted

These bags allow us to implement factors that enforce the constraints that (1) only one Wikipedia page mention can be coreferent to an entity and (2) only one mention in each document can link to an entity.³

In summary, our approach to joint-linking in discovery is essentially hierarchical coreference with the following differences:

- We are not allowed to construct trees that contain more than one mention from the target KB (e.g., Wikipedia page mentions).
- We are not allowed to propose moves that change the parents of mentions whose true entity are known in advance (e.g., the anchor text mentions in Wikipedia are permanently linked to their parent Wikipedia page).

3.7 Experiments

In this section, we empirically study the hierarchical coreference model. First, we evaluate the scalability of the hierarchical model by comparing it with a pairwise model. We further demonstrate scalability to PubMed (67 million authors) and Web of Science (150 million authors). Next, we compare the hierarchical model to fixed-depth alternatives, and find that the hierarchy is indeed necessary for when the model lacks pairwise (between mentions) factors.

3.7.1 Data

For these experiments we employ two labeled data sets. The first dataset is a subset of the Rexa dataset that contains 2,833 labeled mentions⁴. The second dataset is a highly ambiguous subset of Rexa that contains only half the number of mentions (1,459). How-

³Occasionally useful in some entity-linking problem settings.

⁴<http://www2.selu.edu/Academics/Faculty/aculotta/data/rexa.html>

name	#entities	#mentions
D. Allen	44	72
A. Blum	6	201
S. Jones	76	471
L. Lee	79	216
J. McGuire	8	19
A. Moore	28	227
H. Robinson	14	31
S. Young	34	222
Total	280	1459

Table 3.3: REXA dataset.

Dataset	No. mentions	Inference time	No. Machines	No. Cores (total)
BibTeX	1.3 million	1-2 hours	1	1
DBLP	5 million	2-3 hours	1	1
REXA citations	20 million	3 days	3	48
PubMed	67 million	2 days	3	48
WoS	148 million	2 days	3	24

Table 3.4: Large-scale hierarchical coreference runs.

ever, these mentions all refer to an author entity that has a common first-initial last-name combination (See Table 3.3 for details).

3.7.2 Scalability

In order to evaluate the scalability of the hierarchical coreference model, we compare it to a pairwise model on the problem of author coreference. For our first evaluation, we combine conference and journal papers from DBLP⁵ with BibTeX files culled from the web to obtain a dataset with 5.7 million mentions. We also include 2,833 labeled author mentions for the purpose of evaluation.⁶ In these experiments, we evaluate the model using the pairwise F1 defined in Section 2.2.5.

⁵<http://www.informatik.uni-trier.de/~ley/db/>

⁶<http://www2.selu.edu/Academics/Faculty/aculotta/data/rexa.html>

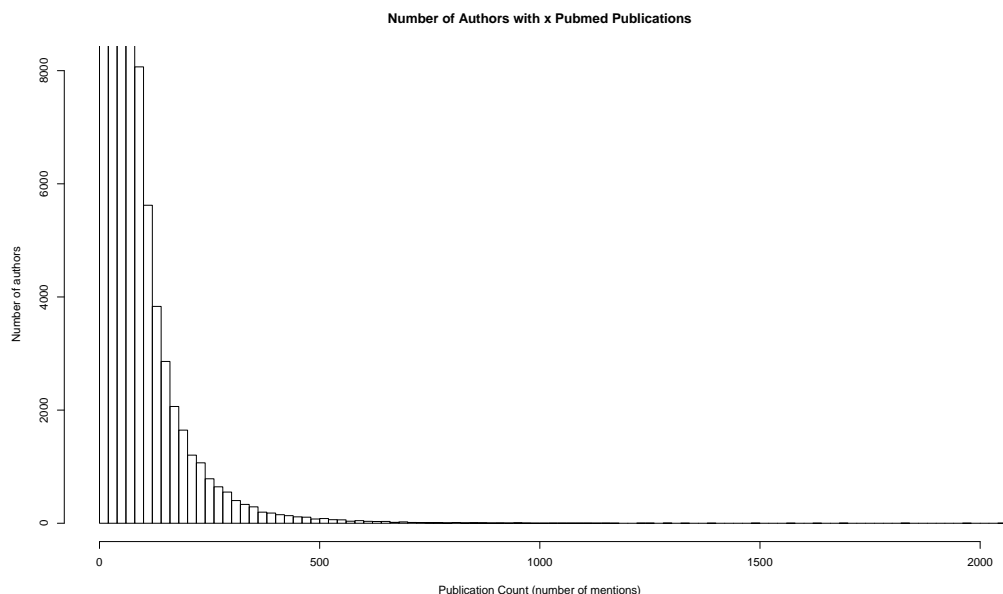


Figure 3.5: Coreference of 67 million author mentions from PubMed enables us to attribute papers to real-world authors. Shown here is the distribution over publication counts for PubMed authors (i.e., the frequency of an author with x publications) resulting from our inference.

We set the log-factor potentials of the two models manually using development data (see Table 3.1), and use the inference procedure described in Section 3.5. Manually tuning these models is time-consuming, but we found that the process could be made feasible if we restrict the weights to be powers of two. Note that after the final manual tuning, both the hierarchical and pairwise models achieve similar accuracy on the set of 2,833 labeled mentions. For the inference in the pairwise model, we use a proposal distribution that moves a single mention in each sample (See Chapter 2.2.2).

We find that the hierarchical model can reach 90% accuracy on this dataset (more than five million mentions) in under four hours. Unfortunately, the pairwise model cannot scale to this size, so we use a 1.3 million mention subset of the data. Figure 3.6a shows that sampling in the hierarchical model is much faster than pairwise and Figure 3.6b shows that the this faster sampling rate results in better accuracy over time. For example, the hierarchical model is 72 times faster than the pairwise model at achieving an accuracy of 60%. Further,

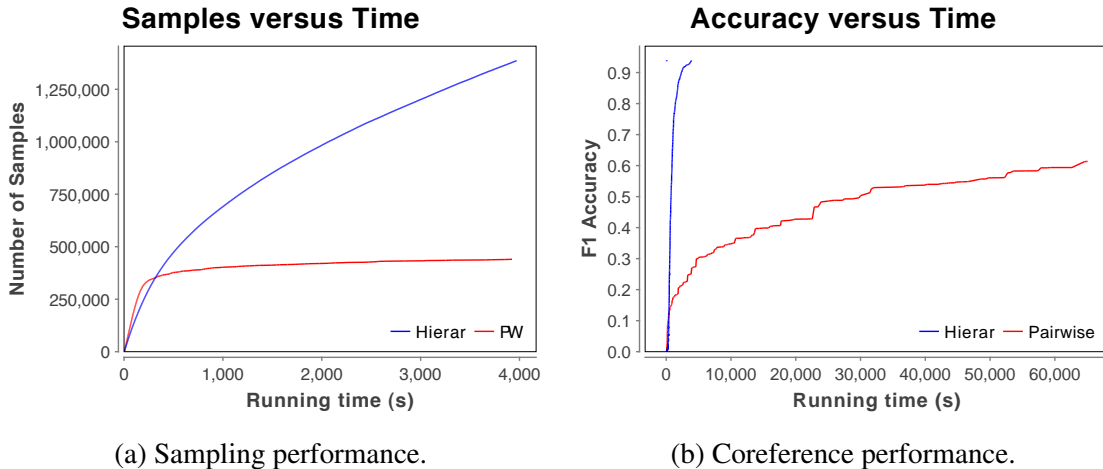


Figure 3.6: Traditional KB (top) vs. epistemological KB (bottom).

we have run the hierarchical coreference model on large collections of mentions including REXA,⁷ PubMed,⁸ and Web of Science (WoS).⁹ We summarize these runs in Table 3.4. Unfortunately we lacked labeled data and could not precisely evaluate the accuracy of these runs. However, manual inspection of the output reveals reasonable clusterings of mentions into entities. For example, in PubMed, we correctly disambiguate the mentions of two “L. M. Burke” entities. We discover one L.M. Burke who has authored 104 publications on the topics of exercise science and sports nutrition, including titles such as:

- “Nutrition for travel”
- “Placebo effect of carbohydrate feedings during a 40-km cycling time trial”
- “Fueling strategies to optimize performance training high or training low”
- “Carbohydrates and fat for training and recovery”
- “Sports nutrition. Approaching the 90s”

⁷rexa.info

⁸<http://www.ncbi.nlm.nih.gov/pubmed>

⁹<http://thomsonreuters.com/thomson-reuters-web-of-science/>

and we discover a second L.M. Burke who has authored four publications on the topic of education in nursing:

“Preceptorship and post-registration nurse education”

“Education purchasers views of nursing as an all graduate profession”

“Teacher self-evaluation, an assessment using Delamonts beyond Flanders fields technique”

“Research utilization and improvement in outcomes after diagnostic cardiac catheterization”

A caveat of this type of manual inspection is that it is more difficult to assess recall than precision (especially since, in contrast to authors in computer science, many of the authors in PubMed and WoS lack public homepages with comprehensive publication lists). Therefore, to supplement our manual inspection, we display the distribution over the size of predicted entities in Figure 3.5. More specifically, the x -axis is the number of author mentions, and the y -axis is the number of entities that have that number of mentions. Thus, another way of interpreting this bar plot is as a distribution over publication counts for all inferred PubMed authors. Intuitively, the curve behaves as we would expect (a power-law) in which many authors have a small number of publications, but only a few authors have many publications. For example, only a few hundred authors have 500 or more publications. Note, that our model is not perfect; for example, it predicts one author with 2500 publications. This is likely an error because it is estimated that even Erdos himself authored only 1,500+ papers.¹⁰ However, such errors are inevitable; overall, the model’s predictions on the large dataset are reasonable.

3.7.3 Studies on the role of the hierarchy

We also study the importance and role of the hierarchy in hierarchical coreference. The primary benefits that the hierarchy provides are:

¹⁰Source: Jerrold Grossman, Erdos Number project <http://www.oakland.edu/enp/pubinfo/>.

- *Block moves* allow large changes to the state space during inference that could potentially overcome local optima.
- *Child-parent* factors (instead of pairwise) allow the computations involved in each move to be more efficient.
- *Improved modeling power* over flatter hierarchies (and compression that adapts to the size of the entities).

The first two benefits concern the efficiency of *inference* while the last benefit concerns the representational power of the *model*. We are interested in understanding the role of the hierarchy from both an inference and modeling perspective. To this end, we compare the following coreference systems:

- *Hierarchical coreference model*. The model presented in this section (arbitrarily deep entity trees and inference using Algorithm 1 with Algorithm 4 for implementing *nextNodePair* and Algorithm 3 for generating the set of moves in each step).
- *Greedy hierarchical coreference model*. Same model as above, but with a greedy inference procedure that cannot undo coreference decisions. In particular, we modify Algorithm 1 so that it will never propose a move that detaches a sub-entity (or mention) from its parent. We happen to implement this modification by appropriately modifying *proposeMoves*, but this could instead be implemented in *nextNodePair* for further computational efficiency if desired.
- *Flat entity-mention model (FEM)*. A variant of the hierarchical model that is one level deep and therefore lacks sub-entities. MCMC inference moves single mentions from one entity to another.
- *FEM with block moves*. Same model as above, but MCMC inference is also permitted to move all the mentions in an entity in a single move (single mention moves are still permitted).

- *Two-tiered hierarchical model.* A variant of the hierarchical model that has one level of mentions, one level of sub-entities, and one level of entities. This model can also be interpreted as a variant of Singh et al. [88] in which the pairwise factors are removed in order to increase efficiency, and the canopies are interpreted as super-entities. For this model, we employ their layer-by-layer inference strategy: in the first step, we perform MCMC inference to cluster mentions into sub-entities; in the second step, we perform MCMC inference to cluster sub-entities into entities.

These systems span three different models, which are all versions of the hierarchical model in which the structure of the entity trees are limited to varying degrees (flat, two-tiered, unlimited). Therefore, we are able to employ the same factor functions for each model. For each model, we also control how “aggressively” the model merges mentions into entities by including an additional penalty on the existence of an entity. A positive penalty causes the model to prefer coreference configurations with fewer entities because it penalizes configurations for each inferred entity. Similarly, a negative penalty causes the model to prefer coreference configurations with more entities because it rewards configurations for each inferred entity. Thus, by varying the aggressiveness, we can compare the models across a wide spectrum of the precision/recall trade-off.

In order to control for any differences in wall-clock running time, we run each model on each aggressiveness for a fixed number of samples (one-million). In Figure 3.8 we show the precision, recall, and F1 for four of the systems¹¹ as a function of the aggressiveness (aggressiveness of 0 is the default setting for the entity penalty (1)). Varying the aggressiveness does indeed trade-off precision and recall for the different models, thus representing a broad range of coreference models. As we initially speculated, the hierarchy appears to be crucial for modeling coreference in the absence of pairwise factors. Indeed, the flat structured model shown in Figure 3.7b is incapable of modeling the Rexa author coreference

¹¹We omit the greedy hierarchical coreference system for the purpose of this study.

dataset for any setting to the aggressiveness penalty (precision plummets before recall begins improving). Although the MCMC block moves improve the accuracy of the flat model (Figure 3.7c), the model still performs poorly across its entire aggressiveness spectrum (this indicates that the problem is likely due to the structure of the model rather than local optima in inference). In Figure 3.7d we show the 2-tiered model. The layer of sub-entities in the 2-tiered model dramatically improves performance of the flat model; however, the accuracy is still significantly lower than the fully deep hierarchical model. We compare the F1 accuracies of the four models directly in Figure 3.8.

Thus, we conclude that hierarchical model’s success is not just due to its inherent inference benefits: a non-trivial hierarchy is indeed a *modeling* necessity. Although the fixed-depth (depth=2) model is capable of performing reasonably well on the Rexa data, it still performs worse than the fully hierarchical model. We hypothesize that as the number of mentions per entity increases, the number tree levels for a child-parent model must also increase. However, further experimentation is necessary to confirm this statement.

In order to further investigate the modeling benefits of the hierarchy, we also study if the structure of the subentities can positively influence coreference accuracy. In Figure 3.5 we compare three greedy strategies for inferring better quality subentities than the default hierarchical coreference sampler. Each of these systems first run 500k steps of inference, then reorganizes the children of each node into sub-entities by greedily clustering the children at each layer according to the specified strategy, and finally runs an additional 500k steps of inference. The topic and co-author strategy cluster children into sub-entities according to what their highest weighted topic or co-author is respectively. The k -means strategy randomly picks a k between 2 and $\log_2(\text{numchildren})$ and then runs k -means using sum of the cosine similarities of the different bags (topics, co-authors, etc.). We see from these results that re-organizing the tree into more semantically meaningful sub-entities (e.g., via topics or k -means) increases accuracy over the baseline model. We are encouraged that such a simple strategy of reorganizing the trees can lead to a bump in accuracy; in fu-

ture work we could consider adding additional MCMC moves that organize the children of nodes into meaningful subentities.

Finally, in Table 3.6 we compare the greedy and non-greedy inference for hierarchical coreference; we report the average F1 and standard error over the course of three runs with different initial random seeds; we use one-million samples for this experiment in order to ensure that inference stabilizes to a local optima. Although greedy coreference is generally faster than non-greedy coreference (run-time are not reported), we can see that the greedy algorithms inability to revisit past decisions results in lower accuracy across the entire aggressiveness spectrum. Further, we observe that the greedy algorithm is less stable in the sense that it is more sensitive to the initial random seed than the non-greedy algorithm (see the column marked “relative std. err” where we see that the greedy algorithm can be up to 5.63 times less stable). The reason is that the greedy algorithm is that all coreference decisions in the greedy algorithm are final and thus earlier decisions will have a greater impact on the final optima to which it converges. These results also partially support our earlier hypothesis that our set of MCMC moves for hierarchical coreference is particularly robust to local optima.

Method for organizing subentities	Precision	Recall	F1
Default h-coref	80.9 ± 0.35	71.2 ± 0.35	75.7 ± 0.47
Cluster on topics	82.1 ± 0.53	73.1 ± 0.26	77.3 ± 0.38
Cluster on co-authors	72.8 ± 6.22	76.6 ± 0.36	74.2 ± 3.37
K-Means	81.7 ± 0.35	75.5 ± 0.78	78.5 ± 0.52

Table 3.5: The effect of structuring subentities on coreference accuracy.

3.8 Conclusion and future work

In this chapter, we presented a new model for large-scale coreference that organizes each entity as a tree. The advantages of the hierarchy are that it provides (1) compact summarization of redundant mentions (2) natural block moves, and (3) richer representations

Model Aggr.	F1 of non-greedy	F1 of greedy	relative std. err (greedy/non-greedy)
-8	20.78 \pm 0.54	30.24 \pm 1.06	1.96
-4	55.38 \pm 0.79	55.63 \pm 0.78	0.99
-2	69.81 \pm 0.69	68.30 \pm 0.46	0.67
-1	74.03 \pm 0.26	73.50 \pm 0.42	1.60
0	75.72 \pm 0.47	75.09 \pm 0.58	1.23
1	79.87 \pm 0.57	74.61 \pm 0.81	1.43
2	78.58 \pm 0.54	62.85 \pm 3.03	5.63
4	58.45 \pm 0.17	55.44 \pm 0.49	2.89
8	54.43 \pm 0.71	53.45 \pm 0.25	0.35

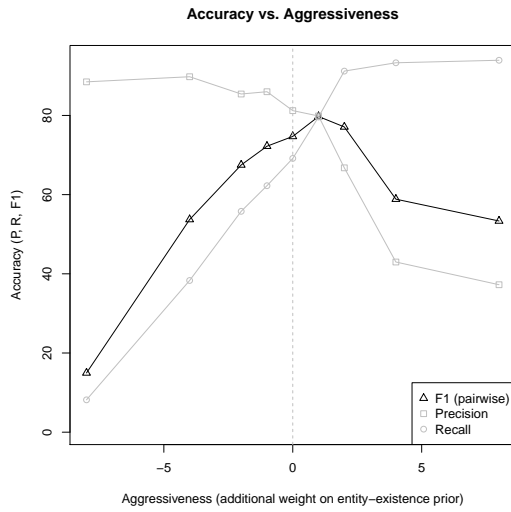
Table 3.6: Greedy vs. non-greedy inference on Rexa author coref (one-million samples). Aggressiveness (how aggressively the model wishes to merge mentions into entities) is varied across rows (0 is the cross-validated value; rows highlighted in gray are aggression values for which the model performs well). For all but one value in this range, greedy inference is more unstable than non-greedy inference (measured by relative standard error).

of entities. We demonstrated that the hierarchical model is much more scalable than the traditional pairwise model allowing us to scale to nearly 150 million author mentions from Web of Science. We explored the role and importance of the hierarchy and found that the hierarchical structure is a modeling necessity for models that do not contain pairwise factors. We also proposed a new approach for entity-linking that jointly solves the problems of linking and discovery (we save evaluation of entity-linking for Chapter 6).

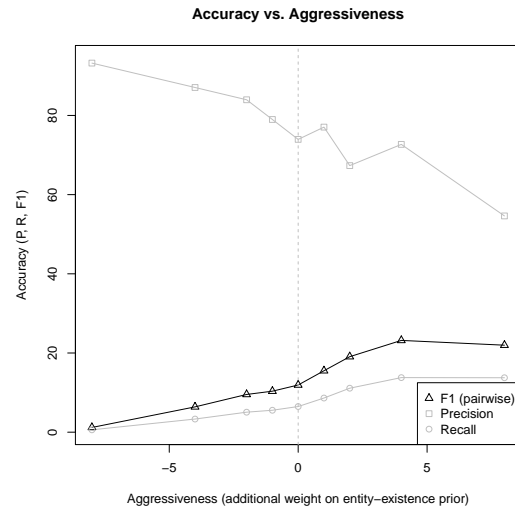
Our results also provide some support for the hypothesis and assumptions upon which epistemological DBs are based. Specifically, the fact that non-greedy MCMC-based algorithms perform better than greedy algorithms (that cannot revisit past conclusions) indicates that future information (in these experiments, the new information was the entities that arose during inference) can indeed inform past integration decisions. In the penultimate chapter (Chapter 6) we find further support for the hypothesis that more evidence improves integration inference, and those results depend crucially on non-greedy algorithms such as the MCMC procedure introduced in this chapter.

In future work we would like to apply hierarchical coreference on other coreference problems. Joint coreference of citations, venues, titles, authors, and grants is an especially

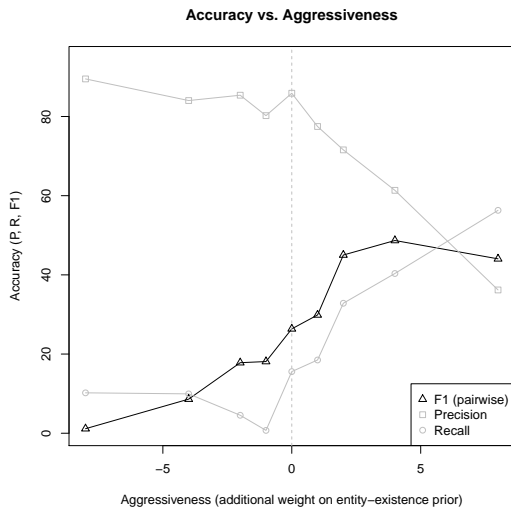
appealing application. We would also like to consider alternative inference algorithms that are more deterministic than MCMC. Finally, exploring alternative entity representations such as embeddings has the potential not only to improve accuracy, but to make hierarchical coreference an order of magnitude faster (because dense low-dimensional vectors are much faster to maintain than sparse high-dimensional vectors).



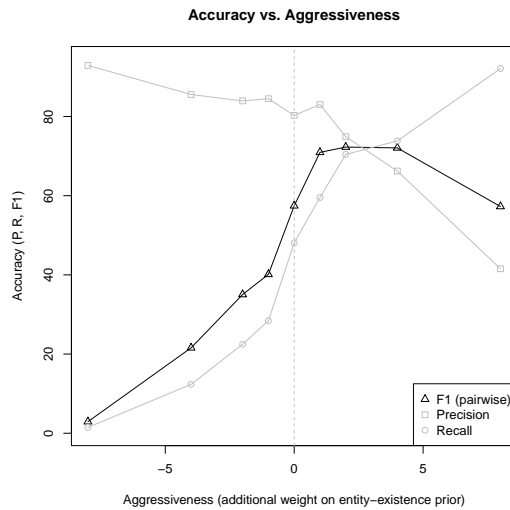
(a) Hierarchical model.



(b) Flat entity model (FEW).



(c) FEW with block moves.



(d) 2-tiered FEM.

Figure 3.7: A comparison of the infinitely deep hierarchical coreference model with a flat entity-based model (with and without block moves). These results confirm our analysis in which we conclude that a hierarchical structure is a modeling necessity in the absence pairwise factors. We vary the magnitude of an additional entity penalty factor and record the impact on precision, recall, and F1. The original setting of the entity penalty factor is 1 (determined on held-out data), and the point $x = 0$ (no additional penalty) corresponds to this parameter setting. Larger penalties encourages the model to predict fewer entities and obtain higher recall, hence the term “aggressiveness.” The F1 of these model are compared directly in Figure 3.8

Coref Model Comparison

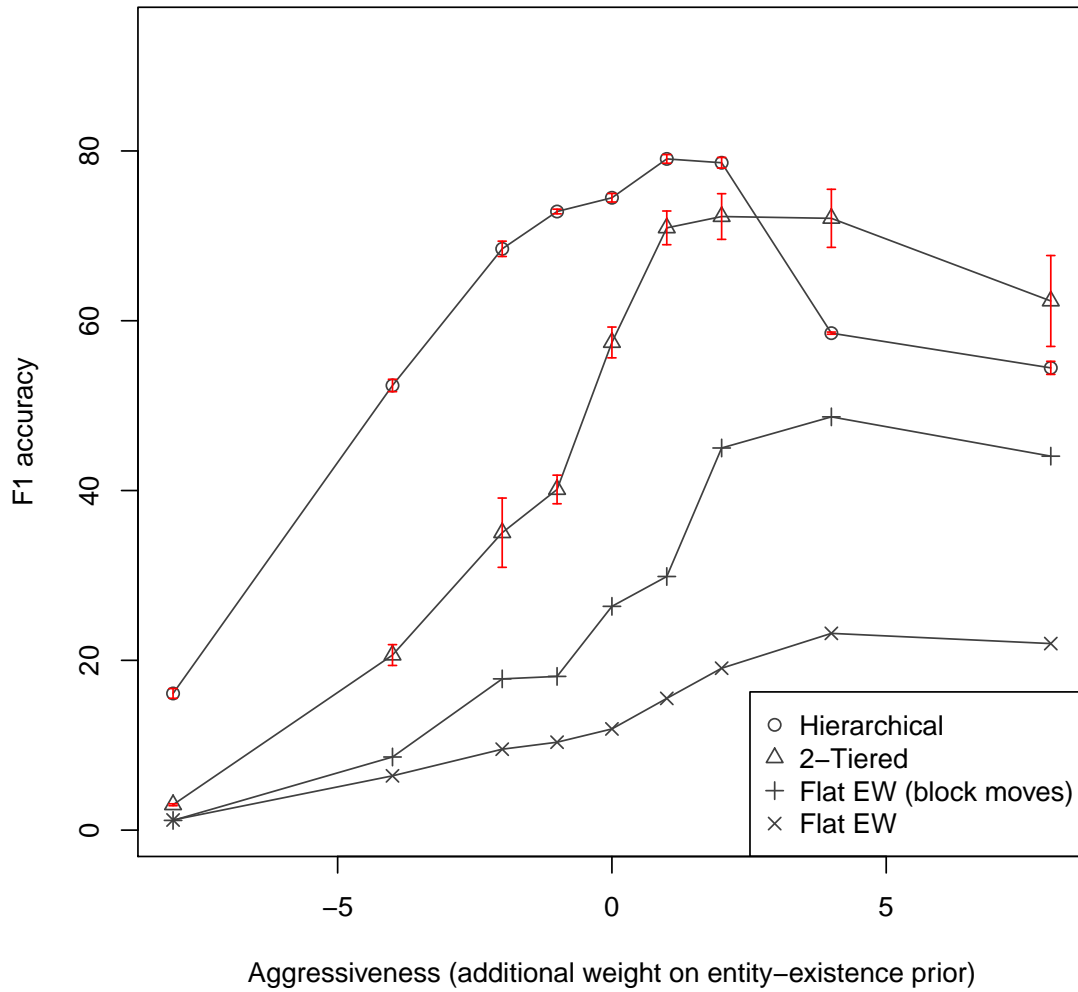


Figure 3.8: A comparison of different coref models. The curves for the hierarchical and two-tiered are the average of three runs; error bars for these two models are the standard error.

CHAPTER 4

INFERENCE WITH PRIORITY-DRIVEN MCMC

4.1 Overview

Markov chain Monte Carlo is the *sine qua non* for performing inference on statistical models that capture complex dependencies amongst the hidden variables. However, in a large-scale epistemological database even MCMC may be expensive because there are more variables than can fit in memory, and furthermore, user queries may be time-sensitive and only focus on a small subset of the variables. For example, in a large bibliographic database with millions of author entities, a user might want to know the probability that two researchers co-authored a paper together (i.e., the query concerns only a single pair of authors from a set of one-million-choose-two binary author-pair variables). In this case, it would be wasteful to sample each variable with equal importance, because the binary variable governing whether or not the two specific authors has an order of a 1 out-of-one trillion chance of being selected for sampling. Moreover, in the incremental KB construction setting—in which new data continues to arrive for integration—the epistemological DB must integrate the new data as efficiently as possible. The new data might provide evidence that causes the model to change previous integration decisions, causing cascading changes to the KB. However, the new evidence is still only likely to effect the assignment to a small percentage of the random variables that govern integration decisions. Thus, prioritizing which variables to select for sampling is an important problem; surprisingly, a problem which has been largely overlooked by the machine learning and statistics communities.

In this chapter we present our work on query-aware-MCMC (QAM), an MCMC sampling method that more frequently selects variables for sampling that are likely to have a larger influence on the answer to the query. We present several heuristics for choosing these priorities based on the topology of the graphical model as well as the statistical dependencies between variables. We show that in finite time, even a basic query-aware sampler can achieve smaller error than than a traditional sampler, and derive a fixed point equation to identify the time-step in which this occurs. Experimentally, we evaluate the query-aware sampler on synthetic graphs in which we can exactly construct and apply the MCMC transition kernels, as well as real-world data in which we must perform sampling. In both cases, we show that a query-aware sampler can be substantially faster. The work in this section is published in Wick and McCallum [117].

4.2 Related work

Despite the prevalence of probabilistic queries, the machine learning and statistics communities have devoted little attention to the problem of query-specific inference. The only existing papers of which we are aware both build upon loopy belief propagation [16, 14]; however, for many inference problems, MCMC is a preferred alternative to LPB because it is (1) able to obtain arbitrarily close approximations to the true marginals and (2) is better able to scale to models with large or real-valued variable domains that are necessary for state-of-the-art results in data integration [21], information extraction [68], and deep vision tasks with many latent layers [81]. Furthermore, MCMC is better suited for epistemological databases because its intermediate result of inference is a single possible world, allowing us to take advantage of decades of engineering on traditional DBMSs.

The decayed MCMC algorithm for filtering [51] can be thought of as a special case of our method in which the model is a linear chain, and the query is for the last variable in the sequence. That paper proves a finite mixing time bounds on infinitely long sequences. In contrast we are interested in arbitrarily shaped graphs and in the practical consideration

of large finite models. MCMC has also recently been deployed in probabilistic databases [109] in which it is possible to incorporate the deterministic constraints of a relational algebra query directly into a Metropolis-Hastings proposal distribution to obtain quicker answers [108, 103].

Bulatov [10] proves bounds on the amount of influence a far-away observation can exert on the marginal of a variable as a function of their topological distance (i.e., path length) in a graphical model. These results provide insight and motivation for query-specific inference.

4.3 Query-aware MCMC

In this section we investigate priority-driven inference for answering probabilistic queries on graphical models (this section is based on [117]). In contrast to the previous hierarchical coreference section, in which we use MCMC as a stochastic local search algorithm for MAP inference, here we focus on marginal inference (see Section 2.1.1.2 for preliminaries on marginal inference and probabilistic query answering).

Given a query $Q = \langle \mathbf{Y}, \mathbf{Z}, \mathbf{X} \rangle$, and a probability distribution π encoded by a graphical model \mathcal{G} with factors Ψ and random variables \mathbf{V} , the problem of query specific inference is to return the highest fidelity answer to Q given a possible time budget. We can put more precision on this statement by defining “highest fidelity” as closest to the true marginal distribution $\pi(\mathbf{Y}|\mathbf{X}) = \sum_{\mathbf{Z}} \pi(\mathbf{Y}, \mathbf{Z}|\mathbf{X})$ in total variation distance. In this section we will use V_i to refer to any type of variable (either a query, latent, or evidence variable).

Our approach for query specific inference is based on the Metropolis Hastings algorithm described in Section 2.1.1.4. A simple yet generic case of the Metropolis Hastings proposal distribution T (that has been quite successful in practice) employs the following steps:

- 1: Beginning in a current state s (assignment to all the variables), select a random variable $V_i \in \mathbf{V}$ from a probability distribution p over the indices of the variables $(1, 2, \dots, n)$.

Algorithm 5 Query specific MCMC

- 1: select some $\mathbf{v} \in \mathcal{V}$
 - 2: **for** $t = 0, \dots, t_\beta$ **do**
 - 3: sample a random variable:
 $x_i \sim p_{q_s} \propto \mathcal{J}_\rho(x_q, x_i)$
 - 4: sample a new value for the random variable:
 $v'_i \sim q(\mathcal{V}_i)$
 $x_i \leftarrow v'_i$
 - 5: accept according to the Metropolis Hastings rule, let:
 $\alpha \leftarrow \min \left(1, \frac{\prod_{\psi \in \text{scope}(i)} \psi(v'_i, \cdot) q(v_i | v'_i)}{\prod_{\psi \in \text{scope}(i)} \psi(v_i, \cdot) q(v'_i | v_i)} \right)$
 $d \sim \text{Bern}(\alpha)$
 - 6: **if** $d > 0.5$ **then**
 - 7: $\mathbf{v} \leftarrow \mathbf{v}'$
 - 8: **end if**
 - 9: **end for**
-

- 2: Sample a new value v_j for V_i according to some distribution $q(\mathcal{V}_i)$ over that variable's domain, leave all other variables unchanged and return the new state s' .

In brief, this strategy arrives at a new state s' from a current state s by simply updating the value of one variable at a time. In traditional MCMC inference, in which the marginal distributions of all variables are of equal interest, the variables are usually sampled in a deterministic order, or selected uniformly at random; that is, $p(i) = \frac{1}{n}$ induces a uniform distribution over the integers $1, 2, \dots, n$.

However, given a query Q , it is reasonable to choose a p that more frequently selects the query variables for sampling (this is further justified because the statistical dependence between variables in a graphical model decay exponentially fast as a function of their topological distance [10]). Clearly, the query variable marginals depend on the remaining latent variables, so we must tradeoff sampling between query and non-query variables. A key observation is that not all latent variables influence the query variables equally. A fundamental question raised and addressed in this section is: *how do we pick a variable selection distribution p for a query Q to obtain the highest fidelity answer under a finite time budget.* We propose to select variables based on their *influence* on the query variable according to the graphical model.

4.3.1 Prioritization via influence

We assign each variable a priority based on how much influence it exerts on the query variables. The more influence a variable has on the query, the more frequently it should be selected for sampling. We begin by defining a general measure of influence between two variables.

Influence Let V_i and V_j be two random variables with marginal distributions $\pi(V_i, V_j), \pi(V_i), \pi(V_j)$. Let $f(\pi_1(\mathbf{V}), \pi_2(\mathbf{V})) \mapsto r, r \in \mathbb{R}_+$ be a non-negative real-valued divergence between probability distributions. The influence $\iota(x, y)$ between x and y is

$$\iota(V_i, V_j) := f(\pi(V_i, V_j), \pi(V_i)\pi(V_j)) \quad (4.1)$$

If we take f to be the KL divergence then influence is simply the mutual information. However, since it is more common to assess MCMC convergence with total variation distance, we instead use the total variation influence $\iota_{\text{tv}}(V_i, V_j) := \|\pi(V_i, V_j) - \pi(V_i)\pi(V_j)\|_{\text{tv}}$. We can then construct the selection distribution p to be proportional to this metric

$$p_{\text{tv}}(i) \propto \iota_{\text{tv}}(Y, V_i) \quad (4.2)$$

An interesting property of the total variation influence (between a particular variable and the query) is that it is equal to the error incurred from neglecting to sample that variable.

Proposition 4.3.1 *If $p(i) = \mathbb{1}(i \neq l) \frac{1}{n-1}$ induces an MH kernel that neglects variable x_l , then the expected total variation error ξ_n of the resulting MH sampling procedure under the model is the total variation influence ($\iota_{\text{tv}}(x_l, x_q$) between the query and the ignored variable .*

Proof The resulting chain has stationary distribution $\pi(x_q|x_l = v_l)$. The expected error is:

$$\begin{aligned}
\mathbb{E}_\pi[\xi_{\text{tv}}] &= \sum_{v_l \in \mathcal{V}_l} \pi(x_l=v_l) \|\pi(x_q|x_l=v_l) - \pi^{(t)}(x_q)\|_{\text{tv}} \\
&= \sum_{v_l \in \mathcal{V}_l} \pi(x_l=v_l) \frac{1}{2} \sum_{v_q \in \mathcal{V}_q} |\pi(x_q|x_l=v_l) - \pi^{(t)}(x_q)| \\
&= \frac{1}{2} \sum_{v_l \in \mathcal{V}_l} \sum_{v_q \in \mathcal{V}_q} |\pi(x_q|x_l=v_l)\pi(x_l=v_l) - \pi^{(t)}(x_q)\pi(x_l=v_l)| \\
&= \frac{1}{2} \sum_{v_l \in \mathcal{V}_l} \sum_{v_q \in \mathcal{V}_q} |\pi(x_q, x_l) - \pi^{(t)}(x_q)\pi(x_l)| \\
&= \iota_{\text{tv}}(x_q, x_l)
\end{aligned}$$

■

Since computing $\iota_{\text{tv}}(V_i, V_j)$ is as difficult as inference, we instead use a computationally tractable variant that approximates this value by using only the factors along a path between V_i and V_j .

4.3.2 Convergence rates

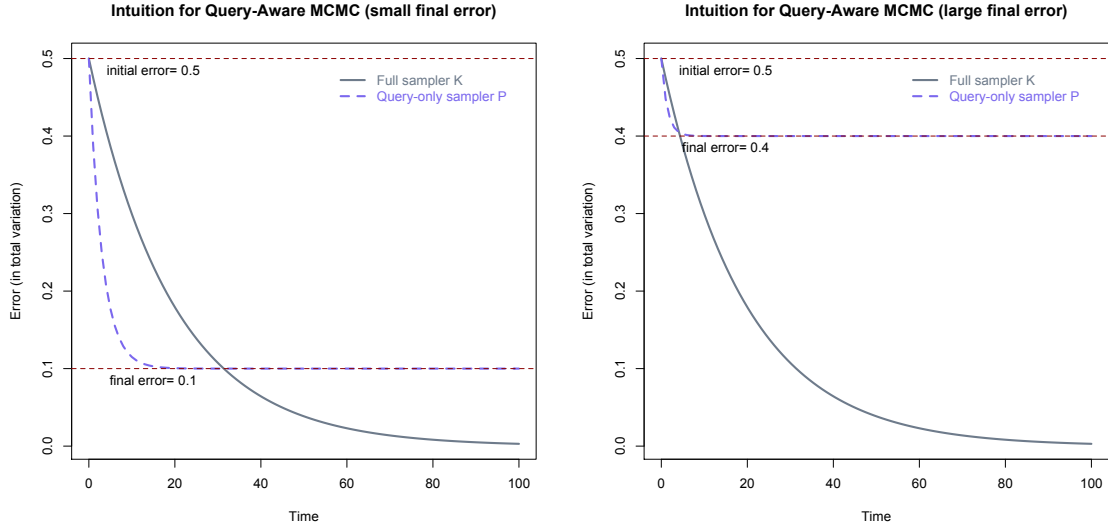
Current conventional wisdom holds that query-aware MCMC is unlikely to perform better than traditional MCMC in the general case. For example, Gil [31] shows that lack of convergence in one dimension implies lack of convergence in all dimensions. Indeed, the marginal distribution of the query variables depends statistically on the other variables in the graph. However, in our setting, time is a scarce commodity; thus neither a query-aware nor conventional chain are likely to truly ‘converge’ to stationarity. This begs the question: which sampler gets closest given the time budget, and under what conditions?

Although we are not yet able to answer this question for our general query-aware MCMC algorithm, we provide insight and analysis for a special case. We are able to show—in the finite time regime, and under certain assumptions on the convergence rates of the chains, the initial distribution over the state space, and the amount of influence between

query and non-query variables—that a query-only MCMC sampler (that is, a sampler that only samples the query variables) can indeed be faster in finite time. Under these conditions, we reveal a fixed point equation for the number time-steps for which query-only MCMC outperforms traditional MCMC. We consider an extreme special case, a query-aware sampler that only samples the query variables and leaves all other variables unchanged (and thus does not converge to the correct joint distribution).

The intuition behind these query-aware MCMC results is that rapid convergence to an incorrect distribution is better than slow convergence to the correct distribution (as long as the error between the true and approximate distribution is not too great). For example, imagine two samplers initialized at some initial distribution π^0 which has some amount of initial error (e.g., the total variation distance) to the target distribution π^* . Suppose that one of these chains converges slowly, but to the correct distribution π^* , while the other chain converges quickly, but to the incorrect distribution $\hat{\pi}$. Which sampler should we choose? Given infinite time, we would of course choose the correct sampler; however, given a fixed time budget, it may be better to choose the incorrect sampler, especially if the error between \hat{p}_i and π is small.

In Figure 4.1 we plot the hypothetical convergence of two such MCMC samplers. The “query-only” sampler is a sampler that only samples the query variables (and thus converges to an incorrect distribution), while the “full” sampler samples all the variables (and thus converges to the correct distribution). If we measure error in terms of convergence in joint distribution, we would expect the plot on the right (large final error) in which the query-only sampler performs uniformly worse the traditional joint sampler; however, if we measure error in terms of convergence in the marginal distribution of a single variable, we might expect the plot on the left (small final error) in which the query-only sampler outperforms the joint sampler for a large number of time steps. Next, we formalize the conditions under which such behavior occurs; our analysis differs from traditional MCMC convergence analysis on the following aspects



(a) Small (final) error regime.

(b) Large (final) error regime.

Figure 4.1: Intuition for why query aware MCMC is likely to perform better than traditional MCMC: fast convergence to the incorrect distribution is often more desirable than slow convergence to the correct distribution. These hypothetical scenarios illustrate the importance of error and convergence rates of the two chains. The figure on the right demonstrates a scenario in which query-aware MCMC would not work well (strong statistical dependencies might give rise to such a case).

- error - our measure of error differs in that we are primarily concerned with approximating the marginal distribution of a subset of the variables rather than the joint distribution of all the variables. Thus, we focus on total variation distance between the true marginal distribution and chains marginal distribution.
- time - we are concerned with error over an initial finite-time window rather than asymptotic convergence.

thus allowing us to demonstrate that query-aware MCMC is viable.

We formalize the intuition from the above paragraph in Proposition 4.3.2

Definition A query-only kernel L corresponds to a sampler that only samples the query variables Y (conditioned on the remaining variables). Let K be a kernel that corresponds to a conventional sampler that samples all variables with equal probability.

Assumption 1 Both L and K are uniformly ergodic:

$$\begin{aligned}\|\pi^0 L^t - \pi_K\|_{tv} &\leq a\gamma_K^t \\ \|\pi^0 K^t - \pi_L\|_{tv} &\leq b\gamma_L^t\end{aligned}$$

Assumption 2 The query aware chain converges much faster to its stationary distribution than the conventional chain in the sense that $\gamma_L \gg \gamma_K$.

Proposition 4.3.2 Let L kernel be the query-only transition kernel defined above. Let K be the conventional transition kernel with invariant distribution $\pi(Y, Z|X)$. If K is slower than L (Assumption 2), then the amount of time for which the query-only sampler is better is given by the fixed point equation (for simplicity $\alpha = \beta = 1$)

$$t = \max\left(\frac{\log(\gamma_l^t + \iota_{lv})}{\log \gamma_k}, 0\right) \quad (4.3)$$

yielding a one-step bound

$$t \geq \frac{\log(\gamma_l + \iota_{lv})}{\log \gamma_k} \quad (4.4)$$

Proof The inequality $\gamma_L^t + \iota_{lv} \leq \gamma_K^t$ lower bounds the amount of time that the query-only kernel (plus the additional error incurred from not sampling the non-query variables) has smaller total error to the stationary distribution than the conventional kernel. We want to know the number of time-steps for which this inequality holds. Solving for t yields the result. ■

Thus, given sufficiently small error, we would expect the query-only kernel to be faster than the conventional kernel for a positive number of time-steps. The size of the error and validity of our assumptions is largely problem-dependent. We discuss these assumptions in the remarks below, and then empirically compare the convergence rates of various query-aware samplers on both synthetic and real-world data.

Remarks:

- *Uniform ergodicity (Assumption 1).* Convergence of such nature is surprisingly commonplace, especially for chains over discrete, finite state spaces. Established examples include the independence Metropolis Hastings sampler (IMHA) [78], slice samplers [64], and several variations of the discrete-space Gibbs sampler [40].
- *Convergence rates.* Tools such as the drift and minorization conditions allow us to establish the relative rates of the two samplers [78].
- *The slowness of the conventional sampler (Assumption 2).* Our analysis in this section assumes that the conventional sampler is slow to converge, but there are relatively few tools for lower bounding convergence in the literature. Although this is not true in general, we conjecture that—under most normal conditions—as the number of variables increase, the convergence rate decreases. The drift and minorization frameworks provide some intuition for why this might be the case. The idea of drift and minorization is that if we can identify a subset of the state space C (with a minorization constant ϵ which governs the convergence rate) to which the chain tends to frequently drift, we can then use (1) the rate at which the chain drifts to C and (2) the minorization constant constant of C to establish an upper bound on convergence. However, it is not possible for the conventional sampler to have a set C that is larger than the number of variables and also have a non-trivial minorization constant (i.e., $\epsilon > 0$); it is unlikely that a drift constant that implies rapid convergence would exist for such a small subset of the state space. If we were to instead consider the full-sweep variant of the conventional sampler (in which we sample every variable in each time step) then it would indeed be possible to establish a non-trivial minorization constant. However, the reduction from the single-variable variant to full-sweep variant scales with the number of variables in the model (i.e., via n applications of the single-variable kernel). Note that in auxiliary variable methods, adding more

variables could increase the convergence rate, but in our case, the variables are not specially designed to have such effect on convergence (the number of variables increase with the amount of data, and are not artificially introduced with the specific intention of increasing the convergence rate).

- *Error of the query-only sampler.* *Prima facie*, the fact that the query-only sampler ignores all the variables except the query variables appears to be an egregious approximation. However, it can be shown that as long as there are no deterministic dependencies in the model, the mutual information between any two variables decays exponentially fast in the topological distance in the graphical model [10]. Thus, for many types of graphical models (such as chains and grids), most variables actually do not have much of an effect on the query variable; consequently, we expect that the error of the query-only sampler would not be too substantial.

4.4 Experiments

4.4.1 Synthetic experiments

In this section we evaluate our query-aware MCMC on six graphical models with diverse structure ranging from models in which the variables are independent (fully factorized) to models in which all the variables are completely dependent through a single factor (fully joint). For this experiment, we limit ourselves to models with sixteen binary variables so that we can (1) compute the full marginal and joint distributions of each model and (2) construct the exact transition kernels for each MCMC sampler (each cell in the matrix is given by Equation 2.8). The former allows us to exactly compute the ground-truth from which we measure the total variation distance, and the latter allows us to exactly compute the intermediate distributions at each step of MCMC.

Given an initial distribution π_0 , we can obtain the next distribution π_1 (via a single step of MCMC) by multiplying an MCMC transition matrix K by the vector representation of this distribution $\pi_1 = K\pi_0$. Repeatedly multiplying the transition kernel yields a

sequence of intermediate distributions $\{\pi_i \mid \pi_i = K\pi_{i-1}\}$ from which we can measure the error of each step of MCMC. For our experiments, we set the initial distribution π_0 to be uniform (each state has equal probability), and construct the transition matrices for each of our samplers.

We evaluate four different query-aware samplers based on various choices for the variable selection distribution p , including topological distance (*QAM-PolyI*), exact mutual information (*QAM-MI*), exact total variational distance (*QAM-TV*), and approximate total variational distance (*QAM-TV-g*) and compare them to two baselines (a traditional MH sampler that selects variables for sampling with uniform probability (*uniform*), and a query-only MH sampler that only samples the query variables (*qo*), on six different graphical models with parameters generated from a beta(2,2).¹ distribution (shown in Figure 4.2). These models range from a fully-independent model in which the variables are statistically independent, to a fully-joint model in which all the variables are statistically dependent via a single factor. We show the rate of convergence of the various samplers to the distribution of the query variables in Figure 4.3, and also to the full joint distribution in Figure 4.4. Note that the x-axis for these figures is time and that a single unit of time is taken to be a single application of the full transition kernels (to the previous time’s distribution).

In order to assist in interpretation of the results we make the following remarks

- For each model, the factors that govern the dependence between the variables have shared parameters (that is, these factor functions are identical to each other), but each variable by default has a unique randomly generated factor. This set-up best resembles how models are used on real-world problems (observations differ, but parameters are shared).

¹this helps ensure an interesting dynamic range over the state space, while limiting the amount of determinism in the model (e.g., in contrast to parameters generated from a uniform distribution).

- On the fully independent model the query variable is completely independent from the other variables; thus, all the query-aware methods are equivalent and exact on this model.
- The linear-chain, hoop, and grid models, are the only models for which the topological distance is non-trivial (because on the fully independent, pairwise, and jully joint models all the nodes have the same topological distance).
- The dynamic range of (and thus, the amount of determinism in) each model varies widely depending on the number of factors. For example, the pairwise model has a quadratic number of factors and tends to have a higher dynamic range (and thus more determinism) than the fully-connected joint model which has only a single factor. This is an important remark because determinism is known to negatively impact the converge rate of MCMC algorithms.
- The topological distance heuristics (Poly1 and Poly2) prioritizes a variable inversely proportional to its topological distance from the query (the number indicates the degree of the polynomial). In contrast, the variable dependence-based strategies (e.g., measured by mutual information) decays priorities at an exponential rate [10]. As a result, the mutual information based method behaves similar to the query-only sampler on certain models.

With these observation in mind, we now discuss the results. First, notice that the empirical results reveal behavior that corroborates Proposition 4.3.2: initially, the query-only sampler (and the other query-aware samplers) converges more rapidly to the stationary distribution than the traditional MCMC sampler, but after a certain point in time, the traditional sampler becomes faster. Such results indicate that the length of the time budget should play a role in determining which sampler to choose. Further, these results suggest that a hybrid algorithm, that initially applies a query-aware sampler, but transitions to a traditional sampler, could be even faster (for example, adding a temperature parameter to the variable priori-

ties would allow annealing between the query aware sampler (lower temperatures) and the uniform sampler (higher temperatures)).

Next, we compare the query-aware samplers to the traditional sampler on models with different numbers of random variables. In this experiment, we focus on the hoop model and vary the number of variables between 4 and 20; we compare the samplers on each of these models and display the results in Figure 4.5. We would expect that as the number of variables increase, the advantage of the query aware samplers (over the traditional sampler) would also increase. The reason is that as the number of variables increase, the number of variables that do not influence the query also increase, and the traditional sampler will waste an increasing amount of time sampling them. Indeed, our experiment shows that this is the case (for hoops). Note, that for the fully-connected model, the effect might not be as dramatic since every variable has topological distance of one to the query.

4.4.2 Real-world data

In this section, we evaluate query-aware MCMC on real-world data (named entity recognition (NER)). Given a collection of sentences, NER is the task of predicting which the entity class to which each word in the sentence belongs (or "OTHER") if the word is not an entity. For example, given labels in the set "PER", "ORG", "LOC", "MISC", "OTHER", assign a label to each word in the sentence

Barack Obama returned to the White House on Monday.

The correct assignment of types to tokens would be: Barack=PER, Obama=PER, returned=OTHER, to=OTHER, the=OTHER, White=LOC, House=LOC, on=OTHER, Monday=MISC.

We investigate two models: a linear-chain, and a skip chain. In the case of the linear-chain we can compute the query marginals exactly using the forward-backward algorithm. In the case of skip-chain, we compute approximate the query-marginals via sampling. The linear-chain system is a model for which we would expect our assumptions to hold; however, it is not immediately clear that the assumptions would hold on the skip-chain model

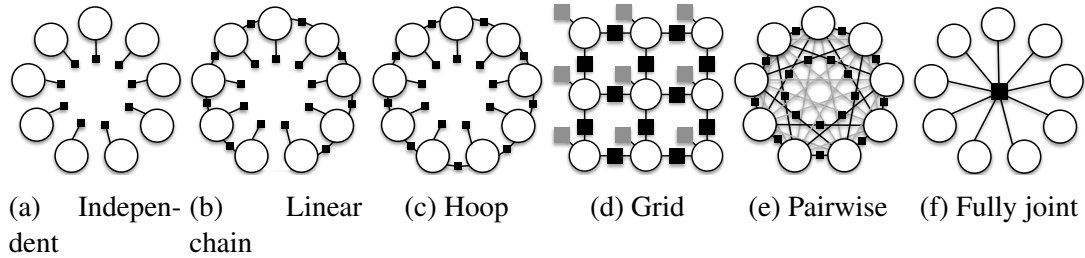


Figure 4.2: Graphical models for evaluating QAM

(there is no bound on the number of edges per variable because there can be large number of edges for frequently occurring words).

4.5 Conclusion

In this chapter we proposed a query-aware MCMC sampler for answering probabilistic queries on graphical models. Since the query-aware setting deviates substantially from traditional machine learning and statistics problems—in which the variables are equally important—we provided analysis and discussion on the conditions under which such a sampler is superior to a conventional MCMC sampler, and empirically validated these claims on both synthetic and real-world data. We conclude that query-aware MCMC is a promising approach to answering user’s probabilistic queries on uncertain knowledge bases. Although theoretical results are currently limited, we hope that advances in finite-time analysis of Markov chains and lower bounds on convergence rates will allow us to provide more thorough analysis in future work.

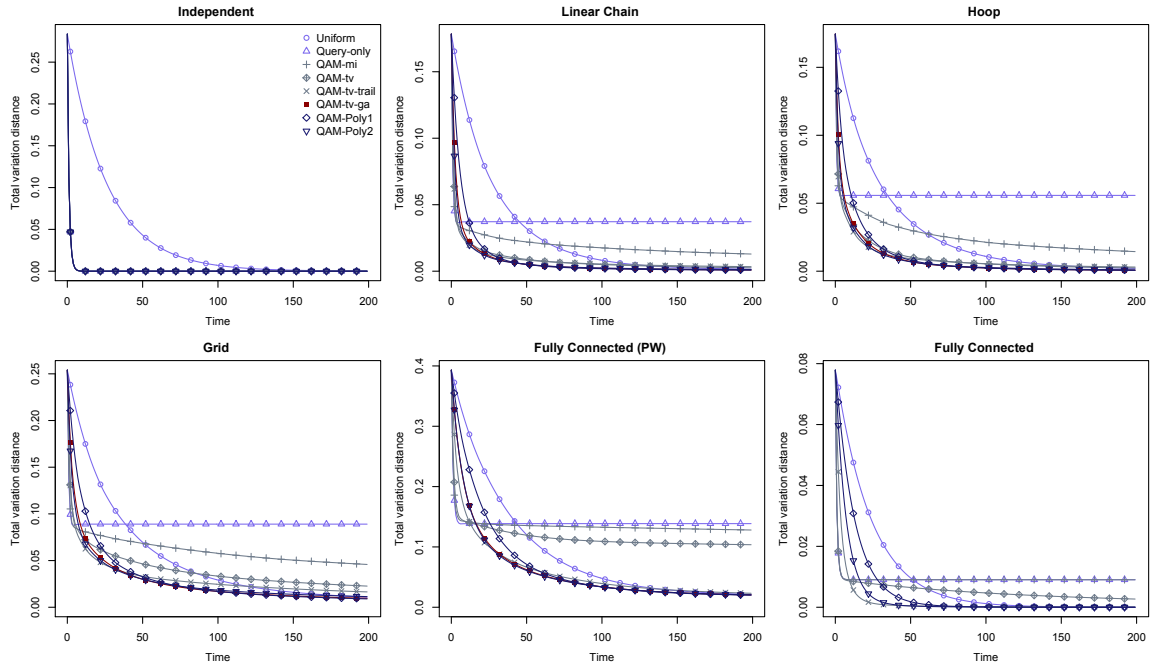


Figure 4.3: Convergence to the query marginals of the stationary distribution from an initial uniform distribution.

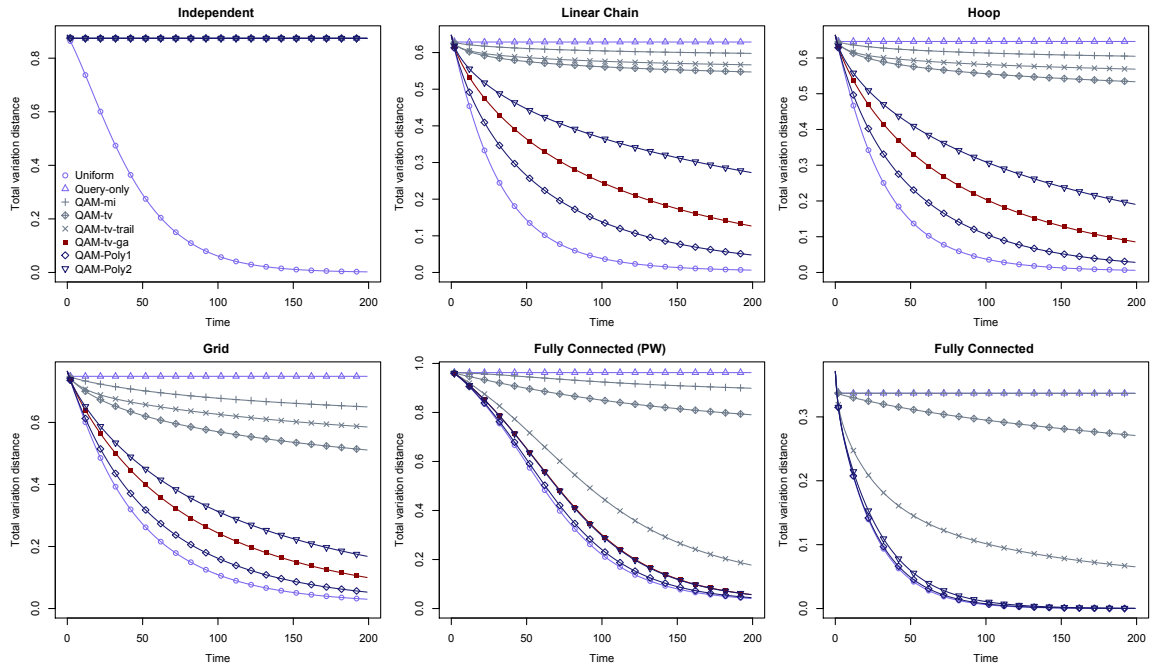


Figure 4.4: Convergence to the full joint stationary distribution from an initial uniform distribution.

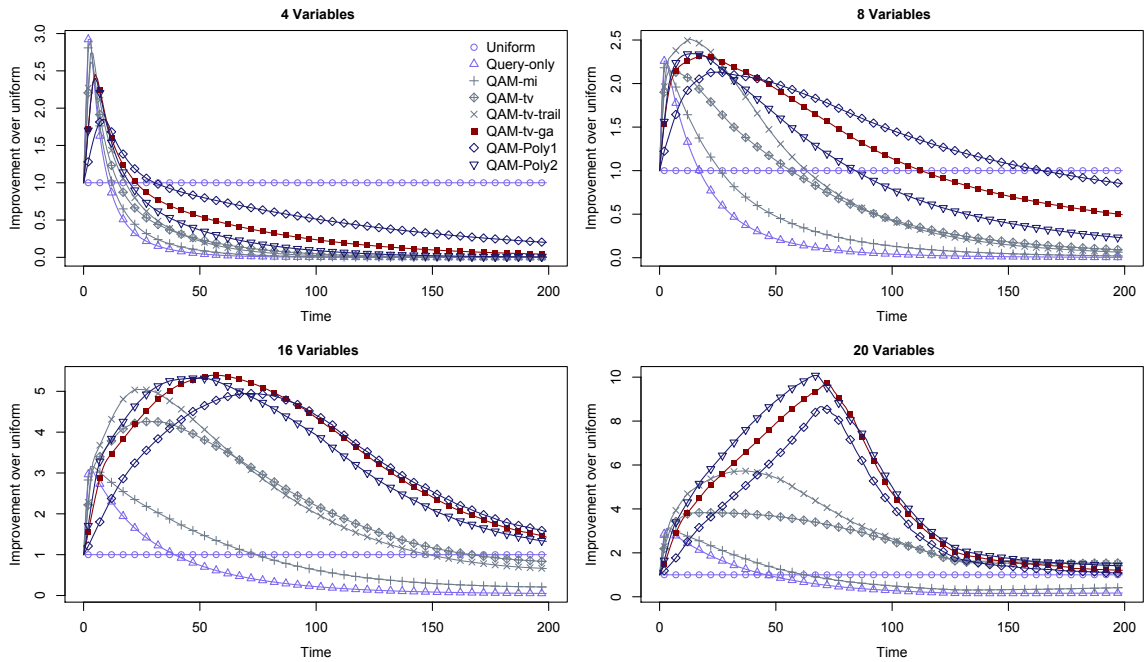
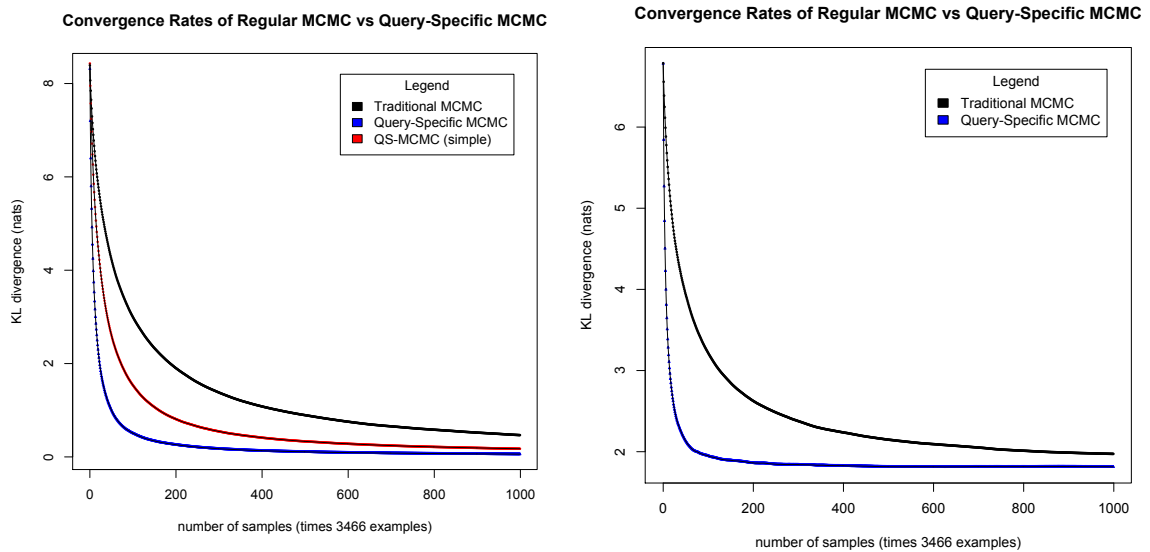


Figure 4.5: Improvement over uniform p as the number of variables increases. Above the line $x = 0$ is an improvement in marginal convergence, and below is worse than the baseline. As number of variables increase, the improvements of the query specific techniques increase.



(a) Convergence to query marginal in linear chain NER. (b) Convergence to query marginal in skip-chain NER.

Figure 4.6: Query-aware sampling on two NER models. Linear-chain model (left) and skip-chain model (right).

CHAPTER 5

PARAMETER ESTIMATION WITH SAMPLERANK

5.1 Overview and related work

Over the past decade, the models required for state-of-the-art structured prediction accuracy in natural language processing and information extraction have become increasingly complex, in order to capture rich problem-specific dependencies. Initially, models such as linear chain CRFs [48], projective first-order dependency parsers, and classifiers [93] were considered suitable for solving tasks such as named entity recognition, parsing, and coreference. These models were popular because their simple structures allow exact learning and inference to be both tractable and efficient. However, the need to include increasingly complex sets of features combined with the desire to perform joint inference across multiple tasks [86, 68, 27], has led to a proliferation in new models for which exact inference and learning are no longer tractable. Examples of such models include skip-chain CRFs for named entity recognition [95], parsers with higher-order dependencies [17, 123], and partition-wise models of coreference [21].

Such models are critical for building the extraction and integration components of probabilistic KBs, but traditional learning algorithms are insufficient because they require expensive inference procedures as subroutines (e.g., marginals are required for maximum likelihood gradients, and MAP inference is required in margin methods to identify constraint violations). Although approximations exist, it has been shown that the use of approximate inference during learning can often lead to surprisingly poor parameter estimates [47, 28]. Additionally, many structured prediction models for extraction/integration contain (1) a high number of interdependent variables, (2) variables with exponentially large

domains, (3) tens-of-millions of parameters [95, 21]. These characteristics make even approximate message passing algorithms such as loopy belief propagation intractable; thus, MCMC has become the *sine qua non* for achieving practical inference in these models.

Contrastive divergence (CD) is a viable approach to parameter estimation with MCMC that has been successfully applied to both Markov random fields [121] and deep belief networks [36, 101]. CD computes inexpensive gradients between the ground-truth and samples along an MCMC chain yielding a stochastic approximation algorithm with asymptotic convergence to the maximum likelihood estimate (MLE). Unfortunately, CD is known to be sensitive to the shape of the underlying probability distribution—making it difficult to apply in certain situations—requiring the support of advanced MCMC procedures [80].

Furthermore, contrastive divergence approximates maximum likelihood, which is not always the best choice for structured prediction problems which are commonly assessed with domain specific evaluation metrics such as F1 or BLEU score. In particular, recent work has articulated the importance of incorporating these rich signals into the learning objectives because they yield better performance [100, 83, 39, 52]. Unfortunately, many of these approaches depend on loss-augmented decoding, which not only limits the class of evaluation metrics, but also limits scalability to complex models.

In this chapter we present a stochastic learning algorithm, SampleRank [21, 19, 110], for rapid parameter estimation in large graphical models with rich evaluation metrics. Rather than performing a multi-step inference routine between parameter updates, SampleRank embeds parameter updates within each step of MCMC inference. In this aspect, SampleRank appears similar to CD. However, to make better use of each generated sample, the learning objective enforces *ranking constraints* between neighboring configurations encountered during inference. That is, rather than simply optimizing the score of the true configuration, the learning objective encourages the proper ranking of pairs of (possibly) incorrect configurations according to any user-provided loss function. Not only do these

additional constraints improve the quality of the model, but they also provide computational efficiency over CD because the corresponding parameter updates tend to be sparser.

The primary use-case for SampleRank is for learning the weights of discriminative structured prediction models in a supervised setting. In the first part of this chapter, we describe this variant of SampleRank and report results on four such structured prediction models: a linear chain CRF, a fully connected pairwise CRF for multi-label classification, a partition-wise CRF for within-doc coreference resolution, and the hierarchical model for coreference resolution. In the second part of this chapter, we describe a variant of SampleRank that is able to train restricted Boltzmann machines (RBMs). Unlike the structured prediction models, RBMs are generative latent variable models that are learned in an unsupervised fashion.

In this chapter, we present the following contributions

- We propose the SampleRank algorithm for estimating parameters in models for which exact inference is intractable. SampleRank learns weights by ranking consecutive states in an MCMC chain.
- We derive the objective function that the SampleRank algorithm minimizes, and show that it is a generalized version of the primal optimization problem for SVM [110]. In initial work [21, 19], we only understood the mechanics of a general SampleRank procedure and not the mathematical objective of a specific variant.
- We show that the objective function minimizes an upper bound on the empirical training risk.
- Using the insight that the primal SVM objective is a special case of the SampleRank objective function, we propose a variant of SampleRank that optimizes the SVM objective by using MCMC to approximately search for the state that maximizes the hinge-loss.

- We reveal a connection between SampleRank and temporal difference methods from reinforcement learning, allowing us to incorporate delayed reward into the MCMC training of graphical models.
- We provide empirical evidence that SampleRank learns better discriminative structured prediction models than other MCMC-based training algorithms such as contrastive divergence.
- We show that SampleRank performs competitively with exact training algorithms on models for which exact training is tractable.
- We show that SampleRank can be used to train unsupervised generative models with latent variables such as restricted Boltzmann machines.
- We show that SampleRank is competitive with (and in some cases superior to) contrastive divergence and persistent contrastive divergence for training RBMs.
- We perform system empirical studies upon which we are able to recommend specific algorithmic parameters for successfully training RBMs with SampleRank.

5.2 SampleRank

In this section we describe our MCMC-based SampleRank algorithm [110], which adjusts parameters at each MCMC step such that the ranking of possible worlds according to the model matches the ranking according to a user-defined loss. For example, in coreference resolution, we can evaluate the quality of a predicted coreference hypothesis using a measure such as F1 accuracy. SampleRank estimates parameters by running a Markov chain on the space of possible coreference predictions. Each sample from the chain produces a pair of coreference predictions $\mathbf{y}^{(t)}, \mathbf{y}^{(t+1)}$. We can score and rank these two configurations with both the F1 accuracy and the model. If the model disagrees with the ranking of the evaluation metric (F1) accuracy, then the parameters are updated to correct this error.

We identify and formalize the objective function for the SampleRank algorithm by generalizing the structured SVM loss, and show that SampleRank is a stochastic approximation (i.e., a Robbins-Monro algorithm [75]) method on this objective. We have also shown that like SVM, SampleRank minimizes an upper bound on the empirical risk of making an incorrect prediction on the training set. More details can be found in Wick et al. [110].

Informally, the structured SVM objective function minimizes margin violations between the ground-truth configuration and the remaining configurations. The intuition behind SampleRank is that these types of ground-truth constraints—that exist in SVM and other machine learning methods—can be deconstructed into atomic constraints between neighbors on a local search space. We use the term “atomic” in the sense that (1) the constraint occurs between the atomic steps of search requiring no inference and (2) the constraints serve as the basic building blocks for larger constraints of interest.

For example, consider the type of constraints satisfied by an SVM under separability. Let $\mathbf{y} \in \mathcal{Y}(\mathbf{x})$ be an arbitrary configuration such that $\omega(\mathbf{y}) < \omega(\mathbf{y}_x^*)$. Assuming separability, structured SVM satisfies the constraint $\boldsymbol{\theta}'\phi(\mathbf{y}_x^*) - \boldsymbol{\theta}'\phi(\mathbf{y}) \geq \omega(\mathbf{y}_x^*) - \omega(\mathbf{y})$. Now let us assume there exists a path $\mathbf{y}^{(0)}, \mathbf{y}^{(1)}, \dots, \mathbf{y}^{(p)}$ on a local search space from $\mathbf{y}=\mathbf{y}^{(0)}$ to $\mathbf{y}_x^*=\mathbf{y}^{(p)}$ of length p such that the path is monotonic in ω : $\omega(\mathbf{y}^{(i)}) < \omega(\mathbf{y}^{(i+1)})$. Assume that we are able to satisfy the local constraints along the path: $\{\boldsymbol{\theta}'\phi(\mathbf{y}^{(i)}) - \boldsymbol{\theta}'\phi(\mathbf{y}^{(i-1)}) \geq \omega(\mathbf{y}^{(i)}) - \omega(\mathbf{y}^{(i-1)})\}_1^p$. Then, we also satisfy the SVM constraint: $\boldsymbol{\theta}'\phi(\mathbf{y}_x^*) - \boldsymbol{\theta}'\phi(\mathbf{y}) \geq \omega(\mathbf{y}_x^*) - \omega(\mathbf{y})$. Note however, that the converse is not necessarily true: satisfying the SVM constraints does not guarantee that each of the pairwise constraints are satisfied. Thus, SampleRank is not equivalent to SVM.

The idea that these ground-truth constraints can be distributed across a local search space is the underlying intuition for how we can achieve rapid learning with inference-free gradients. However, our primary goal is not to simply replicate structured SVM; rather we take a more general approach based on a new family of objective functions that can potentially result in higher quality models.

5.2.1 Pairwise Objectives

As stated previously, SVM minimizes margin violations between each incorrect configuration and the ground-truth. Consider instead a larger family of objective functions that minimize margin violations between arbitrary configuration pairs. We can obtain such objectives by replacing the maximization ξ_{svm} over ground-truth violations in SVM (Equation 2.12), with a maximization over configuration pairs $\mathcal{P}(x)$ determined by ω :

$$\xi_{\text{sr}}(x) = \max_{\langle \mathbf{y}_i, \mathbf{y}_j \rangle \in \mathcal{P}(x)} [\Delta(\mathbf{y}_i, \mathbf{y}_j) - \boldsymbol{\theta}'\phi(\mathbf{y}^+) + \boldsymbol{\theta}'\phi(\mathbf{y}^-)]_+ \quad (5.1)$$

We allow $\mathcal{P}(\mathbf{x})$ to be any subset of $\mathcal{Y}(\mathbf{x}) \times \mathcal{Y}(\mathbf{x})$ subject to $\omega(\mathbf{y}_i) \neq \omega(\mathbf{y}_j) \forall \langle \mathbf{y}_i, \mathbf{y}_j \rangle \in \mathcal{P}(\mathbf{x})$. Note that these new objective functions preserve the structured SVM property of upper bounding the empirical training risk defined in Equation 2.11. We can state this more precisely as follows:

Proposition 5.2.1 *Let $\mathcal{P}_{\text{svm}}(\mathbf{x}) = \{\langle \mathbf{y}_x^*, \mathbf{y} \rangle \mid \mathbf{y} \in \mathcal{Y}(\mathbf{x}) \setminus \mathbf{y}_x^*\}$. If $\mathcal{P}(\mathbf{x}) \supseteq \mathcal{P}_{\text{svm}}(\mathbf{x}) \forall \mathbf{x} \in \mathcal{D}$ then the SampleRank objective upper bounds the empirical risk.*

Proof This follows directly from the fact that $\mathcal{P}(\mathbf{x})$ is a superset of $\mathcal{P}_{\text{svm}}(\mathbf{x})$. Let $\ell(\mathbf{y}_i, \mathbf{y}_j; \boldsymbol{\theta}) := [\Delta(\mathbf{y}_i, \mathbf{y}_j) - \boldsymbol{\theta}'\phi(\mathbf{y}^+) + \boldsymbol{\theta}'\phi(\mathbf{y}^-)]_+$:

$$\begin{aligned} \xi_{\text{sr}} &= \max_{\mathcal{P}(\mathbf{x})} \ell(\mathbf{y}_i, \mathbf{y}_j; \boldsymbol{\theta}) \\ &= \max\left\{ \max_{\mathcal{P}_{\text{svm}}(\mathbf{x})} \ell(\mathbf{y}_i, \mathbf{y}_j; \boldsymbol{\theta}), \max_{\mathcal{P}(\mathbf{x}) \setminus \mathcal{P}_{\text{svm}}(\mathbf{x})} \ell(\mathbf{y}_i, \mathbf{y}_j; \boldsymbol{\theta}) \right\} \\ &= \max\left\{ \xi_{\text{svm}}(\mathbf{y}_x^*, \hat{\mathbf{y}}_x), \max_{\mathcal{P}(\mathbf{x}) \setminus \mathcal{P}_{\text{svm}}(\mathbf{x})} \ell(\mathbf{y}_i, \mathbf{y}_j; \boldsymbol{\theta}) \right\} \\ &\geq \xi_{\text{svm}}(\mathbf{y}_x^*, \hat{\mathbf{y}}_x) \end{aligned}$$

We have that $\xi_{\text{sr}} \geq \xi_{\text{svm}}(\mathbf{y}^*, \hat{\mathbf{y}}) \geq \Delta(\mathbf{y}^*, \hat{\mathbf{y}}) \forall \mathbf{x} \in \mathcal{D} \therefore \frac{1}{n} \sum_{\mathbf{x} \in \mathcal{D}} \xi_{\text{sr}} \geq \frac{1}{n} \sum_{\mathbf{x} \in \mathcal{D}} \Delta(\mathbf{y}_x^*, \hat{\mathbf{y}}_x)$ ■

Corollary 5.2.2 *If there exists an ω -monotonic path for all $\mathbf{y} \in \mathcal{Y}(\mathbf{x}) \forall \mathbf{x} \in \mathcal{D}$ then SampleRank (Algorithm 6) upper bounds the empirical risk.*

Corollary 5.2.3 *If ω is the Hamming loss and the MCMC transition function is irreducible, then SampleRank upper bounds the empirical risk.*

Note that this family of objective functions allow the specification of explicit constraints between partially correct configurations. We conjecture that objectives that include these additional constraints provide better generalization than objectives that only include constraints involving the ground-truth. The reason is that structured prediction models inevitably commit errors; that is, by assigning the incorrect labels to some of the variables. Intuitively, if the model learns in the presence of errors at train-time, then if inference commits an error at test-time, inference is better positioned to infer the correct assignment to the remaining variables.

In order to derive a stochastic approximation algorithm, we first reformulate SampleRank’s pairwise objective as a saddle point optimization problem:

$$\min_{\boldsymbol{\theta}} \max_{\langle \mathbf{y}_i, \mathbf{y}_j \rangle} \sum_{\mathbf{x} \in \mathcal{D}} [\Delta(\mathbf{y}_i(x), \mathbf{y}_j(x)) - \boldsymbol{\theta}' \phi(\mathbf{y}_x^+) + \boldsymbol{\theta}' \phi(\mathbf{y}_x^-)]_+ \quad (5.2)$$

where $\langle \mathbf{y}_i, \mathbf{y}_j \rangle = \langle \langle \mathbf{y}_i(x_1), \mathbf{y}_j(x_1) \rangle, \dots, \langle \mathbf{y}_i(x_n), \mathbf{y}_j(x_n) \rangle \rangle$ is a vector of configuration pairs (one pair per instance) such that each component $\langle \mathbf{y}_i(x_k), \mathbf{y}_j(x_k) \rangle \in \mathcal{P}(x_k)$; that is, the max is a vector of all the per-instance maxima.

5.2.2 Optimization

Obtaining the exact solution to Equation 5.2 is in general intractable due to the combinatorial maximization over each $\mathcal{P}(x)$. Fortunately, there are known stochastic approximation procedures (i.e., Robbins-Monro) for finding saddle point solutions of the form $\langle a^*, b^* \rangle = \min_{a \in \mathcal{A}} \max_{b \in \mathcal{B}} f(a, b)$ where we only have access to noisy estimates of the function $f(a, b)$ and its partial derivatives $\zeta_a(a, b) \cong \frac{\partial}{\partial a} f(a, b)$ and $\zeta_b(a, b) \cong \frac{\partial}{\partial b} f(a, b)$ for a given point $(a, b) \in \mathcal{A} \times \mathcal{B}$. If $\Pi_{\mathcal{A}} : \star \rightarrow \mathcal{A}$ and $\Pi_{\mathcal{B}} : \star \rightarrow \mathcal{B}$ project their arguments onto the convex sets \mathcal{A} and \mathcal{B} respectively, then according to Nemirovski and

Rubinstein [65], the saddle point solution can be found by beginning with a feasible point $(a_0, b_0) \in \mathcal{A} \times \mathcal{B}$ and iterating the following update rules:

$$a_t = \Pi_{\mathcal{A}} [a_{t-1} - \eta_t \zeta_a(a, b_{t-1})] \quad (5.3)$$

$$b_t = \Pi_{\mathcal{B}} [b_{t-1} + \eta_t \zeta_b(a_{t-1}, b)] \quad (5.4)$$

The final solution $(a^*, b^*) \in \mathcal{A} \times \mathcal{B}$ is given as $a^* = \frac{1}{T} \sum_{t=0}^T a_t$, $b^* = \frac{1}{T} \sum_{t=0}^T b_t$. Under relatively mild conditions, the stochastic approximation saddle point (SASP) algorithm converges. In the next section we introduce the SampleRank algorithm and its relation to SASP.

5.2.3 SampleRank Algorithm

We first describe a general family of SampleRank algorithms for learning pairwise objectives functions such as Equation 5.2, and then identify the specific MCMC variant of SampleRank that we advocate in this section. The general SampleRank method constructs a sequence of configuration pairs $(y_0, y_1), (y_2, y_3), \dots$ from a mechanism \mathcal{G} defined over the set $\mathcal{P}(\mathcal{D}) = \bigcup_{x \in \mathcal{D}} \mathcal{P}(x)$. For any pair in the sequence, define $y^+ = \operatorname{argmax}_{y \in (y_i, y_{i+1})} \omega(y)$ and $y^- = \operatorname{argmin}_{y \in (y_i, y_{i+1})} \omega(y)$; if $\boldsymbol{\theta}'(\phi(y^+) - \phi(y^-)) - \Delta(y_i, y_j) < 0$ then the weights are corrected: $\boldsymbol{\theta}_{t+1} \leftarrow \boldsymbol{\theta}_t + \eta_t(\phi(y^+, x) - \phi(y^-, x))$, where η_t is the learning rate at time t . After T time-steps, SampleRank estimates the parameters with the average weight vector: $\frac{1}{T} \sum_{t=1}^T \boldsymbol{\theta}_t$. Under separability, and assuming the random mechanism can return all configurations with positive probability, then this general form of SampleRank can be shown to converge [77]. Furthermore, since updates are performed in isolation using individual configuration pairs, SampleRank can be effortlessly parallelized.

The general form of SampleRank alternates between producing a configuration pair (y_i, y_{i+1}) (resp. maximizing Equation 5.2 w.r.t. (y_i, y_j)) and updating a weight vector $\boldsymbol{\theta}$ (resp. minimizing Equation 5.2 w.r.t. $\boldsymbol{\theta}$). In order to produce an algorithm that is both practical and simple for a wide variety of problems, we advocate a specific variant of Sam-

Algorithm 6 SampleRank with MCMC

```
1: Inputs:  
    $q : \mathcal{Y} \rightarrow \mathcal{Y}$ : proposer (MCMC transition kernel)  
    $\omega : \mathcal{Y} \rightarrow \mathbb{R}$ : performance metric (e.g., F1)  
    $\mathcal{D}$ : the training set  
2: Output:  $\frac{1}{T} \sum_{t=1}^T \theta_t$   
3: Initialization:  $\theta_0 \leftarrow \mathbf{0}$   
4: for  $x \in \mathcal{D}$  do  
5:    $y_0$ : initial configuration in  $\mathcal{Y}(x)$   
6:   for  $t = 0, 1, 2, 3, \dots$  #samples do  
7:     Attempt an MCMC walkstep (or local search move):  
        $y_{t+1} \leftarrow q(\cdot|y_t)$   
8:     Let:  
        $y^+ = \operatorname{argmax}_{y \in \{y_t, y_{t+1}\}} \omega(y)$  and  
        $y^- = \operatorname{argmin}_{y \in \{y_t, y_{t+1}\}} \omega(y)$  and  
        $\hat{\nabla} = \phi(y^+) - \phi(y^-)$  (use Equation 5.6)  
9:     if  $\theta' \hat{\nabla} < \omega(y^+) - \omega(y^-)$  and  $\omega(y_t) \neq \omega(y_{t+1})$  then  
10:       $\theta_{t+1} = \theta_t + \eta_t \hat{\nabla}$   
11:     end if  
12:     if  $(-\text{accept}(y_{t+1}, y_t, \theta))$  then  $y_{t+1} \leftarrow y_t$   
13:   end for  
14: end for
```

pleRank in Algorithm 6 that harnesses MCMC. Lines 4-14 (for simplicity only a single epoch is shown) iterate over the dataset, and then for each instance, an MCMC chain is run for a predetermined number of steps. Lines 7 generates a local-search move (resp. a subgradient maximization step corresponding to SASP update Equation 5.4), and then lines 9-12 perform a minimization with respect to the parameters (resp. SASP minimization Equation 5.3). The conditional in Line 9 is for the hinge-loss and enforces the property that $\langle y_i, y_j \rangle \in \mathcal{P}(x) \implies \omega(y_i) \neq \omega(y_j)$. This property simply states that the model should be agnostic to the relative ordering of y_i, y_j when ω has no preference for one or the other.

5.2.4 Sample Complexity

One concern with the SampleRank objective function is the exponential number of constraints; indeed the objective can be up to quadratic in the SVM objective which is already $O(|\mathcal{Y}(x)|)$. In this section we show that a high quality model can be learned from

a polynomial number of samples using analysis from randomized constraint sampling [26] and sampled convex programs [11].

The idea behind these bounds is to construct a relaxed optimization problem by sampling a manageable set of constraints from a full optimization problem with a distribution ρ . The solution to the full problem is approximated by solving the relaxed sampled problem. Faris and Roy [26] have shown for a problem with k variables, and N sample constraints, where

$$N = O\left(\frac{1}{\epsilon} \left(K \ln \frac{1}{\epsilon} + \ln \frac{1}{\delta}\right)\right) \quad (5.5)$$

that any optimal solution to the relaxed problem with a probability at least $(1 - \delta)$ violates a set of constraints \mathcal{V} with measure $\rho(\mathcal{V}) \leq \epsilon$, where $\rho(\cdot)$ is a probability distribution over the constraint space from which *i.i.d.* sample constraints are generated.

We show that SampleRank can be described as the following SCP [76]:

$$\begin{cases} \min \boldsymbol{\theta}^T \boldsymbol{\theta} \\ \text{s.t. } \Delta(y^+, y^-) + \boldsymbol{\theta}' (\phi(y^-, x) - \phi(y^+, x)) \leq 0, \forall \mathcal{P}_\rho(\mathcal{D}) \end{cases}$$

Where $\mathcal{P}_\rho(\mathcal{D})$ is a set of constraints derived by sampling configuration pairs from $\mathcal{P}(\mathcal{D})$ with distribution ρ . Taking $K = |\boldsymbol{\theta}|$ reveals that a good quality model can be learned from a polynomial number of samples.

5.2.5 Implementation Details for Efficient Learning

In this section we discuss implementation optimizations that are specific to SampleRank. Not only is SampleRank a stochastic gradient algorithm, but the gradients are especially sparse because they are computed from the atomic steps of inference. First we discuss an implementation optimization based on the locality of MCMC and factor graphs, and then we discuss how to exploit the sparsity of the gradients to efficiently implement parameter averaging and L2 regularization.

5.2.5.1 Exploit Locality of MCMC

When implementing SampleRank, we simultaneously exploit the factorization of the graphical model and the “diff”-structure of local search to circumvent exhaustive gradient computations. That is, we can avoid having to fully evaluate $\phi(y^+)$ and $\phi(y^-)$ to compute $\hat{\nabla}$, and similarly avoid having to fully evaluate $\omega(y^+)$ and $\omega(y^-)$ to compute Δ . Since y^+ and y^- differ by just a single step of local search, they only disagree on a small handful of variable assignments. If we take δ to be the set of variables that were changed by the atomic local search step, let δ^+ be the assignment to those variables in y^+ , δ^- be the assignment to those variables in y^- , let $\mathcal{N}(\delta^+)$ be the function that enumerates the set of factors neighboring δ^+ and let $\mathcal{N}(\delta^-)$ be similarly defined, then we efficiently compute $\hat{\nabla}$ with:

$$\hat{\nabla} = \sum_{\psi^r \in \mathcal{N}(\delta^+)} \phi^r(y_j^r) - \sum_{\psi^r \in \mathcal{N}(\delta^-)} \phi^r(y_i^r) \quad (5.6)$$

In addition, by decomposing ω into a product of factors, an analogous evaluation of $\Delta(y_{t+1}, y_t)$ is possible.

We can now perform the computations in Algorithm 6 more efficiently as follows. In Line 8 compute the quantities $\hat{\nabla}$ and $\Delta(y_{t+1}, y_t) = |\omega(y_{t+1}) - \omega(y_t)|$ using the technique exemplified in Equation 5.6. Next compute y^+ and y^- by checking the signum of $\Delta(y_{t+1}, y_t)$: if $\Delta(y_{t+1}, y_t) \geq 0$ set $y^+ \leftarrow y_{t+1}$, $y^- \leftarrow y_t$, otherwise set $y^+ \leftarrow y_t$, $y^- \leftarrow y_{t+1}$. In Line 9, we can substitute $\Delta(y_{t+1}, y_t)$ for $\omega(y^+) - \omega(y^-)$. Further, taking the inner product $\theta' \hat{\nabla}$ yields the log of the model-ratio term in the Metropolis-Hastings acceptance ratio enabling the same type of efficient sampling performed by BLOG [58].

While Equation 5.6 applies generally to all graphical models, we should note that in many real-world problems, additional factors cancel for various model-specific structural reasons, often leading to a full-degree polynomial reduction in computation. In applications where the graphical model’s fan-out is bounded, it can be shown that the number of factors required to perform a SampleRank update is linear in the size of the MCMC step and therefore independent of the number of variables in the model.

5.2.5.2 Efficient Parameter Averaging

We describe an efficient implementation of parameter averaging that exploits sparsity in the gradient, and is applicable for cases in which we do not know the number of SampleRank updates n in advance, a situation that frequently arises if an adaptive stopping criterion is used (e.g., stop when accuracy reaches a certain threshold on a held-out validation set). However, it requires some minor bookkeeping: let τ be a dense vector of non-negative integers with the same number of dimensions as θ . We will use this vector to keep track of—for each element in the weight vector—the most recent time step in which that element was updated. This method is most easily described with the pseudo-code shown Algorithm 7; in pseudocode we notate the i th element of a vector \mathbf{v} as $\mathbf{v}[i]$. Notice that Line 4 loops over the sparse vector $\hat{\nabla}$ rather than the dense gradient θ . This is a crucial implementation detail that takes advantage of sparsity to achieve efficiency.

Algorithm 7 Sparse parameter averaging algorithm for SampleRank.

```

1: for  $t = 0, \dots$ , (until some stopping criteria is met) do
2:    $\hat{\nabla} \leftarrow$  compute SampleRank gradient
3:   denseIndexVector  $\leftarrow$  DENSE_IDX( $\hat{\nabla}$ )
4:   for  $i \leftarrow 0$  until  $\hat{\nabla}$ .numElements do
5:     idxDense  $\leftarrow$  denseIndexVector[ $i$ ]
6:     timeElapsed  $\leftarrow t - \tau$ [idxDense]
7:      $\theta$ [idxDense]  $\leftarrow \theta$ [idxDense]
8:      $\bar{\theta}$ [idxDense]  $\leftarrow$  timeElapsed  $\times \eta \times \hat{\nabla}$ [ $i$ ]
9:      $\tau$ [idxDense]  $\leftarrow t$ 
10:  end for
11: end for
12:  $\bar{\theta} \leftarrow \frac{1}{n} \bar{\theta}$ 

```

Hal Daume’s dissertation contains an alternative method with less bookkeeping for cases in which the number of samples is known *a priori* [22]:

$$\theta \leftarrow \theta + \eta \hat{\nabla} \tag{5.7}$$

$$\bar{\theta} \leftarrow \bar{\theta} + \left(\frac{n-t}{n} \right) \eta \hat{\nabla} \tag{5.8}$$

5.2.5.3 Efficient Implementations of L2 Regularization

The SampleRank objective contains an optional L2 regularization term. Such regularization is known to provide better generalization to unseen test data. Since SampleRank optimizes the objective function in the primal—and because of SampleRank’s connection with structured support vector machines—we rely on the PEGASOS [84] algorithm for efficiently applying regularization to SampleRank. The key observation in PEGASOS [84] is that the L2 regularized weight vector must lie in an \mathbb{R}^{k_θ} ball whose radius is determined by C . Thus, we can apply a projection step after each SampleRank update to ensure that the weights are always inside the appropriate ball. In stochastic approximation, this step is justified because intuitively this projection can be thought of as an error term that goes to zero along with the learning rate. The updates are as follows:

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \eta \hat{\nabla} \tag{5.9}$$

$$\boldsymbol{\theta} \leftarrow \min \left(1, \frac{1}{\sqrt{C} \|\boldsymbol{\theta}\|_2} \right) \boldsymbol{\theta} \tag{5.10}$$

Where the minimization in the second update enforces that the weights are only scaled if the weight vector lies outside the ball. As we can see, the main problem with this computation is that computing the L2 norm of the weight vector seems *prima facie* a dense computation. However, as it turns out, we only need to compute this exhaustively once (or not at all if weights are initialized to $\mathbf{0}$), and then we can sparsely and incrementally maintain the L2 norm as the parameters are updated. We will define $w \leftarrow \|\boldsymbol{\theta}\|_2$. What we want is a way to incrementally compute $w' \leftarrow \|\boldsymbol{\theta} + \hat{\nabla}\|_2$. Simple algebra reveals:

$$w' = \|\boldsymbol{\theta} + \hat{\nabla}\|_2 \tag{5.11}$$

$$= \left((\boldsymbol{\theta} + \hat{\nabla})^T (\boldsymbol{\theta} + \hat{\nabla}) \right)^{1/2} \tag{5.12}$$

$$= \left(\|\boldsymbol{\theta}\|_2^2 + \|\hat{\nabla}\|_2^2 + 2\hat{\nabla}^T \boldsymbol{\theta} \right)^{1/2} \tag{5.13}$$

Thus giving us a recursive definition of the norm, where all summations are now over the sparse gradient $\hat{\nabla}$ instead of the dense parameter vector θ . Of course, we cannot use w to explicitly re-normalize the weight vector because that requires iterating over all the parameters. We will instead modify the SampleRank update rule (Lines 9-11 of the Algorithm 6) as follows:

- 1: **if** $\frac{1}{\sqrt{Cw}}\theta^T\hat{\nabla} < \omega(y^+) - \omega(y^-)$ **then**
- 2: $\theta \leftarrow \theta + \eta\hat{\nabla}$
- 3: $\theta \leftarrow \min\left(1, \frac{1}{\sqrt{Cw}}\right)\theta$
- 4: **end if**

Note the subtle change: we simply keep track of the scalar w separately from the unscaled weights and then multiple w into unscaled weights to calculate the true weights.

5.2.6 Extensions to SampleRank

5.2.6.1 SampleRank for SVM

SampleRank utilizes domain-specific evaluation metrics to generate an enriched set of constraints for rapid learning. However, not all structured prediction problems are evaluated with such rich metrics. For example, in multi-label classification problems, the loss metric cannot indicate a preference between configurations that differ by a choice of incorrect labels (that is, all classification errors are treated equally). In such cases, running SampleRank as described in Algorithm 6 will result in wasted samples if the proposer generates pairs with no preferences under ω . While this problem can be alleviated by designing q to better agree with ω , this is not always straightforward.

Alternatively, we can modify SampleRank to target the structured SVM objective yielding an algorithm similar to persistent contrastive divergence. However, instead of always updating the parameters after each MCMC step, the parameters are only updated if a margin violation is incurred. The gradients between the truth and each configuration can be computed incrementally with Equation 5.6: as the chain wanders from the truth we ac-

accumulate the gradients between neighboring accepted configurations. This accumulated gradient is used in place of the atomic gradient in lines 9-11 of the SampleRank algorithm.

5.2.6.2 SampleRank with delayed reward

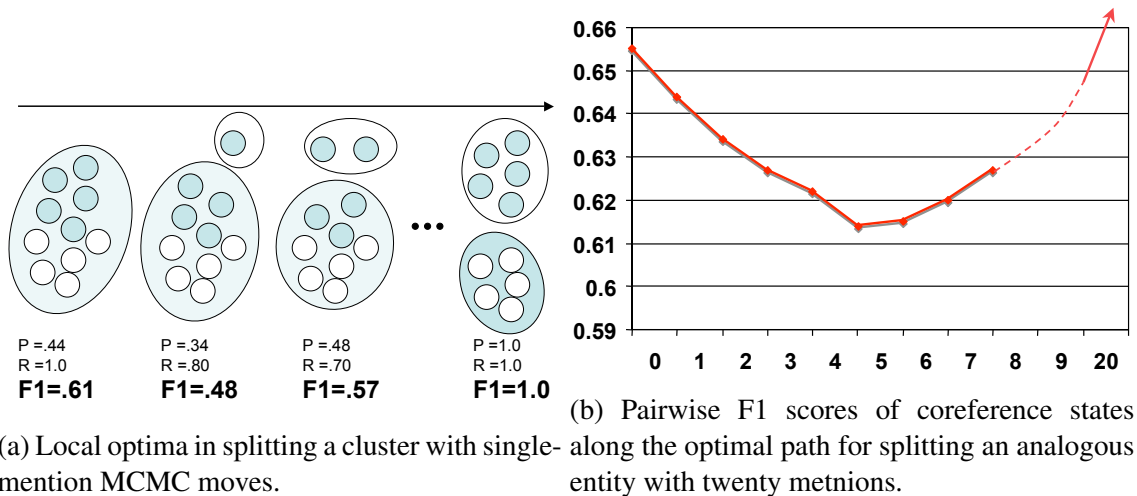


Figure 5.1: Local optima in coreference examples (left: with ten mentions, right: with twenty mentions). In the left figure, small circles represent mentions and their color is their ground-truth entity label; big ovals represent hypothesized entities.

One issue with SampleRank (and MCMC in general) is local optima. For example, if we were to use the pairwise F1 score for ω in SampleRank training of a pairwise coreference model, then this choice of ω would create local optima in the search space. For example, consider the MCMC moves required to split an incorrect entity that contains five mentions of *Washington the state* and five mentions of *Washing the person* into two correct entities. We show the sequence of moves and their corresponding pairwise F1 scores in Figure 5.1b. What we can see is that we must accept several MCMC proposals that decrease the F1 score before the F1 score begins increasing. This is potentially problematic for SampleRank because during learning, SampleRank would learn to rank the initial incorrect state higher than the intermediate states that are required to eventually split the entity.

One approach for addressing this problem is to modify SampleRank so that it captures delayed reward. Although we will not go into details here, we describe such an approach in recent work [113]. The basic idea is to cast learning in a graphical model as reinforcement learning; specifically, as temporal difference (TD- γ) [96] learning in an Markov decision process (MDP).¹ Some key observations in mapping graphical model learning into reinforcement learning are that (1) the linear function approximator for the value-function in reinforcement learning has the same form as the log-linear weights of the graphical model (2) the MCMC proposer defines the action space of the MDP, and (3) the training signal ω is a shaped reward function for the MDP. In our previous paper, we demonstrated that capturing delayed reward can improve accuracy on the task of ontology alignment. However, in comparison to SampleRank, the method is more complicated to implement and contains many parameters that are difficult to set (e.g., setting the learning rate is notoriously difficult for many reinforcement learning algorithms).

5.2.7 Experiments

We evaluate SampleRank parameter estimation on four structurally diverse models for four important real-world problems: with-in document coreference resolution, multi-label classification, named entity recognition, and hierarchical cross-document coreference resolution. First, we compare SampleRank with other MCMC based learning algorithms on a model of noun-phrase coreference (Table 5.1). Second, we compare to a wider range of recent approximate algorithms on a more tractable pairwise model of multi-label classification (Table 5.2). Third, we compare SampleRank to exact methods for linear-chain models of named entity recognition (Table 5.3). Finally, we study SampleRank’s ability to train the hierarchical coreference model weights. We find that SampleRank is better at training structured prediction models than other MCMC alternatives such as persistent contrastive

¹In essence, the basic SampleRank algorithm is similar to TD learning learning with $\gamma = 0$ (in other words, no delayed reward).

divergence. Further, we find that SampleRank is competitive with exact learning algorithm on model that permit exact inference. Detailed explanations of each experiment follow.

5.2.7.1 Noun-phrase Coreference

Noun-phrase coreference is an information extraction problem for which traditional learning and inference algorithms are intractable. The problem of noun-phrase coreference to cluster each noun-phrase (mention) into sets such that all the mentions in a given set refer to the same entity. For example, the cluster corresponding to *Michelle Obama* would contain mentions such as “First Lady,” “Michelle Obama,” and “she.” For this experiment, we employ a partition-wise model of coreference [21], which includes a compatibility function between each pair of mentions.

In order to perform inference, we initialize the model to the singleton configuration (each mention is in a set that contains only itself), and perform a low temperature ($\tau = 0.001$) Metropolis-Hastings inference procedures with the following proposal distribution. First, pick two mentions m_i, m_j uniformly at random from the set of mentions M_d in document d . If m_i and m_j are in different entities, then move m_j to be in the same entity as m_i . Otherwise, the mentions are in the same entity: separate them by moving m_j into a new singleton entity.

We implement three types of MCMC-based training algorithms that use this MCMC procedure: SampleRank, contrastive divergence, and persistent contrastive divergence. We also implement the SVM version of SampleRank, and implement a large-margin variant of CD termed “SampleRank-SVM-1”. Persistent contrastive divergence uses a constant learning rate, and always performs an update after each step of inference by (1) increasing the weights by the sufficient statistics of the ground-truth and (2) decreasing the weights by the sufficient statistics of the current state. Contrastive divergence 1 performs a similar procedure, except the chain is reinitialized to the ground-truth after each sample.

We experiment with two learning rates for SampleRank: an adaptive learning rate called (MIRA [18]), and a constant perceptron learning rate of 1. For training, we perform ten loops over all the documents in the training set, running 100,000 samples for each document. We then evaluate the performance after ten rounds of training on the test set and report the results in Figure 5.1. We find that SampleRank significantly outperforms the learning algorithms.

5.2.7.2 Multi-Label Classification

Multi-label classification is similar to classification, but instead of predicting a single label for each instance, the goal is to predict a subset of labels for each instance [84, 28, 56]. Multi-label classification is a structured prediction problem because the presence of one label might influence the presence or absence of other labels (that is, the classification predictions are made jointly). Thus, we model the problem with a fully-connected pairwise CRF [28] in which there are hidden variables $Y = \{y_i\}_1^N$ for each of the N possible labels, and factor functions $\Psi = \{\psi(y_i, y_j)\}_{\forall i, j: i < j}$. Each $y_i \in \{0, 1\}$ is a Bernoulli random variable where $y_i = 1$ indicates that label i is “on” for a given input $x \in \mathbb{R}^P$. Our feature functions consist of $\psi_i(y_i, x)$ and $\psi_{ij}(y_i, y_j)$ with corresponding weights θ . We define the loss function ω as the Hamming accuracy, and use a Gibbs sampler for inference.

We compare SampleRank and SampleRank-SVM to persistent contrastive divergence (PCD) and PEGASOS [84] (which uses linear programming (LP) solvers to perform loss-augmented inference). Each algorithm performs updates on a single training example at a time. The results are shown in Table 5.2. SampleRank performs significantly better than persistent contrastive divergence on the yeast dataset, and performs competitively to the LP-based training algorithms (which are often exact) on both datasets. The fact that the non-SVM variant of SampleRank performs worse than its SVM counterpart is not surprising in this case because many of the Gibbs sampling updates do not transition between states that differ in accuracy. This is because states that differ in the setting to a single incorrect

variable have the same Hamming accuracy. In these cases, SampleRank-SVM is able to make an update, but the gradient for SampleRank is zero.

5.2.7.3 Named Entity Recognition

In this section we evaluate SampleRank on a linear-chain CRF for named entity recognition (NER), thus allowing us to compare with exact learning algorithms such as SVM and maximum likelihood. NER is the problem of labeling the tokens in a sentence with labels corresponding to entity types. For example, given the sentence

Barack Obama returned to the White House.

We would label the tokens “Barack” and “Obama” as *PERSON* and “White” and “House” as *LOCATION*. The model is a linear chain in which features of the tokens x_1, x_2, \dots, x_n are the observed input variables, and the labels y_1, y_2, \dots, y_n are the output variables. The y variables take on a values from the set $Labels = \{B-PER, I-PER, B-ORG, I-ORG, B-LOC, I-LOC, B-MISC, I-MISC, O\}$. The x variables are each vectors taking on values in $k - dimensional$ binary features space $Tokens = \{0, 1\}^k$. We employ two feature spaces, a basic space which includes the value of the tokens, a normalized variant of the string, the capitalization patterns of the words, etc. And a more advanced set of features which also includes the feature values of previous and next tokens. We use the following set of features:

- Markov transition features: $\phi(y_i, y_{i+1}) \in Labels \times Labels$
- Emission features: $\phi(y_i, x_i) \in Labels \times Tokens$
- Bias features: $\phi(y_i) \in Labels$
- Previous token features (advanced): $\phi(y_i, x_{i-1}) \in Labels \times Tokens$
- Next token features (advanced): $\phi(y_i, x_{i+1}) \in Labels \times Tokens$

- Emission features: $\phi(y_i, x_i) \in Labels \times Tokens$

We report the results in Table 5.3. On both feature sets SampleRank performs competitively to the exact learning algorithms. In fact, somewhat surprisingly, SampleRank performs slightly better than the exact algorithms. We speculate that one reason for this is that the extra ranking constraints in SampleRank (between two incorrect configurations) allows SampleRank to learn models that are better at making predictions in the presence of errors. Furthermore, these additional constraints also function as a regularizer by distributing the mass on the weight vector across the many different parameters.

Table 5.1: A comparison of SampleRank with other MCMC learning algorithms on an entity-wise model of coreference.

Method	B-cubed F1	Time (s)
SampleRank (MIRA)	80.04	3115
SampleRank	79.23	3064
SampleRank-SVM	73.89	3399
persistent contrastive divergence	73.27	11078
contrastive divergence-1	74.24	3326
SampleRank-SVM-1	74.84	2988

5.2.7.4 Hierarchical coreference

In traditional supervised learning for structured prediction, it is assumed that the ground-truth labels are available for each training example. However, in the hierarchical model, we only have labels for the mentions, and not the latent sub-entities. Thus, we propose a label-inheritance approach in which the labels of sub-entities inherit the labels of their children. Since an inferred (incorrect) sub-entity might comprise of many mentions each with different ground-truth labels, we allow the labels on subentities to be “soft.” More specifically, we associate with each node a “bag-of-labels” that represents the count of different ground-truth entity levels at the leaf-level mentions. Then, at the root of each tree, we can examine how these label bags change during training time, and extract a signal

Table 5.2: Labeling error rates of various algorithms on multi-label classification data sets averaged over 10 random runs. Results in bold indicate significant error reduction ($p = 0.05$).

Method	Scene (6 labels)	Yeast (14 labels)
PCD	9.86 \pm .20	26.79 \pm .49
Pegasos	9.57 \pm .10	21.06 \pm .19
SampleRank	9.71 \pm .14	21.48 \pm .13
SampleRank-SVM	9.49 \pm .15	20.58 \pm .35

Table 5.3: SampleRank and exact parameter estimation algorithms on CoNLL NER.

Method	Test-A F1	Test-B F1	Time (s)
BASIC FEATURES			
SR (10 epochs)	0.855	0.795	429 sec
SR (5 epochs)	0.848	0.790	183 sec
SR (1 epoch)	0.822	0.766	65 sec
SVM ^{hmm} (c=0.1)	0.834	0.771	90 sec
SVM ^{hmm} (c=1)	0.843	0.766	90 sec
SVM ^{hmm} (c=10)	0.843	0.766	90 sec
MaxLike-L2(L-BFGS)	0.852	0.780	7687 sec
ADVANCED FEATURES			
SR (10 epochs)	0.913	0.851	663
SR (1 epoch)	0.888	0.825	82
MaxLike-L2	0.896	0.832	15,890
SVM (c=0.1,1,10)	0.886	0.828	90
SVM (c=1/ \mathcal{D})	0.637	0.623	54
SVM (c=10/ \mathcal{D})	0.798	0.759	75

ω for SampleRank. Specifically, we define ω in terms of the pairwise true-positives and false-positives described in the coreference evaluation section (Section 2.2.5):

$$\omega = \text{tp} - \text{fp} \tag{5.14}$$

We choose this signal because it is easy to compute efficiently with a single factor template on the root-level entity’s bags-of-labels. The computation is efficient because it only scales linearly with the number of entity-labels assigned to the entity’s mentions (usually, the

number of mistakes is bounded by a small constant). It also has the desirable property that the ground-truth clusterings of mentions into entities has the highest possible ω score.²

We evaluate SampleRank training for hierarchical coreference on the BoNY (Boston/NY) subset of the Wikilinks corpus described in the previous section. In these experiments, we train on the Boston data (200k samples for training) and evaluate on the NY data (200k samples for testing). We find that SampleRank achieves similar accuracy to the manually tuned model on the basic feature set, but the accuracy of the manually tuned model. However, SampleRank training is automatic and only takes 2-3 minutes on this data. In contrast, the manual method is much more time-consuming.³ Furthermore, SampleRank allows us to introduce a set of more advanced features that comprise higher-order polynomials (up to third degree) of the original features, along with pairwise and triple-wise feature conjunctions. The advanced features boosts accuracy, and still only takes minutes to train (the number of features is prohibitively expensive for manual training).

We also evaluate contrastive divergence (CD) on this data. However, we find that CD is unable to learn good weights in the hierarchical model because it severely overfits to the ground-truth configuration. We speculate that the reason for this is that the MCMC proposal distribution is highly biased towards proposing moves that split an entity apart when initialized in the ground-truth configuration on this dataset. Consequently, CD incorrectly learns a high weight on the feature that encourages the model to keep entities merged together. As seen in results table, CD achieve nearly 100% recall because the resulting model merges everything into a single cluster (except the Wikipedia mentions because only one is allowed per entity). We provide the trained weights for the basic feature set in Table 5.5.

²There are multiple states that achieve this score because there are multiple ways of organizing an entity’s mentions into trees, and each of these trees would have the same score. This further highlights the advantage of automatic learning methods such as SampleRank.

³Note that the manually tuned model uses a different mention-processing pipeline that tokenizes and weights the bags-of-words differently; when employing the new mention-processing strategy the accuracy of the manually tuned model plummets. After spending several hours manually tuning the model, we were unable to bring the accuracy up to SampleRank’s level on the newly processed mentions.

The weights that cause the CD model to over-merge are the positive weight on the bias feature, the relatively high positive weights on each BoW, and the negative weight on the sub-entity existence penalty (causing more sub entities). Although the entity-existence penalty weight of -0.371 is relatively small compared to the other methods, it is still not small enough to compensate for the other weights.

Method	Prec	Rec	F1
SampleRank (advanced features; new processing)	99.9	95.1	97.5
SampleRank (basic features; new processing)	99.9	94.3	97.1
CD (advanced features; new processing)	23.3	99.9	37.8
CD (basic features; new processing)	23.3	99.9	37.8
Manually tuned (basic features; new processing)	85.5	92.9	89.0
Manually tuned (basic features; old processing)	-	-	97.3

Table 5.4: SampleRank training of the hierarchical coreference model on the BoNY subset of Wikilinks.

Features	Weights		
	SampleRank	Manual	CD
combined-bow	0.008	2.0	0.633
subentity-exists	0.041	0.25	-0.997
mention-bow	0.080	2.0	0.656
topic-bow	0.319	2.0	1.532
name-bow	1.441	2.0	2.337
bias	-1.594	-1.5	0.31
entity-exists	-1.197	-0.5	-0.371
context-bow	-0.280	2.0	1.131

Table 5.5: Weights on basic features.

5.3 Training Restricted Boltzmann Machines with SampleRank

Deep learning is an active area of research because these models have been successful at learning feature representations on a variety of vision [38] and NLP tasks [82]. One of the key building blocks of these models is the restricted Boltzmann machine (RBM)

[38, 81]. RBMs are a two layer neural network in which a layer of visible units $\mathbf{x} = \{x_i\}$ (representing the input) are fully connected to a layer of unobserved “hidden” units $\mathbf{z} = \{z_i\}$, but no connections exist between nodes in the same a layer. For example, the visible units might each correspond to the pixels of an image, and the hidden units correspond to latent features (or filters) for representing the image. The connections between these two types of units are the weights. These weights are trained to model the input data, allowing the unsupervised discovery of higher order features from unlabeled data.

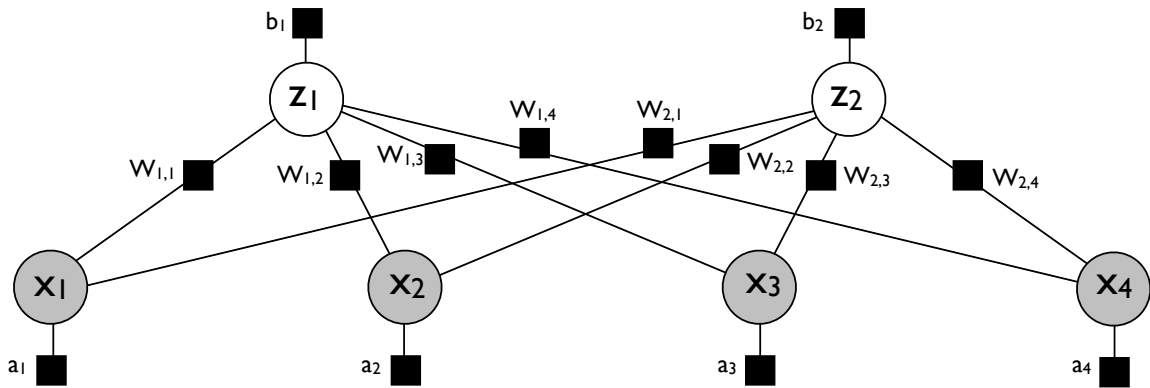


Figure 5.2: A factor graph representation of an RBM as a Markov random field with weights between the visible and hidden units, and bias weights on the visible and hidden units.

RBMs can also be viewed as generative Markov random fields (See Figure 5.2). Formally, they model the probability distributions of the data as follows:

$$\pi(\mathbf{x}) = \exp(-F(\mathbf{x}) - Z), \quad F(\mathbf{x}) = \sum_{\mathbf{z}} E(\mathbf{x}, \mathbf{z}) \quad (5.15)$$

where $F(\mathbf{x})$ is the free energy, E is the negative energy, and Z is the partition function that normalizes over the visible units. Since the model contains a weight between each hidden unit and each observed unit, the energy function takes the following form

$$E(x, h) = \mathbf{x}^T W \mathbf{z} + \mathbf{a}^T \mathbf{x} + \mathbf{b}^T \mathbf{z} \quad (5.16)$$

where W is a matrix that stores the weights between the hidden and observed units, a is the vector of bias weights for the visible units and b is the vector of bias weights for the hidden units. This corresponds to a factor graph with factors: $\Psi = \{\psi_{ij}(x_i, z_j) \mid \forall i, j\} \cup \{\psi_i(x_i) \mid \forall i\} \cup \{\psi_j(z_j) \mid \forall j\}$.

The problem of learning is to find a suitable setting to the parameters $\theta = \langle W, a, b \rangle$ that model the data $\mathcal{D} = \{\mathbf{x}_i\}$; for example, by maximizing the log likelihood

$$\mathcal{L}(D, \theta) = \sum_{\mathbf{x} \in \mathcal{D}} \log \pi(\mathbf{x}; \theta) \quad (5.17)$$

Maximum likelihood learning in this model is intractable because we must marginalize over the hidden variables in order to compute gradients, thus gradients are typically approximated by using Gibbs sampling.

A Gibbs sampler generates a sequence of states $s^{(0)}, s^{(1)}, \dots$ with $s^{(t)} = \langle \mathbf{x}^{(t)}, \mathbf{z}^{(t)} \rangle$. Traditionally, the Gibbs sampler is initialized to an input data example $\mathbf{x}^{(0)} = \mathbf{x}$ with the initial hidden units sampled from $\mathbf{z}^{(0)} \sim \pi(\cdot | \mathbf{x}^{(0)})$. Then, given this initialization, each sample $s^{(t+1)}$ is generated as follows

$$\begin{aligned} \mathbf{x}^{(t+1)} &\sim \pi(\cdot | \mathbf{z}^{(t)}) = \prod_{x_i \in \mathbf{x}} \pi(x_i | \mathbf{z}^{(t)}) \\ \mathbf{z}^{(t+1)} &\sim \pi(\cdot | \mathbf{x}^{(t)}) = \prod_{z_i \in \mathbf{z}} \pi(z_i | \mathbf{x}^{(t)}) \end{aligned}$$

and since the hidden and visible units are binary

$$\begin{aligned} \pi(x_i | \mathbf{z}) &= \text{sigmoid}(W_i^T \mathbf{z} + a_i) \\ \pi(z_i | \mathbf{x}) &= \text{sigmoid}(W_i \mathbf{x} + b_i) \end{aligned}$$

where $\text{sigmoid} : \mathbb{R} \rightarrow (0, 1)$ s.t. $\text{sigmoid}(r) \mapsto \frac{1}{1+e^{-r}}$ is the logistic sigmoid function.

Contrastive divergence (CD) [36] and persistent contrastive divergence (PCD) [101] use this full-sweep Gibbs sampler to approximate the log-likelihood gradient. In particular,

CD-k runs k steps of Gibbs sampling and then uses the k th sample to compute a gradient against the data. Persistent contrastive divergence runs k steps of Gibbs sampling, but uses each of the $1, 2, \dots, k$ samplers to update the weights inside of sampling.

5.3.1 SampleRank for RBMs

Since we can represent an RBM as a graphical model, we could in theory learn its weights using SampleRank. However these models differ from the types of models used in the structured prediction setting in four fundamental ways (1) RBMs are generative models of the data (2) RBMs are unsupervised representation learners (3) RBMs contain latent variables (4) there are no direct interactions (factors) amongst the output variables. Not only are the models different between these two settings, but the MCMC samplers are also substantially different because each iteration of Gibbs samples a new value for every variable (as opposed to the value of a single variable). This is cause for concern because states that differ in a large number of variables may not be meaningful to compare. These differences raise a number of questions:

- Q1: Which state pairs should we consider for ranking during learning?
- Q2: Given a state pair ($s^{(t+1)}$ and $s^{(t)}$), how do we determine which state the model should rank higher?
- Q3: Are consecutive Gibbs states, which potentially differ in the setting to a large number of variables, problematic for SampleRank learning?
- Q4: Is the non-convexity due to the hidden variables problematic for SampleRank learning?

We cannot simply answer these questions in isolation, because the answer to each question is intertwined with the answers to the others. For example, if our answer to Q2 is to rank states according to how similar the visible units in that state are to the original input data, then our answer to Q1 could not include state pairs in which the two states differ in the

setting to a single variable (as is traditional for the discriminative version of SampleRank). The reason is the evaluation objective is a function of the visible units and not the hidden units. Thus, hidden and visible units must change together in order to produce state-pairs that give rise to useful gradients. Given these questions, it is not immediately clear whether SampleRank could actually learn reasonable weights for the RBM. In this section, we describe a version of SampleRank that is suitable for this problem setting and we investigate SampleRank’s ability to train RBMs.

First, we must specify the set of constraints we wish to satisfy by defining the set of state pairs \mathcal{P} (in order to answer Q1) and training signal ω (Q2). Answering these two questions is a matter of choosing an appropriate MCMC procedure and training function $\omega(s^{(t)})$ (to score each MCMC state). For the purpose of direct comparison with CD and PCD, we use the traditional RBM Gibbs sampler described in the previous section; this sampler defines the set \mathcal{P} and addresses Q1. Next, we must address Q2 by choosing an ω for determining ranking constraints. Clearly the model should rank the data state higher than the sampled states; however, given two sampled states, which should the model rank higher? We choose to rank the state that is more similar (measured by total variation distance) to the data state. That is, we define $\omega(s^{(t)})$ as follows:

$$\omega(s^{(t)}) = -\|\mathbf{x}^{(0)} - \mathbf{x}^{(t)}\|_{\text{tv}}$$

Note that even though our training objective only specifies preference over the visible units, we still obtain gradients over the hidden units because the Gibbs sampler has the ability to change both the hidden and visible units in each step. However, this ability might also have negative consequences. For example, the Gibbs sampler can transition between states that differ in an arbitrary number of variable settings in a single step, potentially forcing the algorithm to rank states that are highly dissimilar from each other (a concern exemplified in the first row of Figure 5.5). However, in practice, we expect that as learning progresses, the Gibbs sampler will eventually generate states that are similar to the data, causing the

model to rank states that we believe would be better for learning (indeed such behavior does occur, as seen by the last three rows in Figure 5.5).

We also consider modifications to the Gibbs sampler that are more suitable for SampleRank (and other large margin objectives). For example, since SampleRank only updates the weights if there is a margin violation, a Gibbs sampler might generate states that the model already knows how to rank, and cause learning to stagnate. Thus, we introduce a *temperature* parameter that governs the greediness of the sampler. Intuitively, a lower temperature causes the sampler to generate states with lower energy. Since the energy difference helps determine the margin (Equation 5.1), these states are likely to have larger margin violations than states sampled from higher temperatures. However, the difference in energy is only part of the margin violation calculation; we must also consider the difference in objective scores. Therefore, we modify the Gibbs sampler to sample proportional to the difference in energy and objective. Since our training objective is only a function of the visible units, and fully factorizes over these units, we can modify the Gibbs sampling distribution $\pi(\mathbf{x}_i|\mathbf{z})$ as follows

$$\pi(x_i^{(t+1)}|\mathbf{z}) = \text{sigmoid} \left((W_i^T \mathbf{z} + a^i + (0.5 - x_i^{(t)})/\tau) \right)$$

This modification causes the sampler to select states that differ from the current state $\mathbf{x}^{(t)}$. That is, if $x_i^{(t)}$ is 0 then the pre-sigmoid energy function is increased, thus encouraging the algorithm to sample 1 in the next step; conversely, if $x_i^{(t)}$ is 1, then the pre-sigmoid energy function is decreased, thus encouraging the algorithm to sample a 0 in the next step. Intuitively, the extra term approximately maximizes the difference-in-loss ($\omega(\mathbf{x}^+) - \omega(\mathbf{x}^-)$) term of the margin violation. That is, each visible unit changed corresponds to an opportunity to incur an additional loss of 0.5. Furthermore, loss augmented inference potentially addresses a serious flaw in local smoothness of learnt neural networks in which small perturbations of the input lead to large changes in the output [98].

Finally, we must decide whether to use the maximization over the energy instead of the free-energy to compute the SampleRank loss function. Choosing the former would be more convenient because it would allow us to replace a marginalization over the hidden units with a maximization over the hidden units. However, choosing the latter would produce an algorithm more similar to latent structured SVMs. For the purpose of our investigation, we choose the former. We justify this choice because (1) maximizing over the variables is more convenient and (2) maximization is similar to marginalization because the weights are sufficiently peaked [119].

In summary, we described a SampleRank training procedure for RBMs that employs the existing Gibbs sampling and contrastive divergence machinery. Although similar in algorithmic structure, SampleRank differs from persistent contrastive divergence in the following ways:

- **Loss function:** SampleRank’s objective function involves a hinge loss instead of a log loss. Thus, SampleRank performs an update only if a ranking violation is incurred.
- **Cost-sensitive:** SampleRank’s loss has an evaluation function used for determining ranking constraints. This allows domain-specific cost structures such as accuracy or F1 to be optimized during learning. These cost functions translate into loss-augmented inference during training.
- **Inference:** SampleRank does not need to draw true samples from the model, thus (1) we can use loss-augmented inference during learning and (2) we can use a low temperature Gibbs sampler for learning to greedily find violated constraints.
- **Regularization:** minimizing SampleRank’s objective function involves minimizing the L2 norm of the weight vector; thus, SampleRank can optionally include a projection step onto the L2 ball after each update (as done for SVM learning [84]).

- Initialization: SampleRank does not need to be initialized to the ground-truth. Loss augmented inference will help drive the chain towards the data. We do not investigate the effect of initialization in our experiments.

In the next section we evaluate SampleRank’s ability to learn RBMs via an empirical comparison with contrastive divergence.

5.3.2 RBM Experimentements

Algorithm 8 RBM Experimental Training Procedure

```

inputs:
n //number of rounds to train
k //number of steps per example
algorithm //learning algorithm: CD, PCD, SampleRank, PSVM, etc.
rbm //RBM with randomly initialized weights
for round = 1 to n do
  for trainingExample in  $S_{train}$  do
    resetChain(trainingExample) //re-initializes the chain’s state to the data.
    doKStepsOfLearning(rbm,algorithm,trainingExample,k)
  end for
  writeReconstructionError(rbm, S_{test})
end for

```

The goal of our experiments is to investigate SampleRank’s ability to train an RBM to learn useful representations of the data. For our data, we use the MNIST digit recognition dataset, which comprises 70,000 black and white images of handwritten digits (0-9). The training set comprises 60,000 examples, and the test set comprises 10,000 examples. Each pixel takes a value in the range of 0-255, but we normalized the values to be between 0-1, and then converted them into binary bitmaps by rounding the value to 0 or 1. Depending on the experiment, we use a subset of the available training data (1000, 10k, 60k). First, we learn models on the training data, and then we evaluate the quality of the learned representation by measuring the reconstruction-error variant on the held-out test data. Specifically, we measure the total variation distance between each input image, and the model’s reconstruction of that image, averaged over all images in the testing set.

We set-up each of the training algorithms using the incremental gradient update scheme shown in Algorithm 8. That is, beginning with an randomly initialized RBM, we run n rounds of learning. Each round consists of a sweep over all the training examples, and for each training example we first reset the chain’s state to the input image and then do k steps of a Gibbs-sampling based learner (SampleRank, CD, PCD, etc.). Note that because we are resetting the chain both SampleRank and PCD are only persistent for ten samples (note that this introduces some bias to PCD). In our experiments $k = 10$ and $n = 50$, and we use a learning rate of $0.1/|S_{train}|$ where $|S_{train}|$ is the number of training examples⁴. We also use 100 hidden binary units in our RBM.

We compare the following algorithms: SampleRank, CD, and PCD. For SampleRank, we explore three experimental variables that determine the nature of the algorithm: *LossAugmentedInference* \times *Temperature* \times *Constraints*. When *LossAugmentedInference* is set to true, we employ loss-augmented inference by using the probability distribution defined Equation 5.3.1 to update the visible updates, when false, we use the traditional probability distribution to update the visible units. The variable *Temperature* $\tau \in \{0.01, 0.1, 1.0\}$ determines the greediness of the inference during learning. A low temperature (0.01) is almost completely greedy, and a temperature of 1.0 samples exactly from the model conditional. Finally, the variable *Constraints* = {data-only, pairwise-only, full} determines which constraints we explicitly update during inference. Data-only means that SampleRank algorithm only performs updates between the data state and each sampled chain (the max-margin (SVM) analog of PCD); thus, we call this variant of SampleRank PSVM. The pairwise-only condition only performs updates between consecutive states in the Gibbs chain. The full condition performs updates in both cases. Note, that different settings to these variables produce different learning algorithms. For example, the condition in which we use a low-temperature, no loss augmented inference, and data-only constraints, cor-

⁴We make the learning rate proportional to the number of training examples because we are measuring quality of the model after each pass through all the training examples

responds to the PercLoss algorithm for training conditional RBMs [62]. Informally, the SampleRank variants that employ the data-only updates can be viewed as max-margin or latent SVM variants of PCD.

In Figure 5.3 we investigate how the different settings to our three experimental variables (LossAugmentedInference, Temperature, and Constraints) affect the final quality of the learned RBM model. In the figure, “sr-full” corresponds to the full constraint condition, “sr” corresponds to the pairwise-only condition, and “psvm” corresponds to the “data-only” conditions. We find that using both the pairwise and data constraints performs the best, a finding consistent with a recent study that uses SampleRank to discriminatively train higher order dependency parsers [123]. The other two variables Temperature and LossAugmentedInference are labeled in the legend. Temperature and LossAugmentedInference interact in an interesting way. For example, low temperatures appear to hurt performance unless loss-augmented inference is employed, in which case low temperatures work best.

One possible explanation for this behavior is that certain types of approximate MAP inference procedures are known to cause premature convergence in max margin learning algorithms such as structured perceptron [47] and structured SVMs [28]. For example, the low temperature could cause the model to deterministically return a configuration for which there is no ranking violation, causing the learning algorithm to get stuck. However, upon inspection, we observe that close to 100% of the greedy (low temperature) samples result in a parameter update. Instead, we speculate that the reason the greedy sampler performs poorly is that it generates states that are highly similar to the initial data, causing only small updates in the weights. Indeed, we observe that greedy inference causes the model to generate states that differ in about 1/3rd as many visible units and about 1/7th as many hidden units as the normal temperature-1 sampler. That is, even though the sampler is making updates, the updates are smaller in that they influence fewer entries in the weight matrix. The loss-augmented inference sampling distribution prevents this problem because it encourages the model to move away from the current state. See Table 5.6 for more details.

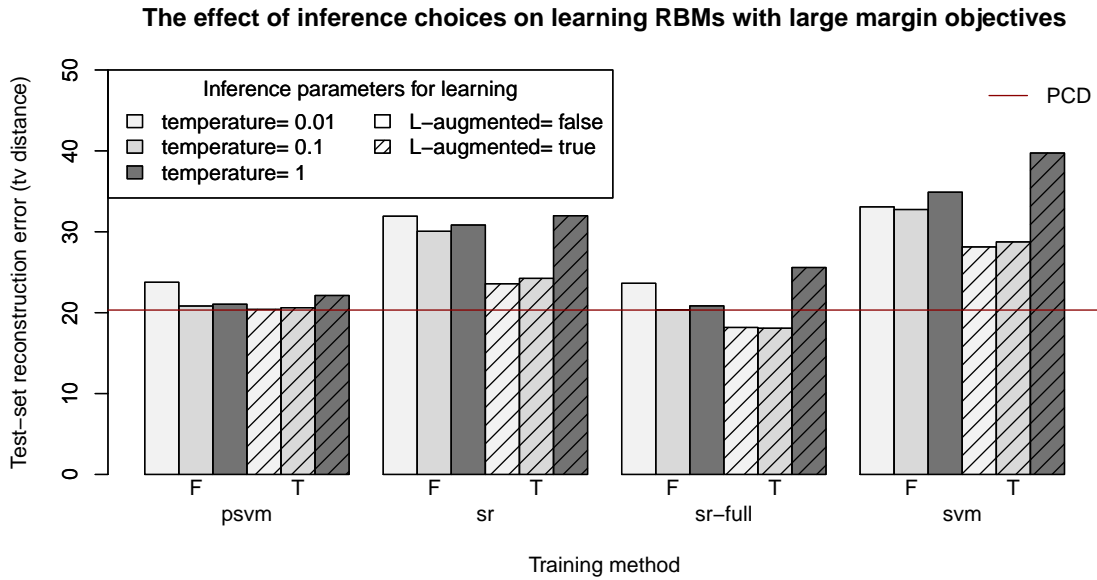


Figure 5.3: A comparison of different SampleRank variations for training RBMs.

In Figure 5.6 we plot test-time reconstruction error as a function of the number of training rounds. SampleRank performs competitively with contrastive divergence, and in some cases, outperforms it completely, especially in the early stages of learning. While this experiment provides strong evidence that SampleRank can learn good RBM representations of data, we caution that there is no consensus in the literature on how to evaluate the quality of these learned models [37]. One reason SampleRank outperforms CD is that it minimizes an loss function that is more similar to reconstruction error than maximum likelihood. For this reason, we also include a visualizations of the SampleRank-trained (with greedy loss-augmented inference, and 40 hidden units) RBM in Figure 5.4. As is typical for RBMs trained on this dataset, some filters correspond to full digits, others corresponds to strokes, or parts/composition of digits. For example, filter 1 appears to be a “0”, filter 35 appears to be a slanted “8,” and filter 3 appears to be a “5.”

Finally in Figure 5.5, we provide a visualization of the ranking constraints encountered during SampleRank (sr-full) training. Each row corresponds to a round of inference initialized at the data (in this example, a “1”), and in each row we show the first five pairwise

constraints encountered in that round. Each constraint is simply a pair of states that appear on the Gibbs chain: the left (odd-numbered) state in each pair is more similar to the data than the right (even-numbered) state. This result partially alleviates our initial concern about the SampleRank procedure generating arbitrary states for ranking (Q3). Note, however, that the variant of SampleRank which ranks against the truth (in addition to ranking consecutive states) performs better than the variant which only ranks consecutive states (see Figure 5.6)

5.3.3 SampleRank for RBM Conclusion

In this section, we demonstrated that SampleRank is capable of training unsupervised generative models such as restricted Boltzmann machines. We showed that SampleRank performs competitively with contrastive divergence, and sometimes learns models that are better able to reconstruct the test-data. We also found that the pairwise constraints improve performance and should be used in combination with the data constraints; unlike the structured prediction setting, pairwise constraints on their own perform poorly for training RBMs.

Based on our empirical study, we make the following recommendations for using SampleRank to train RBMs:

- **Use loss-augmented inference.** We found that loss-augmented inference is crucial for training RBMs with SampleRank. However, this was not the case for training discriminative factor graphs. We conjecture that loss-augmented inference addresses the local-smoothness problem discussed in recent work [98]; in future work we will directly evaluate this hypothesis.
- **Use the full-sweep Gibbs sampler.** Unlike structured prediction problems in which the loss function is defined over all the variables, the loss function for RBMs is only defined on the visible units. Therefore, the sampler must modify both visible and

hidden variables in each step. We found that the Gibbs sampler used for contrastive divergence provided this functionality and also worked well for SampleRank.

- **Use a conservative learning rate.** We found that adaptive learning rates such as MIRA were too aggressive and would overfit to the current training example.
- **Do not use L2 regularization.** We found that regularization was not needed and actually hurt performance. It is a commonly held belief that generative models do not suffer from the same type of overfitting problems from which their discriminative counterparts suffer [37]. Further, the non-convexity of the hidden units causes the model to underfit with respect to the theoretical number of degrees of modeling freedom.
- **Do not use parameter averaging.** In contrast to the results we observed on discriminative models, parameter averaging caused the model to perform poorly. We conjecture that this is in part due to the non-convex nature of RBMs, and in part due to the fact that the initial weights are poor and dominate the average. Another reason related to non-convexity is that the hidden units cause non-identifiability issues in which multiple numberings of the hidden units produce the exact same model. Averaging over these models would then cause the weights to wash-out to the uniform distribution. We conclude—in the context of Q4—that non-convexity is not an issue for SampleRank, but might be a problem for parameter averaging.

Finally, we note that there are many tricks available for training RBMs that were not employed in this study. We imagine that many of the techniques that have been successful for CD/PCD training would also apply to SampleRank. For example, momentum, second-order gradient descent methods, tuned learning rate schedules, mini-batches, and drop-out.

5.4 Conclusion

In this chapter we presented SampleRank, a learning algorithm that embeds parameter updates inside of MCMC. SampleRank is capable of training nearly any graphical model for which MCMC is tractable and efficient; thus, enabling practitioners and researchers to design and train models that capture the complex structures necessary to achieve state-of-the-art accuracy [123].

We demonstrated that SampleRank is capable of training a wide variety of models with varying degrees of structural complexity. In each case, we found that SampleRank was either competitive with state-of-the-art learning algorithms or much better. Even for simple models such as linear chain CRFs, we found that SampleRank learns weights that achieve similar prediction accuracy as exact maximum likelihood and structured SVM, but is orders of magnitude faster at training them. Thus, SampleRank is useful even when exact learning algorithms are available because it allows practitioners to quickly perform many rounds of error analysis and feature engineering (because as we demonstrated, training only takes seconds or minutes on classic NLP datasets).

Applying SampleRank to a new model requires choosing (1) a problem-specific evaluation function such as F1 or accuracy and (2) a suitable MCMC procedure for performing inference in the model. We caution that practitioners should exercise care in making these selections because these two components can interact in unexpected ways. For example, SampleRank might struggle if the evaluation function contains many local optima on the local-search manifold defined by the proposal distribution. Further, if the evaluation metric is not able to distinguish between the intermediate configurations produced by the sampler, then learning becomes slow since the stochastic gradients in these cases are zero. Hamming accuracy and single-site Gibbs sampling is often desirable, because the behavior is easier to understand, but even this combination can suffer from the aforementioned zero-gradient problem if there are a large number of classes for each output variable. In these cases, we can either include additional constraints between the ground-truth configuration

and the current sample to completely circumvent this problem, or initialize the sampler to the ground-truth labels before each round of learning to mitigate the problem.

temperature	loss-augmented	Δ hidden	Δ visible	Δ total	% updated	error
0.01 (greedy)	false	3.0	51.7	100.0%	54.8	23.8
0.01 (greedy)	true	15.9	137.6	100.0%	153.5	20.4
1.0 (sampling)	false	22.3	95.4	117.7	100%	21.0
1.0 (sampling)	true	30.0	126.1	155.0	71.0%	22.1

Table 5.6: Run-time statistics for the SampleRank-data-only algorithm. Δ -hidden is the average number of hidden units that differ between the data and the sample during the course of the run, Δ -visible is analogously defined for the visible units, and Δ -total is their sum. %-updated is the percentage of Gibbs samples that lead to an update (some samples might not lead to an update if the ranking constraint is already satisfied). Error is the reconstruction error on the held out test data.

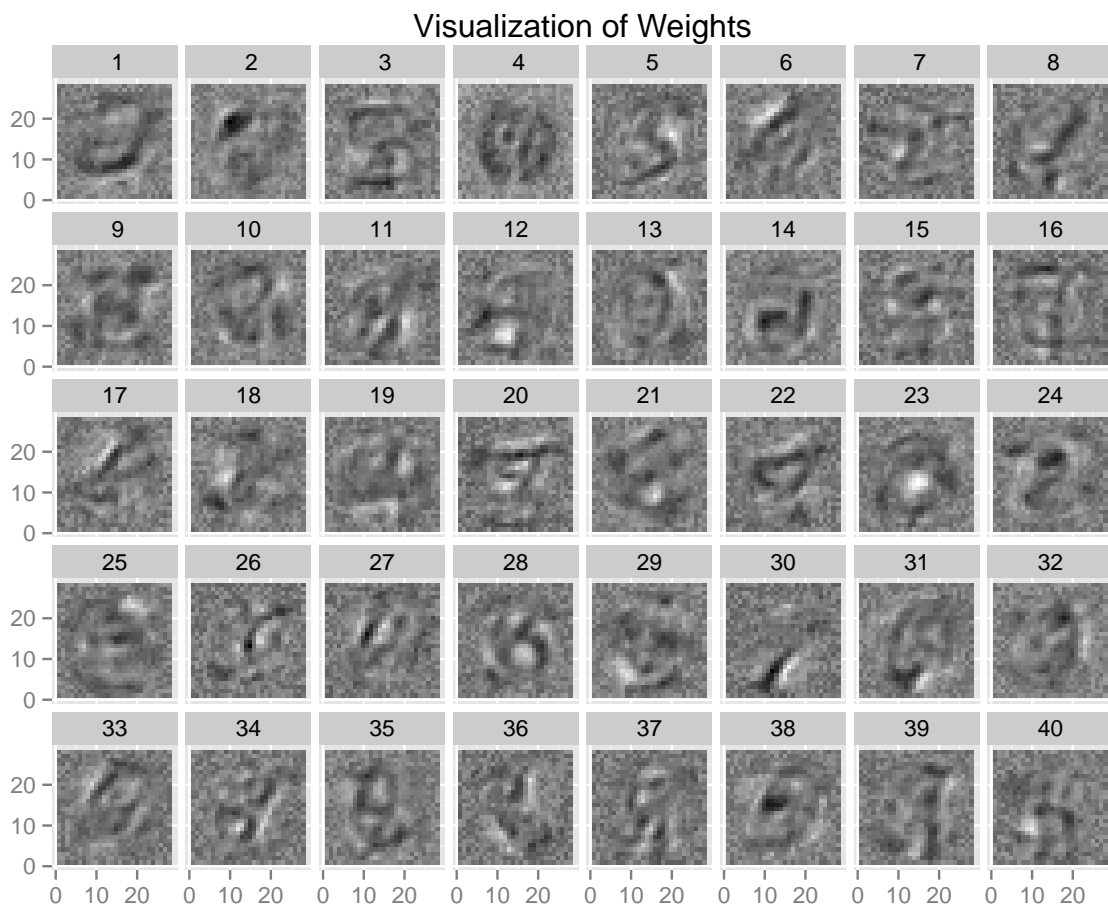


Figure 5.4: RBM filters learned by SampleRank on MNIST digit recognition dataset. The filters are typical for this dataset: some filters represent full digits or composition of multiple digits, others represent edges. The R code for generating these images was adapted from Andrew Landgraf's post: <http://www.r-bloggers.com/restricted-boltzmann-machines-in-r/>

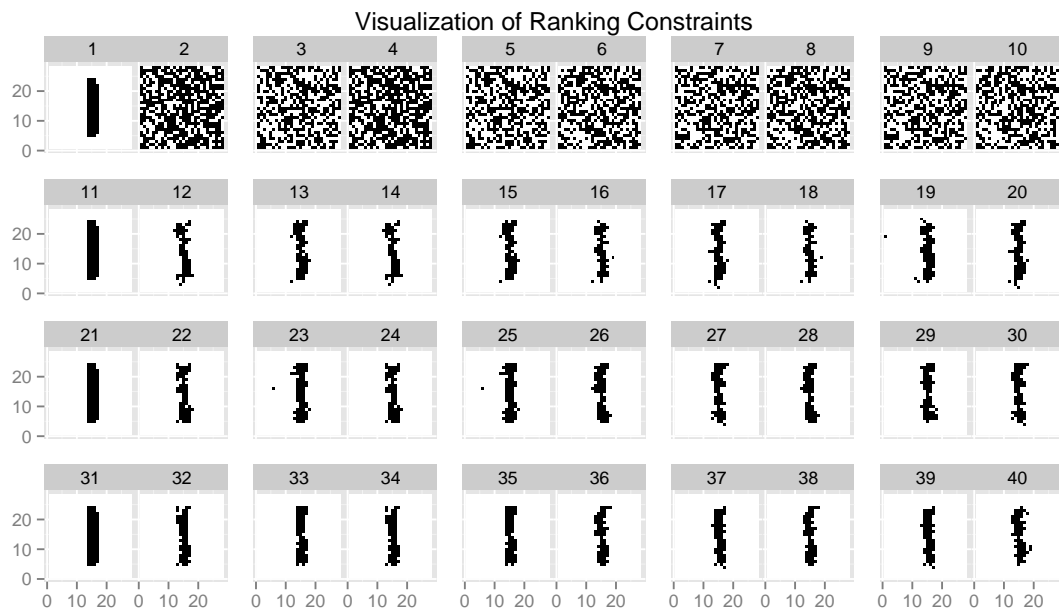


Figure 5.5: Ranking constraints produced by SampleRank during the first four rounds of learning. Each row is a round of learning in which the sampler is initialized to the truth; the first five constraints are shown for each round. The “better” configuration is shown on the left (odd numbers) and the “worse configuration is shown on the right (even numbers). Each row is a new round of learning, initialized at the truth.

Comparison of RBM Training Methods

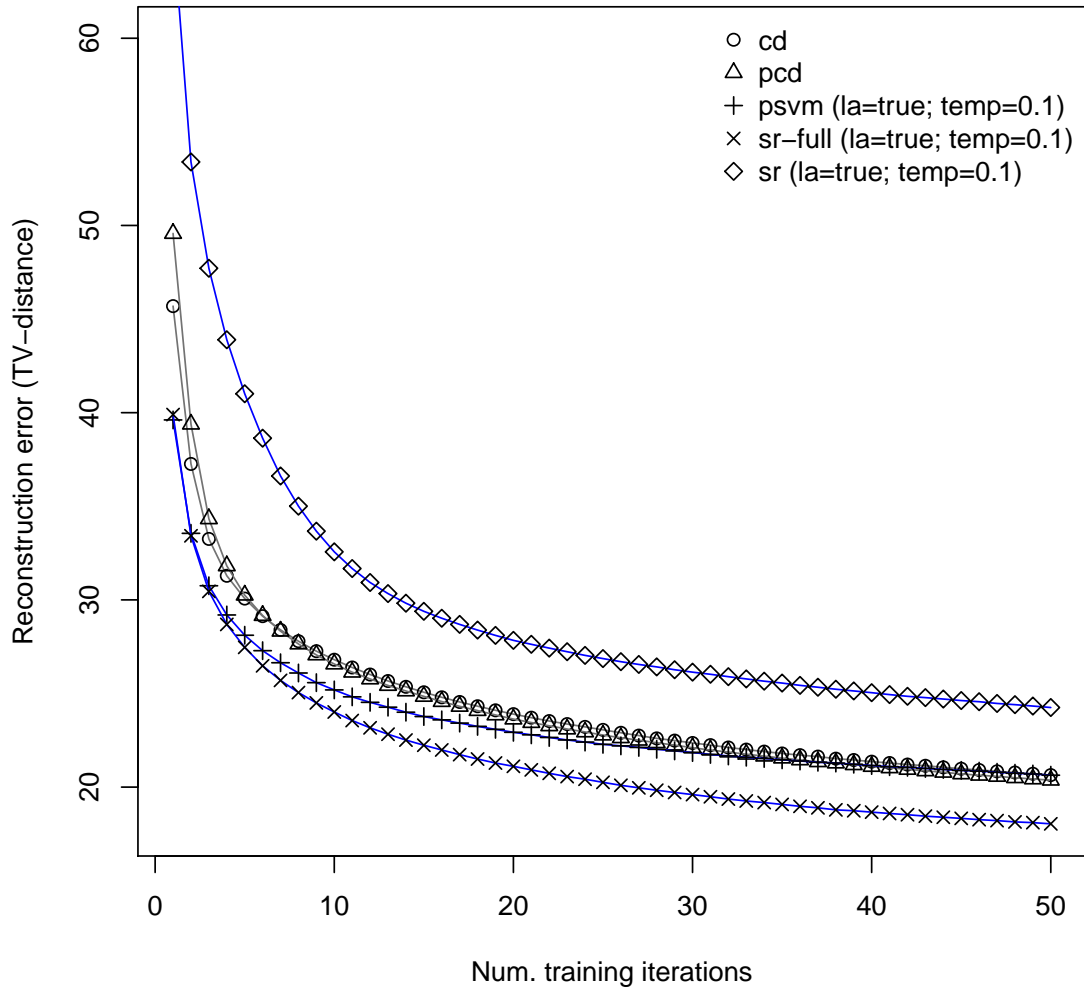


Figure 5.6: Rate at which the different RBM trainers reduce test-time reconstruction error.

CHAPTER 6

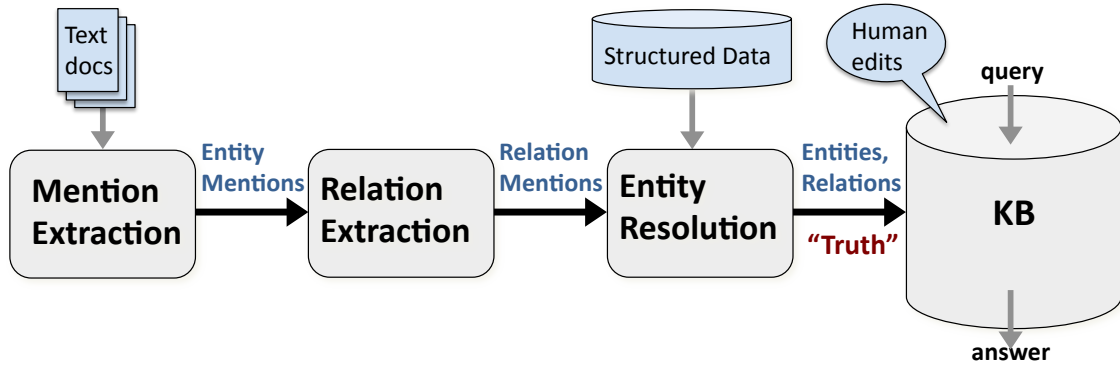
EPISTEMOLOGICAL DBS AND APPLICATIONS

Knowledge base construction is never fully accomplished in a single integration attempt; rather, it is a Sisyphean task which must be solved incrementally as new information becomes available: web spiders continue to provide the KB with raw text, HTML and PDF documents; the KB gradually downloads new rows of external databases; and groups of collaborative users submit suggested edits and changes to the content of the KB. Much of this new evidence is relevant to current extraction and integration predictions, but the traditional approach to knowledge base construction cannot easily revisit these predictions 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 even more likely to have a large effect on previous conclusions. In order to directly handle the unrelenting nature of KB construction, we propose the framework of epistemological databases.

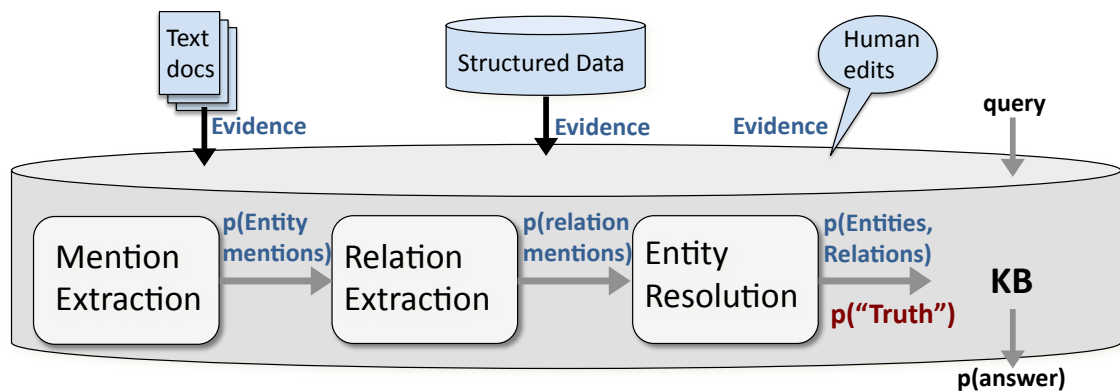
In this chapter, we provide the general definition of epistemological databases, describe a specific implementation thereof that builds upon the work presented in this dissertation, and present experiments that demonstrate the potential utility of such the framework.

6.1 Epistemological databases

An epistemological DB is 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



(a) The traditional approach to automated knowledge base construction where an IE/Integration pipeline injects entities/relations directly into the DB.



(b) An Epistemological Database where the truth is inferred from evidence, not injected directly into the DB.

Figure 6.1: Traditional KB (top) vs. epistemological KB (bottom).

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 6.1). 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?

6.1.1 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) \mathcal{W} endowed with a probability distribution $p : \mathcal{W} \rightarrow [0, 1]$ s.t. $\sum_{w \in \mathcal{W}} p(w) = 1$. 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 employing PDB software directly, 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: inference can lazily materialize other possible worlds as required (e.g., when new evidence causes the model to prefer a new world over the current world) [109].

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 employ 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 [48], coreference resolution [21, 68], information integration [112, 118], and machine vision [104]. Factor graphs are more rigorously defined in the Section 2.1.

The factor function in the epistemological database might include many of 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.

6.1.2 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 primary goal of inference is to find the possible world y^* that is most probable under the model given the evidence (i.e., *maximum a posteriori* (MAP) estimate defined in Equation 2.3).

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, temperature-regulated Markov chain Monte Carlo (MCMC) is a viable approach to inference in complex graphical models [68, 53] that is sufficiently flexible to satisfy these requirements [109], and in some cases, provide provable bounds on the global accuracy of the MAP solution[41]. 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 (as we describe later in this section). We describe MH and MCMC more formally in Section 2.1.1.

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. 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 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 “Assistant Professor” to “Associate Professor”. Note that in practice there can be multiple MCMC inference workers working in parallel [109, 88], with lightweight locking mechanisms for asynchronous inference [85].

Much of the efficiency of MH comes from the form of the acceptance function (Equation 2.2): 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 [88], 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.

6.1.3 Prioritized inference for incremental KB construction

In Chapter 4 we explored the idea of query-aware MCMC, in which sampling is focused on the variables that are important for answering a probabilistic query. In this section, we consider a related problem in which sampling focuses on the variables that are important for integrating new observed evidence. Prioritization is especially important for incremental KB construction applications in which there are a large number of variables, but only a small subset of them are actually affected by the new evidence.

We begin by formalizing the incremental KB construction setting. Let each element $\mathbf{x}^{(t)}$ of the sequence $\mathbf{x}^{(0)}, \mathbf{x}^{(1)}, \dots$ denote the data available for KB construction at that time t . Usually this data accrues gradually over time (i.e., $\mathbf{x}^{(t)} = \mathbf{x}^{(t-1)} \cup \delta_{\mathbf{x}}^{(t)}$); thus for each time step, we assume that data monotonically increases ($\mathbf{x}^{(t-1)} \subseteq \mathbf{x}^{(t)}$), and that the amount of data increases by a small amount ($|\mathbf{x}_{\delta}^{(t)}| \ll |\mathbf{x}^{(t-1)}| \forall t > 0$). The goal of incremental KB construction is to construct and maintain a KB $\mathbf{y}^{(t)}$ out of the available data $\mathbf{x}^{(t)}$ so that the KB is as accurate as possible at each current time step t (e.g., measured by the F1 accuracy of coreference/alignment, classification accuracy of mention types, or utility for solving some real-world task). Note that at time t , we have access to our data integration decisions from time $t - 1$.

For most real-world applications it is prohibitively expensive to re-run inference from scratch each time new data arrives (and wasteful since the amount of new data in each time-step is usually small); therefore, the traditional strategy is to focus integration inference on

the new data, and leave the old integration decisions $\mathbf{y}^{(t-1)}$ unchanged [15]. While this approach is efficient, it may yield poor accuracy for cases in which predictions on the new data are not independent of predictions on the old data (e.g., NLP tasks that require cross-document considerations and/or joint inference).

As argued in this dissertation, Epistemological DBs naturally address the problem of incremental KB construction. The DB contains a set of random variables $\mathbf{Y}^{(t)}$ for the current data $\mathbf{x}^{(t)}$. Moreover, because the DB runs never-ending inference, it also has access to the best estimate of the MAP assignment to the variables from the previous time-step $\mathbf{Y}^{(t-1)} \leftarrow \hat{\mathbf{y}}^{(t-1)}$. Thus, by continuing to run never-ending inference in the epistemological DB, we are able to address the incremental KB construction problem in a way that enables revisitation of previous decisions. Furthermore, because time is limited and not all integration decisions are equally important, we can associate a priority with each integration variable. For example, for situations in which the new data and old data are not highly dependent, we could more aggressively prioritize inference on the new prediction variables than on revisitation of the old prediction variables.

We propose the framework of prioritized inference for addressing this problem. More precisely, given a current estimate $\hat{\mathbf{y}}^{(t-1)}$ of the MAP configuration at time $t - 1$, data $\mathbf{x}^{(t)}$, and a set of prediction variables $\mathbf{Y}^{(t)}$, the goal is to maintain and utilize a set of priorities α for each prediction variable during inference in order to make progress towards the new MAP solution $\hat{\mathbf{y}}^{(t)}$ as quickly as possible. Note that this setting deviates substantially from traditional inference problems in statistics and machine learning in which all variables are considered to be equally important (a small handful of papers on query-specific inference notwithstanding [16, 14]). We describe a prioritized inference algorithm that is specific to hierarchical coreference in Section 6.1.4.

6.1.4 Prioritized MCMC for Coreference

One problem with our basic implementation of the hierarchical coreference MCMC procedure (Algorithms 3&4) from Chapter 3, is that it divides computational resources evenly amongst the coreference decision variables. However, this is not ideal for the incremental KB construction setting because many of the coreference prediction variables associated with existing mentions are unlikely to change; only the subset of variables affected by the new evidence will have the largest impact on the new MAP configuration. Let $M^{(t)}$ be the current set of mentions and $E^{(t)}$ be the set of entities corresponding to the current best known assignment $Y^{(t)}$ of M into entity trees. Given a new set of mentions M_δ , the goal of inference is to identify a new assignment of mentions $M^{(t)} \cup M_\delta$ to entities $E^{(t+1)}$ that has the highest probability in the shortest time possible.

In order to solve this problem with prioritized inference, we seek a set of priorities α_{ij} (each corresponding to a coreference decision variable Y_{ij}) and a modification of our MCMC inference algorithm that is capable of updating and exploiting these priorities to integrate the new mentions as quickly as possible (for example, a procedure that selects a coreference decision variable to change according to its priority). Priorities could be static (computed once upon observing M_δ) or dynamically updated during inference.

One possible solution would be to treat the current knowledge base as a set of “known entities,” and then modify the sampler in the previous section to perform entity-linking between the new mentions M_δ and the existing entities $E^{(t)}$ (that is, assign the priorities for the coreference decision variables between M_δ and $E^{(t)}$ to be 1 and the priorities for all other coreference decision variables to be 0). However, this strategy might fail because the new mentions provide evidence that could potentially change the global state of coreference. For example, a new mention $m \in M_\delta$ might cause a chain reaction: linking a new mention m to an existing entity $e_j \in E^{(t)}$ causes that entity’s attributes to change; the new attributes then cause the model to realize that the entity is actually coreferent with another existing entity $e_k \in E^{(t)}$; merging these two entities together would cause further updates

to the attributes which might cause some erroneous mentions to be removed from the tree. This in turn might cause another new mention $m' \in M_\delta$ to be merged into the entity. Thus, (1) decisions regarding the placement of new mentions can cause far-reaching changes to previous coreference decisions, (2) priorities of variables should change dynamically in response to new inference.

We must modify our MCMC algorithm in a way that causes such a chain reaction to occur as quickly as possible. One possibility would be to *explicitly* introduce a set of auxiliary variables that encode the priority of each coreference decision variable, and then sample coreference decisions variables in accordance to these priorities. During inference, we could dynamically update the values of the priorities as coreference decisions change. For example, if a new mention is merged into an existing entity, we might consider increasing the values of the auxiliary variables between that entity and other existing entities (thus causing the sampler to propose merging the newly modified entity with higher probability).

However, in practice, a method that requires explicitly maintaining a set of priorities for each variable is impractical. First, dynamically updating the priorities (auxiliary variables) every time coreference changes would be slow (for example, when an entity changes during coreference, the decision variables between that entity and all other mentions and entities in the KB would need to be updated). Second, selecting a decision variable in proportion to its priority would be expensive to perform in the inner-most loop of MCMC (although constant time selection algorithms such as the “alias” method exist, they do not apply because the underlying distribution of variable priorities is dynamically changing). Finally, MAP inference would become much more expensive because auxiliary variable schemes in MCMC require marginalizing out the auxiliary variables (i.e., the priorities).

With this discussion in mind, we employ an alternative approach. Rather than explicitly modeling the priorities of each coreference decision, we instead propose a modification our proposal distribution that *implicitly* models a set of dynamically changing priorities on coreference decision variables. Specifically, we change the *nextNodePair* algorithm

Algorithm 9 A “prioritized” implementation of *nextNodePair* for hierarchical coref.

Input: current state $\mathbf{y}^{(t)}$, amount of prioritization b

Output: a pair of nodes $\langle n_i, n_j \rangle_1^m$

$\beta \sim \text{BERNOULLI}(b)$ //Decide whether to sample new mention or old.

//Sample a node by first sampling a mention, then picking a node from the tree.

$$m_i \sim Pr(\cdot | \mathbf{y}^{(t)}, \beta) \quad \text{where } Pr(n_i | \mathbf{y}^{(t)}, \beta) = \begin{cases} \frac{1}{|M^{(t+1)}|_{\#nodes}} & \text{if } \beta = false \\ \frac{1}{|M_\delta|_{\#nodes}} & \text{if } \beta = true \end{cases}$$

$$n_i \sim Pr(\cdot | m_i) \quad \text{where } Pr(n_i | m_i) \propto \begin{cases} \frac{1}{|n_i|_{\#leaves}} & \text{if } n_i \in ancestorsOf(m_i) \\ 0 & \text{otherwise} \end{cases}$$

//Sample the second node in the usual way.

$$c_k \sim Pr(\cdot | n_i) \quad \text{where } Pr(c_k | n_i) = \frac{1}{|canopiesOf(n_i)|}$$

$$n_j \sim Pr(\cdot | c_k) \quad \text{where } Pr(n_j | c_k) = \frac{1}{|c_k|}$$

(*ceteris paribus*) to select a mention m_i from the set of new mentions with probability b or select a mention from the set of all mentions with probability $1 - b$. That is, we substitute Algorithm 9 for Algorithm 4 in the implementation of Algorithm 2. Although this modification appears minor, it causes sophisticated changes in sampling behavior. For example as the new mentions are moved between subtrees during sampling, their presence in those subtrees increases the probability that other mentions in the subtree are selected for sampling. Thus, this modification has the desired effect of catalyzing the aforementioned inference *chain-reactions*.

6.2 Bibliometrics: constructing a database of all scientists in the world

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. Our grand goal is to produce a probabilistic KB containing every scientific researcher in the world.

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 are developing a prototype epistemological DB for bibliographic KB construction. We will focus primarily on supporting coreference resolution (between authors, papers, venues and institutions) inside the DB, and using the epistemological methodology to incorporate human corrections to DB content.

6.2.1 Author mention-finding

Author mention finding is the problem of transforming raw textual citations into records that correspond to a particular author. For example, the mention-finder should convert the citation

Approximate lineage for probabilistic databases. C. Re, D. Suciu. VLDB, 2008

into two author mentions, each of which corresponds to one of the authors:

id	Name	Co-auth	Title	Venue
1	C Re	D Suciu	Approximate lineage for probabilistic DBs.	VLDB
2	D Suciu	C Re	Approximate lineage for probabilistic DBs.	VLDB

In our system, we implement mention-finding as a two stage process. First, we use a linear-chain CRF to segment the fields of the citation on a per-token basis; we use the standard BIO labeling for the author fields (person-first, person-middle, person-last, person-suffix) in order to allow the system to distinguish between multiple authors. Second, we recombine tokens with consecutive labels into fields, and populate the author mentions accordingly.

We use a standard set of features for our CRF. Let L be the domain of the label space and let T be the domain of the token feature space. The features for each token include the token’s word, and various Boolean propositions about the word (e.g., is the word numeric, is the word capitalized, is the word at the start of the citation, etc). Furthermore, each token contains the features of its next and previous token (according to order in the citation). The factors of the linear-chain CRF include the traditional transition factors ($L \times L$), emission factors ($L \times T$), and bias factors L . We learn the weights for each feature using the SampleRank algorithm [110], and use Viterbi for inference.

6.2.2 Author coreference

Given a set of author mentions, author coreference is the problem of partitioning them such that all the mentions in a particular partition refer to the same real-world author. For example, we might partition the “C Re” author mention from the paper “Approximate lineage for probabilistic databases” with the “C Re” author mention associated with the paper “Probabilistic databases: diamonds in the dirt” because we believe both mentions of “C Re” refer to the same person; however, we might place the “C Re” mention associated with the paper “Postnatal depression by another name” into a different partition since it is unlikely that the same person would author papers on such diverse topics (probabilistic databases and medicine). In a relational DB this might be represented by a table with columns for mention ids and entity ids. The set of possible worlds for given mention table would then be all possible assignments of entity ids to mentions.

We model the problem of author coreference using our hierarchical coreference model from Chapter 3. As described in Section 3.6.1, each author mention contains bags-of-words for research paper titles, publication venues, and first-initial-last-names of co-authors. Our hierarchical coreference model includes factors that measure the cosine similarity between topic vectors (inferred with LDA on the publications and venue tokens), and cosine sim-

ilarity between the co-author bags. For a complete list of factors, please refer back to Table 3.1.

6.2.3 Joint Mention-Finding and Coreference

Recent work has demonstrated the importance of joint inference in information extraction; in particular, joint inference is shown to improve the accuracy of citation matching (coreference of research paper titles) and citation segmentation (the primary step in author mention-finding) over approaches that solve the two tasks in isolation [68, 86]. Therefore, we also include factors in our model that span the tasks of author coreference and mention-finding, and are analogous to those used in related work on joint inference. In particular, we augment the mention-finder with factors that inspect the fields of the inferred author entities. If a citation c contains an author mention m which is clustered with an author entity e , then we add a feature for all unigrams, bigrams, and trigrams in c that appear in e 's titles, venue and co-author word bags: the feature indicates the bag in which the n-gram appears, and also includes a weight that reflects the frequency with which it appears in that particular bag. Note that through these joint factors, the problem of citation segmentation (and author mention extraction) is no longer independent on a per-citation basis; that is, mention extraction on newly acquired citations can provide evidence for previous mention extraction predictions. Also note that conditioned on a particular coreference prediction, the model reduces to a linear-chain in which Viterbi exactly finds the best prediction of the labels (but this prediction is conditioned on the current coreference assignment). Thus, to perform joint inference, we can alternate running MCMC sampling for author coreference and Viterbi for mention-finding.

6.3 EpiDB Experiments (bibliographic)

In this section we study epistemological DBs for the problem of incremental KB construction (the setting in which never-ending integration/extraction must incorporate new

data as it becomes available). The data includes both structured (i.e., rows from an existing KB) and unstructured (i.e., raw natural language text). We focus on the tasks of mention extraction and coreference (deduplication of mentions into entities) because these tasks are foundational to KB construction. We also focus on the bibliographic application domain described in the previous section.

6.3.1 Datasets

We use three bibliographic datasets in our experiments: a labeled dataset for evaluating author coreference (unstructured), a labeled dataset for evaluating mention extraction (structured), and a larger unlabeled dataset of author mentions.

The author coreference dataset (Rexa) [20] contains 1459 labeled author mentions of 280 entities with ambiguous author names (see Table 3.3). The author mentions in this dataset are automatically extracted from the bibliography section of research paper PDFs via a linear-chain conditional random field. As a result, this data contains a number of extraction errors, making coreference even more difficult.

The citation dataset (UM-C) [3] contains 1788 citations, 43700 tokens labeled from a space of thirty-six labels (e.g., person-first, person-last, journal, title, date, booktitle, department, etc.).

Finally, we use DBLP [49] as an external database that provides our KB with a source of structured (i.e., manually segmented into mention records) evidence. DBLP contains a total of five-million author mentions, a million of which are directly relevant (i.e., a mention that share a first-initial last name combination with at least one author mention from the citation dataset) to the citation extraction dataset, and twenty-seven-thousand of which are directly relevant to the Rexa dataset.

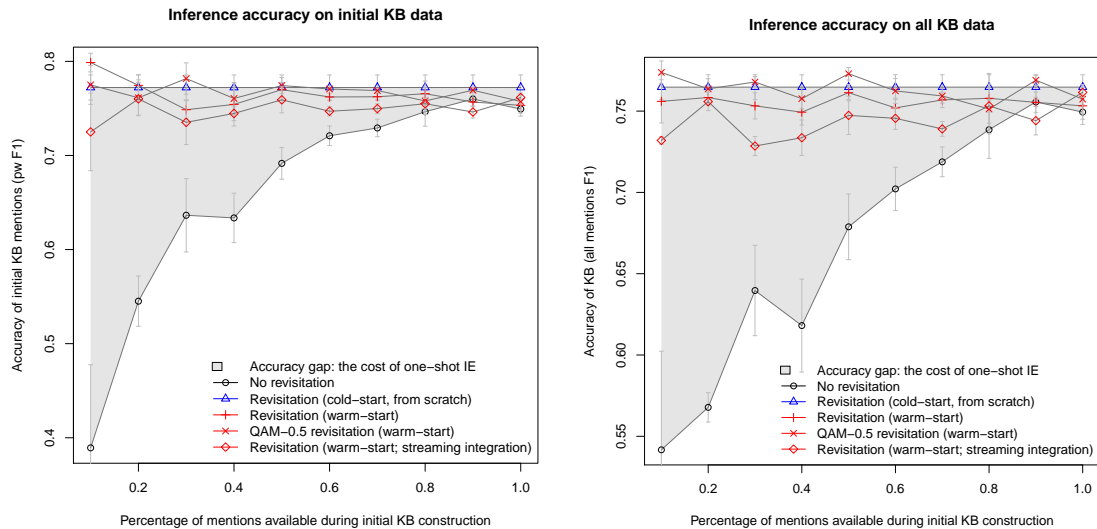
6.3.2 KB management systems and models

We study the following KB management systems

- *One-shot* KB construction. The classic KB construction approach in which the output of an information extraction and integration pipeline is input into the database as the truth. The pipeline processes new data as it arrives, but cannot revisit past inference decisions on existing data.
- *Redo from-scratch (multi-shot)* KB construction. The opposite extreme of one-shot in which inference is re-run from scratch each time new data arrives. This approach is far too expensive to be employed on large KBs, but serves as an ideal upper bound on revisitation accuracy in our experiments.
- *Basic warm-start revisitation for epistemological* KB construction. We term this warm-start because instead of redoing all the predictions from scratch the old predictions are constantly revisited via the never-ending inference procedure.
- *Prioritized warm-start revisitation for epistemological* KB construction. Same as above, except inference is focused in the neighborhood of the new evidence, allowing the evidence’s influence to quickly propagate throughout the KB. The parameter b controls how much the procedure focuses on the influence of the new evidence.

6.3.3 Experimental design

We experimentally simulate the incremental KB construction setting using a two phase experimental design. In the first phase, we present each KB construction/management system with an initial set of data and allow the systems to construct the initial KB; note that there is no difference between the KBs constructed by different systems in this phase. In the second phase, we incrementally provide each system with new unseen data and give the systems the opportunity to integrate the new evidence into to their respective KBs. We evaluate how well each system is able to integrate the new data by evaluating the accuracy of the extractions on both the initial data (i.e., the data provided in the beginning of the experiment) and the full data (i.e., the data available to the system at the current time



(a) Coreference accuracy of initial KB after future mentions become available as evidence. (b) Coreference accuracy of entire KB after future mentions become available as evidence).

Figure 6.2: The benefits of revisitation upon acquisition of new data. New data provides evidence for inference on old data.

step). We distinguish between the initial data predictions and the full data predictions in order to isolate the effects of inference revisitation. In this section, we evaluate both entity resolution and mention finding finding accuracy, but focus on entity resolution in our first set of experiments and mention finding in our last experiment.

6.3.4 Incremental KBC with Epistemological DBs

In our first experiment, we demonstrate that the traditional approach to KB construction makes correctable errors in the incremental KB construction setting, and study the extent to which epistemological DBs can correct these errors. We focus on coreference of automatically extracted author mentions from the Rexa dataset, and adopt the two-phased experimental design described in the previous section. In the first phase, the systems construct their initial KBs using only a randomly selected subset of the automatically extracted author mentions. In the second phase, the systems integrates the remaining mentions. We experimentally control the percentage of mentions available in the first phase. In the con-

text of this experiment, the systems behave as follows. The one-shot “no revisitation” KB construction system runs coreference on the initial set of mentions, and then runs coreference on the remaining set of mentions; however, this system is not capable of revisiting past coreference decisions. That is, it is only able to link the new mentions to old entities, or create new entities from the new mentions. In a sense, this system performs entity-linking in phase 2 against the KB constructed in phase 1. In contrast, the system that redoes inference from scratch is able to completely ignore the initial KB, and simply run coreference on all the mentions. This system functions as an ideal upper-bound on the accuracy we would expect our system to achieve, and the gap between these two curves measures the cost of failing to revisit inference when new data arrives. We vary the portion of initial data and report the accuracy of the predicted author entities for each system (Figure 6.2a displays accuracy of initial data and Figure 6.2b displays accuracy of the full data). Both figures show the average of three random trials with standard error bars.

In Figure 6.2a we evaluate the coreference accuracy of the initial input mentions. Thus, the shaded area represents the cost of one-shot IE; specifically, it is the gap in accuracy between the one-shot IE system which cannot revisit previous conclusions, and the ideal (but prohibitively expensive for large data) system which has the ability to re-run IE from scratch when new data arrives. As expected, the cost of one-shot KB construction increases as the amount of initial data decreases.

In order to understand the extent to which epistemological DB’s can close this accuracy gap through revisitation of past decisions, we also report the accuracies of different revisitation strategies (annotated with “warm-start” in the legend of Figure 6.2a). We find that these systems are able to correct a majority of the initial errors without having to re-run inference from scratch. Furthermore, the effectiveness of the revisitation strategies does not decrease as the amount of initial data decreases. Thus, epistemological DBs are beneficial even for cases in which a substantial portion of the desired pre-integration data is missing during the initial KB construction phase. These results are encouraging because re-running

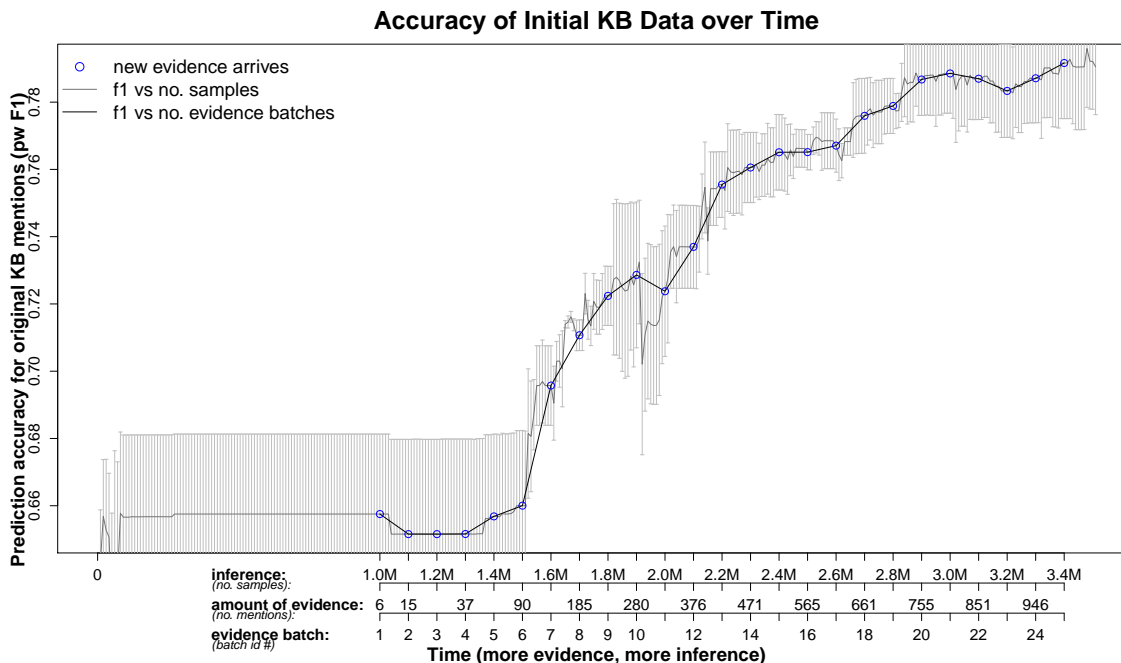


Figure 6.3: Epistemological DB management of an incremental KB construction task. The author coreference accuracy of the original mentions in the KB is plotted as a function of time. Accuracy is recorded every 1000 samples, blue dots indicate a new batch of author mentions. Standard errors reported over three random runs (with different splits of the data).

KB construction from scratch is usually prohibitively expensive, especially for large KBs such as DBLife [23, 15].

We also run a variation of this experiment in which the systems use the entire Rexa dataset to construct the initial KB in phase 1, and then incrementally integrate author mentions from an external database (DBLP) in phase 2. DBLP differs from Rexa in that the author mentions are manually curated and lack extraction errors; we therefore consider DBLP to be “structured” data. Although DBLP is both cleaner and more up-to-date than the Rexa, the dataset only contains an additional five publication seasons (2007-2012). However, even though DBLP contains only a marginal number of additional relevant mentions, we still observe a 5% reduction in error from integrating the DBLP mentions using epistemological DBs.

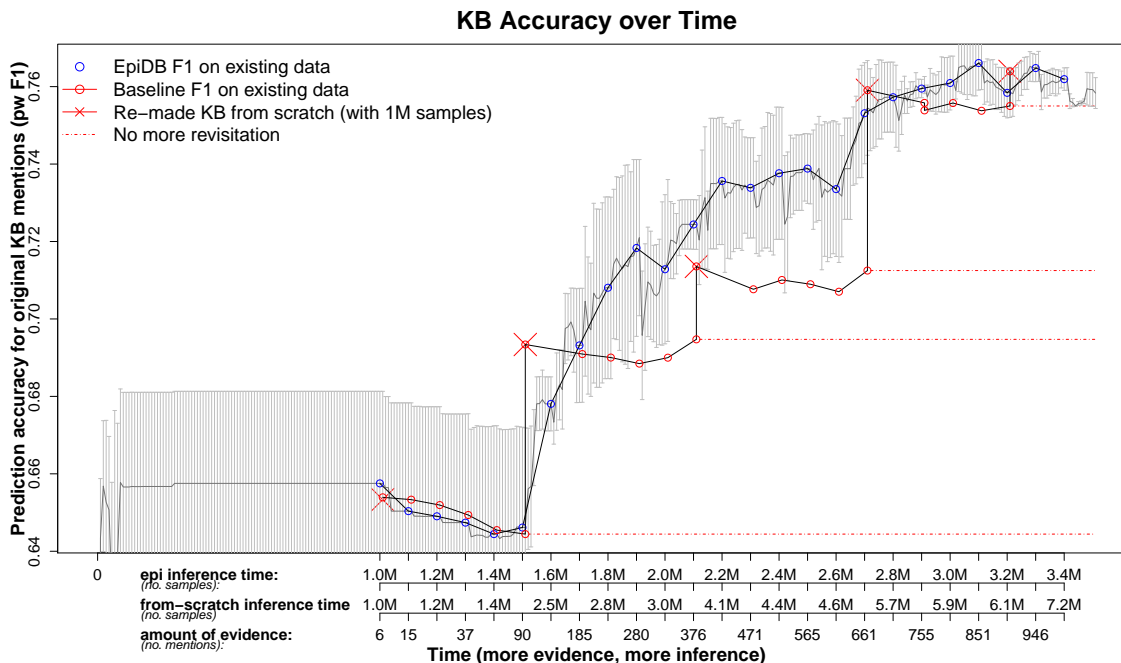


Figure 6.4: Epistemological DB management of an incremental KB construction task vs a baseline approach that periodically re-runs inference from scratch (and runs greedy inference in between these restarts). The author coreference accuracy of all existing mentions in the KB is plotted as a function of time. Accuracy is recorded every 1000 samples, blue/red dots indicate a new batch of author mentions: blue dots show the accuracy for the EpiDB system and red dots show the accuracy for the baseline system. Red X's show accuracy of baseline DB after inference is redone from scratch (1M samples). Dashed relines show the accuracy of the KB if no more inference is done. Standard errors reported over three random runs (with different splits of the data).

Next, we experimentally simulate an incremental KB construction environment in which we can evaluate the ability of the epistemological DB to manage the KB and retroactively correct errors over time. In this experiment, we randomly split the Rexa data such that 1/3 of the mentions are available during the initial KB construction phase, and 2/3 of the mentions become available later during incremental KB construction (we divide this portion of the data into 25 batches, the first five of which only contain mentions of entities not present during the initial KB construction, and the remaining 20 of which contain an equal number of randomly selected mentions). We allow the epistemological DB to begin constructing the initial KB on the initial 1/3 of the data by beginning to run the never-ending MCMC

inference procedure. After one-million steps of MCMC, we begin inputting new mention batches at 100k sample intervals.

We record the accuracy of the initial set of mentions (1/3 of the data) every 1000 steps, and plot the results in Figure 6.3. Note that the first five batches of mentions do not help the system correct errors on the initial set of mentions. This is expected because these first five batches contain only mentions of entities that were not present during the initial construction phase (and are therefore unlikely to provide evidence to the coreference model). However, as the remaining 20 batches of mentions are added, the accuracy of the initial set of mentions sharply rises. Note that error bars are reported every 1000 steps of inference, and are the standard error computed over three random runs. In each of these three runs, we use a different split of the data, and thus the error bars are large in the beginning due to randomness in the initial data, but gradually decrease in magnitude as an increasingly large portion of the entire dataset becomes integrated.

We further compare the epistemological KB maintenance strategy to an exemplary configuration of the traditional KB management approach. In particular, we implement a baseline system that periodically re-runs inference from scratch, but runs greedy inference in between each restart. This baseline system is exemplary in that it combines the complementary strengths of the from-scratch restart system (high accuracy) with the strengths of greedy inference (speed). In Figure 6.4 we compare these KB integration systems on the author coreference problem. We observe that the epistemological system performs better than the baseline in between the baseline's restarts (restarts depicted by red "X"s), but after most restarts the baseline achieves slightly higher accuracy. Of course, in practice, regenerating a KB from scratch at such frequent intervals is computationally expensive and this cost is not visually depicted in Figure 6.4.¹

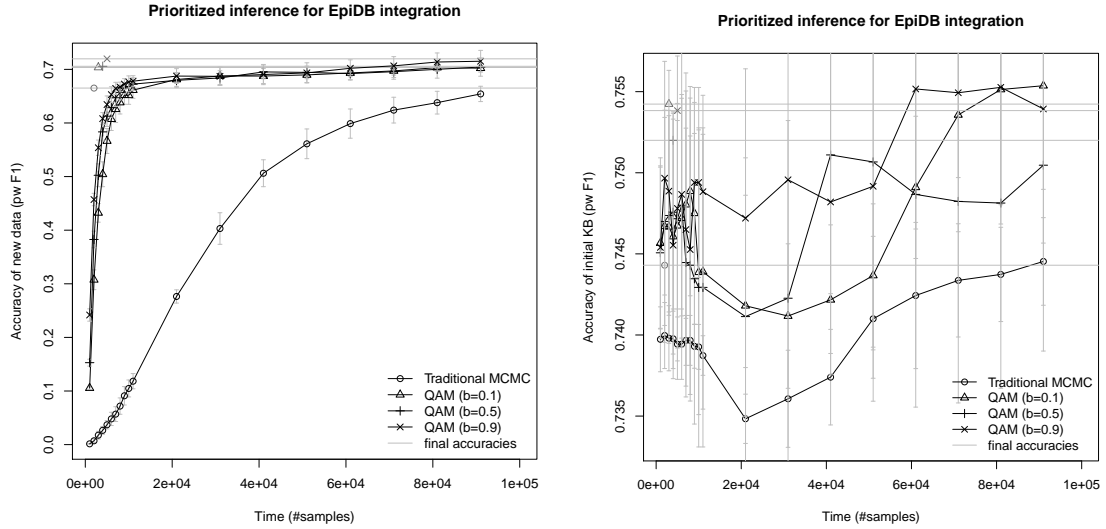
¹Refer to the additional time scale on the x axis to get a sense for the computational differences.

In summary, this first experiment demonstrates that there can indeed be a cost to the traditional one-shot KB construction and that epistemological DBs are effective at overcoming this cost while managing data in an incremental setting. Clearly, the extent to which new data improves the accuracy of past predictions depends on how much novel information that new data provides. Although, we do not systematically investigate this phenomena, we observe its effects in our current experiments. For example, there are few redundant citations present in the Rexa data. As such, when a portion of the Rexa mentions are missing, the accuracy of the initial predictions are affected to a large degree (and the degree to which it is effected depends on how many mentions are missing during the initial KB construction phase). In contrast, there is a substantial amount of redundancy between Rexa and DBLP (in particular, DBLP contains redundant copies of most of the author mentions in Rexa). Therefore, the addition of DBLP as a new source of evidence does not have the same amount of impact as Rexa (only a 5% reduction in error).

6.3.5 Prioritized inference

We also demonstrate the importance of prioritized inference for epistemological DBs. For this experiment, we simulate a setting in which the current KB is large, and only a small number of new mentions arrive. Again, we use the two-phased experimental design. In the first phase, we create an initial KB with all of the relevant DBLP mentions² and half of the Rexa mentions (the Rexa mentions are for evaluation purposes). In the second phase, we add the remaining Rexa mentions, and integrate them using different inference strategies. We compare a prioritized inference strategy (which at each step, either selects from the set of new mentions with probability b , or selects from the set of all mentions with probability $1 - b$) to the baseline strategy of selecting mentions with equal probability at each step. In Figure 6.5a we plot the coreference accuracy of the second half of the Rexa mentions every 1000 steps of MCMC. The prioritized inference strategies are able to corefer the new

²That is, in the same canopy.



(a) How inference prioritization affects the accuracy of the KB. (b) How inference prioritization affects the initial KB accuracy.

Figure 6.5: The benefits of revisitation upon acquisition of new data. New data provides evidence for inference on old data.

mentions much more quickly than the strategy that equally samples all mentions. Next, in Figure 6.5b, we plot the coreference accuracy of the initial set of mentions. Unfortunately, we are unable to demonstrate that prioritized inference improves the accuracy on the old data more quickly than the baseline because the error bars are large relative to the difference in accuracy between the initial and final accuracies. The reason is that the Rexa mentions added in phase 2 are mostly redundant because the publication years covered by Rexa are a subset of those covered in DBLP (Rexa only goes up to 2006, but our version of DBLP goes up to 2012). Thus, we would expect the initial accuracy of the KB to be high with little room to improve.

6.3.6 Joint inference for multi-modal tasks

Finally, we evaluate epistemological DBs for a multi-modal joint inference task that combines mention finding and coreference resolution. We use the model and features described in Section 6.2.3. Using the two-phased experimental design, we demonstrate

how newly acquired mentions can provide evidence that allows the epistemological DB to retroactively correct errors in previous mention-finding decisions. In the first phase, we run joint extraction and coreference on the UMass citation dataset. Then, in the second phase, we include new author mentions from two structured sources: (1) the fully structured (manually segmented) author mentions corresponding to DBLP citation records and (2) author mentions from the fully structured (manually segmented) versions of the UMass citation records. In phase 2 we re-run joint inference (an iteration of coreference followed by segmentation). We find that new input data can cause up to a 20% reduction in citation segmentation error³ (the pre-cursor to author mention finding) when both sources are utilized, and a 5% reduction in error when only DBLP is utilized (the small reduction in error is because DBLP does not overlap much with the citation dataset). In Figure 6.6 we plot the number of errors corrected for each field due to revisitation (alphabetically by field). The field-label “all” is actually a combination of all fields, “joint” is the set of fields that would most directly benefit from joint inference (person-first, person-last, B-person-last, journal, booktitle), and “non_joint” are the remaining fields.

Upon inspection, we determine that the error reduction is indeed due to revisitation during joint inference with new evidence. For example, consider the following citation from the UMass citation dataset:

*Mitra P, Murthy C Pal S: Unsupervised Feature Selection Using Feature Similarity.
Transactions on Pattern Analysis and Machine Intelligence 24 (3), 301–312.*

The underlined segments are errors of the initial citation segmenter, which misclassifies “Similarity” as a Journal and “Analysis and Machine” as a title. Our initial KB thus contains three errorful citations (one for each co-author). Fortunately, DBLP contains ad-

³Reduction in per-label F1 error.

ditional mentions of these authors, including the following records:

Name	Co-authors	Title	Venue
C.A. Murthy	T. Basu	Extraction by Supervised Fea..	ICDM
C.A. Murthy	C. Pal,...	Unsup... using Feature Similarity.	Pattern Analysis...

Despite the extraction errors, coreference correctly resolves the errorful “C Murthy” mention to the manually curated DBLP records. This consequently provides new evidence to the citation segmenter that “Machine” and “Analysis” should actually be venues (because they are venues in the DBLP record) and that “Similarity” should actually be a title. In this particular case, the citation segmenter corrects these errors through successful revisitation. This example also highlights the difference between our incremental KB construction problem setting, and the extraction-over-evolving text problem setting (cf. [15]). In particular, note that the original raw-text of the citation did not change; therefore, the matching algorithm of Cyclex would not have been able to recognize the fact that we need re-run mention-finding on this snippet of text. Instead, Cyclex would have recycled this errorful extraction because the original text remained unchanged.

6.4 Text and Entity Linking Experiments

In Section 3.6.2 we outlined an approach for solving entity-linking with hierarchical coreference. The key idea is to solve entity-linking as a coreference problem (rather than classification) in which all known entities are treated as mentions, and jointly clustered (along with all the other available mentions) into inferred entities. The advantage of this approach is that hierarchical coreference is non-greedy and can retroactively change previous linking decisions as it continues to construct the inferred entities.

Mention-extraction error correction due to multi-modal joint inference on new evide

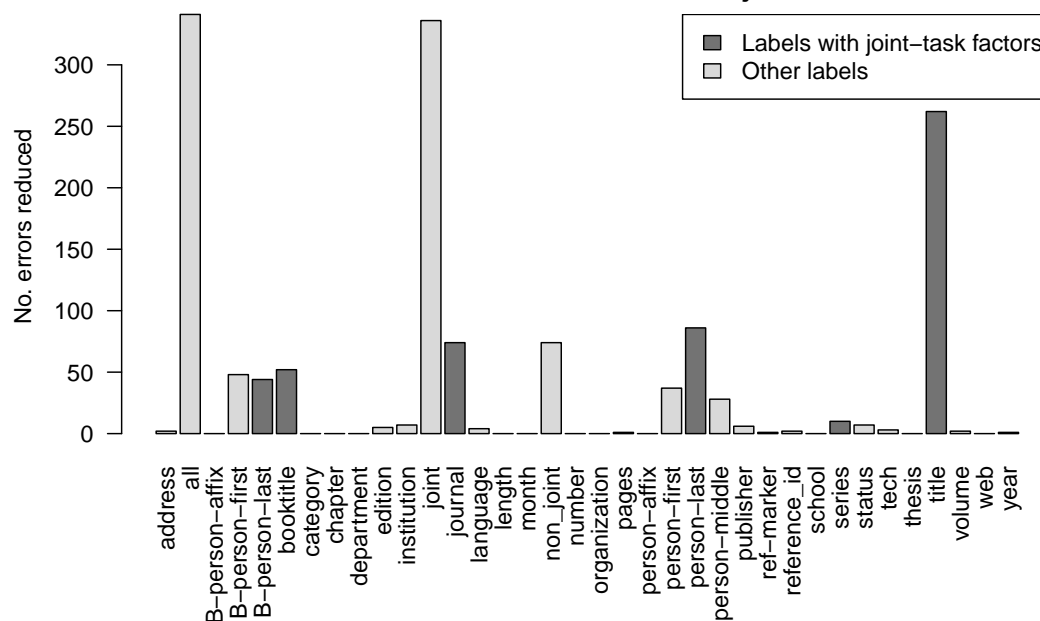


Figure 6.6: New evidence allows multi-modal joint inference to correct citation extraction errors on old data via predictions on new data. Error reduction is 20%.

6.4.1 Data

For the entity-linking experiments, we use the Wikilinks [63] dataset[87] in combination with Wikipedia. Wikilinks is a collection of blogs that contain hyper-links to Wikipedia pages. The anchor texts of these hyper-links are treated as mentions, and the Wikipedia page to which they link is treated as the “ground-truth” entity to which the mention refers.

For each Wikilinks mention we create a record of the context that contains various attributes including (1) a bag-of-context-words of the tokens in the blog from which it was extracted (2) a bag-of-mention-words of the tokens from other mentions in the blog (as identified by a named entity recognition tool), and (3) a bag-of-name-words containing the tokens that appear in the surface form of the mention’s anchor text.

We also process Wikipedia in a similar fashion. First, we employ the Freebase type hierarchy to identify the person, organization, and location entities in Wikipedia. We ex-

tract each of these Wikipedia pages as a *mention* of a real-world entity,⁴ which we populate with a set of features that are homologous to those that we extract for Wikilink mentions. In particular, each Wikipedia mention contains (1) a bag-of-context-words of the tokens from the Wikipedia page (2) a bag-of-mention-words of other anchor texts that appear in that page, and (3) a bag-of-name-words consisting of all the tokens in the Wikipedia title plus all the tokens from anchor texts of other Wikipedia pages that link to this page. For example, if Michelle Obama’s Wikipedia page were to link to Barack Obama’s Wikipedia page via the anchor text “husband,” then we would extract “husband” as an additional name for the name-bag of the Barack Obama Wikipedia mention.

For the purpose of our experiments, we identify two particularly ambiguous subsets of the combined Wikilinks and Wikipedia data. Specifically, we create one dataset of consisting entirely of “Boston” related organizations and another dataset consisting entirely of “New York” related organizations. The Boston dataset contains all the Wikilinks and Wikipedia mentions that refer to the following Wikipedia entities: Boston (the city itself), the Boston Celtics (professional basketball team), the Boston Red Sox (professional baseball team), the Boston Bruins (professional hockey team), and the Boston Globe (newspaper). The New York dataset includes: the New York Yankees (baseball), the New York Knicks (basketball), the New York Rangers (hockey), the New York Giants (Football), and the New York Jets (also Football). Each dataset has approximately 5000 mentions, and each entity has between 500 and 1800 mentions.

We chose these two subsets because they are especially challenging: organizations that are named after the cities to which they belong are ambiguous since they have similar context and overlapping names (e.g., the names of the organizations contain the words “Boston” and “New York” respectively). Furthermore, it is common practice in blogs to refer to a particular sports organization simply by the name of the city from which they are

⁴Yes, each Wikipedia entity is made a “mention” in our system. In this way Wikipedia can be naturally aligned to other pre-structured KBs and mentions from text.

based. For example, “Boston” could refer to the “Boston Celtics,” the “Boston Red Sox,” or the “Boston Bruins” depending on the context. Additionally, sports teams often have overlapping context words such as “beat,” “goal,” and “score.” Finally, sports organizations tend to have many nicknames. For example, the “New York Yankees” are also known as the “Bronx Bombers” and the “NY Highlanders,” and “Boston” is also known as “Beantown.” In comparison, people and most other organizations are on average significantly easier.

6.4.2 Systems and baselines

We manually set the parameters for the model described in Section 3.6.2. For these experiments we tune the parameters on the Boston dataset, and use the New York dataset to evaluate the coreference systems and baselines. We evaluate the following systems:

- **String-match:** this system clusters all mentions that have the same canonical name string.
- **Entity-linking (MCMC)** this system treats the Wikipedia mentions as a set of known entities. During inference, the entity-linking systems only considers MCMC moves that would either add or remove a link between a Wikilinks mention and an entity that contains a Wikipedia mention. This system cannot create new entities.
- **Entity-linking (streaming-k)** same as above, except instead of using MCMC for inference, it makes k passes over all the Wikilinks mentions. It visits each mention (one at a time) and attempts to merge it with the Wikipedia entity for which it has the highest model score (or none if all the scores are negative). A value of $k = 1$ is the traditional streaming setting where the system must make one decision for each mention before moving on to the next [70]. A higher value of k allows the system to revisit an old decision which could be more accurate since more mention context has been aggregated in the entity.

Method	PW F1	Link Acc.
String matching baseline	83.6	91.3
Entity linking (streaming-1)	83.7	92.0
Entity linking (streaming-2)	83.9	92.2
Entity linking (streaming-4)	84.0	92.2
Entity linking (MCMC)	84.0	92.2
Joint linking+discovery	97.3	98.2

Table 6.1: Evaluation of Linking and Discovery using pairwise F1 (PW F1) and linking accuracy.

Pre-known Entities withheld	PW F1
None	97.3
only NY Yankees	96.6
only NY Rangers	96.7
only NY Knicks	96.9
only NY Giants	89.5
only NY Jets	89.1
All	89.8

Table 6.2: Evaluating the ability of our system to discover entities, when the various pre-known (Wikipedia) entities are withheld (metric is pairwise F1)

- **Joint linking+discovery** models entity linking and entity discovery jointly and solves the full coreference problem using MCMC (which in contrast to the entity-linking MCMC algorithm, can also consider merging entity trees which do not contain any Wikipedia mentions) in our hierarchical coreference model from Chapter 3.

6.4.3 Results

In this section we evaluate the joint entity-linking and entity discovery approach. First, in Table 6.1 we compare the joint approach to several commonly employed baselines. We find that solving the full joint coreference problem (evaluating both coreference and entity-linking accuracy) achieves a 75% reduction in error versus the closest approach. This result indicates that current procedures to entity-linking (for example, in TAC-KBP) could be greatly improved by jointly solving the nil-clustering problem rather than deferring it as a post-processing step.

#Mentions			PW F1
additional	seed	total	(on seeds)
0	2275	2275	88.4
759	2275	3034	89.8
1518	2275	3793	95.5
2275	2275	4550	96.6

Table 6.3: Evaluating the effect of additional mentions on the performance of coreference resolution (NY dataset).

Next, we evaluate our system’s ability to perform entity-discovery (that is, coreference of mentions for which we lack known Wikipedia page). We simulate missing entities by withholding Wikipedia pages from the NY dataset and then evaluating our system on the modified data. We report the results in Table 6.2. Note that some entities are more difficult to discover than others; for example, the system performs worse when withholding one of the two football team’s (Jets and Giants) Wikipedia page because the mentions are more contextually similar. However, overall, our system still achieves relatively high accuracy (approximately 90% F1) even when all the Wikipedia pages are withheld.

Finally, we examine how the number of mentions impacts the accuracy of our coreference resolution system. Note that as the number of mentions increases, the size of the search space grows exponentially making coreference more difficult. However, the amount of available information about each entity also increases which should on the other hand have the effect of making coreference easier. In this experiment, we evaluate coreference accuracy on a fixed subset of the mentions, but vary the number of additional mentions input to coreference. Table 6.3 shows that adding additional mentions helps coreference more accurately resolve the fixed set of seed mentions. This result highlights the importance of building scalable coreference systems.

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

6.5 Human machine cooperation

6.5.1 Overview

Automated KB construction with information extraction (IE) is promising because it can scale to large data, but unfortunately lacks the reliability to be trusted by real-world decision makers. Conversely, manual KB construction efforts are highly precise, but lack the coverage of IE. We will combine these two complementary approaches using the framework of epistemological databases: human “corrections” are simply another source of evidence that participates in inference.

Users will communicate corrections to the database using the language of mentions. For example, a user might express that the “Fernando Pereira” that studies NLP is the same “Fernando Pereira” that worked with Prolog. These statements are converted to mentions which are then resolved by hierarchical coreference. Additional potentials allow the model to balance its trust in human edits against other available evidence. To achieve this balance, we propose to augment the model with latent variables representing reliabilities/reputations of users and jointly learn them using MCMC.

We will support edits that (1) correct coreference precision errors, (2) correct coreference recall errors, and (3) correct entity attribute errors. The model will incorporate these edits using the appropriate factors (e.g., should-link/should-not-link constraints between mentions, preferring that canonical entity attributes are derived human-provided mentions, etc.) and use priority-driven MCMC to propagate the influence of the user-introduced mentions. We anticipate that some errors will be difficult to correct due to local optima on the search space, thus it will be necessary to learn specially designed MCMC transitions for decisions involving human-introduced mentions.

6.5.2 Related work

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 [25]. 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 [79]. 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 [23]. Chai et al. [13] 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. [44] 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 [45] 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.

6.5.3 Human edits to coreference predictions

We will use the hierarchical model for coreference between authors, papers, venues and institutions to create a bibliographic database where users will be able to browse author/venue pages, notice coreference errors, and propose corrections. 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 *must-link* and *must-not-link* constraints in constrained clustering problems; however, they are not deterministic, but extra suggestions via factors.

Human-provided coreference edit introduces two new mentions to the DB which are each annotated with the information pertinent to the edit. For example, consider the recall error depicted in Figure 6.7a. 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 do not 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 worked with Prolog is the same Fernando Pereira who works on natural language processing (NLP)”. Presenting this edit to the bibliographic database involves creating two mentions, one with keywords about Prolog 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 Markov chain Monte Carlo (MCMC) inference explores possible worlds by moving mentions between entities, the factor graph rewards possible worlds where the two mentions belong to the same entity. For example, see Figure 6.7b. In our experiments, a similar coreference error is corrected by an edit of this nature.

6.5.3.1 Experiments on author coreference

Our experiments simulate edits by generating them from manually annotated ground-truth coreference data. An advantage of this synthetic data is that (1) we can exert more precise control over the quality of the edits, (2) we can collect larger numbers of edits, and (3) we can avoid any potential biases or limitations introduced via the UI (for example, the deterministic display orders of papers highly biases the users to edit certain mentions more than others).

For these experiments, we generate ten initial KBs by running our coreference model for one-million steps of MCMC sampling with different random seeds. For each KB, we generate user edits by using the ground-truth to determine whether mentions should be added (SL edit) or removed (SNL edit). We run additional experiments using edits generated from the ground-truth annotations. An advantage of this synthetic data is that (1) we can exert more precise control over the quality of the edits, (2) we can collect larger numbers of edits, and (3) we can avoid any potential biases or limitations introduced via the UI (for example, the deterministic display orders of papers highly biases the users to edit certain mentions more than others).

For our synthetic experiments, we generate ten initial KBs by running our coreference model for one-million steps of MCMC sampling with different random seeds. For each KB, we generate user edits by using the ground-truth to determine whether mentions should be added (SL edit) or removed (SNL edit) from each author. If removing or adding to an

entity's subtree increases the F1 score then we classify the edit as corrective, otherwise corruptive. To eliminate ambiguous cases (change in F1 score is not indicative of quality for cases in which an author has a high percentage of incorrect mentions), we only generate synthetic edits to predicted author entities in which 80% of their mentions refer to the same ground truth author entity. We stream the synthetic edits to the KB and report the pairwise F1 accuracy averaged over the ten KBs at each step.) from each author. If removing or adding to an entity's subtree increases the F1 score then we classify the edit as corrective, otherwise corruptive. To eliminate ambiguous cases (change in F1 score is not indicative of quality for cases in which an author has a high percentage of incorrect mentions), we only generate synthetic edits to predicted author entities in which 80% of their mentions refer to the same ground truth author entity. We stream the synthetic edits to the KB and report the pairwise F1 accuracy averaged over the ten KBs at each step.

In Figure 6.10, we study the ability of the epistemological system to handle different types of edits (SL,SNL, corruptive, corrective). As expected, the epistemological system is substantially more robust to corruptive edits than the systems that place complete trust in users (overwrite and max satisfy) (Figures 6.10b,6.10d). Indeed, the probabilistic approach considers multiple sources of evidence and is able to ignore 94.5% of the corruptive edits.

Furthermore, the epistemological system is better at applying the corrective SL edits (Figure 6.10a) than the two systems which place complete trust in the users. At first, it may seem surprising that the probabilistic system, which does *not necessarily apply* every corrective SL edit, could possibly yield a better quality KB than the baseline systems, which apply *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 user 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). However, the epistemological system is not as effective at integrating the corrective SNL edits as the baseline (Figure 6.10a). We hypoth-

esize that this is because splitting entities is more difficult than merging entities due to local optima in our search space, but further investigation is needed to confirm this belief.

In Figure 6.8 we compare the systems' ability to integrate all edit types. Since this experiment combines both corrective and corruptive edits, we are able to simulate users with different reliabilities, and thus include a second epistemological system that models user reliabilities. In particular we simulate ten "users," five of whom are malicious and five of whom benevolent; we randomly assign corrective edits to benevolent users and corruptive edits to the malicious users. Even though 30-50% of the edits are corruptive, both epistemological systems are able to increase the accuracy of the KB from 74.8 to 77.7 (without reliability modeling) and 78.6 (with reliability modeling); in contrast, the systems that completely trusts the users quickly succumb to the corruptive users and rapidly degrade the quality of the KB to 56.0% and 54.0% respectively.

In order to achieve a deeper understanding of the epistemological systems (with and without reliability estimation), we plot their ability to correctly integrate both corrective and corruptive edits into the KBs (Figure 6.9). These curves reveal interesting behavior of the systems. First, notice that corrective edits have an accumulative effect (Figure 6.9b): 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., the KB is more confident when there are multiple edits in support of a particular change). Similarly, corruptive edits also produce an accumulative effect in the epistemological system (Figure 6.9a). Indeed, the accuracy of the sans-reliability 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.

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 6.4: Integration of entity-attribute edits.

6.5.3.2 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 ground-truth 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 one correct edit and one partially correct edit for each of these entities and one malicious edit for 50% of these entities.

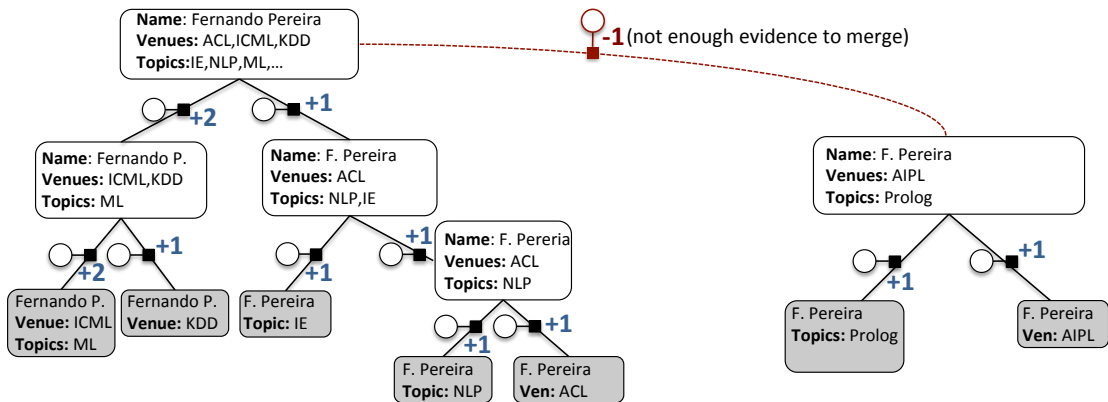
After applying the edits in a random order to the ten KBs, we record the average attribute accuracy, and the average amount of vandalism in Table 6.4. The three systems (resp. overwrite, epistemological with and without reliabilities) all improve the quality of the entity attributes both by reducing 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 the greatest resistance to vandalism (reducing the number of vandalized entities from 50% to 4.03%).

6.6 Conclusion

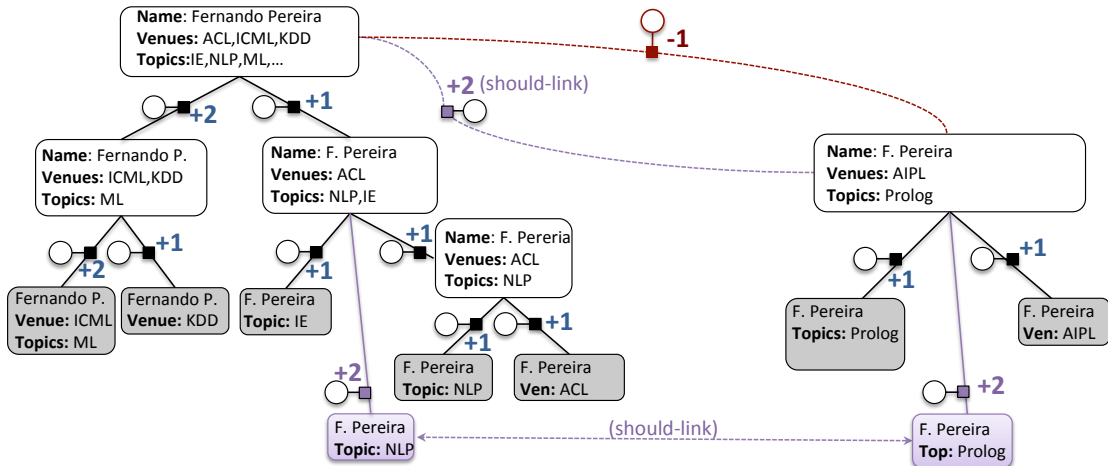
In this section we defined and presented epistemological DBs. The driving philosophical idea behind epistemological DBs is that they never observe the truth about entities and relations, but must instead infer it from available evidence. We proposed a specific implementation of this framework that uses factor graphs to represent the uncertainty in the truth, MCMC inference for discovering the truth when new evidence arrives, and hierarchical coreference for resolving entities.

We found that epistemological DBs are able to effectively integrate a variety of different evidence sources including (1) additional mentions and (2) human edits to the contents of the KB. We also argued that epistemological DBs are well suited for the problem of incremental KB construction, which we formally defined. Not only did we find that the traditional pipelined approaches to KB construction commit correctable errors in this incremental setting, but we further found that epistemological DBs are able to correct these errors (on two separate problem domains: Rexa author coreference and Wikilinks). Finally, we generalized the problem of query-aware MCMC (Chapter 4) to prioritized inference, proposed a specific prioritized sampling method for integrating new mentions in hierarchical entity resolution, and demonstrated it is an order of magnitude faster than the traditional sampler.

In future work we would like to build and deploy an epistemological DB in the wild and study its long-run behavior, especially for the problem of integrating crowd-sourced human contributions with automatically inferred truth values.



(a) A recursive coreference model with two predicted *Fernando Pereira* entities. Black squares represent factors, and the numbers represent their their log scores, which indicate the compatibilities of the various coreference decisions. There is not enough evidence to merge these two entities together.



(b) How a human edit can correct the coreference error in the previous figure. A human asserts that the “Prolog F. Pereira is also the NLP F. Pereira.” This statement creates two mentions with a should-link constraint. During inference, the mentions are first moved into different entities. Then, when inference proposes to merge those two entities, the model gives a small bonus to this possible world because the two should-link mentions are placed in the same entity.

Figure 6.7: A recall coreference error (top), is corrected when a user edit arrives (bottom).

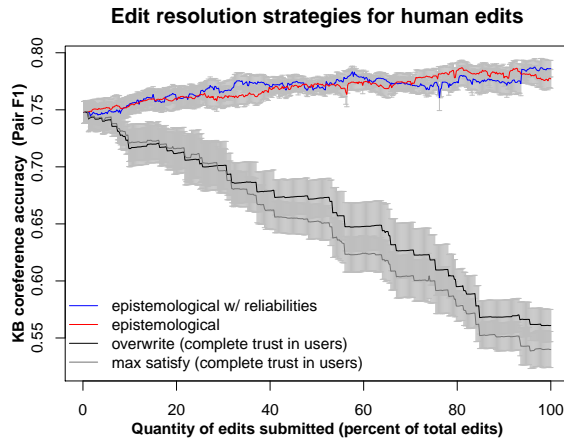
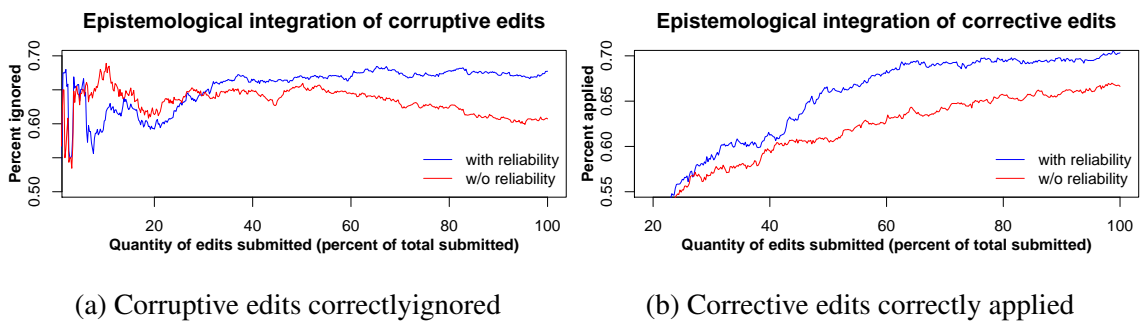


Figure 6.8: Epistemological integration of coreference edit (should-link, should-not-link, corruptive, corrective) with user reliabilities.



(a) Corruptive edits correctly ignored

(b) Corrective edits correctly applied

Figure 6.9: The user reliabilities improve epistemological integration of corrective edits (applies a higher percent) and ignore corruptive edits (ignores a higher percent).

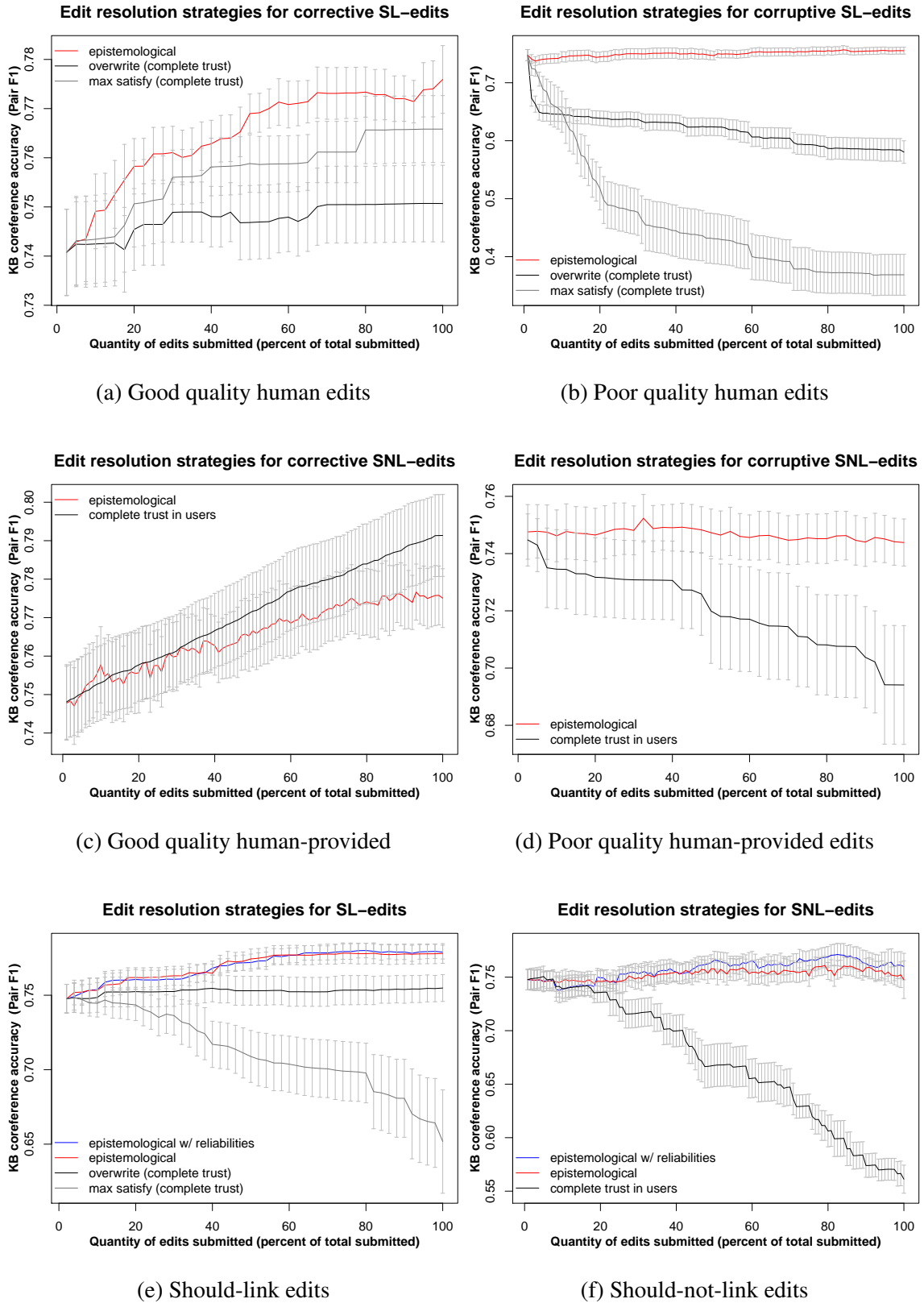


Figure 6.10: A comparison of various approaches to applying human edits to KBs.

CHAPTER 7

CONCLUSION

In this dissertation we tackled a set of core problems for probabilistic KB construction with epistemological databases. We presented solutions to these problems (models, algorithms) in a general way using the formalism of graphical models. Our contributions are summarized next.

7.1 Summary of contributions

- In Chapter 3 we presented a novel hierarchical model for efficiently performing coreference at scale. We found that the model scales much better than a variety of alternatives, including the dominant pairwise approach. We evaluated our model primarily on the problem of author coreference resolution, but also presented ways of adapting our model for the problems of entity linking and wikification.
- In Chapter 4 we posed the problem of focused inference for answering probabilistic queries, and solved it with query-aware MCMC. Unlike most applications of MCMC in which all variables are sampled equally often, query-aware MCMC focuses sampling on the variables that matter most for answering the query. We evaluated a variety of heuristics for tuning which variables to prioritize, and found that under finite time, these methods significantly outperform traditional samplers.
- In Chapter 5 we proposed SampleRank, a parameter estimation algorithm that embeds learning inside of inference. The advantage of this method is that it can be used to train models for which exact learning is not possible; thus, vastly expanding

the set of tractable models for solving real-world problems. We demonstrated that SampleRank better than many other approximate learning methods, and showed it is competitive with exact methods (such as SVM and maximum likelihood) on models for which exact methods are tractable. We also demonstrated that SampleRank can be used to train restricted Boltzmann machines and showed that it is competitive with (if not better than) contrastive divergence.

- In Chapter 6 we formally defined a paradigm-shift for KB construction that we term epistemological DBs. We demonstrated that epistemological DBs are able to effectively infer the truth about entities and relations from available evidence, allowing the system to retroactively correct errors in the incremental KB construction setting. We further demonstrated the epistemological DBs readily support a way of incorporating human edits as yet another source of evidence. We found that the epistemological treatment of user edits is both more accurate and robust than the classic approach of treating user edits as deterministic truths.

Together, these contributions are a concrete step towards building large-scale non-greedy KBs that are able to take advantage of our key assumption: that more data improves the accuracy of probabilistic models of KB construction at inference time. We demonstrate that this assumption holds in two domains (author coreference and newswire entity linking) and for two tasks (coreference in isolation, and joint coreference plus mention-finding), but we conjecture that this assumption will hold in most domains as long as the probabilistic models are sufficiently accurate.

7.2 Potential Impact

Many of the probabilistic methods in this dissertation were formalized in general ways making them suitable for a variety of problem domains, including machine vision, automatic speech recognition, computational biology, and deep learning. For example, we

demonstrated the utility of SampleRank in training complex graphical models on problems in several of these domains. SampleRank’s ability to train models with complex structure will allow practitioners in these domains to design and train even more sophisticated models. The query-aware MCMC inference algorithm was also formalized within the framework of graphical models, making it widely applicable to many problem domains. Finally, the hierarchical coreference problem was demonstrated on three domains: author coreference, Wikilinks coreference, and entity linking. However, the model can be applied to any coreference problem for which rich, disambiguating contextual features exists, and is generally applicable to other clustering problems (especially useful for those in which the number of clusters is unknown and the size of the clusters is heterogeneous). The weights need-not be manually tuned, but can be tuned automatically via SampleRank.

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 transform the way scientific research is conducted. Unfortunately, unlike other domains (e.g., movies and music), there does not exist structured databases that contain accurate and comprehensive lists of the entities apposite to bibliographic data. Many of the ideas presented in this dissertation, such as epistemological DBs and hierarchical coreference provide a concrete step in the direction of eventually constructing such a database (e.g., of all the scientists in the world).

7.3 Limitations, Discussion, and Future Work

Much of the work in this thesis has focused on MCMC. While MCMC is a useful general purpose inference algorithm, for many classes of models, alternatives such as belief propagation or mean field are more appropriate. For example, belief propagation is efficient and exact for linear-chain and poly-tree model structures. The framework of factor graphs, while general, defines course-level abstractions on which general purpose inference algorithms such as MCMC are implemented. However, a finer granularity of abstractions

might be more appropriate for more efficient implementations of inference. For example, knowing that factors are of a particular functional form would allow inference to be implemented much faster (e.g., factors that are convex functions, sub-modular, or exhibit overlapping sub-problems).

A primary focus of this work was the problem of coreference because we believe it to be foundational to KB construction. While coreference is important, other tasks such as relation extraction, mention finding, and attribute extraction are also useful for automatically building KBs, but were not a focus of this dissertation. However, many of the techniques and ideas developed in this dissertation extend beyond coreference. For example, we defined our prioritized inference and parameter learning algorithms generally using the formalism of factor graphs so that they could easily be extended to other KB construction problems in the future.

One of our major application domains was bibliographic data, particularly the problem of building large databases of scientific authors. Indeed, many of the results we presented for our hierarchical coreference model were on author coreference data. Author coreference is an important problem for building bibliograph KBs, and is well representative of a larger general class of coreference problems that includes citation matching, entity-linking, cross-document newswire coreference, among others. First, the types of features useful for disambiguating entities are similar across different coreference domains (e.g., features for capturing variety in entity names, context in which entities are mentioned, extra meta-information, and topical information). Second, when viewing coreference as a clustering problem, the characteristics of the clustering problem are similar across domains: the number of entities (clusters) in author coreference and most other forms of coreference are unknown,¹ and the size of the entities (number of points in each cluster) is heterogeneous and follows a power law. For example, in author coreference some authors

¹Except in entity linking where the number of entities might be known in advance.

are cited more frequently than others and thus there are more mentions of these authors; in entity-linking prominent entities such as Barack Obama are frequently mentioned in newswire articles, but other entities are only mentioned a small handful of times. In either case, the number of prominent entities is small, and there is a long-tail of entities with only a small number of mentions. Third, author coreference spans a range of ambiguity levels, from easy to difficult, to AI-complete. For example, some mentions in author coreference are highly ambiguous because there are many common first-initial/last-name combinations (e.g., “W. Li”), but the problem also contains less common name combinations such as “Simon Peyton-Jones.”

Although author coreference is an archetypal coreference problem, there are also some noteworthy ways in which it is unique. First, the problem of variety is not as extensive in author coreference. For example, although people refer to most authors in a relatively small variety of ways (e.g., “F. Pereira” and “Fernando Pereira,” or “John” and “Jack.”), they refer to the types of entities that appear in Wikipedia in a much larger variety of ways (e.g., “New York,” “the Big Apple,” “NYC,” and “The home of the Mets,” “The Center of the Universe,” “The Five Boroughs,” “The Melting Pot,” “The Capital of the World” are all valid ways of mentioning *New York City*). The lack of variety in author coreference is useful because it makes it easier to distribute the data for parallelized inference. Second, for some author coreference datasets (e.g., DBLP, PubMed, WoS), the number of true mentions per author entity is limited by the number of papers that author has published; thus, the entity sizes are smaller than other types of coreference problems. Consequently, coreference might be faster for these types of datasets. Further, differences in speed and scalability between the hierarchical coreference algorithm and pairwise alternatives are likely to be much larger on coreference problems with larger entities; despite this concern, even on DBLP we found orders of magnitude speed-ups over pairwise coreference. Note that we also evaluate on the REXA dataset which includes citations; thus, the author entities in REXA are much larger, and not limited in size by the publication counts. Finally, author coreference is remarkable

in the sense that there is a dearth of available structured data for assisting the problem. Other problem domains, such as entity-linking and cross-document coreference benefit from existing KBs such as Wikipedia, Freebase, and IMDB. In contrast, there does not exist an accurately maintained KB of all the scientists in the world (or even one with a substantial subset of them). Thus, successful coreference of bibliographic data, especially authors, depends heavily on automated coreference algorithms. On the other hand, when extending our techniques to domains for which structured information is available, algorithms may need to be modified to fully exploit such data. For example, most existing databases are already deduplicated and constraints might need to be added in the coreference model to prevent the model from trying to deduplicate mentions from these sources (as we already have done for Wikipedia mentions in the entity-linking problem).

In summary, the author database domain is representative of a wide variety of KB construction problems in some ways, but not in others. Thus, it is important to keep in mind the aforementioned similarities and differences between domains (along with other properties of the domains) when interpreting the results presented in this dissertation. For example, the lack of variety in author names allowed us to more easily distribute inference in hierarchical coreference. Thus, in order to scale epistemological databases to other domains we must first investigate more general ways of distributing hierarchical coreference (this is an area of ongoing work). Further, focusing on a specific domain such as bibliographics allowed us to more easily build the models required for epistemological databases. Thus, our results might not necessarily extend to more heterogeneous domains in which the accuracy of the models is likely to be worse. For example, if the coreference model performs poorly at disambiguating mentions from a particular domain, then adding mentions of new entities could be harmful since this would increase the amount of ambiguity in the problem. In contrast, this was not a problem in our specific domains: adding mentions of new entities had a positive impact on the model's accuracy. Additionally, more heterogeneous domains might exhibit more types of noise (missing context, spelling variations, typos, contextual variety,

irrelevant background text). Such noise could pose new problems for hierarchical coreference and epistemological databases. For example, newswire text contains much more background noise (e.g., stop words, irrelevant tokens and entities) than author coreference mentions which tend to have more relevant context (the words in the titles are essentially keywords). Thus, studying epistemological DBs robustness to more heterogeneous domains is an important area of future work.

Finally, note that our work on epistemological DBs has focused on improving the quality of the KB through never-ending inference on new evidence. However, in this setting, the KB's accuracy is ultimately limited by the model's abilities. That is, we expect that after a certain point, adding more data no longer improves the accuracy of the KB because we have reached the upper limit of the model's capabilities. Human-provided contributions (e.g., user edits) is one way that the accuracy of the KB is able to continue to improve, but individual contributions usually only effect a relatively small portion of the KB—and although we demonstrated that inference propagates the contribution's impact beyond their intended targets, correcting the remaining errors in the KB via human edits might require a full manual effort in the worst case. Further, since the contributions are treated as evidence, their integration still ultimately depends on the model's abilities. Therefore, we see never-ending learning [12] as an important area of complementary work because it would allow the model's accuracy to improve over time. Although we do not study this problem in the dissertation, we have developed some of the tools necessary to explore this line of work. For example, SampleRank can run inside the never-ending MCMC procedure and treat human-contributions as evidence for both inference and for learning. However, in order to accomplish this at scale, we must develop ways of running SampleRank in the context of parallel and distributed MCMC.

In future work we would like to make further steps towards building a KB of all the scientists in the world, but consider additional bibliographic entity types (e.g., papers, authors, venues, grants). There are strong relations between authors, venues, grants, funding

agencies, and research papers that can easily be derived from joint coreference of these various entity types. Further, exploiting the structure of these relations, including the citation graph, could inspire a more domain-specific approach to solving joint, bibliographic coreference. We are also interested in exploring more general domains, such as Wikipedia and beyond. For these problems we are investigating new algorithms for distributed asynchronous hierarchical coreference that promise to scale additional orders of magnitude. We are also interested in developing more theory for SampleRank in order to better understand the impressive results it achieves on tasks ranging from coreference and mention finding to representation learning in restricted Boltzmann machines, the building block of deep belief networks.

BIBLIOGRAPHY

- [1] Ackley, David H., Hinton, Geoffrey E., and Sejnowski, Terrence J. A learning algorithm for boltzmann machines. *Cognitive science* 9 (1985).
- [2] Amit, B., and Baldwin, B. Algorithms for scoring coreference chains. In *Proceedings of the Seventh Message Understanding Conference (MUC7)* (1998).
- [3] Anzaroot, Sam, Passos, Alexandre, Belanger, David, and McCallum, Andrew. Learning soft linear constraints with application to citation field extraction. In *ACL* (2014).
- [4] Bagga, Amit, and Baldwin, Breck. Cross-document event coreference: annotations, experiments, and observations. In *Proceedings of the Workshop on Coreference and its Applications* (Stroudsburg, PA, USA, 1999), CorefApp '99, Association for Computational Linguistics, pp. 1–8.
- [5] Bengston, Eric, and Roth, Dan. Understanding the value of features for coreference resolution. In *Empirical Methods in Natural Language Processing (EMNLP)* (2008).
- [6] Bhattacharya, Indrajit, and Getoor, Lise. A latent dirichlet model for unsupervised entity resolution. In *SDM* (2006).
- [7] Bilenko, Mikhail, Kamath, Beena, and Mooney, Raymond J. Adaptive blocking: Learning to scale up record linkage. In *Proceedings of the Sixth International Conference on Data Mining* (Washington, DC, USA, 2006), ICDM '06, IEEE Computer Society, pp. 87–96.
- [8] Bollacker, Kurt, Evans, Colin, Paritosh, Praveen, Sturge, Tim, and Taylor, Jamie. Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data* (New York, NY, USA, 2008), SIGMOD '08, ACM, pp. 1247–1250.
- [9] Bremaud, Pierre. *Markov Chains: Gibbs Fields, Monte Carlo Simulation, and Queues*. Springer, 1999.
- [10] Bulatov, Yaroslav. Importance of far away observations in a symmetric binary hmm. 2007.
- [11] Calafiore, G., and Campi, M. C. Uncertain convex programs: Randomized solutions and confidence levels. *Mathematical Programming* 102 (2005), 25–46.

- [12] Carlson, A., Betteridge, J., Kisiel, B., Settles, B., Jr., E.R. Hruschka, and Mitchell, T.M. Toward an architecture for never-ending language learning. In *Proceedings of the Conference on Artificial Intelligence (AAAI)* (2010), AAAI Press, pp. 1306–1313.
- [13] Chai, Xiaoyong, Vuong, Ba-Quy, Doan, AnHai, and Naughton, Jeffrey F. Efficiently incorporating user feedback into information extraction and integration programs. In *SIGMOD Conference* (2009), pp. 87–100.
- [14] Chechetka, Anton, and Guestrin, Carlos. Focused belief propagation for query-specific inference. In *International Conference on Artificial Intelligence and Statistics (AI STATS)* (2010).
- [15] Chen, Fei, Doan, AnHai, Yang, Jun, and Ramakrishnan, Raghu. Efficient information extraction over evolving text data. In *ICDE* (2008), pp. 943–952.
- [16] Choi, Arthur, and Darwiche, Adnan. Focusing generalizations of belief propagation on targeted queries. In *Association for the Advancement of Artificial Intelligence (AAAI)* (2008).
- [17] Collins, Michael, and Koo, Terry. Discriminative reranking for natural language parsing. *Comput. Linguist.* 31, 1 (Mar. 2005), 25–70.
- [18] Crammer, Koby, and Singer, Yoram. Ultraconservative online algorithms for multi-class problems. *J. Mach. Learn. Res.* 3 (Mar. 2003), 951–991.
- [19] Culotta, Aron. *Learning and inference in weighted logic with application to natural language processing*. PhD thesis, University of Massachusetts, May 2008.
- [20] Culotta, Aron, Kanani, Pallika, Hall, Robert, Wick, Michael, and McCallum, Andrew. Author disambiguation using error-driven machine learning with a ranking loss function. In *Sixth International Workshop on Information Integration on the Web (IIWeb-07)* (Vancouver, Canada, 2007).
- [21] Culotta, Aron, Wick, Michael, Hall, Robert, and McCallum, Andrew. First-order probabilistic models for coreference resolution. In *HLT/NAACL* (2007), pp. 81–88.
- [22] Daume, Hal. *Practical Structured Learning Techniques for Natural Language Processing*. PhD thesis, University of Southern California, Los Angeles, CA, august 2006.
- [23] DeRose, Pedro, Shen, Warren, Chen, Fei, Lee, Yoonkyong, Burdick, Douglas, Doan, AnHai, and Ramakrishnan, Raghu. Dblife: A community information management platform for the database research community. In *CIDR* (2007), pp. 169–172.
- [24] Doan, AnHai, and Halevy, Alon Y. Semantic integration research in the database community: A brief survey. *AI Magazine* 26, 1 (2005), 83–94.

- [25] Doan, AnHai, Ramakrishnan, Raghu, and Halevy, Alon Y. Crowdsourcing systems on the world-wide web. *Commun. ACM* 54, 4 (2011), 86–96.
- [26] Farias, D. P. De, and Roy, B. Van. On constraint sampling in the linear programming approach to approximate dynamic programming. *Mathematics of Operations Research* 29, 3 (August 2004), 462–478.
- [27] Finkel, Jenny Rose, and Manning, Christopher D. Joint parsing and named entity recognition. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics* (Stroudsburg, PA, USA, 2009), NAACL '09, Association for Computational Linguistics, pp. 326–334.
- [28] Finley, T., and Joachims, T. Training structural SVMs when exact inference is intractable. In *Proc International Conference on Machine Learning (ICML)* (2008), pp. 304–311.
- [29] 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).
- [30] Giles, C. Lee, Bollacker, Kurt D., and Lawrence, Steve. Citeseer: An automatic citation indexing system. In *Proceedings of the Third ACM Conference on Digital Libraries* (New York, NY, USA, 1998), DL '98, ACM, pp. 89–98.
- [31] Gill, Jeff. Is partial-dimension convergence a problem for inferences from MCMC algorithms? *Political Analysis* 16, 2 (2008), 153–178.
- [32] Givoni, I.E., Chung, C., and Frey, B.J. Hierarchical affinity propagation. In *UAI* (2011).
- [33] Haghighi, Aria, and Klein, Dan. Unsupervised coreference resolution in a nonparametric bayesian model. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics* (Prague, Czech Republic, June 2007), Association for Computational Linguistics, pp. 848–855.
- [34] Haghighi, Aria, and Klein, Dan. Coreference resolution in a modular, entity-centered model. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics* (Stroudsburg, PA, USA, 2010), HLT '10, Association for Computational Linguistics, pp. 385–393.
- [35] Hernández, Mauricio A., and Stolfo, Salvatore J. The merge/purge problem for large databases. In *Proceedings of the 1995 ACM SIGMOD international conference on Management of data* (New York, NY, USA, 1995), SIGMOD '95, ACM, pp. 127–138.
- [36] Hinton, Geoffrey E. Training products of experts by minimizing contrastive divergence. *Neural Computation* 14, 8 (2002), 1771–1800.

- [37] Hinton, Geoffrey E. A practical guide to training restricted boltzmann machines. In *Neural Networks: Tricks of the Trade (2nd ed.)*. 2012, pp. 599–619.
- [38] Hinton, Geoffrey E., Osindero, Simon, and Teh, Yee-Whye. A fast learning algorithm for deep belief nets. *Neural Comput.* 18, 7 (July 2006), 1527–1554.
- [39] Joachims, Thorsten, Finley, Thomas, and Yu, Chun-Nam John. Cutting-plane training of structural svms. *Machine Learning* 77, 1 (2009), 27–59.
- [40] Johnson, Alicia A. *Geometric Ergodicity of Gibbs Samplers*. PhD thesis, Macalester College, 2009.
- [41] Jung, Kyomin, Kohli, Pushmeet, and Shah, Devavrat. Local rules for global map: When do they work? In *NIPS* (2009).
- [42] Kambhatla, Nanda. Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations. In *ACL* (2004).
- [43] Kasneci, Gjergji, Ramanath, Maya, Suchanek, Fabian M., and Weikum, Gerhard. The yago-naga approach to knowledge discovery. *SIGMOD Record* 37, 4 (2008), 41–47.
- [44] Kasneci, Gjergji, Van Gael, Jurgen, Herbrich, Ralf, and Graepel, Thore. Bayesian knowledge corroboration with logical rules and user feedback. In *Proceedings of the 2010 European conference on Machine learning and knowledge discovery in databases: Part II* (Berlin, Heidelberg, 2010), ECML PKDD’10, Springer-Verlag, pp. 1–18.
- [45] Koch, Christoph, and Olteanu, Dan. Conditioning probabilistic databases. *Proc. VLDB Endow.* 1 (August 2008), 313–325.
- [46] Kosik, Kenneth S. The wikification of knowledge. *Neiman Reports* (2008).
- [47] Kulesza, Alex, and Pereira, Fernando. Structured learning with approximate inference. In *Adv. Neural Inf. Proc. Sys.* (2007).
- [48] Lafferty, John, McCallum, Andrew, and Pereira, Fernando. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. 18th International Conf. on Machine Learning* (2001), Morgan Kaufmann, San Francisco, CA, pp. 282–289.
- [49] Ley, Michael. The dblp computer science bibliography: Evolution, research issues, perspectives. In *SPIRE* (2002), pp. 1–10.
- [50] Liu, Jun S., Liang, Faming, and Wong, Wing Hung. The multiple-try method and local optimization in metropolis sampling. *Journal of the American Statistical Association* 96, 449 (2000), 121–134.

- [51] Marthi, Bhaskara, Pasula, Hanna, Russell, Stuart, and Peres, Yuval. Decayed MCMC filtering. In *Conference on Uncertainty in Artificial Intelligence (UAI)* (2002), pp. 319–326.
- [52] McAllester, David, Hazan, Tamir, and Keshet, Koseph. Direct loss minimization for structured prediction. In *Proc International Conference on Machine Learning* (2010).
- [53] McCallum, A., Schultz, K., and Singh, S. Factorie: Probabilistic programming via imperatively defined factor graphs. In *Neural Information Processing Systems Conference (NIPS)* (2009).
- [54] McCallum, Andrew, and Wellner, Ben. Conditional models of identity uncertainty with application to noun coreference. In *NIPS17*, Lawrence K. Saul, Yair Weiss, and Léon Bottou, Eds. MIT Press, Cambridge, MA, 2005.
- [55] McCallum, Andrew K., Nigam, Kamal, and Ungar, Lyle. Efficient clustering of high-dimensional data sets with application to reference matching. In *Proceedings of the Sixth International Conference On Knowledge Discovery and Data Mining (KDD-2000)* (Boston, MA, 2000).
- [56] Meshi, Ofer, Sontag, David, Jaakkola, Tommi, and Globerson, Amir. Learning efficiently with approximate inference via dual losses. In *ICML* (2010), pp. 783–790.
- [57] Mihalcea, Rada, and Csomai, Andras. Wikify!: linking documents to encyclopedic knowledge. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management* (New York, NY, USA, 2007), CIKM '07, ACM, pp. 233–242.
- [58] Milch, Brian, Marthi, Bhaskara, and Russell, Stuart. *BLOG: Relational Modeling with Unknown Objects*. PhD thesis, University of California, Berkeley, 2006.
- [59] Milch, Brian, Marthi, Bhaskara, Russell, Stuart, Sontag, David, Ong, Daniel L., and Kolobov, Andrey. BLOG: Probabilistic models with unknown objects. In *IJCAI* (2005).
- [60] Miller, George A. Wordnet: A lexical database for english. *Commun. ACM* 38, 11 (Nov. 1995), 39–41.
- [61] Milne, David, and Witten, Ian H. Learning to link with wikipedia. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management* (New York, NY, USA, 2008), CIKM '08, ACM, pp. 509–518.
- [62] Mnih, Volodymyr, Larochelle, Hugo, and Hinton, Geoffrey E. Conditional restricted boltzmann machines for structured output prediction. *CoRR abs/1202.3748* (2012).
- [63] Murray, Iain, and Ghahramani, Zoubin. Bayesian learning in undirected graphical models: Approximate mcmc algorithms. In *UAI* (2004).

- [64] Neal, Radford. Slice sampling. *Annals of Statistics* 31 (2000), 705–767.
- [65] Nemirovski, Arkadi, and Rubinstein, Reuven Y. An efficient stochastic approximation algorithm for stochastic saddle point problems. Tech. rep., Technion, 1996.
- [66] Ng, Vincent. Machine learning for coreference resolution: From local classification to global ranking. In *ACL* (2005).
- [67] Pelleg, Dan, and Moore, Andrew. Accelerating exact k-means algorithms with geometric reasoning. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (New York, NY, USA, 1999), KDD '99, ACM, pp. 277–281.
- [68] Poon, Hoifung, and Domingos, Pedro. Joint inference in information extraction. In *Proceedings of the 22nd national conference on Artificial intelligence - Volume 1* (2007), AAAI Press, pp. 913–918.
- [69] Rahman, Altaf, and Ng, Vincent. Supervised models for coreference resolution. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2 - Volume 2* (Stroudsburg, PA, USA, 2009), EMNLP '09, Association for Computational Linguistics, pp. 968–977.
- [70] Rao, Delip, McNamee, Paul, and Dredze, Mark. Streaming cross document entity coreference resolution. In *COLING (Posters)* (2010), pp. 1050–1058.
- [71] Ratnoff, Lev, Roth, Dan, Downey, Doug, and Anderson, Mike. Local and global algorithms for disambiguation to wikipedia. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1* (Stroudsburg, PA, USA, 2011), HLT '11, Association for Computational Linguistics, pp. 1375–1384.
- [72] Ravin, Yael, and Kazi, Zunaid. Is Hillary Rodham Clinton the president? disambiguating names across documents. In *Annual Meeting of the Association for Computational Linguistics (ACL)* (1999), pp. 9–16.
- [73] Richardson, Matthew, and Domingos, Pedro. Markov logic networks. *Machine Learning* 62 (2006), 107–136.
- [74] Riedel, Sebastian, Yao, Limin, McCallum, Andrew, and Marlin, Benjamin M. Relation extraction with matrix factorization and universal schemas. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 9-14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA* (2013), pp. 74–84.
- [75] Robbins, Herbert, and Monro, Sutton. A stochastic approximation method. *Annals of Mathematical Statistics* (1951).

- [76] Rohanimanesh, Khashayar, Wick, Michael, and McCallum, Andrew. Inference and learning in large factor graphs with adaptive proposal distributions. Tech. Rep. UM-CS-2009-008, University of Massachusetts, 2008.
- [77] Rohanimanesh, Khashayar, Wick, Michael, and McCallum, Andrew. Inference and learning in large factor graphs with a rank based objective. Tech. Rep. UM-CS-2009-08, University of Massachusetts, Amherst, 2009.
- [78] Rosenthal, Jeffrey S. Minorization conditions and convergence rates for Markov chain Monte Carlo. *Journal of the American Statistical Association* 90, 430 (1995), 558–566.
- [79] Russell, Bryan C., Torralba, Antonio, Murphy, Kevin P., and Freeman, William T. LabelMe: A database and web-based tool for image annotation. *Int. j-comp-vis* 77 (May 2008), 157–173.
- [80] Salakhutdinov, Ruslan. Learning in markov random fields using tempered transitions. In *Adv. Neur. Inf. Proc. Sys.* (2009).
- [81] Salakhutdinov, Ruslan, and Hinton, Geoffrey. Deep Boltzmann machines. In *Proceedings of the International Conference on Artificial Intelligence and Statistics* (2009), vol. 5, pp. 448–455.
- [82] Salakhutdinov, Ruslan, and Hinton, Geoffrey E. Replicated softmax: an undirected topic model. In *NIPS* (2009), pp. 1607–1614.
- [83] Sarawagi, Sunita, and Gupta, Rahul. Accurate max-margin training for structured output spaces. In *Proc. International Conference on Machine Learning* (2008).
- [84] Shalev-Shwartz, Shai, Singer, Yoram, and Srebro, Nathan. Pegasos: Primal estimated sub-gradient solver for svm. ICML 2007, Ed. A fast online algorithm for solving the linear svm in primal using sub-gradients.
- [85] Singh, Sameer. *Scaling MCMC Inference and Belief Propagation for Large, Dense Graphical Models*. PhD thesis, PhD Thesis, University of Massachusetts, 2014.
- [86] Singh, Sameer, Schultz, Karl, and McCallum, Andrew. Bi-directional joint inference for entity resolution and segmentation using imperatively-defined factor graphs. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases: Part II* (Berlin, Heidelberg, 2009), ECML PKDD '09, Springer-Verlag, pp. 414–429.
- [87] Singh, Sameer, Subramanya, Amarnag, Pereira, Fernando, and McCallum, Andrew. Wikilinks: A large-scale cross-document coreference corpus labeled via links to Wikipedia. Tech. Rep. UM-CS-2012-015, 2012.
- [88] Singh, Sameer, Subramanya, Amarnag, Pereira, Fernando C. N., and McCallum, Andrew. Large-scale cross-document coreference using distributed inference and hierarchical models. In *ACL* (2011), pp. 793–803.

- [89] Singh, Sameer, Wick, Michael, and McCallum, Andrew. Monte carlo mcmc: Efficient inference by approximate sampling. In *Empirical Methods in Natural Language Processing (EMNLP)* (2012).
- [90] Singh, Sameer, Wick, Michael L., and McCallum, Andrew. Distantly labeling data for large scale cross-document coreference. *arXiv abs/1005.4298* (2010).
- [91] Singla, Parag, and Domingos, Pedro. Discriminative training of markov logic networks. In *AAAI* (Pittsburgh, PA, 2005).
- [92] Singla, Parag, and Domingos, Pedro. Lifted first-order belief propagation. In *Proceedings of the 23rd national conference on Artificial intelligence - Volume 2* (2008), AAAI'08, AAAI Press, pp. 1094–1099.
- [93] Soon, Wee Meng, Ng, Hwee Tou, and Lim, Daniel Chung Yong. A machine learning approach to coreference resolution of noun phrases. *Comput. Linguist.* 27, 4 (2001), 521–544.
- [94] Suchanek, Fabian M., Kasneci, Gjergji, and Weikum, Gerhard. Yago: A large ontology from wikipedia and wordnet. *J. Web Sem.* 6, 3 (2008), 203–217.
- [95] Sutton, Charles, and McCallum, Andrew. Collective segmentation and labeling of distant entities in information extraction. Tech. Rep. TR # 04-49, University of Massachusetts, July 2004.
- [96] Sutton, Richard S. Learning to predict by the methods of temporal differences. *Machine Learning* (1988), 9–44.
- [97] Swendsen, R.H., and Wang, J.S. Nonuniversal critical dynamics in MC simulations. *Phys. Rev. Lett.* 58, 2 (1987), 68–88.
- [98] Szegedy, Christian, Zaremba, Wojciech, Sutskever, Ilya, Bruna, Joan, Erhan, Dumitru, Goodfellow, Ian, and Fergus, Rob. Intriguing properties of neural networks. In *ICLR* (2014).
- [99] Tang, Jie, Zhang, Jing, Yao, Limin, Li, Juanzi, Zhang, Li, and Su, Zhong. Arnetminer: Extraction and mining of academic social networks. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (New York, NY, USA, 2008), KDD '08, ACM, pp. 990–998.
- [100] Taskar, Ben, Guestrin, Carlos, and Koller, Daphne. Max-margin markov network. In *Adv. Neur. Inf. Proc. Sys.* (2003).
- [101] Tieleman, Tijmen. Training restricted boltzmann machines using approximations to the likelihood gradient. In *Proc. International Conference on Machine Learning* (2008), pp. 1064–1071.
- [102] Tsochantaridis, I., Hofmann, T., Joachims, T., and Altun, Y. Support vector machine learning for interdependent and structured output spaces. In *Proc. Int. Conf. Mach. Learn.* (2004), pp. 104–112.

- [103] Wang, Daisy Zhe, Franklin, Michael J., Garofalakis, Minos, Hellerstein, Joseph M., and Wick, Michael L. Hybrid in-database inference for declarative information extraction. In *Proceedings of the 2011 international conference on Management of data* (New York, NY, USA, 2011), SIGMOD '11, ACM, pp. 517–528.
- [104] Weinman, Jerod, Learned-Miller, Erik, and Hanson, Allen. Scene text recognition using similarity and a lexicon with sparse belief propagation. *PAMI* (2009).
- [105] Wick, Michael, Culotta, Aron, Rohanimanesh, Khashayar, and McCallum, Andrew. An entity-based model for coreference resolution. In *SIAM International Conference on Data Mining* (2009).
- [106] Wick, Michael, Kobren, Ari, and McCallum, Andrew. Probabilistic reasoning about human edits in information ingegration. In *ICML Workshop on Machine Learning Meets Crowdsourcing* (2013).
- [107] Wick, Michael, and McCallum, Andrew. Advances in learning and inference for partition-wise models of coreference. Tech. Rep. UM-CS-2009-028, University of Massachusetts, Amherst, 2008.
- [108] Wick, Michael, McCallum, Andrew, and Miklau, Gerom. Representing uncertainty in probabilistic databases using scalable factor graphs. Master’s thesis, University of Massachusetts, 140 Governor’s Drive, 2009.
- [109] Wick, Michael, McCallum, Andrew, and Miklau, Gerome. Scalable probabilistic databases with factor graphs and mcmc. *Proc. VLDB Endow.* 3 (September 2010), 794–804.
- [110] Wick, Michael, Rohanimanesh, Khashayar, Bellare, Kedar, Culotta, Aron, and McCallum, Andrew. Samplerank: Training factor graphs with atomic gradients. In *Proceedings of the International Conference on Machine Learning (ICML)* (2011).
- [111] Wick, Michael, Rohanimanesh, Khashayar, Culotta, Aron, and McCallum, Andrew. Samplerank: Learning preferences from atomic gradients. In *NIPS Workshop on Advances in Ranking* (2009).
- [112] Wick, Michael, Rohanimanesh, Khashayar, McCallum, Andrew, and Doan, AnHai. A discriminative approach to ontology alignment. In *In proceedings of the 14th NTII WS at the conference for Very Large Databases (VLDB)* (2008).
- [113] Wick, Michael, Rohanimanesh, Khashayar, Singh, Sameer, and McCallum, Andrew. Training factor graphs with reinforcement learning for efficient map inference. In *NIPS* (2009).
- [114] Wick, Michael, Schultz, Karl, and McCallum, Andrew. Human machine cooperation: Supporting human corrections to automatically constructed kbs. In *NAACL WS on Automatic Knowledge Base Construction (AKBC)* (2012).

- [115] Wick, Michael, Singh, Sameer, and McCallum, Andrew. A discriminative hierarchical model for fast coreference at large scale. In *Association for Computational Linguistics (ACL)* (2012).
- [116] Wick, Michael, Singh, Sameer, Pandya, Harshal, and McCallum, Andrew. A joint model for discovering and linking entities. In *Automated Knowledge Base Construction Workshop* (2013).
- [117] Wick, Michael L., and McCallum, Andrew. Query-aware mcmc. In *Advances in Neural Information Processing Systems 24*, J. Shawe-Taylor, R.S. Zemel, P. Bartlett, F.C.N. Pereira, and K.Q. Weinberger, Eds. 2011, pp. 2564–2572.
- [118] Wick, Michael L., Rohanimanesh, Khashayar, Schultz, Karl, and McCallum, Andrew. A unified approach for schema matching, coreference and canonicalization. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining* (New York, NY, USA, 2008), KDD '08, ACM, pp. 722–730.
- [119] Yang, Jimei, Safar, Simon, and Yang, Ming-Hsuan. Max-margin boltzmann machines for object segmentation. In *Computer Vision and Pattern Recognition (CVPR)* (2014).
- [120] Yang, Xiaofeng, Su, Jian, Lang, Jun, Tan, Chew Lim, Liu, Ting, and Li, Sheng. An entity-mention model for coreference resolution with inductive logic programming. In *ACL* (2008), pp. 843–851.
- [121] Younes, Laurent. Estimation and annealing for gibbsian fields. *Annales de l. I. H. P, Section B* (1988).
- [122] Zhang, Tian, Ramakrishnan, Raghu, and Livny, Miron. Birch: An efficient data clustering method for very large databases. In *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data* (New York, NY, USA, 1996), SIGMOD '96, ACM, pp. 103–114.
- [123] Zhang, Yuan, Lei, Tao, Barzilay, Regina, Jaakola, Tommi, and Globerson, Amir. Steps to excellence: Simple inference with refined scoring of dependency trees. In *ACL* (2014).